

A Development Perspective of Point-Based Computer Graphics

Ajahati Mukti*

Faculty of Computer Science, University of Brawijaya, Jl. Veteran, Ketawanggede, Kec. Lowokwaru, Kota Malang, Jawa Timur 65145, Indonesia

*: All corresponding should be sent to Dr. Ajahati Mukti.

Author's Contact: Ajahati Mukti, E-mail: amukti@gmail.com

DOI: <https://doi.org/10.15354/si.21.re231>

The authors declare no competing interest.

Every object has its characteristic shape, appearance and responses to physical interactions. Computer graphics center on those three components of an object to bring them onto the computer display. With the rapid development of three dimensional (3D) printing technology, the accuracy of the focused object's geometry was put forward. Point-based graphing is a way to taking the role in rendering the huge 3D sampled data. Based on the digital geometry processing of point-sampled model, various algorithms were reviewed, and some related key techniques were compared with the potential perspective of the future work in this area was also presented.

Keywords: Point-based computer graphics, Point-sampled model, Rendering, Point-based modeling
Science Insights, 2021 October 30; Vol. 39, No. 2, pp.343-352.

© 2021 Insights Publisher. All rights reserved.



Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the [Creative Commons Attribution-NonCommercial 4.0 License](https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed by the Insights Publisher.

POINT is the simplest and most basic geometric entity. In the 1980s, Levoy and Whitted proposed to directly use points as basic primitives to draw complex three dimensional (3D) geometric models (1). At that time, it did not arouse too much attention. In recent years, with the dramatic increase in the complexity of model polygons, the advantages of point models have become more apparent. Point-based computer graphics, which uses point elements as the research object, has been attracting attention and becoming a hotspot in graphics.

The digital geometry processing flow of the point model can be summarized into three stages: point acquisition, point processing and modeling, and point model drawing. Each stage involves many detailed techniques. Based on the point model digital geometry processing flow, we in this review analyze in detail the main progress made at each stage, compare the key technologies involved, and look forward to its development trends.

Point Acquisition and Preprocessing Technology

The main source of the point model is the raw data generated by the 3D scanner, including the depth map generated by the depth camera and a large number of spatial 3D point positions obtained by the laser 3D scanner. With the development of hardware, the acquisition of point models has become easier and more convenient. Another source of point models is the existing models. Most geometric models such as polygon mesh models, implicit surfaces, etc. can be easily converted to point models (2). Since the data obtained by the 3D scanner generally cannot be directly used for the processing of the point model, it must be pre-processed.

Registration

When scanning an entity, due to the occlusion of the model itself, the limitation of the line of sight of the scanner, etc., the acquisition of complete data on the surface of the object requires multiple measurements. Then a coherent 3D model is synthesized based on the results of multiple scans. Because the point data obtained by each measurement often only covers part of the surface of the object, and may be misaligned. Therefore, in order

to obtain the point model of the complete surface of the entity, it is necessary to register these local point data.

3D data registration is mainly divided into two categories: machine registration and automatic registration. Automatic registration has nothing to do with hardware, so it has been widely used. Automatic registration is generally divided into initial registration and precise registration. The methods of initial registration mainly include: (i) Labeling method. In the measurement, some feature points are artificially pasted, and these feature points are used for positioning. The method relies on measurement and instrumentation. (ii) Feature extraction method. It extracts contour lines as a benchmark for alignment. In the early days of accurate registration, the Iterative Closest Point (ICP) algorithm was the most widely used, but the algorithm is low in efficiency. Chen and Medioni proposed to use the tangent plane of the point to approximate the point cloud, and finally summarized as finding the least square distance from the point to the tangent plane, but the calculation speed of this method was slow (3). Blais and Levine combined the reverse calibration method and random search method to improve efficiency, but it has an impact on the accuracy of registration (4). He et al. proposed Iterative Closest Line (ICL), which directly connects the points in two point clouds and finds the corresponding line segments for registration (5).

Denoising

The point-based model is usually obtained by a 3D scanner, so the sampling points contain a certain amount of noise and need to be denoised in preprocessing. What needs attention in the denoising process is how to effectively maintain the characteristics of the model while denoising. Alexa uses the Moving Least-Squares (MLS) surface to approximate the original point set model, and then moves the points to the corresponding surface to remove the noise (6). This method was not efficient. Bacciaglia and colleagues proposes a 3D Meanshift process based on all directions the denoising algorithm of heterogeneous point model, which takes point normal and curvature as characteristic components, vertex coordinate value as spatial component, finds its local mode through Meanshift process, and proposes a trilateral filter for vertex estimation based on this method (7). The denoising effect can also effectively guarantee the characteristics. Fleishman proposed a robust MLS surface reconstruction method, which can reconstruct piecewise linear smooth surfaces from noisy point models (8). This improved MLS reconstruction method does not require the topological connection information of the model, so it not only can effectively denoise, but also can effectively reconstruct sharp features from the point model. A non-local denoising method of the point model was proposed and this method expressed the local geometric features at each sampling point as "geometric gray value", and uses all possible local similar areas on the surface of the model for self-prediction to achieve the purpose of smoothing and denoising (9).

Fan et al. proposed a point model denoising algorithm based on sampling fidelity (10). The algorithm calculates the fidelity of each sampling point through the MLS surface, uses the improved bilateral filtering operator to obtain the filtering direction of each sampling point, and combines the fidelity and

characteristic to denoise the point model. Schall et al. used the non-local domain filtering algorithm in image processing to denoise the point model, and achieved good results, and it also has the ability to restore the geometric features of the point model (11). With the development of 3D scanning technology, although the accuracy of the measuring instrument is continuously improving, the error caused by the measurement and the data noise caused by the limitation of the scanner itself cannot be ignored. At present, denoising, which is the basic work of digital geometric processing of point models, is still one of the focuses of research.

Repair

When the 3D scanner samples from the surface of the object, there will be holes in the data obtained due to the self-occlusion of the model surface and the mirror material. Similarly, when the point model data is edited for large-scale deformation, the model will also have holes or cracks. Therefore, repairing the model becomes an important process after obtaining the 3D data. Buhmann used the radial basis function with global compact support to fit the discrete point set (12). Wang and Oliveira reconstructed the MLS surface to realize the repair of the hole (13). The common feature of these algorithms is that they are simple and easy to implement, but the constructed hole surface is smooth, lacking the geometric characteristics of the surface, and appears unnatural. The goal of repair is not only to repair holes and cracks, but also to reproduce as much detail as possible in the repaired area. Based on the goal of repair, Park and colleagues proposed a context-based point model repair method (14). This method is based on the principle of neighborhood similarity, and finds the sampling point block with the highest matching degree around the hole as the geometric block to fill the hole in other parts of the sampling point model. This method has achieved good results, but due to the need to use Multi-level Partition of Unity (MPU) to locally reconstruct the surface, the calculation is huge and unstable. **Figure 1** is a schematic diagram of the point model repair on the context.

For a model with color texture, not only the geometric information needs to be repaired, but also the texture information on the surface of the model. Further, Park et al. constructed a local parameterization for the area to be repaired, and then extended the image Poisson equation to the point model (14). Although this method obtains geometric and color restoration results, it needs to parameterize the point model and solve equations. Based on the idea of texture synthesis, Sharf et al. proposed a new algorithm for surface color texture restoration based on global optimization and context-based geometric restoration on the point sampling geometry (16). This method realizes the texture and geometric restoration of the point model, but it requires the user to interactively specify the texture and geometric samples on the model surface, and the base surface needs to be constructed during geometric restoration. Ji et al. proposed a geometric restoration algorithm for generating details of point models (17). The method firstly detects holes' boundary of the point model and determines its influence area. According to the sampling points in the influence area, the Radial Basis Function (RBF) surface reconstruction is performed, and the three-sided filter operator is used to estimate the geometric details of each

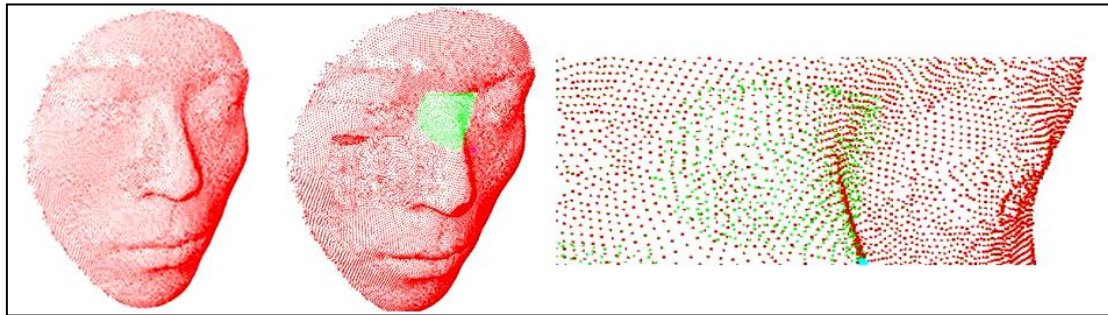


Figure 1. Schematic Diagram of the Context-Point Model Repair.
 (Modified from: Gai et al. *J Opt Soc Am A* 2019; 36:A39-A46)

sampling point. Finally, the geometric details of the holes' surface are reconstructed based on the global optimization texture synthesis method with constraints.

The repair of the point model not only makes the filled hole's surface reach a smooth geometric connection with the affected area surface at the boundary, but also makes the hole's surface maintain the consistency of geometric characteristics with the original model as a whole. Therefore, the geometric repair of the point model has always been one of the thorny problems and research focuses in surface design.

Point-Based Modeling Technology

The sampling point data after the preprocessing stage can be modeled. The goal of point-based modeling is to construct a continuous surface model from the original point cloud. There are many techniques involved in point-based modeling, such as surface reconstruction, surface simplification, geometric attribute analysis, feature extraction, and resampling. Here we focus on curved surface reconstruction and curved surface simplification.

Curved Surface Reconstruction Technology

Curved surface reconstruction refers to the process of reconstructing a 3D model from discrete scan point data. The Radial Basis Function (RBF) curved surface reconstruction method uses multi-harmonic radial basis functions to reconstruct a smooth streamlined surface from a non-uniformly distributed point cloud:

$$s(x) = p(x) + \sum_{i=1}^N \lambda_i \varphi(|x - x_i|)$$

But this method is quite computationally expensive. Yang and Qian successfully applied MLS to the surface reconstruction of point models (17). MLS curved surfaces have been widely used in point-based modeling and drawing. Zeng and colleagues used the relationship between the normal vector field of the curved surface and the gradient of the index function that characterizes the curved surface (18); the normal vector field establishes the index function gradient field, and then uses the least squares to solve the Poisson equation according to the gradient field:

$$\vec{\Delta x} = \nabla \cdot \vec{V}$$

Based on this, the index function is obtained, and finally the curved surface is reconstructed by the function by extracting its isosurface. Kolluri based on the sequence of generalization first and refinement later in MLS, and proposed a curved surface representation method for progressive points (19).

Simplification of Curved Surfaces

The point model obtained using 3D usually has a high degree of complexity. In order to make the large-scale data model conform to geometric processing and rendering, the data model must be simplified. Zhu et al. applied several grid-related clustering methods to the surface of the sampling point model, and achieved the same simplification effect as the surface of the grid model (20). However, this method cannot use a global error to control the simplification process in advance like a mesh model surface. Du et al. proposed a simplified method specifically designed for surface splats. This method fully considers the linear geometric characteristics of the surface splat, and can use a certain global error to control the formation and simplification of the surface element (21). Hu et al. proposed a new large-scale point model simplification algorithm, which is divided into two stages (22). In the first stage, the point model is divided into multiple slices similar to the plane according to the principle of geometric approximation; in the second stage, the hierarchical spatial clustering simplification is performed on the basis of the slices. Wang and colleagues proposed a point model simplification method based on geometric images and a point model simplification algorithm based on similarity (23). The point model simplification algorithm based on geometric image maps the spherical polar coordinates of the point model to the plane to construct its geometric image. By defining the simplification density as the maximum pixel search radius in the order of simplification, the curvature threshold and the simplification density are combined to realize the curvature adaptive simplification of the point model. The similarity-based point model simplification algorithm uses the normal tensor voting method to divide the point model into strong feature edge and non-strong feature edge; the MeanShift clustering method is used to perform surface area

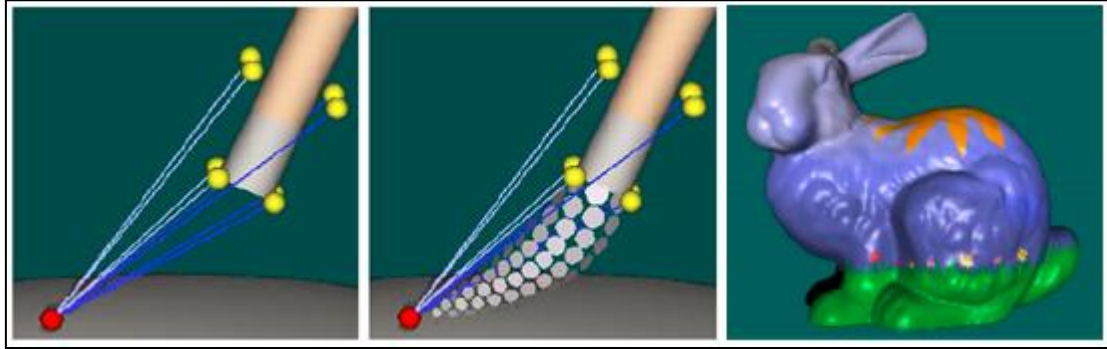


Figure 2. Brush and Coloring Effect Based on Point Model.

(Modified from: Gross. *Point Based Graphics-State of the Art and Recent Advances*. SIGGRAPH 2009. ETH Zurich.)



Figure 3. Using Boolean Operations to Construct Geometric Entities.

(Modified from: Gross. *Point Based Graphics-State of the Art and Recent Advances*. SIGGRAPH 2009. ETH Zurich.)

geometric feature similarity on the non-strong feature edge part Clustering. On this basis, simplifications are performed separately to maintain the feature boundaries and the details of the curved surface.

Point Model Editing and Geometric Modeling Technology

Point model processing is generally divided into two parts: pre-processing and post-processing. The point model pre-processing technology and point-based modeling technology introduced above are all pre-processing, and their main goal is to construct a surface continuous point model from the original point data obtained from the scanner. The post-processing of the point model is to perform further modeling processing on the point model, generally including editing, deformation, Boolean operations, etc., to eventually obtain a model that meets the

requirements of the customers.

Point Model Editing

The editing of the point model refers to the processing of the appearance characteristics such as the color and texture of the point model and the normal vector. The PointShop3D system is an important achievement of point model editing. This system is similar to the PhotoShop image processing system and can perform various operations such as texture drawing, parameterization, displacement mapping, and carving on the surface of the sampling points. PointShop3D uses effective parameterization technology to establish the mapping between the 3D simplified model and the 2D parameter domain, and directly applies the processing method of the parameter domain to the 3D simplified model. Adams et al. proposed a 3D object surface shading system based on a point model. It uses a point model to represent

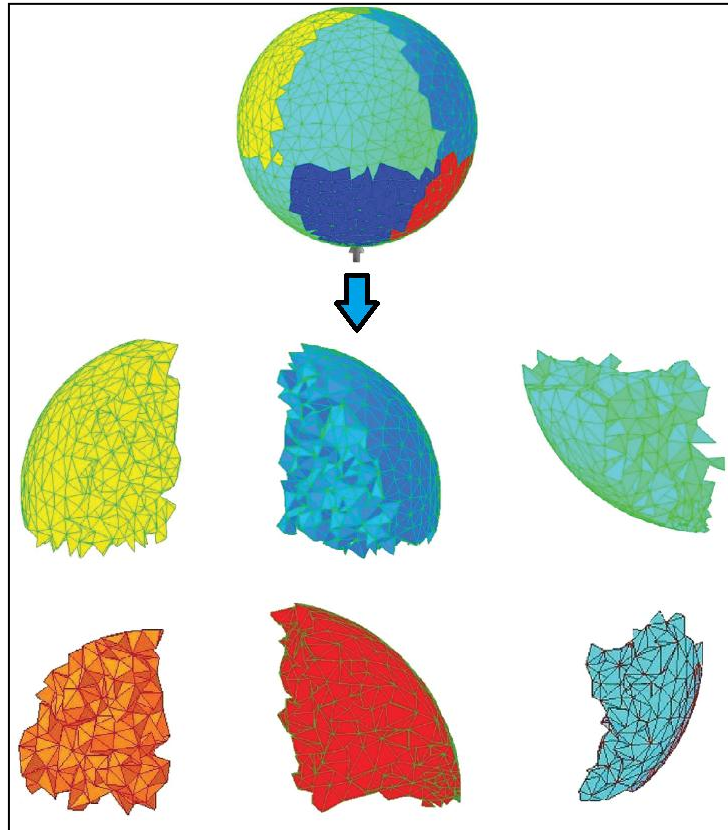


Figure 4. Object Fragmentation Simulation: Shape of Fragments from the Direct Fragmentation of a Sphere.

(Modified from: Paluszny et al. *Int J Solid Struct* 2016; 80:38-51.)

the surface of the object and the surface of the brush, which overcomes the shortcomings of the mesh model that needs to be parameterized. **Figure 2** is a drawing of the brush coloring effect of the point model.

In terms of texture synthesis, Huang et al. studied how to perform finite element geometric processing on the surface of a point model (24). They gave a simple pixel-based texture synthesis result. Wu and Kobbelt proposed a globally optimized texture synthesis algorithm that can be directly used on the 3D discrete point sampling surface (25). The synthesized texture has visual continuity and smoothness.

Geometric Modeling

The geometric modeling technology of the point model includes a wide range of categories, such as the construction of geometric entities, deformation and animation.

Constructive Solid Geometry

Constructive Solid Geometry (CSG) is a technology for constructing complex models based on Boolean operations of simple entities, that is, generating complex point models by performing Boolean operations on multiple simple point models. As shown in **Figure 3**, Boolean operation is the key technology of

solid geometric modeling. By performing merge, intersection, and difference operations on simple entities, new and more complex solid models can be generated.

Taubin realized the Boolean operation of the point model by fitting the curved surface to the surface of the point model (26). Pereira et al. proposed another method of constructing geometric entities, which does not require the reconstruction of the local curved surface of the point model surface, and directly performs Boolean operations on the point model represented by the Surfel (27). In the follow-up work, the implementation of CSG operation of the point model on the GPU greatly improved the calculation speed.

Free Deformation

The free deformation technique first generates a displacement function $d:R^3 \rightarrow R^3$, and then implements the transformation $p_i \rightarrow d(p_i)$ on each point P on the surface (i.e., the center of the mesh vertex or splat). Hu and others combined the particle system and MLS to achieve point model bending, stretching, torsion, compression and other deformation operations (28). The level-set representation based on the point model was realized and presented as the geometric deformation operation of the point model. Biancolini et al. used the cubic harmonic radial



Figure 5. Rapid Modeling of Deformable Shapes and Their Motion.

(Modified from: Adams et al. *Comput Animat* 2008; 2008:77-86.)

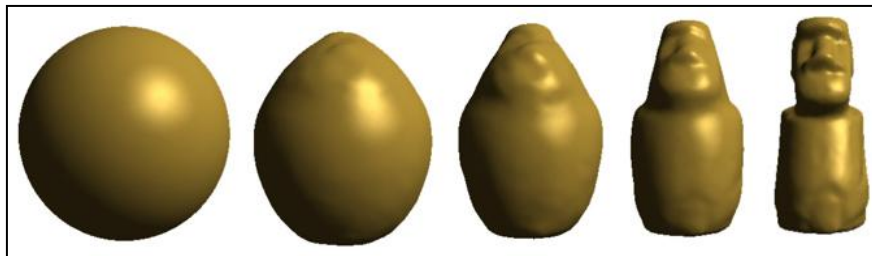


Figure 6. Point-Based Model Morphing.

(Modified from: Bao et al. *Comput Animat Virt World* 2005; 16(3-4):509-518.)

basis function to realize the real-time free deformation of the mesh model and the point model (29).

Point Model Animation

Mesh Free Method (MFM) is a new method of mechanical analysis. Researchers combined MFM and point models to propose point-based animation. For the first time, an MFM was established based on continuum mechanics to simulate elastic, plastic, and soluble objects. Based on the Kirchhoff law of thin disks and slices, Chen et al. simulated the deformation effect of the point model, and gave a point sampling thin plate spline animation (30).

Keiser et al. combined solid mechanics equations and fluid mechanics Navier-Stokes equations to extend the method of mesh-free point model animation to the simulation of fluid-related viscous materials (31). Pauly et al. studied the meshless animation of elastic and plastic materials in the process of crushing and fracture (32). Subsequently, Yang and colleagues proposed an expansion method for elastic object fragmentation (33), as shown in **Figure 4**.

Based on the overall conformal parameterization, Simonsen and Li proposed a physical simulation method of mesh-free model to simulate the elastic deformation and fracture effect of

thin slices (34). Xie and Niu discussed the collision detection problem of the point model (35). According to MLS, a classifier is defined to determine whether a sampling point is inside the point model, so as to realize the collision detection and self-intersection judgment between the point models. Akgunduz et al. discussed the collision detection and processing of the quadratic stiffness point model (36). Adams et al. proposed an interactive model deformation and key frame interpolation framework based on the Meshless Finite Element formula, as shown in **Figure 5**.

Point-Based Morphing Technology

Point model morphing technology is an important branch based on point model animation. The characteristics of the point model determine that it has obvious advantages over the morphing based on the grid model in model morphing. However, since the point model does not provide surface analytical expressions and parameterized information, it increases the difficulty of morphing. As shown in **Figure 6**, the morphing of the point model is mainly divided into physics-based morphing and geometry-based morphing.

Physics-Based Morphing

Physically-based morphing requires the establishment of related physical models, and the intermediate transition point model is solved according to the energy equation. This morphing simulates real physical phenomena to a certain extent. Wang et al. analyzed in detail model morphing and physics-based meshless methods, and proposed a physics-based point model morphing method (38). With the help of dynamic meshless method to simulate point sampling curved surface, the concept of differential geometry and continuum mechanics is used to analyze the deformation of curved surface during morphing, but this method involves the solution of complex boundary value partial differential equations.

Geometry-Based Morphing.

Geometry-based morphing obtains the intermediate transition model by geometrically transforming the source and target models, and its computational complexity is not as large as that of physics-based morphing. Hassanat analyzed the clustering point models by establishing a binary tree for the source and target models respectively, and then establish a corresponding relationship between the nodes of the two binary trees (39). However, this method will produce a lot of cracks and holes during the morphing process. Immonen proposed a point model morphing algorithm based on plane parameterization (40). Using the mesh-free parameterization method proposed, the zero genus point model is mapped to the circular parameter domain, and then the characteristic points are aligned by rotating the parameter domain and the corresponding relationship between the source and target models is established (41). The premise of the plane parameterization method is to determine a boundary on the point model, which is not an easy task. Xiao et al. proposed a robust point model morphing algorithm based on spherical parameterization (42). The source and target models are firstly parameterized on the spherical surface, and the parameterized model is embedded on the unit sphere. After adaptively aligning the corresponding feature points between the models on the spherical surface, the spherical surface is mapped to the rectangular parameter domain, and the corresponding relationship between the sampling points between the models is established based on the rectangular domain. In the morphing process, the Laplacian operator is used to solve the intermediate point model to maintain the details of the model.

Point Model Drawing Technology

After processing and modeling the point model, the results need to be displayed on the display device. The technology involved at this stage is the drawing technology of the point model. The drawing algorithm of the point model is generally divided into two types, namely the drawing algorithm based on efficiency and the drawing algorithm based on quality.

Drawing Algorithm Based on Efficiency

The efficiency-based rendering algorithm uses an opaque square to represent each point. Although efficient, it will produce graphics aliasing. Qsplat technology is a representative point drawing technology developed by Rusinkiewicz and Levoy of Stanford University (43). It uses a tree-like hierarchy to enclose the ball data structure. Each node in the tree contains the posi-

tion and radius of the ball, the normal vector at each point, the width and color value of the normal cone. The hierarchical tree is traversed recursively according to the depth-first method during drawing. For each intermediate node, first determine whether the ball is completely off the screen or completely facing away, for visibility selection. If at least a part of the child nodes of the node are visible, the projection size of the node on the screen is compared with a threshold. If it is greater than the threshold, it will continue to downward; if it is less than the threshold or the leaf node has been reached, a small area is drawn on the screen determined by the ball position and radius of the node. Since the Qsplat algorithm is based on hierarchical structure traversal, it cannot be accelerated by programmable hardware. At the same time, since Qsplat is mainly oriented to the rapid display of models with large data volumes, the data files of each model are still large despite the considerable compression rate. Rusinkiewicz and Levoy then proposed Streaming Qsplat, which is an improvement on the Qsplat method (44). This method serializes the model into a linear structure, so that the point model can be quickly transmitted in the network. The drawing is still traversed through the hierarchical tree, so it can only be realized by software. Subsequently, Srivastava et al. proposed a method to serialize the point model hierarchy into a linear array, thereby realizing the GPU hardware acceleration of Qsplat (45).

Drawing Algorithm Based on Quality

Wicke et al. proposed to use a point face (surfel) to represent a sampling point (46). This point is represented as a disk located tangentially upward, and the disks at each point overlap each other and form a compact surface of the object. This method uses a data structure called Layered Depth Cube (LDC) to store sampling points. When sampling, light rays are projected from three directions perpendicular to each other, and the intersection point with the object is recorded along each light ray, and normal vector, color and material information are recorded at each intersection point. When drawing, the occluded points are first removed by visibility preprocessing, and then the visible points are projected to the screen space, and two-dimensional image reconstruction is performed on the screen space to obtain a smooth image. The quality of the image drawn by this method has been significantly improved, but there is a problem of discontinuous coloring. Thalmann and Thalmann further discussed the point-based texture representation and anti-aliasing (47). Later, Zwicker et al. proposed a high-quality anisotropic anti-aliasing algorithm, which is also called the Surface Splatting method (48). The algorithm gives each footprint a Gaussian filter kernel with a symmetric radius, and reconstructs the continuous surface through the weighted average of each footprint data, and the quality of the drawn image is good. However, the algorithms are all implemented by software, so the running efficiency is low when drawing highly complex models. Later, Botsch and Kobbelt implemented a surface footprint method based on Gaussian filtering through graphics hardware, which significantly improved the rendering speed (49). However, since the current graphics hardware is optimized specifically for triangle-based rendering, in terms of rendering performance, hardware-accelerated points are not as good as hardware-accelerated

triangles to a certain extent.

Point-Based Rendering Processing Platform

In the early days, attention was mainly focused on the algorithmic research of each stage of the point model digital geometry processing flow. Later, I began to think about how to construct a point-based 3D geometry processing application platform to better display and utilize existing results. Pauly et al. developed the PointShop3D system, which is interactive point sampling geometry processing system software (50). The platform software is completely open source and available to other researchers for further research and development. The latest research results of researchers can be added to the platform in the form of plug-ins, so PointShope3 has great versatility and scalability. But PointShop3D cannot realize the processing and drawing of large-scale point model data. Qsplat gives a large-scale point model drawing method, but it is not a general software platform.

Perspectives

Point-based computer graphics has developed rapidly. In the future, there are still many valuable research work that we need to further in-depth: (i) Point-based natural scene modeling. According to the characteristics of the point model, there is no need

to establish a topological relationship between points. You can consider using points to construct some irregular natural scenery or phenomena, such as clouds, rain, and water waves. (ii) Animation based on the point model. The mesh-free method is more advantageous than the finite element method in solving unconventional problems such as large deformation and fracture, and the point model is the most natural representation of the mesh-free model. With the development of the digital entertainment field, the application of mesh-free technology for point model deformation will be an important and meaningful research direction. (iii) Simplification, compression and transmission of point models. The huge amount of data of the point model poses a challenge to the processing capabilities of the current graphics engine. How to effectively simplify the model is a valuable work. On the other hand, with the development of the Internet, the dot model may be applied as a media resource, so it is also a meaningful work to consider the transmission and compression method of the dot model in the network. (iv) Large-scale point model drawing based on GPU. With the development of programmable graphics hardware, the speed and quality of graphics processing have been greatly improved. The application of GPU programming to the drawing of large-scale point models is an inevitable trend of development. ■

References

1. Levoy M, Whitted T. The Use of Points as Display Primitives. TR 85-022. Department of Computer Science, The University of North Carolina at Chapel Hill, 1985.
2. Yngve G, Turk G. Roust creation of implicit surfaces from polygonal meshes. *Vis Comput Graph IEEE Transact* 2002; 8:346-359. DOI: <https://doi.org/10.1109/TVCG.2002.1044520>
3. Chen Y, Medioni G. Object modelling by registration of multiple range images. *Image Vision Comput* 1992; 10(3):145-155. DOI: [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C)
4. Blais G, Levine MD. Registering multiview range data to create 3D computer objects. *IEEE Trans Pattern Anal Mach Intell* 1995; 17(8):820-824. DOI: <https://doi.org/10.1109/34.400574>
5. He Y, Liang B, Yang J, Li S, He J. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. *Sensors (Basel)* 2017; 17(8):1862. DOI: <https://doi.org/10.3390/s17081862>
6. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Computing and rendering point set surfaces. *IEEE Trans Visual Comp Graph* 2003; 9(1):3-15. DOI: <https://doi.org/10.1109/TVCG.2003.1175093>
7. Bacciaglia A, Ceruti A, Liverani A. Surface smoothing for topological optimized 3D models. *Struct Multidisc Optim* 2021; In press. DOI: <https://doi.org/10.1007/s00158-021-03027-6>
8. Fleishman S, Cohen-Or D, Silva CT. Robust moving least-squares fitting with sharp features. *ACM Trans Graph* 2005; 24:544-552. DOI: <https://doi.org/10.1145/1186822.1073227>
9. Jin Y, Jiang W, Shao J, Lu J. An improved image denoising model based on nonlocal means filter. *Math Prob Eng* 2018; 2018: 8593934. DOI: <https://doi.org/10.1155/2018/8593934>
10. Fan L, Zhang F, Fan H, Zhang C. Brief review of image denoising techniques. *Vis Comput Ind Biomed Art* 2019; 2:7. DOI: <https://doi.org/10.1186/s42492-019-0016-7>
11. Schall O, Belyaev A, Seidel HP. Adaptive feature-preserving non-local denoising of static and time-varying range data. *Comput Aid Design* 2008; 40(6):701-707. DOI: <https://doi.org/10.1016/j.cad.2008.01.011>
12. Buhmann MD. A new class of radial basis functions with compact support. *Math Comput* 2001; 70. DOI: <https://doi.org/10.1016/j.cad.2008.01.011>

- <https://doi.org/10.1090/S0025-5718-00-01251-5>
13. Wang J, Oliveira M. Filling holes on locally smooth surfaces reconstructed from point clouds. *Imag Vis Comput* 2007; 25:103-113. DOI: <https://doi.org/10.1016/j.imavis.2005.12.006>
 14. Park S, Guo X, Shin H, Qin H. Shape and appearance repair for incomplete point surfaces. *Proceed IEEE Int Conf Comput Vis* 2005; 2:1260-1267. DOI: <https://doi.org/10.1109/ICCV.2005.218>
 15. Sharf A, Alexa M, Cohen-Or D. Context-based surface completion. *ACM Trans Graph* 2004; 23:878-887. DOI: <https://doi.org/10.1145/1015706.1015814>
 16. Ji C, Li Y, Fan J, Lan S. A novel simplification method for 3D geometric point cloud based on the importance of point. *IEEE Access* 2019; pp.1-1. DOI: <https://doi.org/10.1109/ACCESS.2019.2939684>
 17. Yang P, Qian X. Direct computing of surface curvatures for point-set surfaces. *Eurographics Symposium on Point-Based Graphics* 2007; pp.29-36. DOI: <https://doi.org/10.2312/SPBG/SPBG07/029-036>
 18. Zeng Q, Qin Y, Chang W, Luo X. Correlating and evaluating the functionality-related properties with surface texture parameters and specific characteristics of machined components. *Int J Mechan Sci* 2018; 149:62-72. DOI: <https://doi.org/10.1016/j.ijmecsci.2018.09.044>
 19. Kolluri R. Provably good moving least squares. *ACM Trans Algor* 2008; 4. DOI: <https://doi.org/10.1145/1361192.1361195>
 20. Zhu L, Kukko A, Virtanen J-P, Hyyppä J, Kaartinen H, Hyyppä H, Turppa T. Multisource point clouds, point simplification and surface reconstruction. *Remote Sens* 2019; 11(22):2659. DOI: <https://doi.org/10.3390/rs11222659>
 21. Du X, Yin B, Kong D. Adaptive out-of-core simplification of large point clouds. *2007 IEEE International Conference on Multimedia and Expo* 2007; pp.1439-1442. DOI: <https://doi.org/10.1109/ICME.2007.4284931>
 22. Hu Q, Feng D, Zhang H, Yao Y, Aburaia M, Lammer H. Oriented to multi-branched structure unsupported 3D printing method research. *Materials* 2020; 13(9):2023. DOI: <https://doi.org/10.3390/ma13092023>
 23. Wang Y, Zheng J, Wang H. Fast mesh simplification method for three-dimensional geometric models with feature-preserving efficiency. *Sci Program* 2019; 2019:4926190. DOI: <https://doi.org/10.1155/2019/4926190>
 24. Huang Z, Gong G, Han L, Song Z. Geometry surface simplification and reconstruction of FEM point cloud. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)* 2010; pp.V5-357-V5-360. DOI: <https://doi.org/10.1109/ICCASM.2010.5620131>
 25. Wu J, Kobbelt L. Optimized sub-sampling of point sets for surface splatting. *Comput Graph Forum*. 2004; 23:643-652. DOI: <https://doi.org/10.1111/j.1467-8659.2004.00796.x>
 26. Taubin G. An improved algorithm for algebraic curve and surface fitting. *1993 (4th) International Conference on Computer Vision*, 1993; pp.658-665. DOI: <https://doi.org/10.1109/ICCV.1993.378149>
 27. Pereira A, Arruda M, Miranda A, Lira W, Martha L. Boolean operations on multi-region solids for mesh generation. *Eng Comput* 2012; 28:1-15. DOI: <https://doi.org/10.1007/s00366-011-0228-8>
 28. Hu Y, Fang Y, Ge Z, Qu Z, Zhu Y, Pradhana A, Jiang C. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans Graph* 2018; 37:1-14. DOI: <https://doi.org/10.1145/3197517.3201293>
 29. Biancolini ME, Chiappa A, Cella U, Costa E, Groth C, Porziani S. Radial basis functions mesh morphing. In: Krzhizhanovskaya V. et al. (eds) *Computational Science – ICCS 2020*. ICCS 2020. Lecture Notes in Computer Science, vol 12142. Springer, Cham. 2020. DOI: https://doi.org/10.1007/978-3-030-50433-5_23
 30. Chen Y, Zhao J, Deng Q, Duan F. 3D craniofacial registration using thin-plate spline transform and cylindrical surface projection. *PLoS One* 2017; 12(10): e0185567. DOI: <https://doi.org/10.1371/journal.pone.0185567>
 31. Keiser R, Adams B, Gasser D, Bazzi P, Dutre P, Gross M. A unified Lagrangian approach to solid-fluid animation. *Proceedings Eurographics/IEEE VGTC Symposium Point-Based Graphics*, 2005; pp.125-148. DOI: <https://doi.org/10.1109/PBG.2005.194073>
 32. Pauly M, Keiser R, Adams B, Dutré P, Gross M, Guibas LJ. Meshless animation of fracturing solids. *ACM Trans Graph* 2005; 24(3):957-964. DOI: <https://doi.org/10.1145/1186822.1073296>
 33. Yang P, Liu Y, Zhang X, Zhou X, Zhao Y. Simulation of fragmentation with material point method based on Gurson model and random failure. *CMES - Comput Model Eng Sci* 2012; 85: 207-237. DOI: <https://doi.org/10.3970/cmesci.2012.085.207>
 34. Simonsen BC, Li S. Mesh-free simulation of ductile fracture. *Int J Num Meth Eng* 2004; 60. 1425-1450. DOI: <https://doi.org/10.1002/nme.1009>
 35. Xie M, Niu X. A 3D roaming and collision detection algorithm applicable for massive spatial data. *PLoS One* 2020; 15(2):e0229038. DOI: <https://doi.org/10.1371/journal.pone.0229038>
 36. Akgunduz A, Banerjee P, Mehrotra S. A linear programming solution for exact collision detection. *ASME J Comput Inf Sci Eng* 2005; 5(1):48-55. DOI: <https://doi.org/10.1115/1.1846053>
 37. Adams B, Ovsjanikov M, Wand M, Seidel HP, Guibas LJ. Meshless modeling of deformable shapes and their motion. *Comput Animat* 2008; 2008:77-86. DOI: <https://doi.org/10.2312/SCA/SCA08/077-086>
 38. Wang S, Xiang N, Xia Y, You L, Zhang J. Real-time surface manipulation with C1 continuity through simple and efficient physics-based deformations. *Vis Comput* 2021; 37:2741-2753. DOI: <https://doi.org/10.1007/s00371-021-02169-4>
 39. Hassanat ABA. Two-point-based binary search trees for accelerating big data classification using KNN. *PLoS One* 2018; 13(11):e0207772. DOI: <https://doi.org/10.1371/journal.pone.0207772>
 40. Immonen E. A parametric morphing method for gen-

- erating structured meshes for marine free surface flow applications with plane symmetry. *Journal of Computational Design and Engineering* 2019; 6(3):348-353. DOI: <https://doi.org/10.1016/j.jcde.2018.11.002>
41. Duan L, Luo X, Ruan L, Gu M. Novel method of boundary-free mesh parameterization. *PLoS One* 2019; 14(6):e0217537. DOI: <https://doi.org/10.1371/journal.pone.0217537>
 42. Xiao C, Zheng W, Peng Q, Forrest A. (2004). Robust morphing of point-sampled geometry. *J Vis Comput Animat* 2004; 15:201-210. DOI: <https://doi.org/10.1002/cav.22>
 43. Rusinkiewicz S, Levoy M. QSplat: A Multiresolution Point Rendering System for Large Meshes. *Proceedings of SIGGRAPH 2001*. DOI: <https://doi.org/10.1145/344779.344940>
 44. Rusinkiewicz S, Levoy M. Streaming QSplat: A viewer for networked visualization of large, dense models. *proceedings of the 2001 symposium on interactive 3D graphics*, 2001. DOI: <https://doi.org/10.1145/364338.364350>
 45. Srivastava DJ, Vosegaard T, Massiot D, Grandinetti PJ. Core Scientific Dataset Model: A lightweight and portable model and file format for multi-dimensional scientific data. *PLoS One* 2020; 15(1):e0225953. DOI: <https://doi.org/10.1371/journal.pone.0225953>
 46. Wicke M, Teschner M, Gross M. CSG tree rendering for point-sampled objects. *12th Pacific Conference on Computer Graphics and Applications*, 2004. PG 2004. *Proceedings.*, 2004; pp.160-168, DOI: <https://doi.org/10.1109/PCCGA.2004.1348346>
 47. Thalmann NM, Thalmann D. Transparency, texture, shadows, and anti-aliasing. In: *Computer Animation. Computer Science Workbench*. Springer, Tokyo. 1990. Print ISBN: 978-4-431-68107-6. DOI: https://doi.org/10.1007/978-4-431-68105-2_9
 48. Zwicker M, Pfister H, Baar JV, Gross MH. Surface splatting. *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive techniques*, 2001.
 49. Botsch M, Kobbelt L. High-quality point-based rendering on modern GPUs. 2003; pp.335-343. DOI: <https://doi.org/10.1109/PCCGA.2003.1238275>
 50. Pauly M, Keiser R, Kobbelt LP, Gross M. Shape modeling with point-sampled geometry. *ACM Trans Graph* 2003; 22(3):641-650. DOI: <https://doi.org/10.1145/882262.882319>

Received: June 10, 2021 | Revised: September 21, 2021 | Accepted: September 27, 2021
