Point-Based Computer Graphics

Hanspeter Pfister
Mitsubishi Electric Research Laboratories

Markus Gross
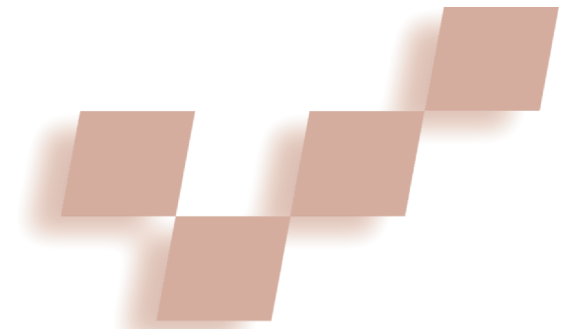*ETH Zurich*

Point primitives have experienced a major renaissance recently, and considerable research has focused on the efficient representation, modeling, processing, and rendering of point-sampled geometry. There are two main reasons for this new interest in points. On one hand, we have witnessed a dramatic increase in the polygonal complexity of computer graphics models. The overhead of managing, processing, and manipulating very large polygonal-mesh connectivity information has led many leading researchers to question the future utility of polygons as the fundamental graphics primitive. On the other hand, modern 3D digital photography and 3D scanning systems acquire both the geometry and appearance of complex, real-world objects. These techniques generate huge volumes of point samples, which constitute discrete building blocks of 3D object geometry and appearance—much as pixels are the digital elements for images. This special issue of *IEEE Computer Graphics and Applications* highlights recent advances in point-based computer graphics. It brings together a range of different topics from the main areas of our field: acquisition, modeling, and rendering.

Acquisition

The first article, "A Simple Approach for Point-Based Object Capturing and Rendering" by Sainz et al., describes a simple acquisition system for real-world objects. Their system uses a video camera and turntable to capture images of the object. Using the visual hull and voxel carving techniques, they generate a cloud of points on the object surface. In a postprocessing step, the system smooths the points and their normals, then renders the object interactively using a multiresolution representation. The article provides an overview of the whole content creation pipeline and demonstrates the effectiveness of using point primitives for capturing, processing, and rendering.

In "Image-Based Rendering of Range Data with Estimated Depth Uncertainty," Hofsetz et al. demonstrate the close connection of point-based graphics and image-based rendering. They present a method for automatically extracting per-pixel depth maps from a sequence of calibrated images of a static scene. From the epipolar geometry they compute a region of depth uncertainty around each depth pixel. The points with depth uncertainty are represented by elliptical Gaussian kernels and rendered with point splatting. Their results demonstrate that weighing the contribution of each point splat with uncertainty improves the reconstruction's visual quality. The article shows that a point-based representation is more flexible and robust in the presence of noise because it does not require making specific assumptions about the underlying scene geometry and topology.

Processing and modeling

"Scalar-Function-Driven Editing on Point Set Surfaces" by Guo, Hua, and Qin moves from acquisition to the next step in the content creation pipeline—editing and modeling of point-based surfaces. Starting with a set of unstructured point samples, they reconstruct a surface distance field from the point cloud. The surface can then be changed locally or globally using free-form deformations and level-set editing tools. Dynamic surface resampling ensures that the geometry is preserved without worrying about point connectivity. Their system also incorporates a haptic interface for force feedback during surface editing. This comprehensive system demonstrates how easily you can edit, deform, and process point-based models.

A short article by Jones, Durand, and Zwicker, "Normal Improvement for Point Rendering," addresses the important issue of postprocessing noisy point data from scanners. They use a feature-preserving bilateral filter to denoise the point normals. The filter does not modify sample positions, thereby preserving the original features as much as possible, while the smoothed normals greatly improve the rendering quality. This article clearly demonstrates another trait of point-based graphics: simplicity.

Rendering

Point-based graphics started by focusing on rendering point-based models, which remains an important and

popular topic. "Flexible Point-Based Rendering on Mobile Devices" by Duguet and Drettakis demonstrates that relatively complex objects and scenes can be efficiently rendered on mobile devices using points. They use a hierarchy of packed points based on a recursive grid data structure to deal with the limited memory and display resolution. Their rendering algorithm is efficient and includes shadow map computation. This article addresses many of the important issues of mobile graphics. This line of research may eventually lead to low-power hardware implementations of rendering pipelines that are completely based on point primitives.

Concluding this special issue is an article by Hopf, Luttenberger, and Ertl, "Hierarchical Splatting of Scattered 4D Data," that presents an efficient rendering method for large time-dependent point data. They use a hierarchical data structure that is constructed using principal component analysis clustering. The out-of-core rendering algorithm uses quantized relative coordinates, lossless compression, and fast approximate sorting of semitransparent clusters. Their system can render millions of points at interactive speeds on current graphics hardware. The results include some stunning images and real-time animations of large astronomical data.

The future

In the past four years, point-based graphics has grown tremendously. By the time this issue is published, the first Symposium on Point-Based Graphics in Zurich will have concluded. The large number of submissions (44 papers) for this first-time conference shows the huge interest in this young and exciting field. Many opportunities exist for future research and innovations. For example, points hold a lot of promise for free-viewpoint 3D video and 3D TV, physics-based modeling, compression and transmission, novel rendering architectures, special effects, and new hybrid representations for digital media. We hope that this special issue of *CG&A* stimulates even more ideas in this rapidly moving field. ∎

Hanspeter Pfister *is associate director and senior research scientist at Mitsubishi Electric Research Laboratories (MERL) in Cambridge, Massachusetts. His research interests include computer graphics, computer vision, scientific visualization, and graphics architectures. Pfister received an MS in electrical engineering from the Swiss Federal Institute of Technology (ETH) Zurich and a PhD in computer science from the State University of New York at Stony Brook. He is associate editor of* IEEE Transactions on Visualization and Computer Graphics *and chair of the IEEE Technical Committee on Visualization and Computer Graphics. He is a senior member of the IEEE and a member of the IEEE Computer Society, ACM, ACM Siggraph, and the Eurographics Association.*

Markus Gross *is a professor of computer science and director of the Computer Graphics Laboratory of the Swiss Federal Institute of Technology (ETH) in Zurich. His research interests include point-based graphics, physics-based modeling, multiresolution analysis, and virtual reality. Gross received an MS in electrical and computer engineering and a PhD in computer graphics and image analysis, both from the University of Saarbrucken, Germany. He has been widely published and often lectures on computer graphics and scientific visualization. He wrote* Visual Computing *(Springer, 1994). He is a former associate editor in chief of* IEEE Computer Graphics and Applications *and will be chair of the papers committee for Siggraph 2005. Gross is a senior member of the IEEE and a member of the IEEE Computer Society, ACM, ACM Siggraph, and the Eurographics Association.*

Readers may contact Hanspeter Pfister at MERL, 201 Broadway, Cambridge, MA 02139; pfister@merl.com. Contact Markus Gross at the Institute of Scientific Computing, Dept. of Computer Science, ETH, Zurich, Switzerland; grossm@inf.ethz.ch.

Correction

In the May/June issue of *IEEE CG&A*, two equations in the article by Liu, Zhang, and Shan, "Image-Based Surface Detail Transfer," were printed with missing elements.

The equation on the bottom right hand side of page 30 should have appeared as follows:

$$||\,\overline{\mathbf{n}}(P_1) - \overline{\mathbf{n}}(P_2)\,||\leq\delta$$

The equation numbered 11 on page 31 should have appeared as follows:

$$I'_2(\mathbf{p}) \approx \frac{I_1(\mathbf{p})}{\overline{I}_1(\mathbf{p})}\overline{I}_2(\mathbf{p}) \equiv \left(1+\frac{I_1(\mathbf{p})-\overline{I}_1(\mathbf{p})}{\overline{I}_1(\mathbf{p})}\right)\overline{I}_2(\mathbf{p})$$