Received: 15 May 2019 | Revised: 30 September 2019 | Accepted: 8 October 2019

DOI: 10.1002/cav.1920

RESEARCH ARTICLE

WILEY

Sketch-based shape-constrained fireworks simulation in head-mounted virtual reality

Xiaoyu Cui[1] | **Ruifan Cai**[1] | **Xiangjun Tang**[1] | **Zhigang Deng**[2] | **Xiaogang Jin**[1]

[1]State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

[2]Computer Science Department, University of Houston, Houston, Texas

Correspondence
Xiaogang Jin, State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China.
Email: jin@cad.zju.edu.cn

Funding information
Key Research and Development Program of Zhejiang Province, 2018C01090; National Key R&D Program of China, 2017YFB1002600; National Natural Science Foundation of China, 61972344; Science and Technology Project on Preservation of Cultural Relics, Cultural Heritage Bureau of Zhejiang Province, 2018009

Abstract
In this paper we present a novel shape-constrained fireworks simulation method with rich textures in an HMD (Helmet Mounted Display) virtual environment using sketched feature lines as input. Our approach first retrieves an object from a three-dimensional (3D) model database using a sketch-based 3D shape retrieval algorithm. Then, in order to approximate models with complex structures, we introduce a novel point sampling algorithm based on Gaussian curvatures, which stores not only the positions of the selected vertices but also the texture (UV) coordinates information for texture display. In addition, we introduce a multilevel explosion process so that the fireworks can dynamically form specific, visually pleasing shapes. Through our experiments, we demonstrate that our approach can produce better results than state-of-the-art approaches.

KEYWORDS
particle systems, shape constraints, sketch-based shape retrieval, virtual reality

1 | INTRODUCTION

In recent decades the simulation of natural phenomena such as flames, smokes, etc. has been widely applied in computer games, films, education, virtual reality, and many other fields. Among many existing techniques, particle systems[1] have been widely considered as a powerful technique in computer graphics that uses a large number of small sprites, three-dimensional (3D) models, or other graphical primitives to simulate some fuzzy phenomena.

Some research works have been done to simulate fireworks based on particle systems, whose vibrant hues and dazzling starbursts can bring beauty to the silence and darkness of night, since fireworks are often considered to be the highlights of traditional festivals and other special events. In the recent related work,[2] Zhao et al. presented a fireworks controller to simulate shape-constrained fireworks in real time. With the popularity of consumer-grade virtual reality (VR) devices,[3] especially helmet-mounted display (HMD) devices, ordinary users are affordable to experience the highly immersive experience of watching virtual fireworks at home. However, Zhao et al.'s method[2] has the following limitations when used in HMD virtual environments. First, it relies on the keyboard to input a specific mesh model, which interrupts the user's immersive experience in a HMD virtual environment. Second, it approximates shapes with bunches of fireworks, which may fail to describe geometric features for complex objects. Third, it does not simulate textured objects that can potentially exhibit more diverse and stunning visual effects.

Although a sketch-based system like Teddy can design rotund objects efficiently,[4] it may take a first-time user a few minutes to master the provided operations. For scenarios like VR or science and technology museums, it is still too complex for an ordinary visitor to learn. Moreover, such a system does not support the design of realistic complex objects with rich textures.

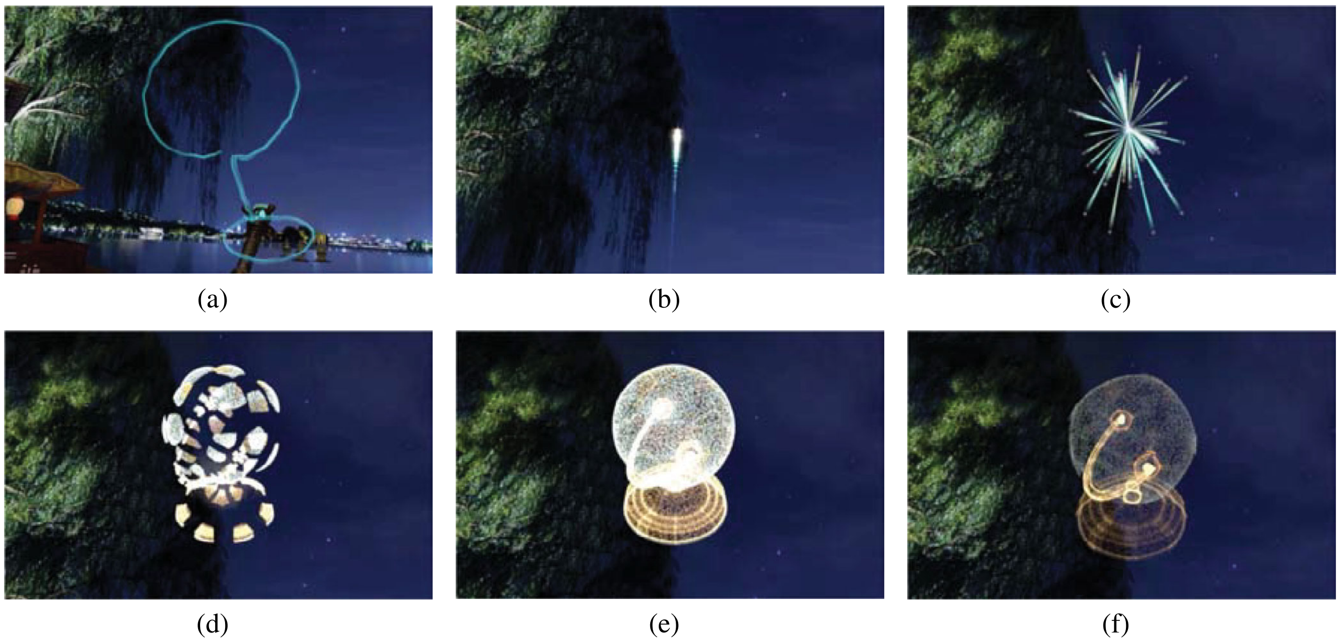


FIGURE 1 The process of the "tellurion" fireworks simulation: a, sketch drawing; b, fireworks emitting from horizontal; c, first-level explosion; d and e, second-level explosion; f, fireworks fading away

Inspired by the above observations, in this work we develop a new sketch-based fireworks simulation system to exhibit 3D complex models with abundant texture information in virtual environments. First, a user sketches the desired model, as seen from a certain viewpoint (Figure 1a). With the sketch as the input for our sketch-based object retrieval system, the resulting model is obtained after comparisons with line drawings of all the objects from probable view directions. Then, to efficiently simulate the shape of the retrieved object, we propose a Gaussian curvatures-based point sampling algorithm, where the positions of the vertices unevenly selected can be taken as the target positions of the simulation process. We record their 3D positions and their texture (UV) coordinates separately to display both the shape and texture information naturally. In addition, we introduce a new concept of multi-level explosions (Figure 1c-f), since it is often infeasible to form complex shapes with a single explosion. To better determine the shape of each level, the K-means clustering algorithm[5] is applied to the sampled vertices. To provide immersive interaction experience for users, we integrate our system into HMD virtual environment so that users are able to sketch what they want with a controller in hand while indulging in the virtual scene.

The contributions of our work can be summarized as follows:

- It introduces a 3D shape retrieval-based shape-constrained fireworks simulation method using sketches in head mounted virtual environments.
- It introduces a new point sampling algorithm based on Gaussian curvatures. Instead of evenly approximating an object, our new sampling method selects a different number of points according to the banding property of the object in order to simulate models with complex silhouettes.
- It exhibits not only the shape but also the texture information of the object retrieved during the exploding of fireworks in order to generate realistic visual effects.

2 | RELATED WORK

2.1 | Shape-constrained animation

In recent years, shape-constrained animation has attracted increasing attentions because it can produce aesthetically pleasing visual effects. Wang et al. proposed a sketch-based shape-preserving tree animation method, which can transform a leafy tree into custom crown shapes defined by sketches.[6] For creating morphing effects between two arbitrary 3D objects, Wang et al. combined the features of 3D morphing and flock animation by taking each tetrahedron decomposed from 3D objects as an agent.[7] Recently Chen et al. proposed a multi-agent model for large-scale controllable shape-constrained simulation of flying insects.[8] It can guide aggregate insects smoothly deforming into a target shape using a force model. Shape-constrained animation has also been used in fluid and smoke simulations. Yang et al. presented a synthesis method to improve small-scale turbulence details for controllable smoke animations constrained by shapes and paths.[9] Hu et al. presented an interface to support two-dimensional (2D) sketch-based fluid design with a perceptual understanding of human sketches.[10]

2.2 | Sketch-based 3D shape retrieval

Sketch-based 3D shape retrieval is to retrieve 3D models based on an input 2D sketch. To compute the similarity between a sketch image and a 3D model, several existing content-based image retrieval methods find the best match based on the projected images of the model. Funkhouser et al. presented a sketch-based matching approach, where thumbnail images are prerendered with the boundary contours of each 3D object from 13 orthographic view directions.[11] Assuming that users usually express their concept of a 3D shape with three 2D views without missing any information, the method by Pu et al.[12] only needs three or six views that can be determined in real time. Their sketch user interface also provides a robust way to allow users to make small mistakes during the sketch drawing. Saavedra et al.[13] efficiently compute the histogram of local edge orientations, which is invariant to scaling and translation transformations. Yoon et al.[14] proposed an approach to compute descriptors by analyzing diffusion tensor fields of suggestive contour images that are rendered from different viewpoints to measure the similarities with the user-drawn sketch.

2.3 | Fireworks simulations

Simulation of fuzzy objects is one important subject in the field of computer graphics. The particle system approach was firstly proposed by Reeves to simulate fuzzy irregular objects (e.g., fires, clouds, water, etc.).[1] A particle system consists of a number of individual particles, each of which has its own properties that can be changed over time. Later, researchers introduced various fireworks simulation methods based on particle systems. Zhao et al. proposed a Graphics Processing Unit (GPU)-based shape-constrained fireworks simulation pipeline, where a dual-depth-peeling-based point sampling algorithm is employed to provide good distributions.[2] Chang et al. introduced a Compute Unified Device Architecture (CUDA)-based framework to generate dynamic fireworks with high efficiency and frame rates.[15] Kim et al. introduced a sample-based control method that enables the formation of a target shape by assembling fire-flakes.[16] However, these methods do not tackle fireworks simulations for 3D models with complex structures. In particular, existing point sampling algorithms in these methods fail to approximate sharp outlines and can only associate one or a limited number of colors with the simulation, which largely undermine the vividness and realism of fireworks effects.

3 | PIPELINE OVERVIEW

In the offline process, we first generate line drawings of each 3D model in our database from 102 different view directions using line rendering techniques. Then, we encode the line drawings with a bag-of-features (BoF) approach through Gabor filters to form local descriptors and quantify these descriptors to achieve the resilience to both global and local deformations.

At runtime, based on the histogram features of an input user-drawn sketch, we retrieve the best matched 3D object from the database, whose projection is the most similar to the input sketch. To curb the shape of fireworks for maximal approximation, we employ a point sampling algorithm based on the Gaussian curvatures of the discrete vertices of the 3D object. Besides storing the positions of the sampled vertices, the texture (UV) coordinates of the sampled vertices are also recorded for texture mapping on particles in our work. After point sampling, the K-means clustering algorithm is applied to the sampled vertices to determine seeds as the firework emitters at the second level of the fireworks explosion. For each vertex, we use a bilinear interpolation algorithm to obtain the filtered pixel color from its corresponding texture image and set it as the color of the particle during the fireworks explosion. Figure 2 shows the pipeline of our fireworks simulation system.

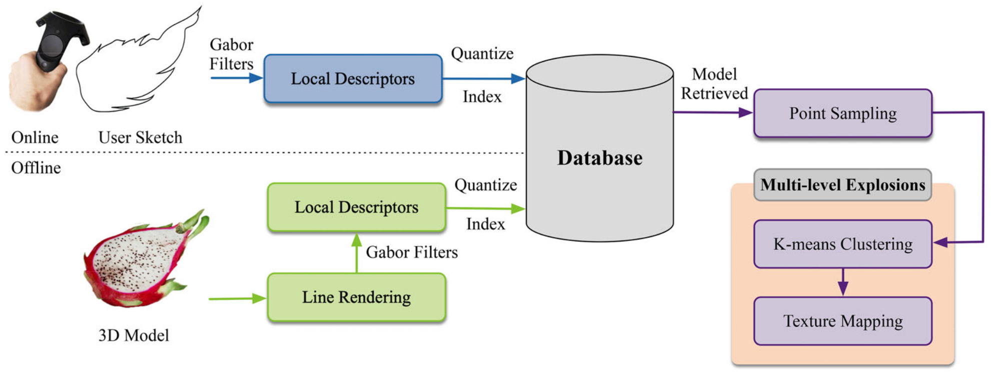


FIGURE 2 The overview of our fireworks simulation system. In the offline process, our framework sets up a three-dimensional model database for the online querying. In the online process, after a user draws a sketch that he/she wants to find in the helmet-mounted display-based VR environment, our approach retrieves the object of interest and performs a visually realistic, shape-constrained fireworks simulation

4 | SKETCH-BASED OBJECT RETRIEVAL

We develop a sketch-based object retrieval system based on the previous work by Eitz et al.,[17] which has been demonstrated to have a high-quality retrieval performance. The core idea of this approach is the BoF model,[18,19] which has been widely used for image classification by treating image features as words. This model can be briefly described as follows: (a) extract features from a training set of images; (b) learn a "visual vocabulary" from these features through a clustering process; (c) quantize the features based on the visual vocabulary; and (d) redefine the images according to the frequencies of "visual words." Thus, the definition of the BoF model can be conceptually regarded as the "histogram representation based on independent features.".[20] However, this model itself is designed to handle 2D image matching, and it has not been applied or extended to solve the problem of sketch-based 3D object retrieval.

To convert the features of 3D objects to those of 2D images, we generate a set of 2D drawings for each 3D model in our dataset. In order to simplify the pipeline, we uniformly select 102 viewpoints on a sphere with the model enclosed to project it into line drawings. With groups of line drawings generated from the 3D objects in the dataset, we extract their features based on a bank of Gabor filters to encode them as local image descriptors. By employing the K-means clustering algorithm, a visual vocabulary is built, where the centroid of each resulting cluster represents a visual word. The number of clusters, as an important parameter that strongly influences the retrieval performance, is empirically set to 2,500 to obtain satisfactory results in our experiments. Through the quantization of local image descriptors for each line drawing, the histograms of visual word frequencies are defined and stored in an inverted index data structure[21] for online query.

At runtime, the user first draws a sketch with a wireless VR controller in hand and finishes his/her submission. Then, our approach at the back-end performs the following operations on the sketch: extracting local image descriptors, quantizing them, and encoding them as a histogram of the occurrence frequencies of the visual words. In this process, the Term frequency-inverse document frequency weighting functions[21] are used to determine the importances of the visual words, and we return the model with the best matched view after computing the similarities between the histogram of the query sketch and those in the data structure of inverted index.

5 | FIREWORKS ANIMATION CONTROL

5.1 | Point sampling

The shape of fireworks is controlled by the triangular mesh of the retrieved model. Using the vertices of the mesh to determine the properties of particles might be a straightforward solution. But for a complex model, its number of vertices may be too large to meet the requirements for real-time fireworks simulation, or not large enough to exhibit details of the fireworks if the model is over-simplified. As a result, point sampling is needed to achieve satisfactory fireworks simulations. Although the efficiency of the dual depth peeling algorithm[2] has been demonstrated for vertice sampling, it is still difficult to deal with complex models even using the dual depth peeling. To tackle this issue, we present a new point sampling algorithm based on Gaussian curvatures. Note that when the retrieved model has been loaded into the system, we first evaluate its maximum distance between any two vertices along the directions of three axes (i.e., x, y, and z directions). Based on the proportions between the maximum distances and the preset standard distances, the model is accordingly scaled to a fixed size.

5.1.1 | Gaussian curvatures

Mapping a surface to the unit sphere sets up the correspondences between the points on the surface and the points on the sphere, which is also called *Gaussian mapping*. Gaussian curvature K is the product of two principal curvatures, κ_1 and κ_2, at a point on the surface,

$$K = \kappa_1 . \kappa_2, \tag{1}$$

the geometric interpretation of which refers to the area of the sphere or the limit of the local area of the surface. Note that the Gaussian curvature reflects the degree of the local curvature of the surface.

Given the positive and negative values of Gaussian curvatures, it is convenient to define the structure of the surface at a point. The Gaussian curvature $K > 0$ represents an elliptic point, $K < 0$ represents a hyperbolic point, and $K = 0$ represents a plane or a parabolic point. Besides, Gaussian curvature is the intrinsic quantity of the surface, which is only related to the first basic type of the surface and has nothing to do with the selection and parameterization of the axes. Therefore, for every vertex on the input triangle mesh, we compute its discrete Gaussian curvature. Based on the obtained curvature value, a corresponding number of points are selected within the range of the triangle mesh whose formation includes particular vertices.

5.1.2 | Approximation in discrete geometry

In recent years researchers have proposed many methods to estimate the micro-components of discrete surfaces. At least four methods have been designed to estimate the discrete Gaussian curvatures for triangle mesh surfaces.[22–25] In particular, the method in Reference 25 is also called the Voronoi method since it uses average Voronoi cells. Based on the research study by Vasa et al.,[26] the Voronoi method is optimal for estimating various curvatures of triangle mesh surfaces.

The basic idea of the Voronoi method is to regard the smooth surface as the limit or linear approximation of a cluster of triangular patches. The measure of each vertex on a triangular patch can be taken as the average measure within a small neighborhood, called Voronoi region (Figure 3). Thus, the discrete form of Gaussian curvatures can be obtained based on discrete differential geometry as follows,

$$K(v) = \frac{1}{A(v)} \left(2\pi - \sum_{v_i \in N(v)} \theta_i \right).$$

Here, $\sum_{v_i \in N(v)} \theta_i$ represents the sum of the angles corresponding to the adjacent triangles, and $A(v)$ is the sum of fragmental areas neighboring to v.

Different discrete methods can be used to compute $A(v)$:

- For an acute triangle (see Figure 4a), connect its circumcenter o_1 with the midpoints p_1, q_1 of the sides except the one opposite to a_1 , and obtain a new area $A_{\text{acute}}(v)$ (grey).
- For an obtuse triangle (see Figure 4b), connect the midpoint o_2 of the side opposite to the obtuse angle with the midpoints p_2, q_2 of the other two sides separately, and obtain a new area $A_{\text{obtuse}}(v)$ (grey).

After computing the Gaussian curvature of every vertex on triangular patches, we determine the number of sampled points randomly in the neighboring triangular areas of a vertex, based on the different levels partitioned corresponding to the absolute value of Gaussian curvature. We explain it further in Section 7.

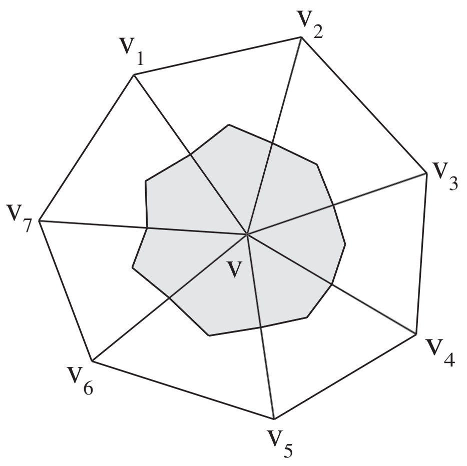


FIGURE 3 An example of Voronoi region which appears in grey

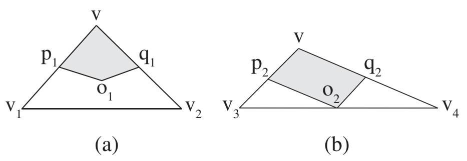


FIGURE 4 Discrete methods to compute $A(v)$: a, for an acute triangle; b, for an obtuse triangle

5.2 | Multilevel Explosions

5.2.1 | Explosion levels

The common approach to approximate the full process of a fireworks explosion is to calculate the emission velocities and directions of the particles' motions based on their final coordinates and the explosion point (generally, all particles in the system have the same explosion point). However, the time from particle emission to the forming of the target shape, about 2 to 3 seconds, is typically too short for fireworks to be watched clearly and sharply. Real-world fireworks often explode several times in order to form the target shape aloft. To increase the ornamental value, we determine an intermediate shape as a transition from the single emitter to the ultimate shape formed by particles and extend the method proposed by Zhao et al.[2] to achieve shape control in the system. The cycle for each fireworks explosion is extended to around 9 seconds in our experiments.

With the coordinates of the emission point where the explosion starts, the number and coordinates of the final points where particles reach at the end, the K-means clustering algorithm is applied to define clustering centers, which together form the intermediate shape after convergence. Assuming N denotes the number of final points, and K denotes the number of selected centroids, the K-means clustering algorithm employed to achieve second-level explosions can be described as follows:

1. Randomly select K from N particles as the initial centroids.
2. For each remaining particle, calculate its distance to every centroid and classify it into the set of the nearest centroid.
3. Recalculate the centroids based on the sets that have been obtained.
4. Repeat steps 2 and 3 until the distance between the new centroid and the centroid of the last update is smaller than a user-specified threshold. The clustering step ends.

Therefore, fireworks explosions are divided into two processes. The explosion process from the emission point (Figure 5a) to the intermediate shape (Figure 5b) is the first-level explosion, and the explosion from the intermediate shape to the refined shape of the retrieved object (Figure 5d) is the second-level explosion.

5.2.2 | Simulate fireworks explosions

The first-level explosion is achieved by a single particle system with K particles, which starts from the fireworks launching point and fall on their corresponding centroids. Let g be the gravitational acceleration, centroid[i] be the target position

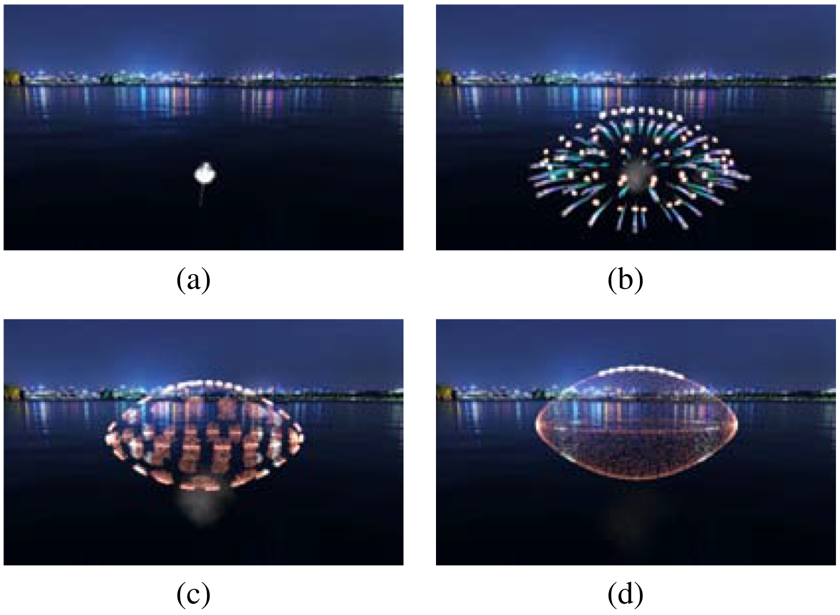


FIGURE 5 Explosion levels. a, the emission point; b, the intermediate shape; c, the transition shape; d, the refined shape of the retrieved object

of particle i, d_i be the distance vector from *mcenter*, which is the launching point (ie, the geometric center of the retrieved 3D model), to *centroid*[i], and *ls* be the duration of the explosion. Then, v_i, the initial velocity vector of particle i, can be calculated as follows:

Algorithm 1. First-level Explosion Simulation

```
1: function FIRSTEXPLODE(centroid)
2:     $n \leftarrow centroid.length$
3:     for $i = 0 \rightarrow n - 1$ do
4:         $d_i \leftarrow centroid[i] - mcenter$
5:         $v_i.x \leftarrow d_i.x/ls$
6:         $v_i.y \leftarrow (d_i.y + 0.5 * g * ls * ls)/ls$
7:         $v_i.z \leftarrow d_i.z/ls$
8:     end for
9: end function
```

Then, assuming $t = 0$ when the first-level explosion starts, p_i^t, the position of particle i at time t can be calculated as follows:

$$p_i^t = \text{mcenter} + v_i * t.$$

The second-level explosion is achieved by K particle systems, in which the number of all the particles is N. The starting point of each second-level particle system is its corresponding centroid in the first level, and each particle in the system will finally arrive at its target position. To this end, the shape of the retrieved 3D object will be formed. The method used to calculate the initial velocity vectors of the particles in the K particle systems is the same as that in the first-level explosion.

In the explosion of our simulated fireworks, each particle is initialized with a high speed and its size grows explosively. Then, the particle's size will shrink slowly and its speed will decrease rapidly because of air resistance. In addition, we flick the brightness of each particle during the whole process in order to improve the realism.

5.2.3 | Map textures

By dividing the explosions into multiple levels, we build an integrated fireworks simulation system conforming to the laws of physics. However, the firework effects achieved through mono-color particles fail to approximate the textures on

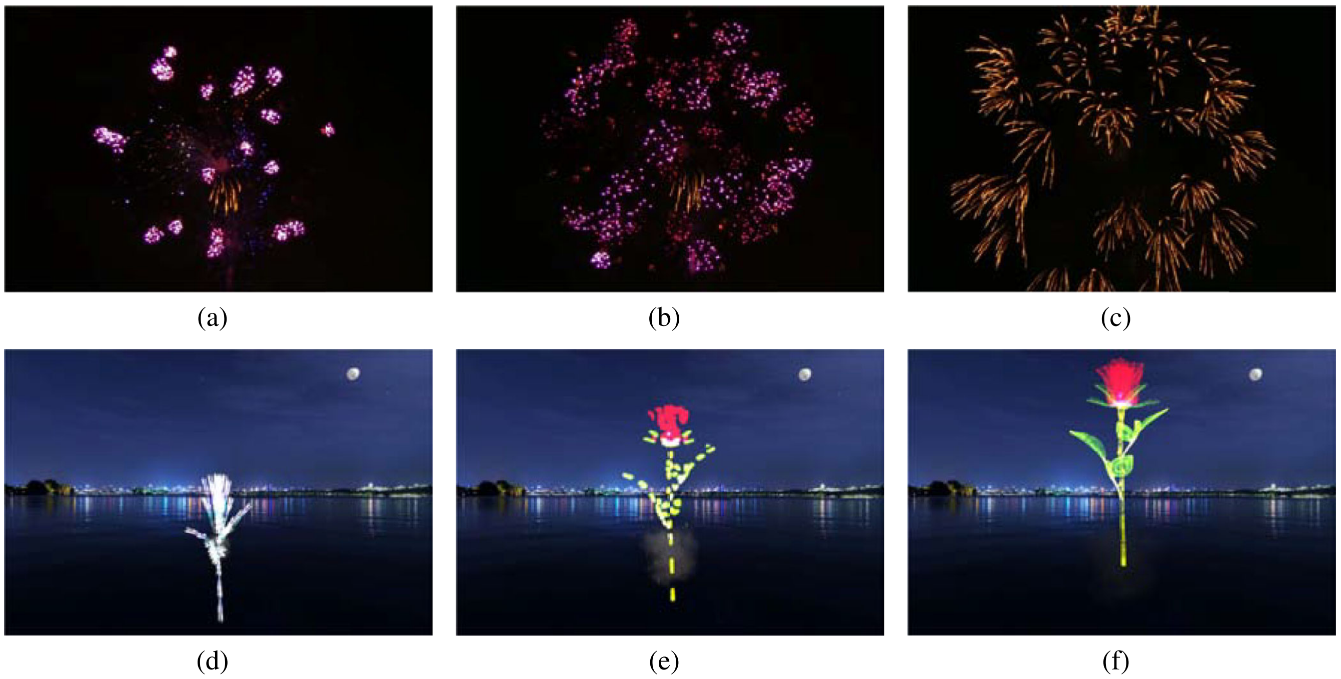


FIGURE 6 The comparison with a real-world fireworks display : a-c, the process of a real firework display; d, the first-level explosion in our simulation system; e, the second-level explosion in our simulation system; and f, the shape of the three-dimensional model “rose” is fully formed

the retrieved 3D model. As aforementioned, at the previous step we have selected a specific number of points from the neighboring area of triangle patches for each vertex, which is described in Section 5.1. In order to map each selected point with the color sampled from the model's texture, we employ the same point sampling method on the UV coordinates of the vertices of each triangle patch, and then use bilinear interpolation algorithm to obtain the precise texture information of a sampled point and set it as the initial color of the particle, which represents the sampled point in the second-level explosion. Figure 6 shows the comparison with a real-world fireworks display.

6 | USER INTERFACES

Pursuing the sense of reality, we integrated our fireworks simulation system into HMD virtual environments. Wearing the HTC Vive headset, users can find themselves immersed in the natural scenery and surrounding sound effects. We built a night scene of a lake with a distant city and mountains scattered under the moonlight, in order to emphasize the stunning flares of fireworks explosions.

Based on the trail renderer component in the Unity3D engine, the controller, represented by a magic wand in the main scene, can be held in hand, with which users can draw numerous curves until they submit the final sketch to trigger the fireworks simulation. Figure 7 shows a user who is wearing the HMD device to draw sketches in virtual environment.

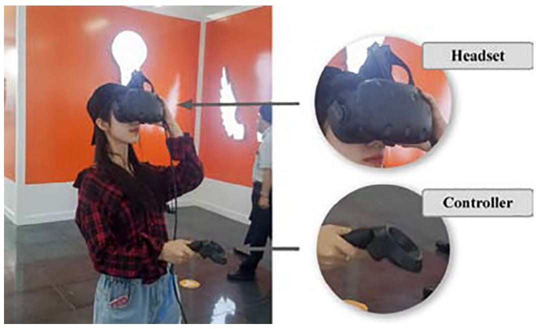


FIGURE 7 A user wears the helmet-mounted display devices to draw sketches in the virtual environment

7 | RESULTS AND DISCUSSION

In our experiments, we set the number of the selected centroids K to 70 and the gravitational acceleration g to 9.81. All our experiments were conducted on a desktop computer with 3.30 GHz Intel Core i7 CPU, 4GB of memory and an NVIDIA GeForce GTX 970 GPU.

In order to verify our point sampling method, we compare our approach to the state-of-the-art methods by Yuksel et al.[27] and the Houdini engine[28] in Figure 8. In our experiments, each model is sampled with the same number of points. Results show that our method (b) can more faithfully preserve the sharp features of input models than the other methods. Although the Yuksel et al.'s method (c) and the Houdini engine (d) can also produce results with good shapes, they may loss some important features, such as the eyes and teeth of the pumpkin cartoon model in Figure 8. Statistical results show that our method is about three times faster than the Houdini engine.[28] Note that our approach needs to have a sufficient number of point samples (about 10,000) in order to depict the underlying shape. Although the Yuksel et al.'s method can generate Poisson disk sample sets with a desired size,[27] it needs to take several seconds for an ordinary model with 10,000 faces (see Table 1), which prevents its use from real-time VR applications. By contrast, it only takes fewer than 0.1 second to perform the point sampling with our approach. The Yuksel et al.'s method[27] first samples $M(M = 3N)$ points from the input mesh and then reduces them to N by a greedy elimination algorithm. Their method's computational complexity is $O(NlogN)$ with a KD-tree while ours is only $O(N)$.

Figure 9 shows a result comparison between our method and the previous shape-constrained simulation approach by Zhao et al..[2] Our result can better preserve the sharp features of the input Stanford bunny model than that by Zhao et al.'s work,[2] which benefits from our new point sampling method based on Gaussian curvatures.

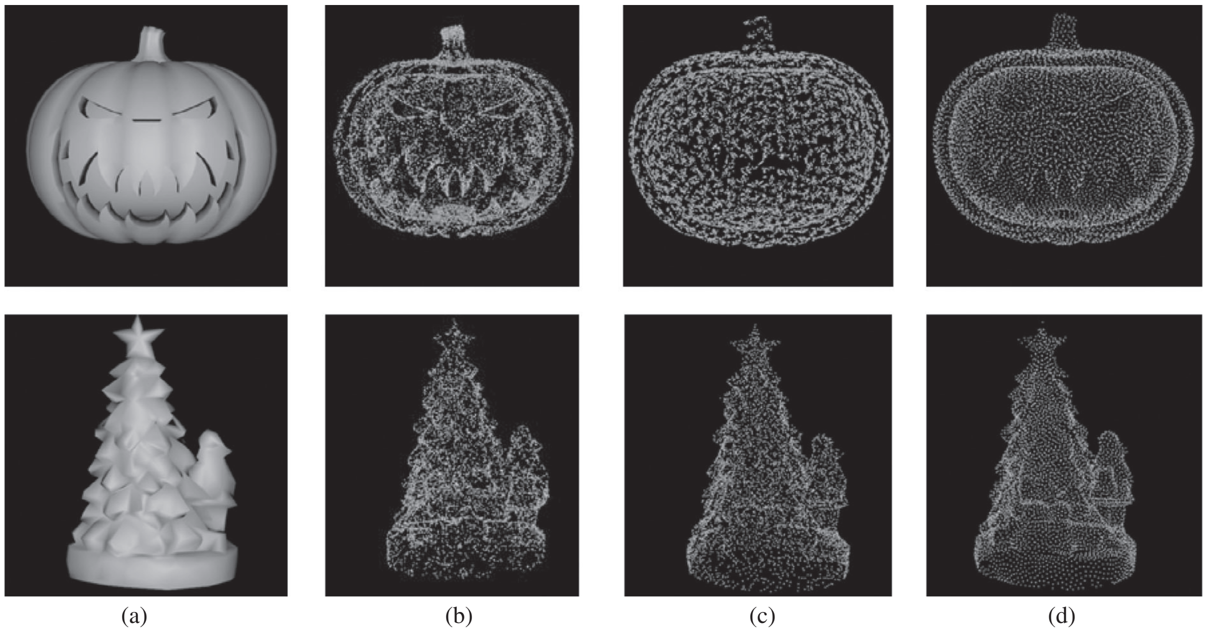


FIGURE 8 Comparisons to the state-of-the-art methods on point sampling. For the three-dimensional models in (a), our sampling method can faithfully capture the sharp features of input models (b), while the results by the Yuksel et al.'s method[27] (c) and the Houdini engine[28] (d) may loss some important features, such as the eyes and teeth of the pumpkin model

TABLE 1 Sampling time for the Yuksel et al.'s method,[27] the Houdini engine[28] and our method. As the Yuksel et al.'s method will take several seconds for an ordinary model with 10,000 faces, it is not appropriate for real-time VR applications. Time is measured in seconds.

Faces	Sampling Points	Time for Yuksel et al.'s method[27]	Time for Houdini	Time for our method
44,295	260,146	25.236	1.126	0.318
18,728	105,992	7.783	0.478	0.140
7,152	30,168	1.510	0.146	0.043
1,322	5,692	0.208	0.053	0.009

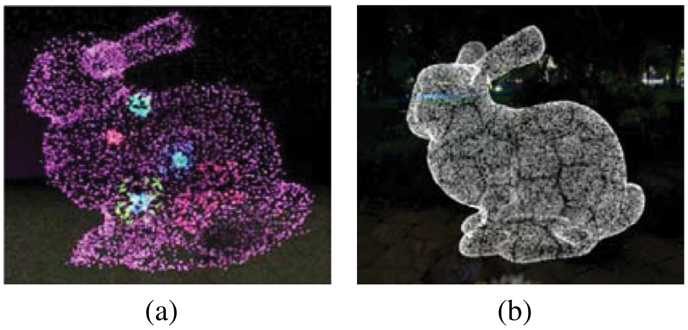


FIGURE 9 The comparison between our method and Zhao et al.'s approach:[2] a, the result by Reference 2; b, is the result by our method

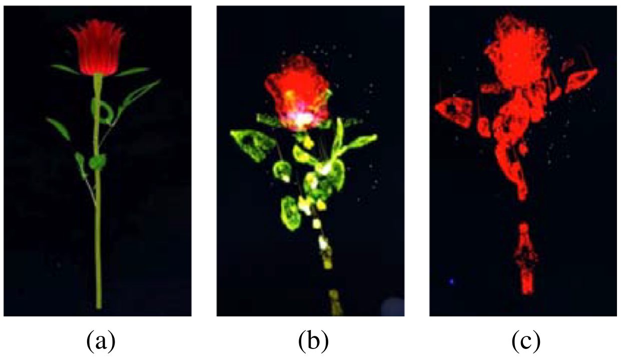


FIGURE 10 For the rose model in (a), our method can capture the texture information (b) realistically, while (c) is the result without texture information

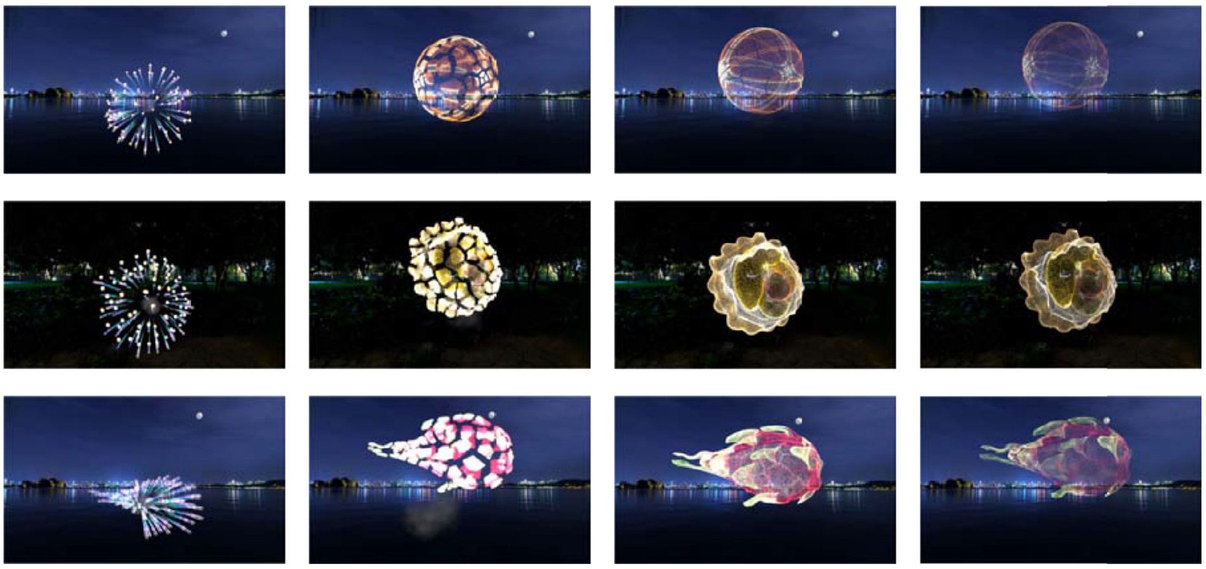


FIGURE 11 Some results by our approach. The first column shows the intermediate shapes generated after the first-level explosion. The second and third columns show the the processes of the second-level explosion, where the refined shapes are gradually formed. The fourth column shows the fading of the simulated fireworks fading after the second-level explosion

Figure 10 shows a comparison of the results by our method without and with texture information. The fireworks simulation result on the rose with its texture expression is more vivid than that with a flat color. In order to measure texture preservation quantitatively, we utilize the Euclidean distance between the color histograms of two images. Taking a rose model in Figure 10(a) as input, the distance between the rendered image of the input model (a) and our result (b) is 0.64, while the distance between (a) and the result without texture information (c) is 1.02. This example validates that our method can better preserve texture information.

Figure 11 shows the results produced by our fireworks simulation system. In this figure, the first column shows the intermediate shapes, which are determined based on the outcome of the K-means clustering algorithm on sample points. The second and third columns show the transformations from the intermediate shapes to the precise shapes of the

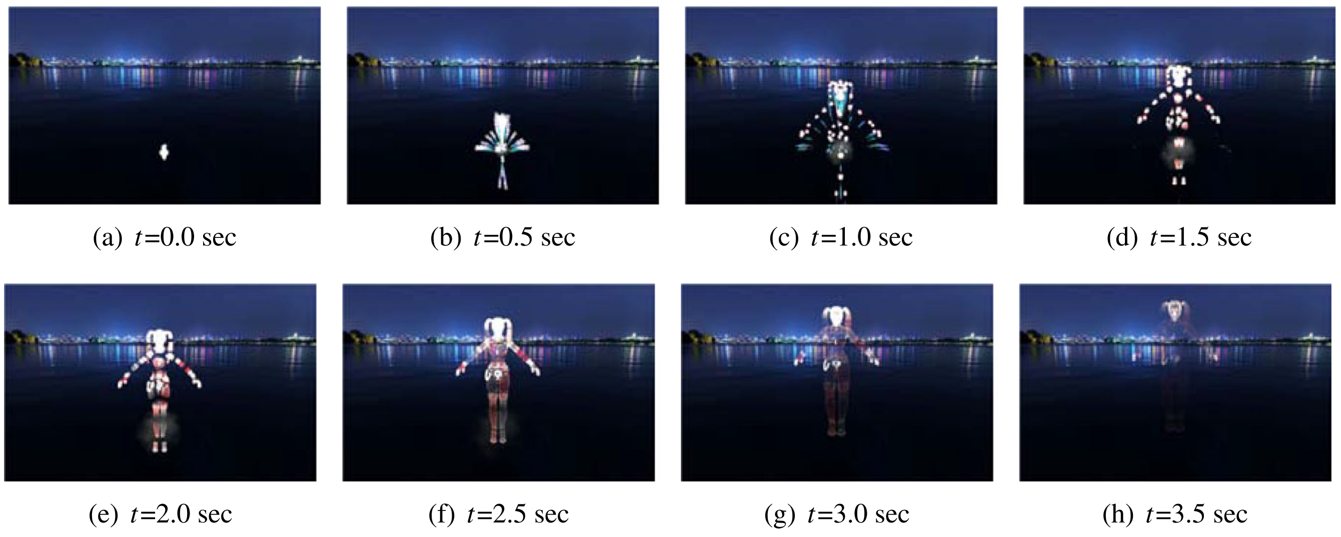


FIGURE 12 Results of the Harley Quinn's fireworks simulation by our approach

TABLE 2 Performance statistics of our fireworks simulation system. Time is measured in seconds

Model	Vertices number	Triangles number	Samples number	Sampling time	Clustering time
Basketball	10,648	18,728	145,192	0.216	2.163
Cupcake	12,389	24,448	150,809	0.253	2.513
Pitaya	16,228	28,702	210,409	0.324	3.597

retrieved 3D models. The last column shows the fading of the fireworks aloft. Please refer to the accompanying demo video for the animation results. We also used a more complicated model, Harley Quinn, to demonstrate firework simulations (Figure 12).

We also show the performance statistics of our fireworks simulation system in Table 2. For a retrieved 3D model, the time complexity of our method is linear. Our approach can generate results in real time in HMD virtual environments.

8 | CONCLUSION

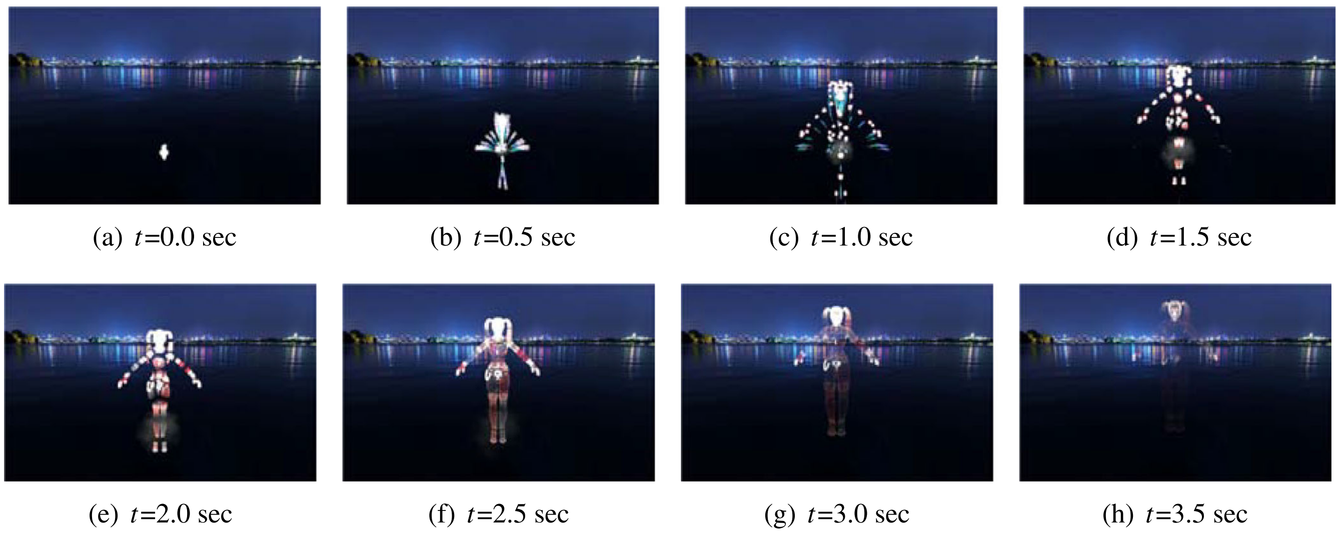
In this paper, we present a novel shape-constrained virtual fireworks simulation approach in HMD virtual environments. Its sketch-based retrieval algorithm can efficiently retrieve the 3D object whose line drawing is the most similar to the user-provided sketch from a specific viewpoint of projection. The point sampling method based on Gaussian curvature, introduced in our work, provides a balanced distribution to simulate the shape of the retrieved 3D model even with complex structures. Based on the outcome of the K-means clustering algorithm on sampled points, multilevel explosions are designed to imitate physical features of fireworks.

Although our approach can produce visually appealing results, certain limitations still exist in our current approach. It does not simulate the smoke released by fireworks explosions, which could potentially influence the degree of verisimilitude. Also, our current approach mainly focuses on one common type of fireworks, Rocket, which reduces the diversity of fireworks display.

In the future, we plan to combine our simulation approach with projection systems by exploring the concept of holographic fireworks displays. Since the scope of user interactions is strictly limited with VR controllers (e.g., HTC Vive devices), fireworks simulation with holographic techniques can be used virtually anywhere, indoor, or outdoor.

ACKNOWLEDGEMENTS

Xiaogang Jin was supported by the National Key R&D Program of China (Grant No. 2017YFB1002600), the Key Research and Development Program of Zhejiang Province (Grant No. 2018C01090), the Science and Technology Project on Preservation of Cultural Relics, Cultural Heritage Bureau of Zhejiang Province (Grant No. 2018009), and the National Natural Science Foundation of China (Grant No. 61972344).



ORCID

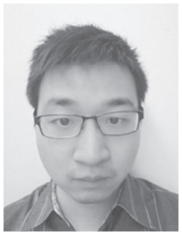
Xiaogang Jin https://orcid.org/0000-0001-7339-2920

REFERENCES

1. Reeves WT. Particle systems–a technique for modeling a class of fuzzy objects. ACM Trans Graph. 1983;2(2):91–108.
2. Zhao H, Fan R, Wang CC, Jin X, Meng Y. Fireworks controller. Comput Animat Virtual Worlds. 2009;20(2-3):185–194.
3. Steed A, Frlston S, Lopez MM, Drummond J, Pan Y, Swapp D. An 'in the wild' experiment on presence and embodiment using consumer virtual reality equipment. IEEE Trans Vis Comput Graph. 2016;22(4):1406–1414.
4. Igarashi T, Matsuoka S, Tanaka H. Teddy: A sketching interface for 3d freeform design. Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99; New York, NY, : ACM Press/Addison-Wesley Publishing Co; 1999. pp. 409-416, .
5. Lloyd SP. Least squares quantization in pcm. IEEE Trans Inf Theory. 1982;28(2):129–137.
6. Wang Y, Wang L, Deng Z, Jin X. Sketch-based shape-preserving tree animations. Comput Animat Virtual Worlds. 2018;29(3-4):e1821.
7. Wang X, Zhou L, Deng Z, Jin X. Flock morphing animation. Comput Animat Virtual Worlds. 2014;25(3-4):351–360.
8. Chen Q, Luo G, Yang T, Jin X, Deng Z. Shape-constrained flying insects animation. Comput Animat Virtual Worlds. 2019;30(3-4):e1902.
9. Yang B, Jin X. Turbulence synthesis for shape-controllable smoke animation. Comput Animat Virtual Worlds. 2014;25(3-4):465–472.
10. Hu Z, Xie H, Fukusato T, Sato T, Igarashi T. Sketch2vf: Sketch-based flow design with conditional generative adversarial network. Computer Animat Virtual Worlds. 2019;30(3-4):e1889.
11. Funkhouser TA, Min P, Kazhdan MM, et al. A search engine for 3d models. ACM Trans Graph. 2003;22(1):83–105.
12. Pu J, Ramani K. A 3D model retrieval method using 2D freehand sketches. Proceedings of the International Conference on Computational Science; Springer; 2005, pp. 343–346.
13. Saavedra JM, Bustos B. An improved histogram of edge local orientations for sketch-based image retrieval. In: Goesele M, Roth S, Kuijper A, Schiele B, Schindler K, editors. Pattern recognition. Berlin, Germany / Heidelberg: Springer, 2010; p. 432–441.
14. Yoon SM, Scherer M, Schreck T, Kuijper A. Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. Proceedings of the 18 ACM International Conference on Multimedia 2010; Oct, Firenze, Italy; 2010. pp. 193–200.
15. Chang H, Peng H, Tsai C. Cuda-accelerated rendering of fireworks in nearly ultra high definition videos. Proceedings of the 2016 IEEE 2nd International Conference on Multimedia Big Data (BigMM); Apr 2016. pp. 251–254.
16. Kim TH, Hong E, Im J, Yang D, Kim Y, Kim C-H. Visual simulation of fire-flakes synchronized with flame. Vis Comput. 2017;33(6-8):1029–1038.
17. Eitz M, Hildebrand K, Boubekeur T, Alexa M. Sketch-based shape retrieval. ACM Trans Graphics (TOG). 2012;31(4):1–10.
18. Squire DM, Müller W, Müller H, Pun T. Content-based query of image databases: inspirations from text retrieval. Pattern Recogn Lett. 2000;21(13):1193–1198.
19. Sivic J and Zisserman A. Video google: a text retrieval approach to object matching in videos. Proceedings of the 9th IEEE International Conference on Computer Vision; vol. 2, Oct 2003. pp. 1470-1477.
20. Li FF, Fergus R, Torralba A. Recognizing and learning object categories. CVPR short course; 2007.
21. Witten IH, Moffat A, Bell TC. Managing gigabytes: compressing and indexing documents and images. Hoboken, NJ: John Wiley & Sons Inc, 1994.
22. Moreton HP. Functional optimization for fair surface design. ACM Siggraph Comput Graphics. 1992;26(2):167–176.
23. Dyn N, Hormann K, Kim SJ, Levin D. Optimizing 3D triangulations using discrete curvature analysis. Math Methods Curves Surfaces. 2001;36(7):2084–2091.
24. Cohen-Steiner D, Morvan JM. Restricted delaunay triangulations and normal cycle. Proceedings of the 9th Annual Symposium on Computational Geometry, SCG '03; New York, NY: ACM; 2003. pp. 312–321.
25. Meyer M, Desbrun M, Schröder P, Barr AH. Discrete differential-geometry operators for triangulated 2-manifolds. Visualiz Math. 2002;3(8-9):35–57.
26. Váša L, Vaněček P, Prantl M, Skorkovská V, Martıćnek P, Kolingerová I. Mesh statistics for robust curvature estimation. Comput Graph Forum. 2016;35(5):271–280.
27. Yuksel C. Sample elimination for generating poisson disk sample sets. Comput Graph Forum. 2015;34(2):25–32.
28. Wikipedia contributors. Houdini (software) — Wikipedia, the free encyclopedia; 2019. [cited 2019 August 19]. Available from: https://en.wikipedia.org/w/index.php?title=Houdini_(software)&oldid=908693177.

AUTHOR BIOGRAPHIES

Xiaoyu Cui is an undergraduate majoring in digital media technology in Zhejiang University, China. She is currently doing research at the State Key Lab of CAD&CG, Zhejiang University. Her main research interests include special effects simulation and virtual reality.



Ruifan Cai received his BSc degree in computer science from Zhejiang University, China, in 2014. He is currently a PhD candidate at the State Key Lab of CAD&CG, Zhejiang University. His main research interests include geometric modeling and indoor scene reconstruction.

Xiangjun Tang is an undergraduate majoring in digital media technology in Zhejiang University, China. He is currently doing research at the State Key Lab of CAD&CG, Zhejiang University. His main research interests include special effects simulation and virtual reality.

Zhigang Deng is currently a Full Professor of Computer Science at the University of Houston (UH) and the Founding Director of the UH Computer Graphics and Interactive Media (CGIM) Lab. His research interests include computer graphics, computer animation, virtual human modeling and animation, and human computer interaction. He earned his Ph.D. in Computer Science at the Department of Computer Science at the University of Southern California in 2006. Prior that, he also completed B.S. degree in Mathematics from Xiamen University (China), and M.S. in Computer Science from Peking University (China). He is the recipient of a number of awards including ACM ICMI Ten Year Technical Impact Award, UH Teaching Excellence Award, Google Faculty Research Award, UHCS Faculty Academic Excellence Award, and NSFC Overseas and Hong Kong/Macau Young Scholars Collaborative Research Award. Besides the CASA 2014 Conference General Co-chair and SCA 2015 Conference General Co-chair, he currently serves as an Associate Editor of several journals including Computer Graphics Forum, and Computer Animation and Virtual Worlds Journal. He is a senior member of ACM and a senior member of IEEE.

Xiaogang Jin is a Professor of the State Key Lab of CAD&CG, Zhejiang University, China. He received his BSc degree in computer science in 1989 and his MSc and PhD degrees in applied mathematics in 1992 and 1995, respectively, all from Zhejiang University. His current research interests include traffic simulation, insect swarm simulation, physically based animation, cloth animation, special effects simulation, implicit surface computing, non-photo realistic rendering, computer-generated marbling, and digital geometry processing. He received an ACM Recognition of Service Award in 2015 and the Best Paper Awards from CASA 2017 and CASA 2018. He is a member of the IEEE and the ACM.

How to cite this article: Cui X, Cai R, Tang X, Deng Z, Jin X. Sketch-based shape-constrained fireworks simulation in head-mounted virtual reality. *Comput Anim Virtual Worlds*. 2020;e1920. https://doi.org/10.1002/cav.1920