Object-Space Point Blending and Splatting

Renato Pajarola
Information and Computer Science
University of California Irvine
pajarola@acm.org

Miguel Sainz
Electrical Engineering and Computer Science
University of California Irvine
msainz@ece.uci.edu

Patrick Guidotti
Mathematics
University of California Irvine
gpatrick@math.uci.edu

Abstract

We present a novel point-based rendering approach based on object-space point interpolation. We introduce the concept of a transformation-invariant covariance matrix of a set of points to efficiently determine splat sizes in a multiresolution hierarchy. We analyze continuous point interpolation in object-space, and define a new class of parametrized blending kernels to achieve smooth blending. Furthermore, we present a hardware accelerated rendering algorithm based on α-texture mapping and α-blending.

1. Introduction

Point-based surface representations have recently been established as viable graphics rendering primitives [Gross 2001]. In particular they lead to compact multiresolution models that can provide efficient level-of-detail (LOD) rendering. Recent efforts in point-based rendering (PBR) have focused on compact and very efficient multiresolution models [Rusinkiewicz and Levoy 2000, Botsch et al. 2002], as well as on high-quality texture sampling and rendering [Zwicker et al. 2001, Ren et al. 2002].

The major challenges are to achieve smooth and continuous surface interpolation from irregular distributed point samples, support correct visibility as well as provide efficient rendering. We propose a novel point blending and rendering technique that is based on the direct interpolation between point samples in 3D. In contrast to previous methods, we define the blending of surface points as a weighted interpolation in object-space. We analyze the smooth interpolation between points in object-space and define a new class of parametrized blending kernels. We also provide an efficient technique to calculate splat sizes in a multiresolution point hierarchy. Furthermore, our approach exploits hardware acceleration. An example rendering result of our approach for the textured David statue of Michelangelo is shown in Figure 1.

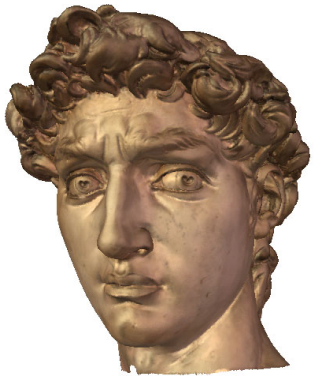

FIGURE 1. The head of Michelangelo's David statue rendered with τ=16 pixels screen tolerance, at 1/4 of the full resolution (510827 out of 2000606 points).

2. Exposition

In this project we consider blending and rendering techniques for surfaces represented as dense sets of point-samples organized in a space-partitioning multiresolution hierarchy (octree). The data set consists of surface point samples (*surfels*) s with attributes for spatial coordinates $\boldsymbol{p}$, normal orientation $\boldsymbol{n}$, surface color $\boldsymbol{c}$ and spatial extent as elliptical disk e centered at $\boldsymbol{p}$ and perpendicular to $\boldsymbol{n}$. These surfels cover the sampled object in object-space without holes.

In the 4-dimensional homogeneous space of points $\boldsymbol{p}_i \in \mathbf{R}^3$, thus $\boldsymbol{p}'^{\mathrm{T}}_i = (\boldsymbol{p}^{\mathrm{T}}_i, 1)$, we define the transformation-invariant *generic homogeneous covariance matrix* $\overline{M}'$ of points $\boldsymbol{p}_i$ by $M' = \sum \boldsymbol{p}'_i \cdot \boldsymbol{p}'^{\mathrm{T}}_i$, and we can express the homogeneous covariance matrix in any local coordinate system given by translation T and rotation R as $M' = n^{-1} R \cdot T \cdot M' \cdot T^{\mathrm{T}} \cdot R^{\mathrm{T}}$. Given two different sets of points $P = \{\boldsymbol{p}_1 .. \boldsymbol{p}_n\}$ and $Q = \{\boldsymbol{q}_1 .. \boldsymbol{q}_m\}$ as well as $M'_P = \sum \boldsymbol{p}'_i \cdot \boldsymbol{p}'^{\mathrm{T}}_i$ and $M'_Q = \sum \boldsymbol{q}'_i \cdot \boldsymbol{q}'^{\mathrm{T}}_i$, the combined generic covariance matrix $\overline{M}'$ of the union $P \cup Q$ is then given by $M' = M'_P + M'_Q$ without the need to recalculate the sum of tensor products of all points $P \cup Q$. This concept of a *generic homogeneous covariance matrix* is used to efficiently calculate tangential bounding ellipses in the hierarchical multiresolution point representation.

We interpret the interpolation of surface parameters such as color in object-space between surfels $s_1 \ldots s_n$ as a weighted sum. In fact we interpolate the color $\boldsymbol{c}_{\hat{\boldsymbol{p}}}$ of a pixel $\hat{\boldsymbol{p}}$ (projection of a point $\boldsymbol{p}$) from surfels s_i whose elliptical disks e_i intersect $\hat{\boldsymbol{p}}$ as $\boldsymbol{c}_{\hat{\boldsymbol{p}}} = \sum \Psi_i(\hat{\boldsymbol{p}}) \cdot \boldsymbol{c}_i$ using conforming blending functions Ψ_i with *local support* over the elliptical disk e_i. Using rotation-symmetric blending kernels $\psi(r) = \mathrm{e}^{(-a \cdot (r/b)^n)/(1-(r/b)^n)}$ we can define a conforming blending function Ψ_i for the surfel s_i as a normalization $\Psi_i(\hat{\boldsymbol{p}}) = \psi_i(\hat{\boldsymbol{p}}) / (\psi_i(\hat{\boldsymbol{p}}) + \sum \psi_j(\hat{\boldsymbol{p}}))$ given its overlapping adjacent surfels s_j. Based on this blending and normalization we developed an efficient rendering algorithm that generates the correct pixel colors $\boldsymbol{c}_{\hat{\boldsymbol{p}}} = (\sum \psi_i(\hat{\boldsymbol{p}}) \cdot \boldsymbol{c}_i) / (\sum \psi_i(\hat{\boldsymbol{p}}))$ by rendering an α-textured polygon for each surfel s_i which represents its blending kernel ψ_i in object-space to get the intermediate result $\boldsymbol{c}'_{\hat{\boldsymbol{p}}} = \sum \psi_i(\hat{\boldsymbol{p}}) \cdot \boldsymbol{c}_i$ and by a post-process per-pixel normalization with the accumulated pixel blending weight $\sum \psi_i(\hat{\boldsymbol{p}})$.

Table 1 shows experimental results of our point rendering algorithm on a 1.5GHz Pentium4 CPU and nVIDIA GeFroce4 Ti4600 GPU. It lists the number of visible splats, the time for LOD selection, the time for blending and visibility splatting, and the time for color normalization all given in seconds per frame.

Model	Tol. τ	#Splats	LOD	Splatting	Normalization	Total
David	0.0%	904121	0.315s	1.386s	0.0002s	1.710s
	0.01%	454656	0.269s	0.791s	0.00019s	1.061s
	0.12%	340537	0.195s	0.612s	0.0002s	0.808s

TABLE 1. Rendering performance is given for each task in seconds used per frame. Tolerance τ in percent of viewport.

References

M. Botsch, A. Wiratanaya, and L. Kobbelt. Efficient high quality rendering of point sampled geometry. In *Proceedings Eurographics Workshop on Rendering*, pages –, 2002.

M. Gross. Are points the better graphics primitives? Computer Graphics Forum 20(3), 2001. Plenary Talk Eurographics 2001.

L. Ren, H. Pfister, and M. Zwicker. Object space EWA surface splatting: A hardware accelerated approach to high quality point rendering. In *Proceedings EUROGRAPHICS 2002*, pages –, 2002. also in Computer Graphics Forum 21(3).

S. Rusinkiewicz and M. Levoy. Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings SIGGRAPH 2000*, pages 343–352. ACM SIGGRAPH, 2000.

M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Surface splatting. In *Proceedings SIGGRAPH 2001*, pages 371–378. ACM SIGGRAPH, 2001.