**Point-Sampled Shape Representations**

**Why Point-Based Graphics?**
- simplicity
- generality
- flexibility
- efficiency?
- point vs. splat-approximation
- GPU processing
- quality?

**Overview**
- point-based representations
  - shape approximation
  - surface topology
- octree point clouds
- optimized splat subsampling

**Point-Based Approximation**
- what is the approximation power?
  - error = $O(h^2)$
  - polygons

- error = $O(h)$
  - points
Point-Based Approximation

- what is the approximation power?

\[ \text{error} = O(h^2) \]

Point-Based Approximation

- what is the required precision?

\[ \text{number} = O(\text{surface area}) \]

Point-Based Approximation

- what is the required precision?

- Points: precision = \( O(\text{sampling density}) \)
  - number = \( O(\text{surface area}) \)

- Splits: precision = \( O(\text{sampling density}^2) \)
  - number = \( O(\text{surface curvature}) \)

Consequences

- pure point-based representations
  - insufficient object space approximation power
  - screen-space dependent sampling resolution
  - screen-space dependent sampling resolution
  - forward mapping techniques independent from scene complexity?
  - efficient culling and adaptive super-sampling techniques required

Consequences

- splat-based representations are (as least) as powerful as polygon meshes
  - locally optimal linear approximation (ellipses)
  - added flexibility (\( C^{-1} \))
  - sharp features can be represented (splat clipping)
Consequences

- splat-based representations are (as least) as powerful as polygon meshes
  - locally optimal linear approximation (ellipses)
  - sharp features can be represented (splat clipping)

Point-Based Surface Topology

- manifold surfaces are at least $C^0$
- locally independent approximation yields $C^0$
- visual continuity through overlapping splats (object vs. image space)
- visual smoothness through normal blending
- topology information embedded in a point cloud?

Point-Based Surface Topology

- $\epsilon$-neighborhood
  - symmetric, non-manifold, uniform
  - super-linear complexity
- uniform sampling vs. "r-sampling"
  - geometrical precision
  - topological precision
- k-nearest neighborhood
  - asymmetric, non-manifold, adaptive
  - linear complexity

Overview

- point-based representations
  - shape approximation
  - surface topology
- octree point clouds
- optimized splat subsampling

Point Clouds

- piecewise constant approximation
  - sampling resolution: $h$
  - $O(h^2)$ sample points
  - $3 \log(h)$ bits per sample
  - total complexity $O(h^2 \log(h))$
  - can we obtain $O(h^2)$ total complexity?
Point Clouds

Octree Point Clouds

Octree Point Clouds

Octree Point Clouds
Zero Tree Coding

O(h²)

Octree Point Clouds

- storage per point
  - 8/4 + 8/16 + ... = 8/3 = 2.67 bit (uncompressed)
  - 1.00 – 1.50 bit (entropy encoded)
- resolution independent : O(h²)
  - coarser octree levels encode many samples
- fast rendering by octree traversal
  - 4 scalar additions and 2 divisions per point
- level of detail representation
Octree Traversal

- fixed translation vectors for cell centers

- leaf node centers

\[ y = x + \sum_{k} v_k \]

- modelview + viewport transformation

\[ \tilde{x} = x + \sum_{k} v_k x_k = x + \sum_{k} x_k \]

- incremental summation during traversal

Level of Detail

8 octree levels
compression factor
\[ \approx 1:24 \]

9 octree levels
compression factor
\[ \approx 1:27 \]

10 octree levels
compression factor
\[ \approx 1:30 \]

11 octree levels
compression factor
\[ \approx 1:33 \]
Level of Detail

Progressive Transmission

Overview

- point-based representations
  - shape approximation
  - surface topology
- octree point clouds
- optimized splat subsampling
Problem Specification
- given:
  - sample points \( p_i \) on a surface
  - approximation tolerance \( \varepsilon \)
- find:
  - minimal set of elliptical splats \( S_j = (c_j, u_j, v_j) \)
  - all samples within \( \varepsilon \)
  - no holes
  - most regular splat distribution

Approximation Error
- distance of a sample point to a set of splats
  \( (\text{minimum projected distance}) \)
- replace each splat by an \( 2\varepsilon \)-cylinder
- splat overlap in object space?
  - union of solids
  - projected overlap

Surface Structure
- surface samples \( p_i \)
- k-nearest neighbor graph \( N(i,j) \)
- estimated normals \( n_i \)
- surface area element \( \omega_i = r^2 \)
- splats \( S_j \)
- coverage relation \( C(i,j) \)
- surface patches \( P_j = C(*,j) \)

Our Approach...
- sub-problems ...
  - global error control
  - prevent holes
  - optimal splat distribution
- techniques ...
  - one-sided Hausdorff distance (splat generation)
  - discrete coverage estimation (set operations)
  - global relaxation (better than greedy)
Splat Generation
• grow a candidate splat for each point $p_i$
  – no least squares fitting
    (fixed normal, maximum deviation)
  – align elliptical splats to principal directions

Coverage Estimate
• each sample has to be assigned to a splat
• guarantee sufficient overlap

Splat Generation
• grow a candidate splat for each point $p_i$
  – no least squares fitting
    (fixed normal, maximum deviation)
  – align elliptical splats to principal directions
  – each selection satisfies error threshold

Coverage Estimate
• each sample has to be assigned to a splat
• guarantee sufficient overlap
• modified coverage relation $C'(i,j)$
Coverage Estimate
- Each sample must be assigned to a splat.
- Guarantee sufficient overlap.
- Modified coverage relation $C'(i,j)$.
- Set operations:
  - Check if active splats cover all samples.
- Complexity depends on
  - Number of active splats.
  - Number of input samples.

Greedy Selection
- Any selection of candidates satisfies the error tolerance.
- Find a selection that covers all points.
- Greedy selection
  - Largest un-covered patch.

Global Relaxation
- Optimize splat distribution.
- Two set operations...
  - Minimize overlap.
  - Remove redundant splats.
- Preserve coverage (local updates only)
  - Kernel of a splat $K_j \subseteq P_j$.
- Iterate over all splats.

Minimize Overlap
- Replace a splat $P_j$ by one of its k-nearest neighbors $P'$.
- Minimize overlap with nearby active splats.
- Preserve full coverage (kernel $K_j$).
- Simple local set operations.

Minimize Overlap
- Replace a splat $P_j$ by one of its k-nearest neighbors $P'$.
- Minimize overlap with nearby active splats.
- Preserve full coverage (kernel $K_j$).
- Simple local set operations.
Minimize Overlap
• replace a splat $P_j$ by one of its $k$-nearest neighbors $P'$
• minimize overlap with nearby active splats
• preserve full coverage (kernel $K_j$)
• simple local set operations

Remove Redundant Splats
• greedy selection causes redundancy
• remove and re-distribute neighboring active splats

Minimize Overlap
• replace a splat $P_j$ by one of its $k$-nearest neighbors $P'$
• minimize overlap with nearby active splats
• preserve full coverage (kernel $K_j$)
• simple local set operations

Remove Redundant Splats
• greedy selection causes redundancy
• remove and re-distribute neighboring active splats
Remove Redundant Splats

• greedy selection causes redundancy
• remove and re-distribute neighboring active splats

Examples

Examples

input 170K error 0.47 %

422 333

Comparison

• greedy vs. global relaxation

734, 0.29 %
Comparison

- splats vs triangles

Advantages

- exploit full flexibility of splat representations (k-nearest neighbors)
- global relaxation leads to better results than greedy selection
- take full splat geometry into account, not just the centers

Visual Approximation Quality

- approximate normal vectors
- known problem of polygons

Visual Approximation Quality

- approximate normal vectors
- known problem of polygons
**Visual Approximation Quality**

- approximate normal vectors
- known problem of polygons
  (where phong shading doesn't help)

**Phong Splatting**

- splat \( S_i = (c_i, u_i, v_i, n_i, \alpha_i, \beta_i, \text{rgb}_i) \)
- \((c_i, u_i, v_i)\) obtained by least squares
  - tangents aligned to principal directions
- \((n_i, \alpha_i, \beta_i)\) obtained by least squares
  - w.r.t. splat parametrization
  - normal vector length doesn't matter

**Examples**

**Examples**

**Comparison**
Comparison

Overview

- point-based representations
  - shape approximation
  - surface topology
- octree point clouds
- optimized splat subsampling

Conclusions

- point-based representations
  - good for screen space blending
  - view-independent sampling causes redundancy
  - hierarchical octree representation
- splat-based representation
  - performance ???

Conclusions

- point-based representations
- splat-based representation
  - same approximation order as polygons
    - ellipses approximate better than triangles
    - overlap more flexible than manifold consistency
  - sharp corners
  - high quality rendering
- performance ???

Conclusions

- point-based representations
- splat-based representation
  - phong splatting improves visual quality
    and allows for sparser representations
  - why is the polygon rate still higher than the splat rate?