Point-Sampled Shape Representations

RWTH AACHEN 1 Computer Graphics Group Leif Kobbelt

Why Point-Based Graphics ?

- simplicity
- generality
- flexibility

- efficiency ?
  – point- vs. splat-approximation
  – GPU processing
- quality ?

RWTH AACHEN 2 Computer Graphics Group Leif Kobbelt

You are here ...

- 1.) Science
- ...
  – 1.1.) Computer Science
  – ...
    - 1.1.1.) Computer Graphics
    - ...
      – 1.1.1.1.) Point-based Graphics

        1.1.1.1.1.) Point-based Rendering Technology
        1.1.1.1.2.) Point-based Geometry Processing

RWTH AACHEN 3 Computer Graphics Group Leif Kobbelt

Overview

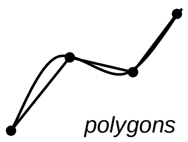
- point-based representations
  – shape approximation
  – surface topology
- octree point clouds
- optimized splat subsampling

RWTH AACHEN 4 Computer Graphics Group Leif Kobbelt

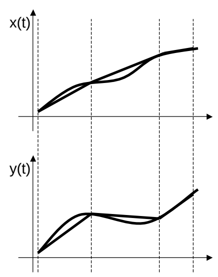
Point-Based Approximation

- what is the approximation power ?

error = $O(h^2)$



polygons

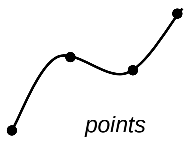


RWTH AACHEN 5 Computer Graphics Group Leif Kobbelt

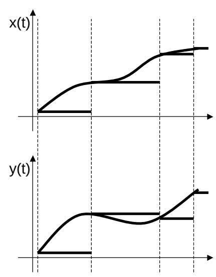
Point-Based Approximation

- what is the approximation power ?

error = $O(h)$



points

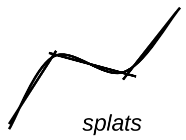


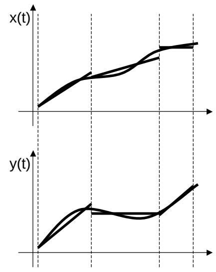
RWTH AACHEN 6 Computer Graphics Group Leif Kobbelt

Point-Based Approximation

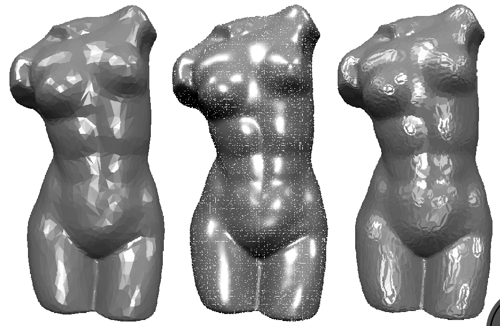
- what is the approximation power ?

$\boxed{\text{error} = O(h^2)}$



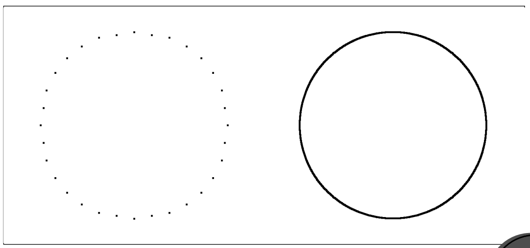


Point-Based Approximation



Point-Based Approximation

- what is the required precision ?



Point-Based Approximation

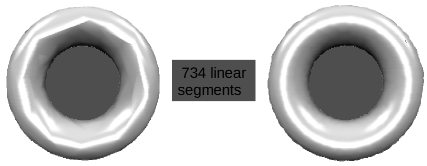
- what is the required precision ?
  - *Points* : precision = O(sampling density)
    number = O(surface area)
  - *Splats* : precision = O(sampling density ^{2})
    number = O(surface curvature)

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

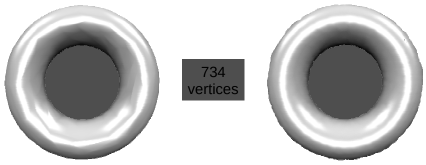
Consequences

- 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)
  - added flexibility („ C^{-1} “)
  - sharp features can be represented (splat clipping)



Point-Based Surface Topology

- manifold surfaces are at least C^0
- locally independent approximation yields C^{-1}
- 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

- ε-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

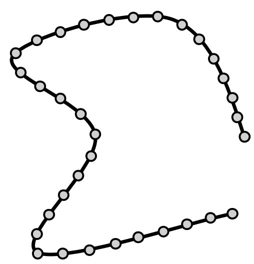
Overview

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

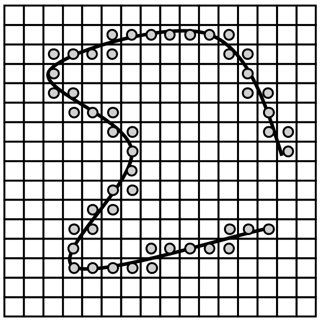
Point Clouds

- 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

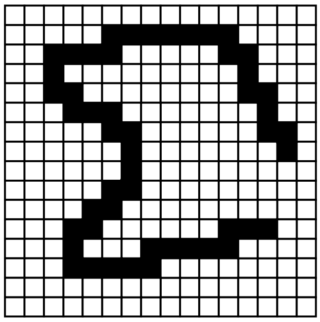


Point Clouds



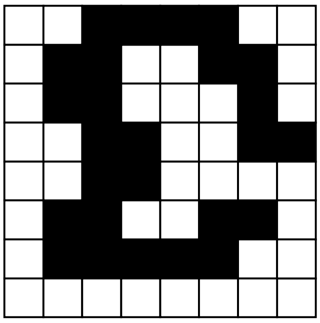
RWTH AACHEN 19 Computer Graphics Group Leif Kobbelt

Point Clouds



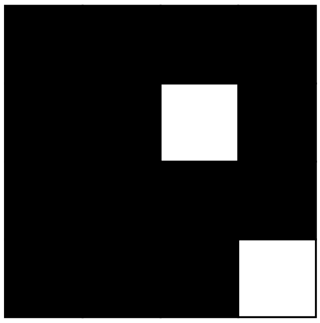
RWTH AACHEN 20 Computer Graphics Group Leif Kobbelt

Octree Point Clouds



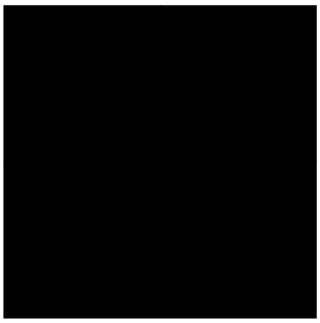
RWTH AACHEN 21 Computer Graphics Group Leif Kobbelt

Octree Point Clouds



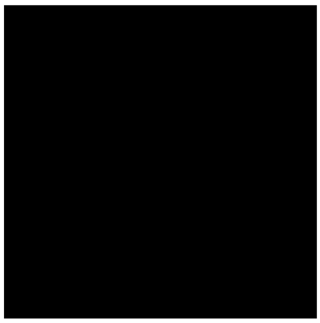
RWTH AACHEN 22 Computer Graphics Group Leif Kobbelt

Octree Point Clouds



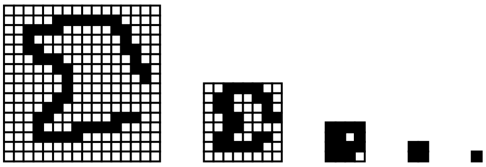
RWTH AACHEN 23 Computer Graphics Group Leif Kobbelt

Octree Point Clouds



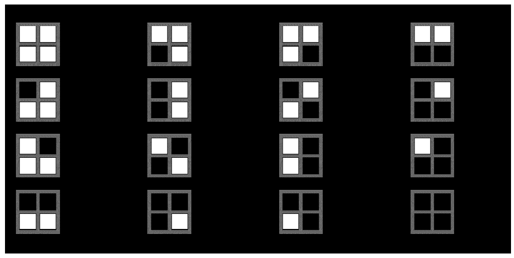
RWTH AACHEN 24 Computer Graphics Group Leif Kobbelt

Zero Tree Coding



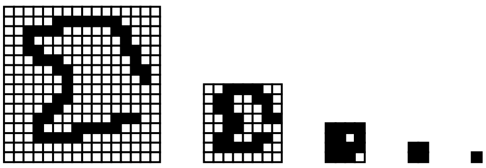
Computer Graphics Group
Leif Kobbelt

Zero Tree Coding



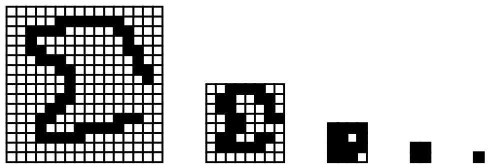
Computer Graphics Group
Leif Kobbelt

Zero Tree Coding



Computer Graphics Group
Leif Kobbelt

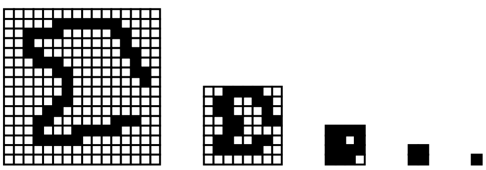
Zero Tree Coding



O(h^3)

Computer Graphics Group
Leif Kobbelt

Zero Tree Coding



O(h^2)

Computer Graphics Group
Leif Kobbelt

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(h^2)
  - coarser octree levels encode *many* samples
- fast rendering by octree traversal
  - 4 scalar additions and 2 divisions per point
- level of detail representation

Computer Graphics Group
Leif Kobbelt

Octree Traversal

- fixed translation vectors for cell centers

Level i   Level i-1

$$d_{i,j} = 2^{i-2}h \begin{pmatrix} \pm 1 \\ \pm 1 \\ \pm 1 \end{pmatrix}$$

Octree Traversal

- leaf node centers

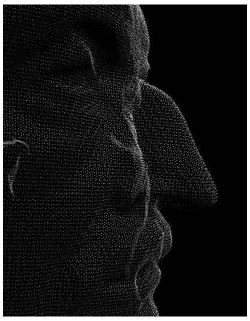
$$p = c + \sum_{i=1}^{k} d_{i,j_i}$$

- modelview + viewport transformation

$$Mp = Mc + \sum_{i=1}^{k} Md_{i,j_i} = c' + \sum_{i=1}^{k} d'_{i,j_i}$$

- incremental summation during traversal

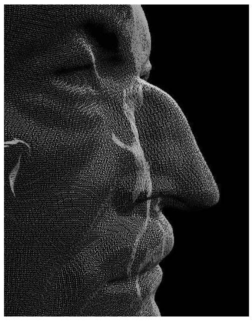
Level of Detail



8 octree levels

compression factor ≈ 1:24

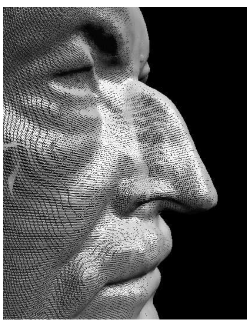
Level of Detail



9 octree levels

compression factor ≈ 1:27

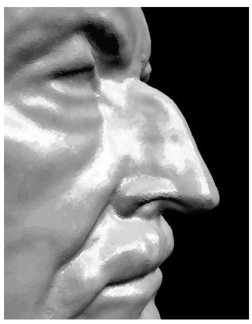
Level of Detail



10 octree levels

compression factor ≈ 1:30

Level of Detail



11 octree levels

compression factor ≈ 1:33

Level of Detail

Level of Detail

Level of Detail

Level of Detail

Progressive Transmission

Overview

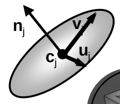
- point-based representations
  - shape approximation
  - surface topology
- octree point clouds
- optimized splat subsampling

Problem Specification

- given:
  - sample points $\mathbf{p}_i$ on a surface
  - approximation tolerance ε

Problem Specification

- given:
  - sample points $\mathbf{p}_i$ on a surface
  - approximation tolerance ε
- find:
  - minimal set of elliptical splats $S_j = (\mathbf{c}_j, \mathbf{u}_j, \mathbf{v}_j)$
  - all samples within ε
  - no holes
  - most regular splat distribution

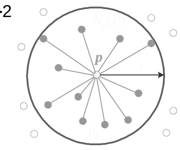


Approximation Error

- distance of a sample point to a set of splats
  (minimum projected distance)
- replace each splat by an 2ε-cylinder
- splat overlap in object space ?
  - union of solids
  - projected overlap

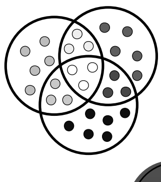
Surface Structure

- surface samples $\mathbf{p}_i$
- k-nearest neighbor graph N(i,j)
- estimated normals $\mathbf{n}_i$
- surface area element $\omega_i = r^2$
- splats S_j
- coverage relation C(i,j)
- surface patches P_j = C(*,j)



Surface Structure

- surface samples $\mathbf{p}_i$
- k-nearest neighbor graph N(i,j)
- estimated normals $\mathbf{n}_i$
- surface area element $\omega_i = r^2$
- splats S_j
- coverage relation C(i,j)
- surface patches P_j = C(*,j)

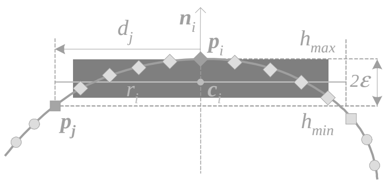


Our Approach ...

- 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)

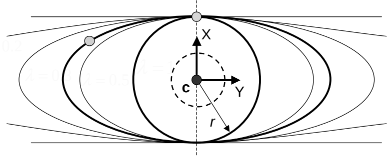
Splat Generation

- grow a *candidate* splat for each point $\mathbf{p}_i$
  - no least squares fitting (fixed normal, maximum deviation)



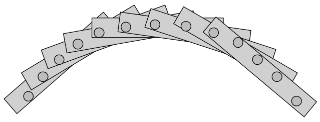
Splat Generation

- grow a *candidate* splat for each point $\mathbf{p}_i$
  - no least squares fitting (fixed normal, maximum deviation)
  - align elliptical splats to principal directions



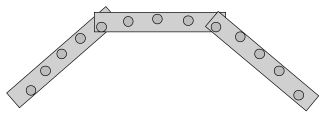
Splat Generation

- grow a *candidate* splat for each point $\mathbf{p}_i$
  - no least squares fitting (fixed normal, maximum deviation)
  - align elliptical splats to principal directions



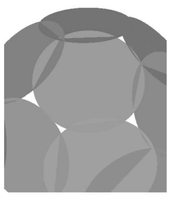
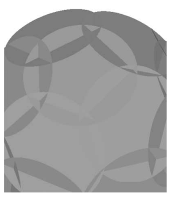
Splat Generation

- grow a *candidate* splat for each point $\mathbf{p}_i$
  - no least squares fitting (fixed normal, maximum deviation)
  - align elliptical splats to principal directions
  - each selection satisfies error threshold



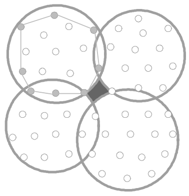
Coverage Estimate

- each sample has to be assigned to a splat
- guarantee sufficient overlap

Coverage Estimate

- each sample has to be assigned to a splat
- guarantee sufficient overlap
- modified coverage relation C´(i,j)



Coverage Estimate

- each sample has to 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

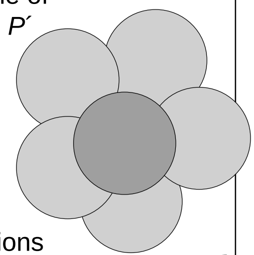
- any selection of candidates satisfies the error tolerance
- find a selection that covers all points
- greedy selection
  – largest un-covered patch

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

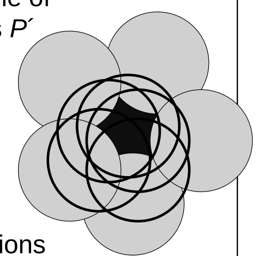
Minimize Overlap

- 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



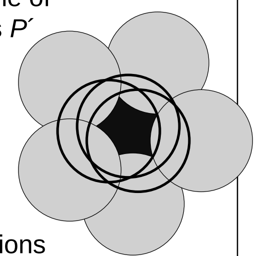
Minimize Overlap

- 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



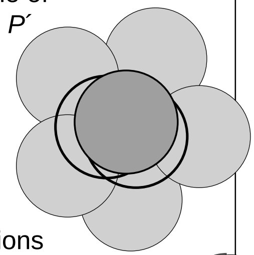
Minimize Overlap

- 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



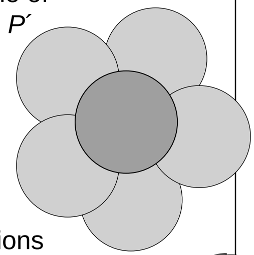
Minimize Overlap

- 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



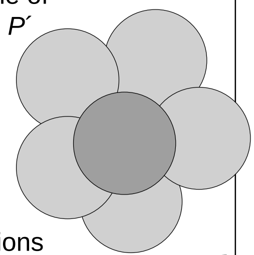
Minimize Overlap

- 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



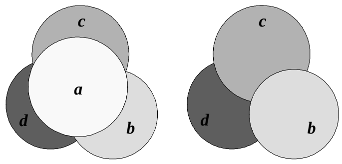
Minimize Overlap

- 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



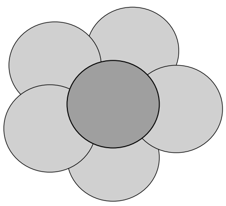
Remove Redundant Splats

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



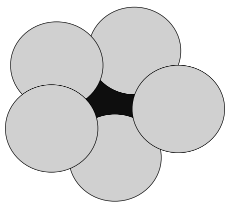
Remove Redundant Splats

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



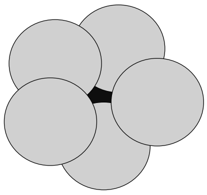
Remove Redundant Splats

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



Remove Redundant Splats

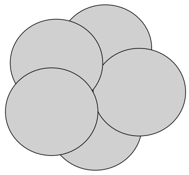
- greedy selection causes redundancy
- remove and re-distribute neighboring active splats



RWTH AACHEN 67 Computer Graphics Group Leif Kobbelt

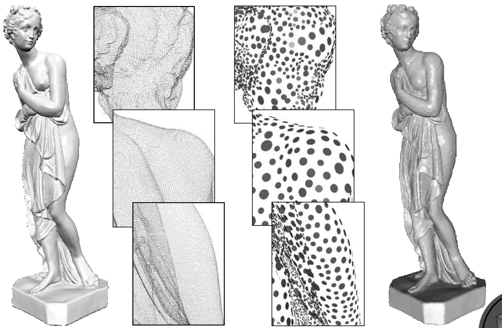
Remove Redundant Splats

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



RWTH AACHEN 68 Computer Graphics Group Leif Kobbelt

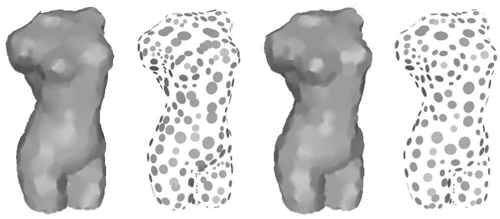
Examples



RWTH AACHEN 69 Computer Graphics Group Leif Kobbelt

Examples

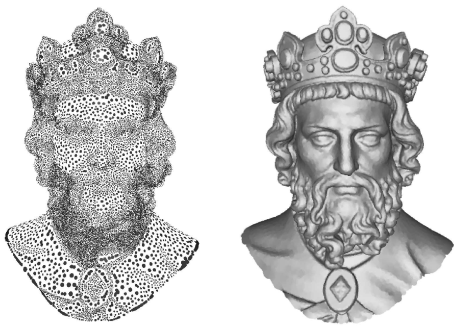
input 170K error 0.47 %



422 333

RWTH AACHEN 70 Computer Graphics Group Leif Kobbelt

Examples



600K

66K

0.03%

RWTH AACHEN 71 Computer Graphics Group Leif Kobbelt

Comparison

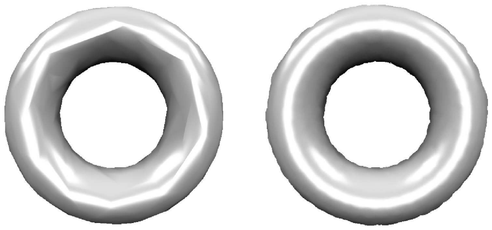
- greedy vs. global relaxation

734, 0.29 % 493, 0.29 %

RWTH AACHEN 72 Computer Graphics Group Leif Kobbelt

Comparison

- splats vs triangles



734 triangles 734 splats

Comparison

- circular vs. elliptical splats

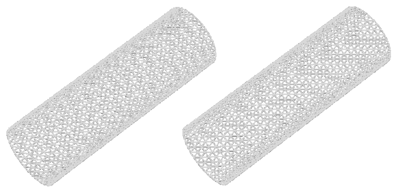
734, 0.2 % 510, 0.2 %

Advantages

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

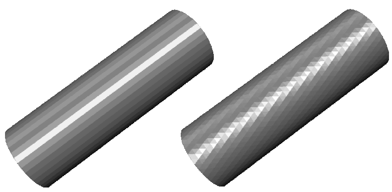
Visual Approximation Quality

- approximate normal vectors
- known problem of polygons



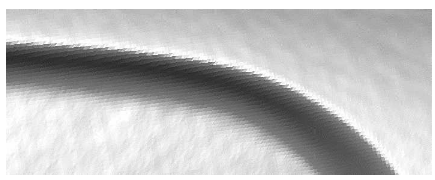
Visual Approximation Quality

- approximate normal vectors
- known problem of polygons



Visual Approximation Quality

- approximate normal vectors
- known problem of polygons



Visual Approximation Quality

- approximate normal vectors
- known problem of polygons

(where phong shading doesn‘t help)

Phong Splatting

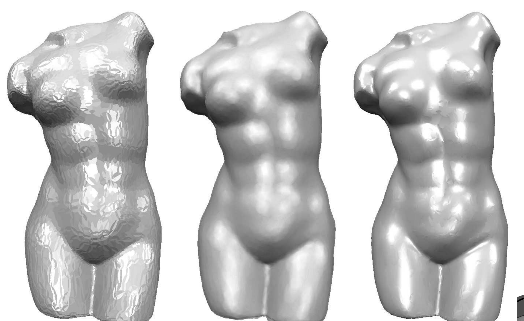
- splat $S_j = (\mathbf{c}_j, \mathbf{u}_j, \mathbf{v}_j, \mathbf{n}_j, \alpha_j, \beta_j, rgb_j)$
- $(\mathbf{c}_j, \mathbf{u}_j, \mathbf{v}_j)$ obtained by least squares
  - tangents aligned to principal directions
- $(\mathbf{n}_j, \alpha_j, \beta_j)$ obtained by least squares
  - w.r.t. splat parametrization
  - normal vector length doesn‘t matter

Examples

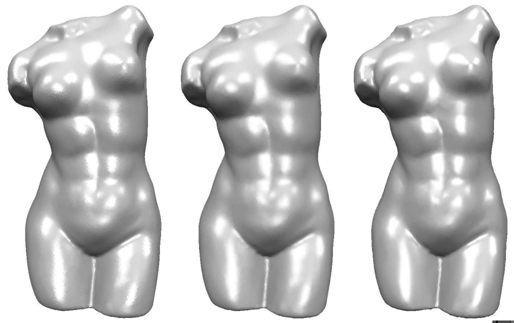
Examples

Examples

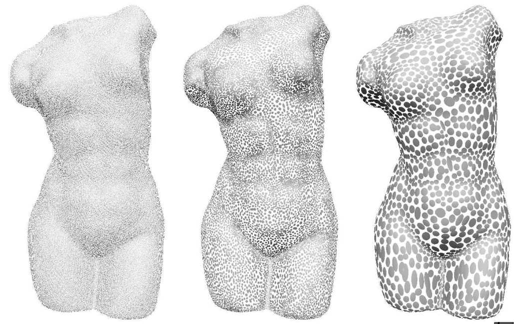
Comparison



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