Mass-Spring Based Point-Sampled Surface Modelling

Zhixun Su, Xiaojie Zhou, Xiuping Liu, Fengshan Liu, Xiquan Shi

Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, P.R. China.
Email: zxsu@comgi.com, xiaojiezhou66@gmail.com, xpliu@comgi.com.
Applied Mathematics Research Center, Delaware State University, Dover, DE 19901, USA.
Email: {fliu,xshi}@desu.edu

Abstract- In this paper, a physically based modelling method for point-sampled surface is proposed based on mass-spring system. First, a Delaunay based simplification algorithm is applied to the original point-sampled surface to produce the simplified point-sampled surface. Then the mass-spring system for the simplified point-sampled surface is constructed by using tangent planes to address the lack of connectivity information. Finally, the deformed point-sampled surface is obtained by transferring the deformation of the simplified point-sampled surface. Experiments on both open and closed point-sampled surfaces illustrate the validity of the proposed method.

1 Introduction

Point based techniques have gained increasing attention in computer graphics. The main reason for this is that the rapid development of 3D scanning devices has facilitated the acquisition of the point-sampled geometry. Since point-sampled objects do neither have to store nor to maintain globally consistent topological information, they are more flexible compared to triangle meshes for handling highly complex or dynamically changing shapes. Since the pioneering report by Levoy and Whitted [11], much work has been focused on both point based rendering[10, 20, 26] and point based modeling [1, 4, 10, 15, 17, 25]. However, physically based modeling of point-sampled objects is still a challenging area.

Physically based modeling has been investigated extensively in the past two decades. Due to the simplicity and efficiency, mass-spring systems have been widely used to model soft objects in computer graphics, such as cloth simulation. We introduce mass-spring system to point-sampled surface simulation. A Delaunay based simplification algorithm is applied to the original point-sampled surface to produce the simplified point-sampled surface. Due to the lack of connectivity information, mass-spring system can not be applied to point-sampled surface directly. By using the tangent plane and projection, mass-spring system is constructed locally for the simplified point-sampled surface. Then the deformed point-sampled surface is obtained by transferring the deformation of the simplified point-sampled surface. The idea of the present method is similar to [17]. They studied curve deformation, while we focus on point-sampled surface simulation. The overview of our method is shown as Figure 1.

The remaining of the paper is organized as follows. Related work is introduced in Section 2. Section 3 explains the Delaunay based simplification algorithm. Section 4 describes the simulation of the simplified point-sampled surfaced based on mass-spring system. Section 5 introduces the displacement transference to the original point-sampled surface. Some experiments results are shown in Section 6. A brief discussion and conclusion is presented in Section 7.

2 Related Work

Point sampled surface representation is the foundation of point based modeling. Purely point-based representations need mush points to describe an object since the approximation power of piecewise constant representation is linear. Pfister et al [25, 18, 26]proposed the surfels as surface elements to store the position, normal, color, size and others. Surfels help to bridge the gap between neighboring point samples, just as in the Pointshop3D system[25]. When large deformation happens, discrete presentations may generate gaps between point samples. The now standard MLS (Moving least squares) surfaces [1] provide a continuous surface approximating the given unstructured point cloud. A MLS surface is defined as the stationary set of a projection operator. Later Fleishman et al [4] proposed a robust moving least-squares fitting with sharp features for reconstructing a piecewise smooth surface from a potentially noisy point clouds. The displacement transference in our method is inspired by the moving least squares projection.

Point-sampled surfaces often consist of thousands or even millions of points sampled from an underlying surface. Reducing the complexity of such data is one of the key processing techniques. Alexa et al [1] described the contribution value of a point by estimating its distance to the MLS surface defined by the other sample points, and the point with the smallest contribution will be removed repeatedly. Pauly et al [16] extended mesh simplification algorithms to point clouds, and presented the clustering, iterative, and particle simulation simplification algorithms. Moenning et al [14] devised a coarse-to-fine uniform or feature-sensitive simplification algorithm with user-controlled density guarantee. We present a projection based simplification algorithm, which is more suitable for the construction of mass-spring system.

Point based surface editing and deformation are popular fields in point based graphics. Pauly et al [17] introduced Boolean operations and free-form deformation of point-sampled geometry. Miao et al [12] proposed a detail-preserving local editing method for point-sampled geometry based on the combination of the normal geometric details and the position geometric details. Xiao et al [23, 24] pre-

International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
First Special Issue on "Computer Graphics and Geometric Modeling" (2008)
Selected Papers from CGGM'2007 Workshop

```
┌──────────────┐ simplification ┌──────────────┐  mass-spring  ┌──────────────────┐ displacement ┌──────────────┐
│original point│───────────────▶│simulation point│───────────▶│deformed simulation│─transference─▶│deformed point│
│sampled surface│               │sampled surface │  simulation │point sampled surface│             │sampled surface│
└──────────────┘                └──────────────┘               └──────────────────┘              └──────────────┘
```
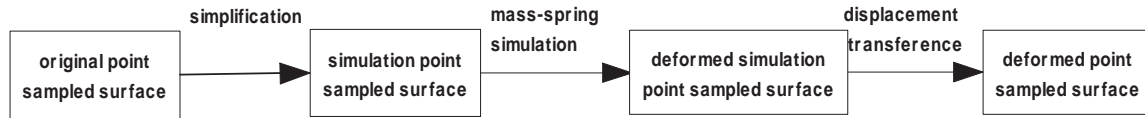
Figure 1: The overview of the present method

sented efficient filtering and morphing methods for point-sampled geometry. Since the pioneering work of Terzopoulos and his co-workers [22], significant research effort has been done in the area of physically based modeling for meshes [5, 19]. Recently, Müller et al [15] presented point based animation of elastic, plastic and melting objects based on continuum mechanics. Clarenz et al [3] proposed framework for processing point-based surfaces via PDEs. Recently, Guo and Qin et al [2, 7, 8, 9] proposed the framework(named P-MASS) of physically based morphing, animation and simulation system. Mass-spring model is one of the most simplest physically-based methods. In the mass-spring system, an object is modelled as a collection of point masses connected by springs in a lattice structure, spring forces and external forces, such as gravity or user applied force, are considered for each single mass point. Since the lack of explicit connectivity information, mass-spring system model for point sampled surface is not a direct extension. Guo et al [6] presented a special mass-spring system for point sampled surface. In this method, the surface is represented as the zero level set of a scalar function, and the springs is for the scalar function not for the spatial position of the points. The deformation of the point sampled surface is achieved by changing the magnitude of the scalar function through the special mass-spring simulation. In this paper, mass-spring system is constructed directly for the simplified point-sampled surface. The idea of the present method is similar to [21]. They studied curve deformation, while we focus on point-sampled surface simulation.

3 Simplification of Point-Sampled Surface

The point-sampled surface consists of n points $P = \{\mathbf{p}_i \in \mathbb{R}^3, i = 1, \ldots, n\}$ sampled from an underlying surface, either open or closed. Since the normal at any point can be estimated by the eigenvector of the covariance matrix that corresponds to the smallest eigen value [16], without loss of generality, we can assume that the normal $\mathbf{n_i}$ at point $\mathbf{p}_i$ is known as input. Traditional simplification algorithms reserve more sample points in regions of high-frequency, whereas less sample points are used to express the regions of low-frequency, which is called adaptivity. However, adaptivity does not necessary bring good results for simulation. An example is shown in Figure 2, 2a) shows the sine curve and the simplified curve, force $\mathbf{F}$ is applied on the middle of the simplified curve, 2b) shows that the simulation based on the simplified curve produce the wrong deformation. We present a Delaunay based simplification algorithm, which is suitable for simulation and convenient for the construction of mass-spring system.

For $\mathbf{p}_i \in P$, the index set of its k-nearest points is denoted by $N_i^k = \{i_1, \ldots, i_k\}$. These points are projected onto the tangent plane at point $\mathbf{p}_i$ (the plane passing through $\mathbf{p}_i$ with normal $\mathbf{n}_i$), and the corresponding projection points are denoted by $\mathbf{q}_i^j, j = 1, \ldots, k$. 2D Delaunay triangulation is implemented on the $k+1$ projection points. There are two possible cases: 1) $\mathbf{p}_i$ is not on the boundary of the surface, 2) $\mathbf{p}_i$ is on the boundary of the surface, shown as Figure 3. Suppose that there are m points $\{\mathbf{q}_i^{j_r}, r = 1, \ldots, m\}$ which are connected with $\mathbf{p_i}$ in the triangle mesh, the union of the triangles that contain $\mathbf{p_i}$ is denoted by R_i whose diameter is d_i. In both cases, if d_i is less than the user-defined threshold, $\mathbf{p_i}$ will be removed. This process is repeated until the desired number of points is reached or the diameter d_i for each point exceeds the threshold. The resulting simplified point set is denoted by $S = \{\mathbf{s}_j, j = 1, \ldots, n_s\}$, and $\mathbf{s}_j$ is called simulation point. It is important to select a proper value of k, too small k may influence the quality of simplification, while too big k will increase the computational cost. In our experiments, preferable k is in the interval $[10 - 20]$.

4 Simulation Based on Mass-Spring System

4.1 Structure of the Springs

Since no explicit connectivity information is known for the simplified point-sampled surface, traditional mass-spring system [19] can not be applied directly. Here the stretching and bending springs are constructed based on the region R_i corresponding to $\mathbf{s}_i$. For $\mathbf{s}_i \in S$, the vertices of the region R_i are $\{\mathbf{q}_i^{j_r}, r = 1, \ldots, m\}$, which are the projection points of $\{\mathbf{s}_i^{j_r}, r = 1, \ldots, m\}$. Assuming that $\mathbf{q}_i^{j_r}$ are sorted counter clockwise. The stretching springs link $\mathbf{s}_i$ and $\mathbf{s}_i^{j_r}$ and the bending springs connect $\mathbf{s}_i^{j_r}$ and $\mathbf{s}_i^{j_{r+2}}$(Figure 4). This process is implemented on each point on S, and the structure of the springs is obtained consequently.

4.2 Estimation of the Mass

The mass of $\mathbf{s}_i$ is needed for simulation. Note that in region of low sampling density, a simulation point $\mathbf{s}_i$ will represent large mass, whereas smaller mass in region of higher sampling density. Since the area of region R_i reflects the sampling density, the mass of $\mathbf{s}_i$ can be estimated by

$$m_i = \frac{1}{3}\rho S_{R_i} \qquad (1)$$

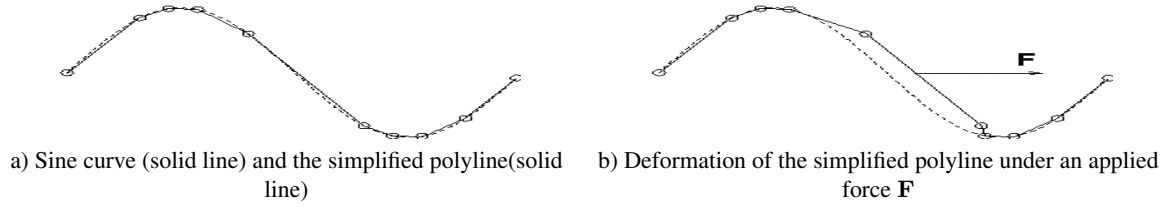where S_{R_i} is the area of region R_i, and ρ is the mass density of the surface.

International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
First Special Issue on "Computer Graphics and Geometric Modeling" (2008)
Selected Papers from CGGM'2007 Workshop

a) Sine curve (solid line) and the simplified polyline(solid line)

b) Deformation of the simplified polyline under an applied force **F**

Figure 2: The effect of simplification to the simulation

a) Delaunay triangulation for case 1)

b) Delaunay triangulation for case 2)

Figure 3: Delaunay triangulation of the projection points on the tangent plane

4.3 Forces

According to the Hooke's law, the internal force $\mathbf{F}_s\left(S_{i,j}\right)$ of spring $S_{i,j}$ linking two mass points $\mathbf{s}_i$ and $\mathbf{s}_j$ can be written as

$$\mathbf{F}_s\left(S_{i,j}\right) = -k_{i,j}^s\left(\mathbf{I}_{i,j} - l_{i,j}^0\frac{\mathbf{I}_{i,j}}{\|\mathbf{I}_{i,j}\|}\right) \tag{2}$$

where $k_{i,j}^s$ is the stiffness of spring $S_{i,j}$, $\mathbf{I}_{i,j} = \mathbf{s}_j - \mathbf{s}_i$, and $l_{i,j}^0$ is the natural length of spring $S_{i,j}$.

In dynamic simulation, a damping force is often introduced to increase the stability. In our context, the damping force is represented as

$$\mathbf{F}_d\left(S_{i,j}\right) = k_{i,j}^d\left(\mathbf{v}_j - \mathbf{v}_i\right) \tag{3}$$

where $k_{i,j}^d$ is the coefficient of damping, $\mathbf{v}_j$ and $\mathbf{v}_i$ are the velocities of $\mathbf{s}_j$ and $\mathbf{s}_i$.

Appling external forces to the mass-spring system yields realistic dynamics. The gravitational force acting on a mass point $\mathbf{s}_i$ is given by

$$\mathbf{F}_g = m_i g \tag{4}$$

where m_i is the mass of the mass point $\mathbf{s}_i$, g is the acceleration of gravity. A force that connects a mass point to a point in world coordinates $\mathbf{r}_0$ is given by

$$\mathbf{F}_r = k_r\left(\mathbf{r}_0 - \mathbf{s}_i\right) \tag{5}$$

where k_r is the spring constant. Similar to [22], other types of external forces, such as the effect of viscous fluid, can be introduced into our system.

4.4 Simulation

The mass-spring system is governed by Newton' s Law. For a mass point $\mathbf{s}_i$, there exists the equation

$$\mathbf{F}_i = m_i\mathbf{a}_i = m_i\frac{\mathrm{d}^2\mathbf{x}_i}{\mathrm{d}t^2} \tag{6}$$

where m_i, $\mathbf{x}_i$, and $\mathbf{a}_i$ are the mass, displacement, acceleration of $\mathbf{s}_i$ respectively. A large number of integration schemes can be used to Eq. (6). Explicit schemes are easy to implement and computationally cheap, but stable only for small time steps. In contrast, implicit schemes are unconditionally stable at the cost of computational and memory consumption. We use explicit Euler scheme for simplicity in our system.

5 Deformation of the Original Point-Sampled Surface

The deformation of the original point-sampled surface can be obtained by the deformation of the simplified point-sampled surface. Let us consider the x-component u of the displacement field $\mathbf{u} = (u, v, w)$. Similar to [15], we compute the displacement of $\mathbf{p}_i$ through the simulation points in its neighborhood. While the simulation points sampled from an underlying surface, it may be singular due to coplanarity if we use moving least square fitting to compute the displacement directly. We treat the tangent plane at $\mathbf{p}_i$ as the reference domain. The simulation points $\mathbf{s}_i^j, j = 1, \ldots, k$ in the neighborhood of $\mathbf{p}_i$ are projected onto the reference plane, with corresponding projection points $\mathbf{q}_i^j, j = 1, \ldots, k$, and $(\bar{x}_j, \bar{y}_j), j = 1, \ldots, k$ are the coordinates of $\mathbf{q}_i^j, j = 1, \ldots, k$ in the local coordinate system with origin $\mathbf{p}_i$. Let the x-component u is given by

$$u\left(\bar{x}, \bar{y}\right) = a_0 + a_1\bar{x} + a_2\bar{y} \tag{7}$$

The parameters $a_l, l = 0, 1, 2$ can be obtained by minimizing

$$E_i = \sum_{j=1}^{k} w\left(r_j\right)\left(u_j - a_0 - a_1\bar{x}_j - a_2\bar{y}_j\right)^2 \tag{8}$$

where r_j is the distance between $\mathbf{p}_i$ and $\mathbf{q}_i^j$, $w\left(\cdot\right)$ is a Gaussian weighting function $w\left(r_j\right) = \exp\left(-r_j^2/h^2\right)$. Eq. (8) can be rewritten in matrix form as

$$E_i = (\mathbf{Pa} - \mathbf{u}^T)\mathbf{W}(\mathbf{Pa} - \mathbf{u}) \tag{9}$$

International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
First Special Issue on "Computer Graphics and Geometric Modeling" (2008)
Selected Papers from CGGM'2007 Workshop

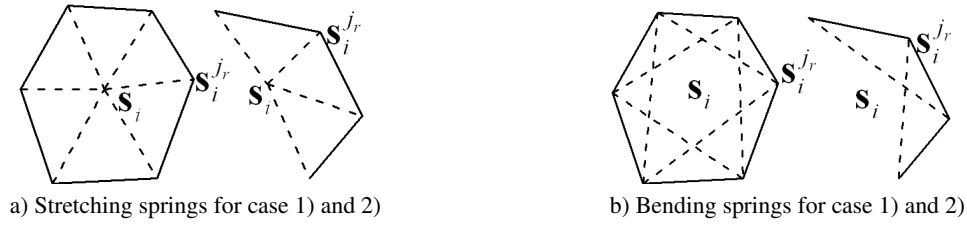a) Stretching springs for case 1) and 2)   b) Bending springs for case 1) and 2)

Figure 4: The spring structures (dashed lines).

where

$$\mathbf{u} = \begin{pmatrix} u_1 \\ \cdots \\ u_k \end{pmatrix}, \mathbf{P} = \begin{pmatrix} 1 & \bar{x}_1 & \bar{y}_1 \\ \cdots & \cdots & \cdots \\ 1 & \bar{x}_k & \bar{y}_k \end{pmatrix}$$

$$\mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \mathbf{W} = \begin{pmatrix} w(r_1) & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & w(r_k) \end{pmatrix}.$$

By minimizing Eq. (9), we have

$$\frac{\partial E_i}{\partial \mathbf{a}} = \mathbf{Aa} - \mathbf{Bu} = \mathbf{0} \tag{10}$$

where $\mathbf{A} = \mathbf{P}^T\mathbf{W}\mathbf{P}$, $\mathbf{B} = \mathbf{P}^T\mathbf{W}$. Then

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{Bu} \tag{11}$$

The x-component u of the displacement of $\mathbf{p}_i$ is

$$u_i = u(0,0) = a_0 \tag{12}$$

Similarly, v and w can be computed. The matrix $\mathbf{A}$, $\mathbf{B}$ can be pre-computed, and used for the computation of v and w. Since the shape of the point-sampled surface is changed due to the displacements of the sample points, the normal of the underlying surface will change consequently. The normal can be computed by the covariance analysis as mentioned above. The point sampling density will be changed due to the deformation, we use the resampling scheme of [1] to maintain the surface quality.

6 Experimental Results

We implement the proposed method on a PC with Pentium IV 2.0GHz CPU and 512MB RAM. Experiments are performed on both closed and open surfaces, shown as Figure 5 and Figure 6. The sphere is downloaded from the website of PointShop3D and composed of 3203 surfels. For the modeling of the hat, the original point-sampled surface is sampled from the lower part of a sphere, and a stretching force acted on the middle of the point-sampled surface produce the hat. We also produce other interesting examples, the sculpting of a character "B" and the logo "CGGM 07", which are both produced by applying force on the point-sampled surfaces (Figure 7, Figure 8). The simplification and the construction of mass-spring system can be performed as preprocess, and the simulation points is much less in the simplified surface than the original point-sampled surface, so the simulation is very efficient. The performance of simulation is illustrated in Table 1. The main computational cost is the transference of the displacement from the simplified surface
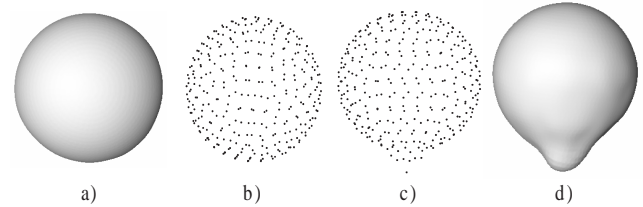

a)   b)   c)   d)

Figure 5: The deformation of a point sampled sphere. a) The original point-sampled surface. b) The simplified point-sampled surface. c) Deformation of the simplified point-sampled surface. d)The deformation of a point sampled sphere.
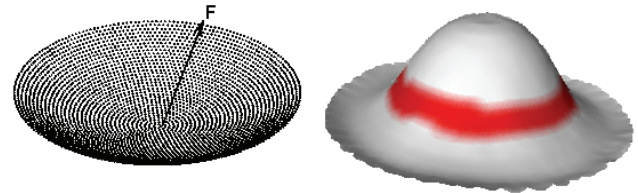

Figure 6: The modeling of a hat.

to the original point-sampled surface and the normal computation of the deformed point-sampled surface. Compared to the global parameterization in [8], the local construction of mass-spring system makes the simulation more efficient. The continuum-based method [15] presented the modeling of volumetric objects, while our method can deal with both volumetric objects using their boundary surface and sheet like objects.
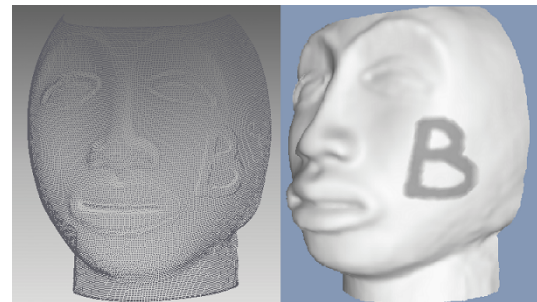

Figure 7: The sculpting of character "B".

International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
First Special Issue on "Computer Graphics and Geometric Modeling" (2008)
Selected Papers from CGGM'2007 Workshop

Figure 8: The "CGGM 07" logo

Table 1: The simulation time

Number of simulation points	85	273	338
Simulation time per step (s)	0.13	0.25	0.42

7 Conclusion

As an easy implemented physically based method, mass-spring systems have been investigated deeply and have been used widely in computer graphics. However, it can not be used to point-sampled surfaces due to the lack of connectivity information and the difficulty of constructing mass-spring system. We solve the problem of the construction of mass-spring system for point-sampled surface based on projection and present a novel mass-spring based simulation method for point-sampled surface. A Delaunay based simplification algorithm facilitates the construction of mass-spring system and ensures the efficiency of the simulation method. Further study focuses on the simulation with adaptive topology. And the automatic determination of the simplification threshold should be investigated to ensure suitable tradeoff between accuracy and efficiency in the future.

Acknowledgement

Bibliography

[1] Alexa M., Behr J., Cohen-Or D., Fleishman S., Levin D., Silva C. T.: Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics **9**(2003) 3-15

[2] Bao Y., Guo X., Qin H.: Physically-based morphing of point-sampled surfaces. Computer Animation and Virtual Worlds **16** (2005) 509 - 518

[3] Clarenz U., Rumpf M., Telea A.: Finite elements on point based surfaces, Proceedings of Symposium on Point-Based Graphics (2004)

[4] Fleishman S., Cohen-Or D., Silva C. T.: Robust moving least-squares fitting with sharp features. ACM Transactions on Graphics **24** (2005) 544-552

[5] Gibson S.F., Mirtich B.: A survey of deformable models in computer graphics. Technical Report TR-97-19, MERL, Cambridge, MA, (1997)

[6] Guo X., Point-Based Modeling, Animation, and Simulation System for Computer Graphics, Ph. D Dissertation, Stony Brook University, 2006.

[7] Guo X., Hua J., Qin H.: Scalar-function-driven editing on point set surfaces. IEEE Computer Graphics and Applications **24** (2004) 43 - 52

[8] Guo X., Li X., Bao Y., Gu X., Qin H.: Meshless thin-shell simulation based on global conformal parameterization. IEEE Transactions on Visualization and Computer Graphics **12** (2006) 375-385

[9] Guo X., Qin H.: Real-time meshless deformation. Computer Animation and Virtual Worlds **16** (2005) 189 - 200

[10] Kobbelt L., Botsch M.: A survey of point-based techniques in computer graphics. Computer & Graphics, **28** (2004) 801-814

[11] Levoy M., Whitted T., The use of points as a display primitive, In Technical Report TR85-022, University of North Carolina at Chapel Hill, 1985.

[12] Miao Y., Feng J., Xiao C., Li H., Peng Q., Detail-preserving local editing for point-sampled geometry. H.-P Seidel, T. Nishita, Q. Peng (Eds), CGI 2006, LNCS 4035 (2006) 673-681

[13] Miao L., Huang J., Zheng W., Bao H. Peng Q.: Local geometry reconstruction and ray tracing for point models. Journal of Computer-Aided Design & Computer Graphics **18** (2006) 805-811

[14] Moenning C., Dodgson N. A.: A new point cloud simplification algorithm. Proceedings 3rd IASTED Conference on Visualization, Imaging and Image Processing, Benalmádena, Spain (2003) 1027-1033

[15] Müller M., Keiser R., Nealen A., Pauly M., Gross M., Alexa M.: Point based animation of elastic, plastic and melting objects, Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2004) 141-151

[16] Pauly M., Gross M., Kobbelt L.: Efficient simplification of point-sampled surfaces. Proceedings of IEEE Visualization (2002) 163-170

[17] Pauly M., Keiser R., Kobbelt L., Gross M.: Shape modeling with point-sampled geometry. ACM Transactions on Graphics **22**(2003) 641-650

[18] Pfister H., Zwicker M., Baar van J. et al Surfelssurface elements as rendering primitives, Proc. SIGGRAPH, (2000) 54-61.

[19] Provot X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. Proc of Graphics Interface (1995) 147-154.

[20] Rusinkiewicz S. and Levoy M., Qsplat: A multiresolution point rendering system for large meshes, Proc. SIGGRAPH (2000) 343-352

[21] Su Z., Li L., Zhou X.: Arc-length preserving curve deformation based on subdivision. Journal of Computational and Applied Mathematics **195** (2006) 172-181

[22] Terzopoulos D., Platt J., Barr A., Fleischer K.: Elastically deformable models. Proc. SIGGRAPH (1987) 205-214

[23] Xiao C., Miao Y., Liu S., Peng Q., A dynamic balanced flow for filtering point-sampled geometry. The Visual Computer **22** (2006) 210-219

[24] Xiao C., Zheng W., Peng Q., Forrest A. R., Robust morphing of point-sampled geometry. Computer Animation and Virtual Worlds **15** (2004) 201-210

[25] Zwicker M., Pauly M., Knoll O., Gross M.: Pointshop3d: An interactive system for point-based surface editing. ACM Transactions on Graphics **21**(2002) 322- 329

[26] Zwicker M., Pfister H., Baar van J., and Gross M., Surface splatting, Proc. SIGGRAPH (2001) 371-378