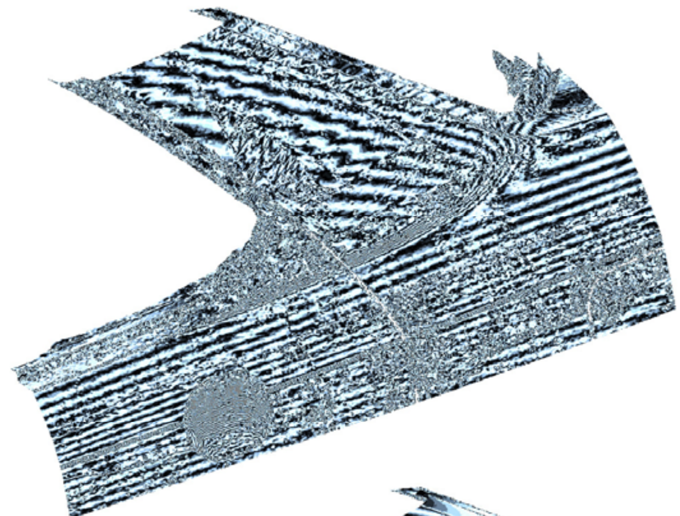
Processing 3D Laser Scanner Data for Automotive Styling Design

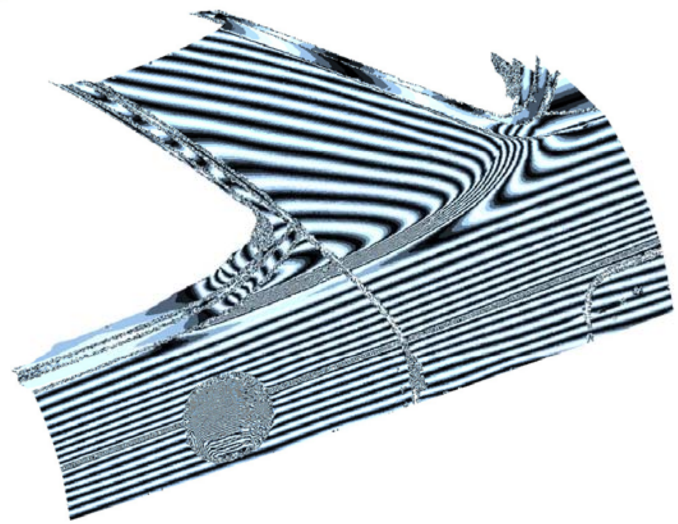
Ph.D. Thesis Proposal

Miguel Vieira

May 6, 2005



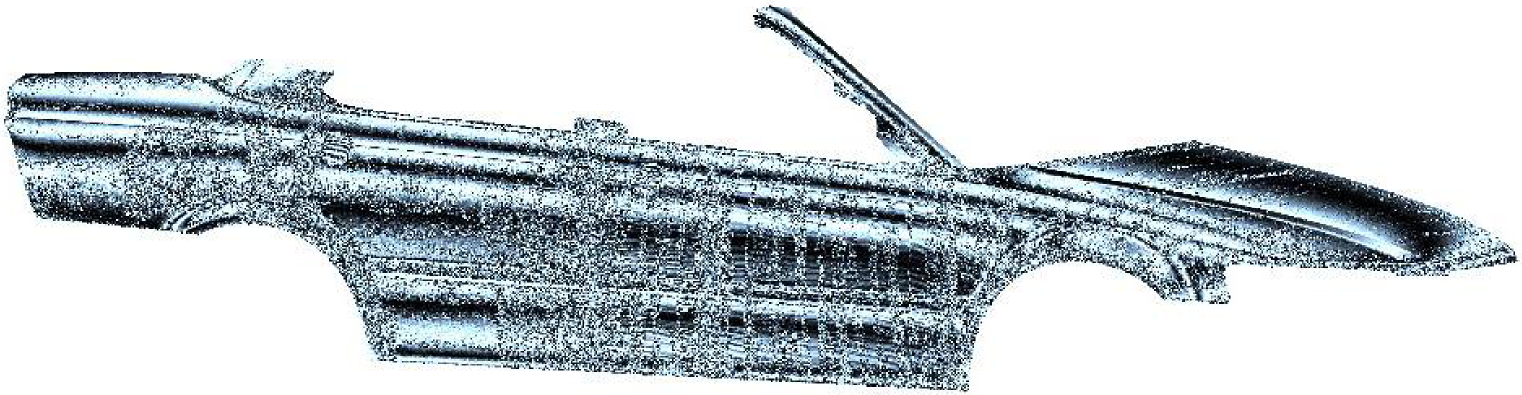
Dr. Kenji Shimada (Adviser)
Dr. Martial Hebert
Dr. William Messner
Dr. Burak Ozdoganlar



Laser Scanner Technology

http://www.steinbichler.de

- Can quickly measure 3D surface coordinates without contact
- Accurate and easy to use
- Used in automotive styling design to digitize models
- Scanned data can be used for:
  - Testing engineering requirements
  - Aesthetic interrogation
  - Surfacing the final part



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Problem Statement

- Laser-scanner data often needs significant preprocessing before it is useful
- Automotive styling design issues are not addressed by existing algorithms
  - How to remove geometric noise from the data without deforming it or smoothing sharp edges?
  - How to group the points into regions closely approximated by single surfaces?
  - Can we create a piecewise-smooth reconstruction of the part that satisfies both aesthetic and engineering constraints?

Automotive Design Requirements

- Design Constraints
  - Scanner accuracy
  - Tolerances required for design and manufacture
  - These bound the deformation of the data and the accuracy of surface fitting and surface reconstruction
- Aesthetic Constraints
  - A k differentiable surface has k-1 differentiable reflection lines: smooth reflection lines require a twice differentiable surface
  - Character lines are sharp edges that create a distinctive shape – these sharp edges must be preserved
  - Principle of simplest shape: the resulting surface must be free of unnecessary undulations

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Thesis Contributions

- Mesh Smoothing
  - Creates an aesthetically pleasing surface
  - Minimizes mesh deformation
  - Preserves sharp edges
- Surface Extraction
  - Estimates curvatures and noise levels on the data
  - Detects sharp edges
  - Segments data into regions approximated by single surfaces
  - Fits surfaces to a specified tolerance
- Surface Reconstruction (proposed)
  - Will reconstruct objects with large smooth surfaces, smooth blends between surfaces, and sharp edges
  - Will bound the accuracy and geometric continuity of the reconstructed object

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Algorithms

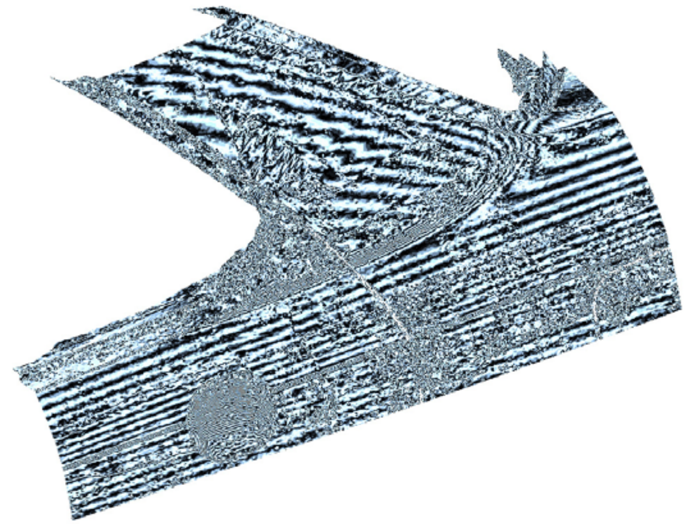
- Mesh Smoothing
- Surface Extraction from Meshes
- Surface Extraction from Point-Sampled Data
- Surface Reconstruction Proposal
- Summary

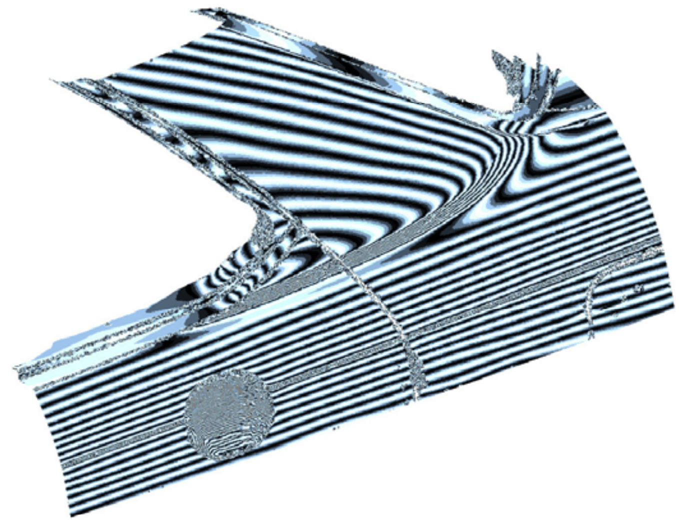
.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Problem Statement

- Goal: remove geometric noise from a mesh with edge preservation and minimal deformation

- Deformation must be within laser-scanner and design tolerances
- Result must be smooth over large scales with sharp edges preserved
- The smoothed mesh will be used for aesthetic interrogation and patch fitting for design

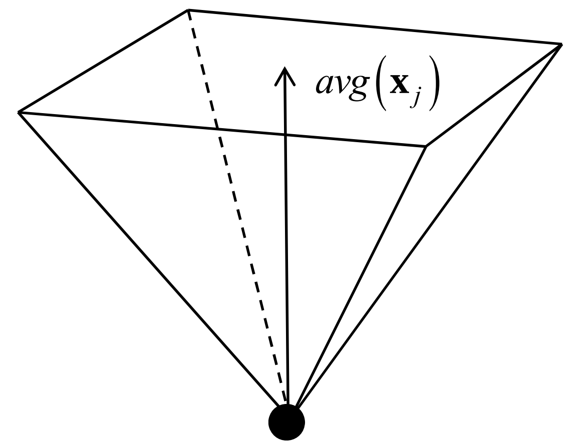


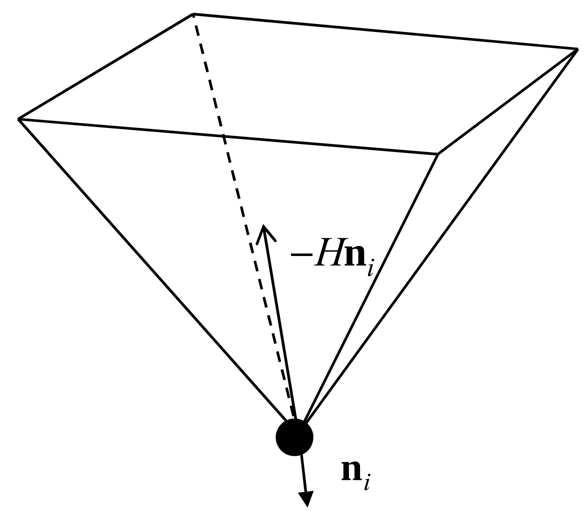


.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Existing Smoothing Methods

- Diffusion: move each vertex to a weighted average of the positions of its neighbors [1]
- Curvature Flow: move each vertex in its normal direction a distance equal to the mean curvature [2]
  - Both these methods can cause distortion, so using them requires a compromise
- Normal Maps: smooth the normals first, then find vertex positions that satisfy the normal field [3]

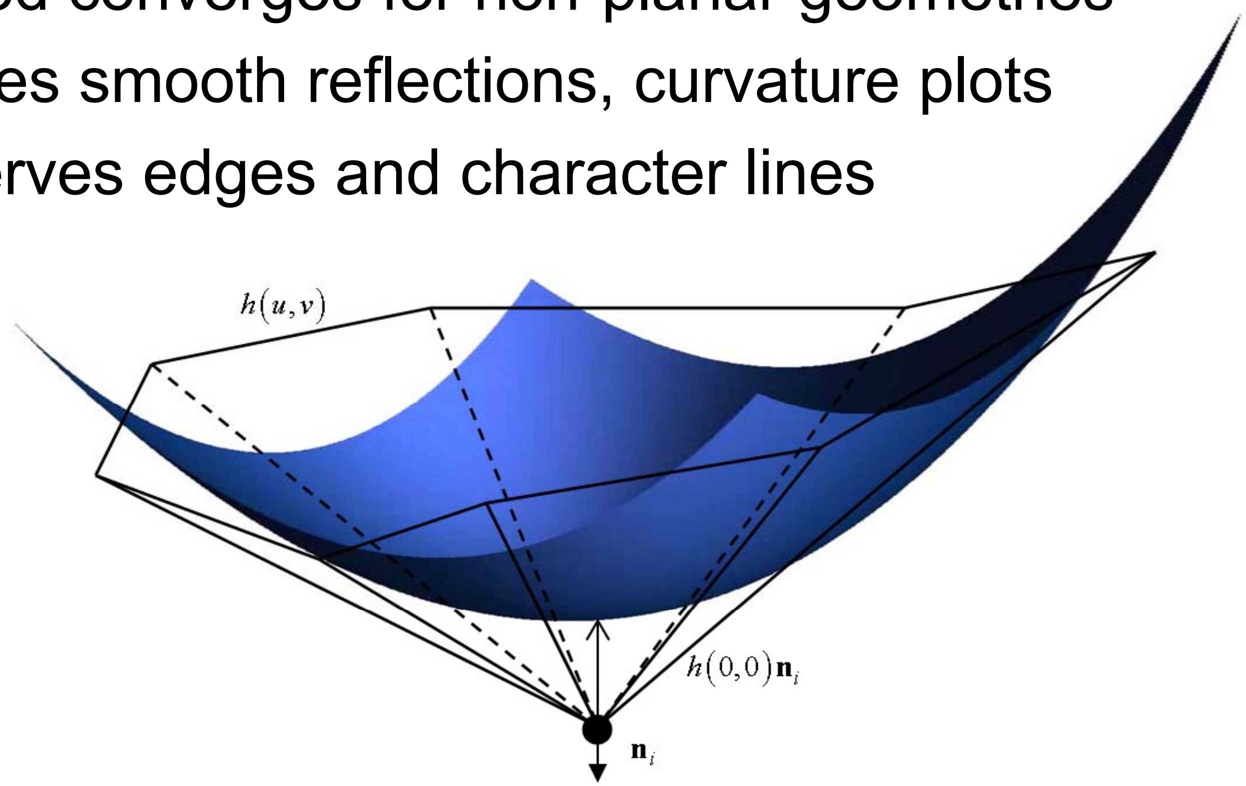




[1] Kobbelt, et al., *Geometric Modeling Based on Polygonal Meshes (tutorial), EUROGRAPHICS 2000.*
[2] Desbrun, et al., *Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow, SIGGRAPH 99.*
[3] Tasdizen, et al., *Geometric Surface Smoothing via Anisotropic Diffusion of Normals, IEEE Visualization 2002.*

Polynomial Fitting Solution

- Each vertex is moved onto a polynomial surface approximating its neighborhood
- User can control polynomial order, neighborhood size, and threshold angle
- Method converges for non-planar geometries
- Creates smooth reflections, curvature plots
- Preserves edges and character lines



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Differential Geometry

- The neighborhood of every point on a smooth surface is defined by a 2D Taylor series in a local coordinate system:

$$h(u,v) = \sum_{k=2}^{n} \frac{1}{k!}\left(u\frac{\partial}{\partial u} + v\frac{\partial}{\partial v}\right)^{k} h(0,0)$$

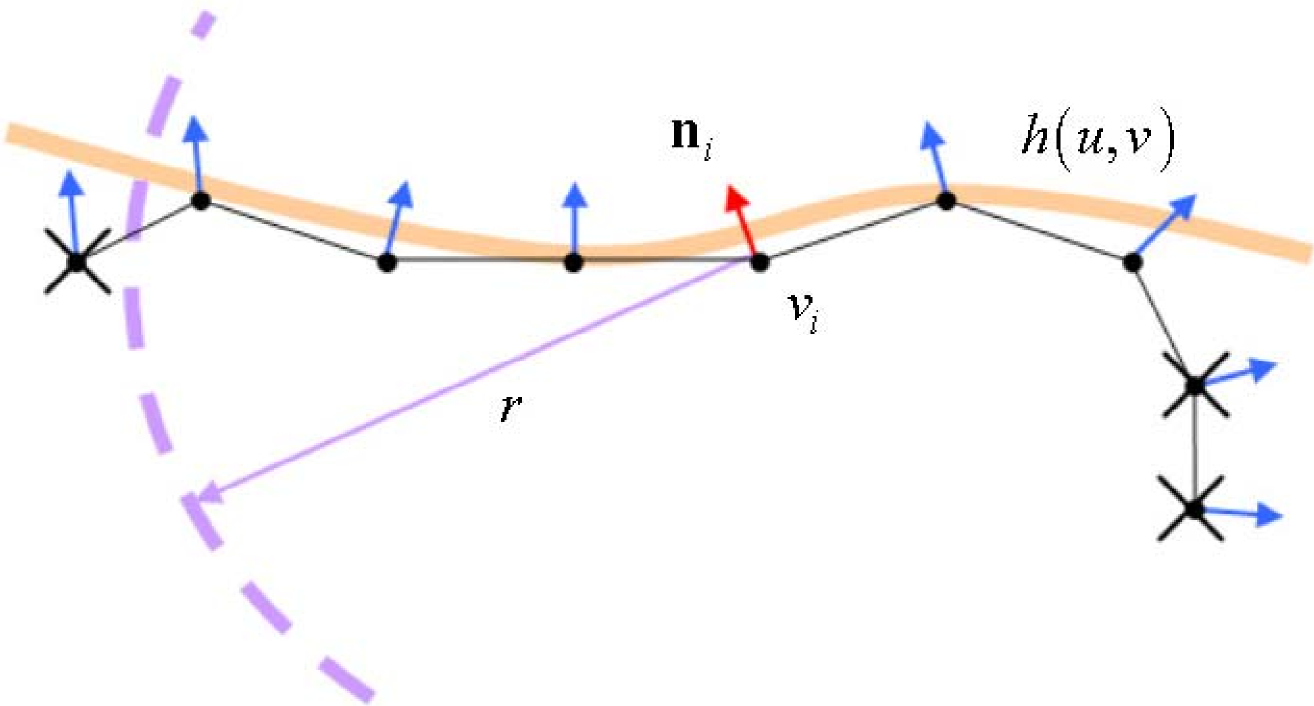
- The Taylor series determines the differentiability of the surface at the point
- We find the coefficients of truncated approximation of the Taylor series

$$h(u,v) = a_{00} + a_{10}u + a_{01}v + a_{20}u^2 + a_{11}uv + a_{02}v^2$$

- Approximating polynomial is found with a linear least-squares problem solved by Cholesky decomposition

Vertex Neighborhoods

- Create a neighborhood by adding vertices with positions and normals close to that of the target vertex
- This creates pseudo-planar neighborhoods that don't include sharp edges



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Implementation

- For each vertex
  - Grow a neighborhood using breadth-first search with distance and normal-deviation constraints
  - Find neighborhood vertex positions in local coordinate system
  - Least squares fit a polynomial to the vertices in the local coordinate system
  - Move the target vertex onto the polynomial
- User selects neighborhood width, polynomial order, and threshold angle

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Automobile C-Pillar

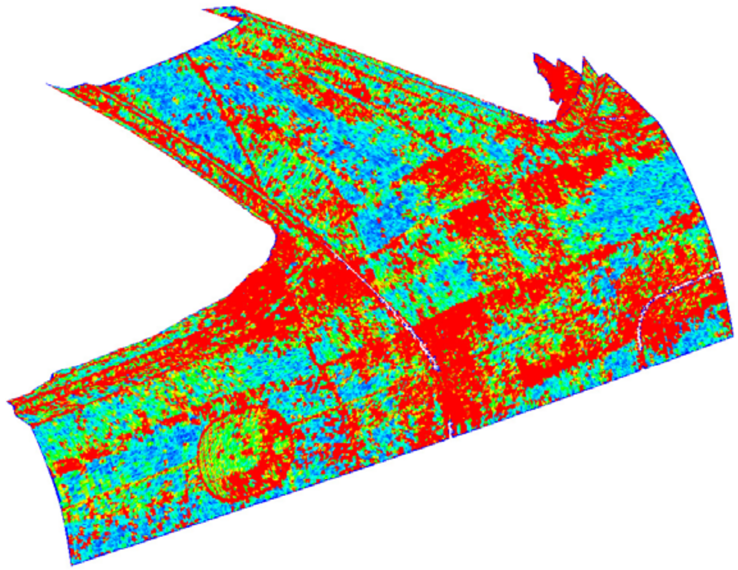
99,790 vertices

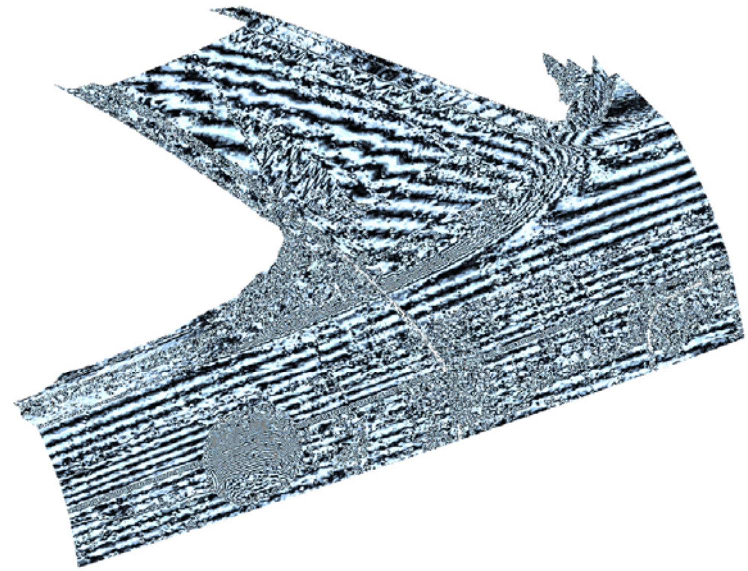
Results

Curvature

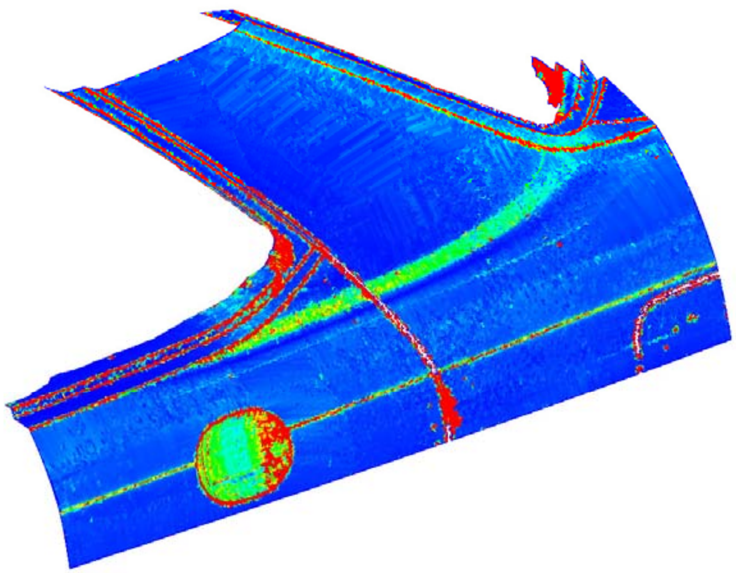
Reflections

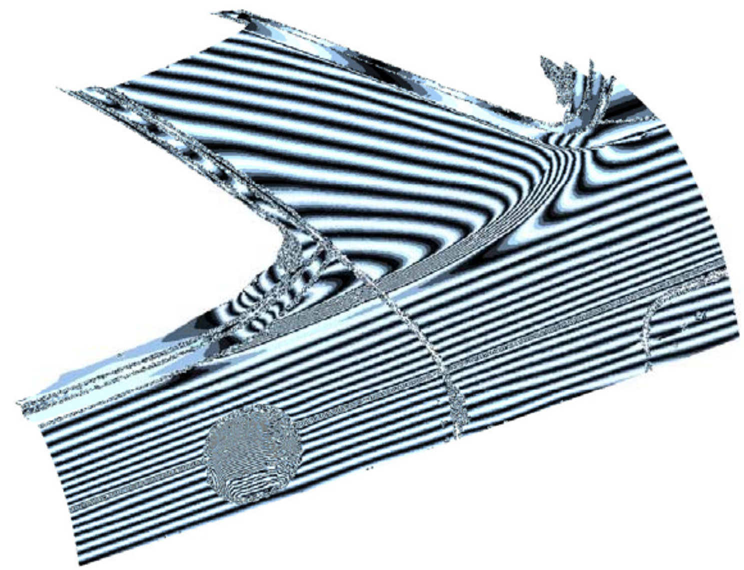
Original





Smoothed

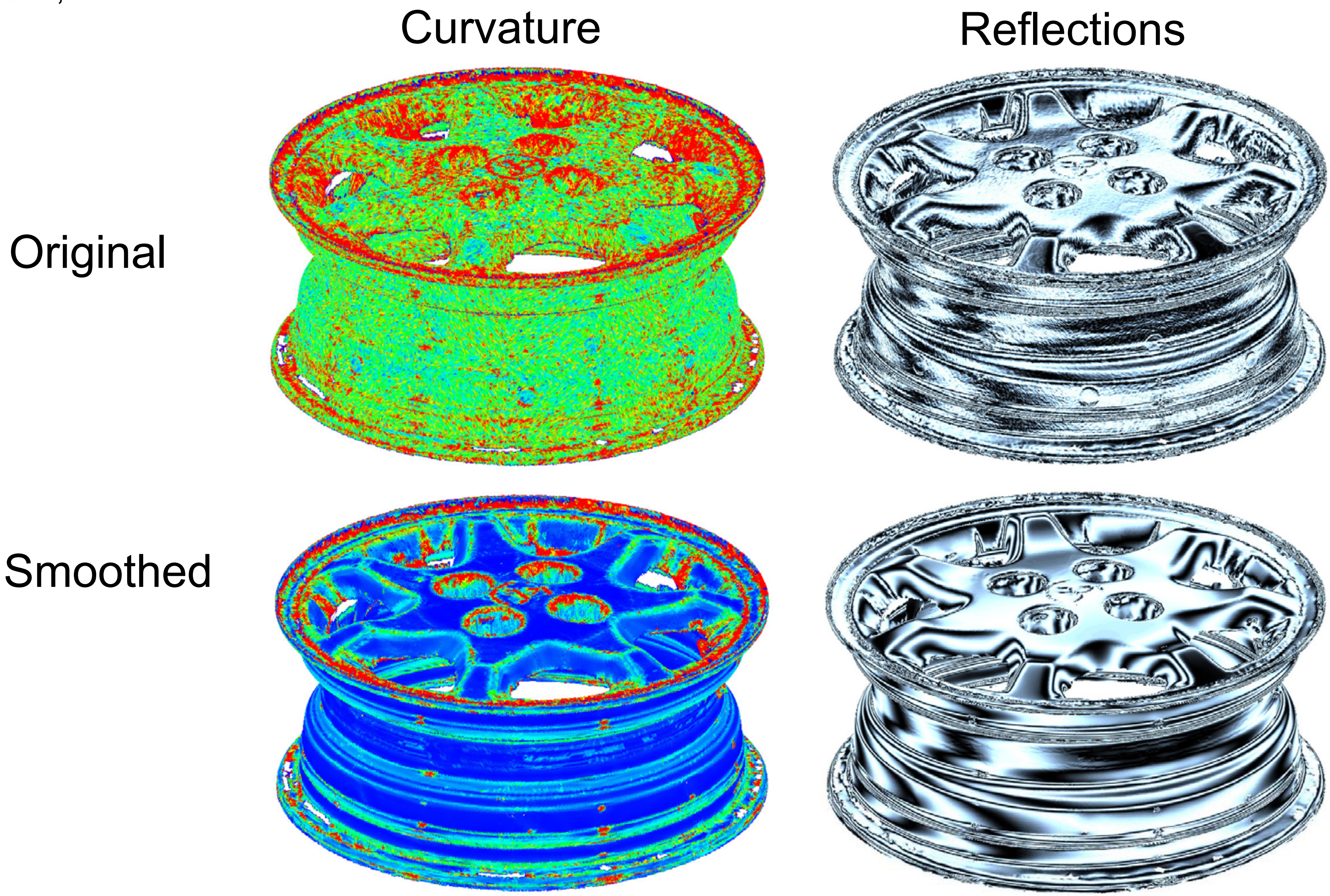




Wheel Rim

366,920 vertices

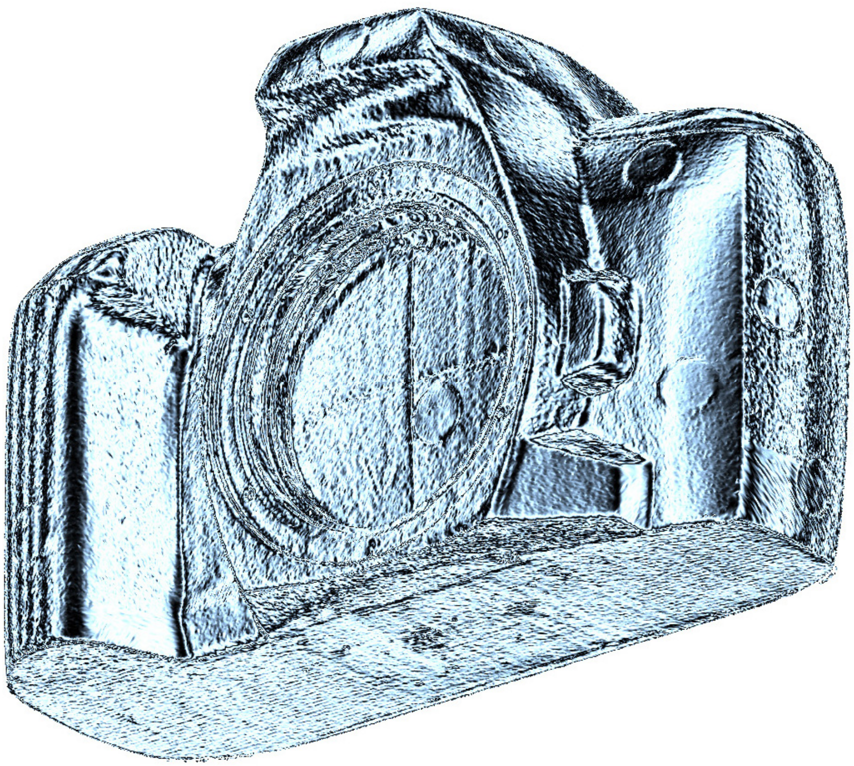
Results



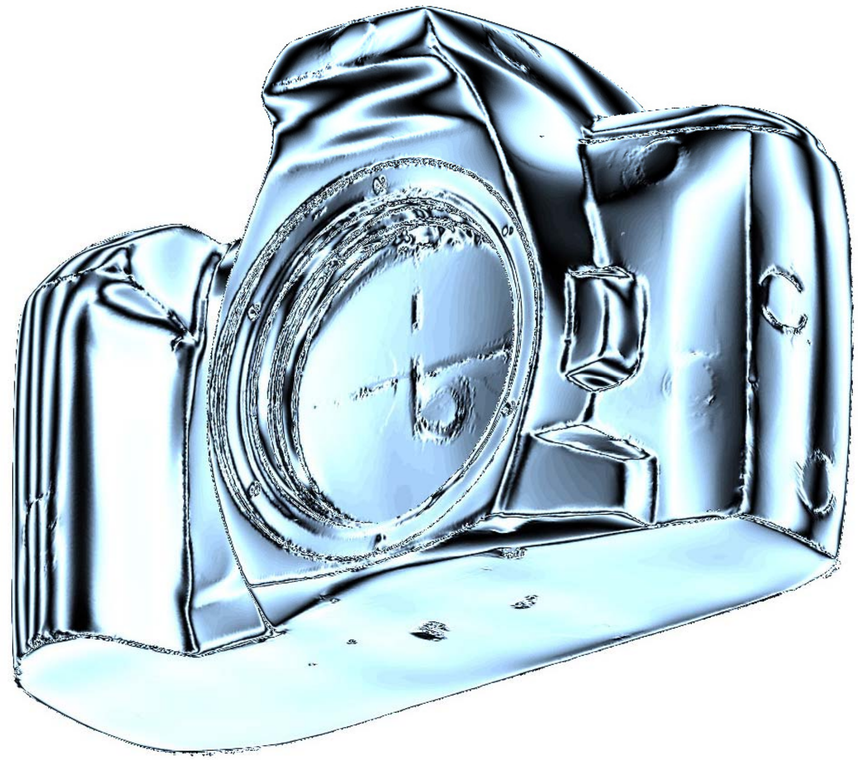
SLR Body

606,254 vertices

Results



Original



Smoothed

Computer Integrated
Engineering Laboratory
Carnegie Mellon

Algorithms

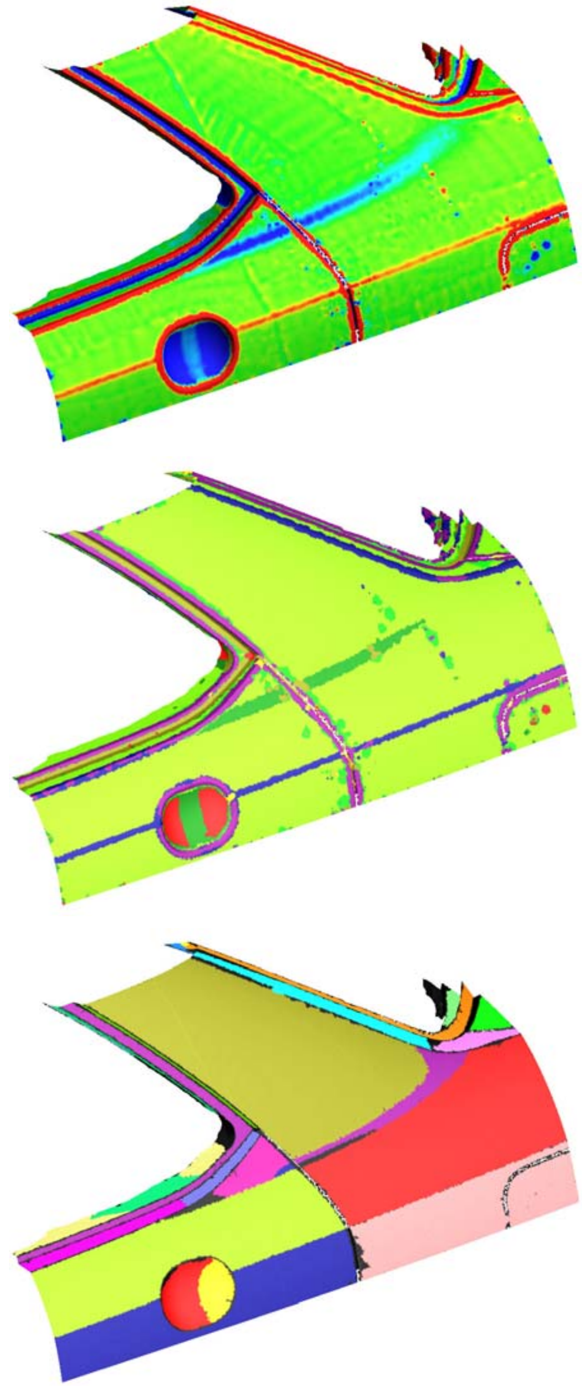
- Mesh Smoothing
- Surface Extraction from Meshes
- Surface Extraction from Point-Sampled Data
- Surface Reconstruction Proposal
- Summary

Problem Statement

- Must create a concise surface-based representation of the object from the scanned data
- This is a manually intensive and time-consuming procedure
- How can we automatically segment a dense, noisy mesh into groups of vertices approximated by single surfaces and then output the surfaces themselves?
  - Segments should be as large as possible
  - Surface approximations must be accurate
- A designer can easily modify the extracted surfaces to create a final design

Algorithm Summary

- Approximate vertex neighborhoods with polynomials
  - Estimate surface curvatures
  - Estimate noise levels
  - Detect sharp edges
- Filter curvatures and label vertices by curvature-signs
- Contract labeled regions to create seed regions
- Grow seed regions by iterating between region growing and surface fitting
- Regions are grown by testing vertices for geometric compatibility with the approximating surfaces



Related Work

- Besl and Jain introduced region growing for images and range data [1]
  - We generalize this approach to dense, noisy, unstructured data
- Direct segmentation: hierarchical subdivision by testing neighborhoods for compatibility with increasingly complex analytic surfaces [2]
  - Not clear how to use with freeform surfaces
- Recover-and-select: simultaneously grow randomly placed regions while culling non-optimal models [3,4]
  - We place seed regions carefully for computation efficiency

[1] Besl and Jain. *Segmentation Through Variable-Order Surface Fitting*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988.
[2] Benkő and Várady. *Segmentation Methods for Smooth Point Regions of Conventional Engineering Objects. Computer-Aided Design Journal*, 2004.
[3] Leonardis, et al. *Superquadrics for Segmenting and Modeling Range Data. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

Noise and Curvature Estimation

- We approximate the points within a ball centered at each vertex with a quadric polynomial

$$h(u,v) = a_{00} + a_{10}u + a_{01}v + a_{20}u^2 + a_{11}uv + a_{02}v^2$$

- From this polynomial we calculate the Gaussian, mean, and principal curvatures
- The G^0 noise is calculated by $\varepsilon_i^0 = |a_{00}|$
- The G^1 noise is calculated by $\varepsilon_i^1 = \cos^{-1}(\mathbf{n} \cdot \mathbf{n}_i)$
- For the G^0 and G^1 noise, we find the value that is larger than that of a certain percentage of the other vertices
- These values will be the compatibility thresholds for region growing

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Sharp Edge Detection

- If the smallest radius of curvature at a vertex is small compared to the mesh density, a sharp edge is indicated
- We do not grow regions from sharp edges
- Mathematically, a sharp edge is likely if $\dfrac{1}{\left|\kappa_{\max,\,i}\right|} < 10\ l_{avg,\,i}$

Computer Integrated Engineering Laboratory
Carnegie Mellon

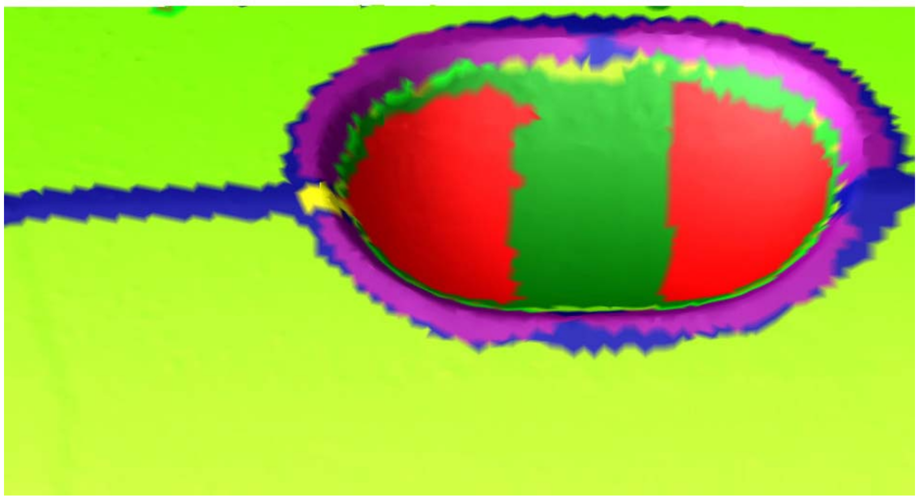
Surface-Type Labeling

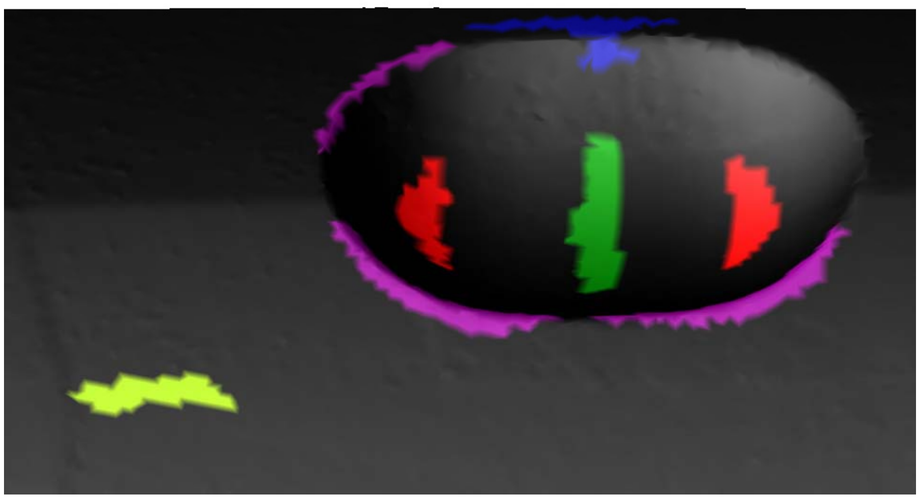
- We apply median and mean filters to the curvatures to despeckle and smooth them
- Then we use the signs of the filtered Gaussian and mean curvature to label the vertices

	$K > 0$	$K = 0$	$K < 0$
$H < 0$	Peak	Ridge	Saddle Ridge
$H = 0$	(none)	Flat	Minimal Surface
$H > 0$	Pit	Valley	Saddle Valley

Seed Regions

- Region growing is most effective when vertices on the boundary of the growing region are geometrically compatible with those already in the region
- Cluster points with the same surface-type labels
- Contract these clusters to form seed regions

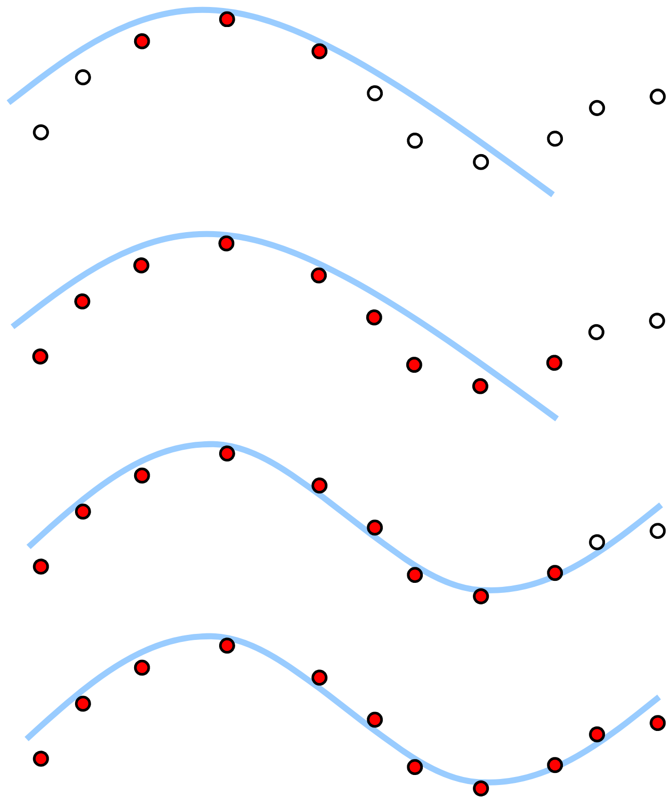




.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Region Growing

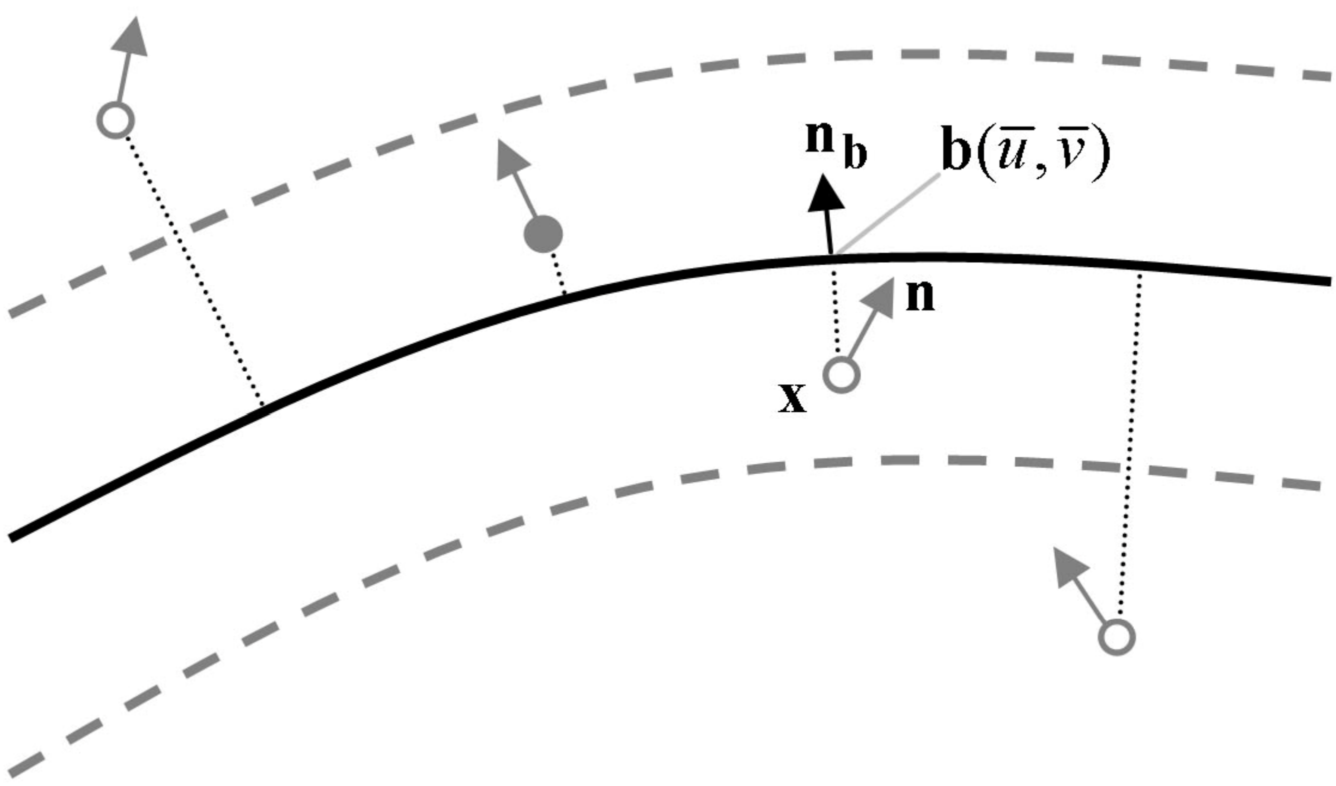
- Seed regions lie within large groups of vertices with identical geometric properties
- A surface approximating the seed region likely approximates the nearby vertices well



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Geometric Compatibility

- We fit a surface to the seed region
- Given a point $\mathbf{x}$ and a parametric surface $\mathbf{b}(u,v)$
  - $\mathbf{x}$ is G^0-compatible if $\| \mathbf{x} - \mathbf{b}(\bar{u},\bar{v}) \| < \varepsilon^0$
  - $\mathbf{x}$ is G^1-compatible if $\cos^{-1}(\mathbf{n},\mathbf{n_b}) < \varepsilon^1$



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

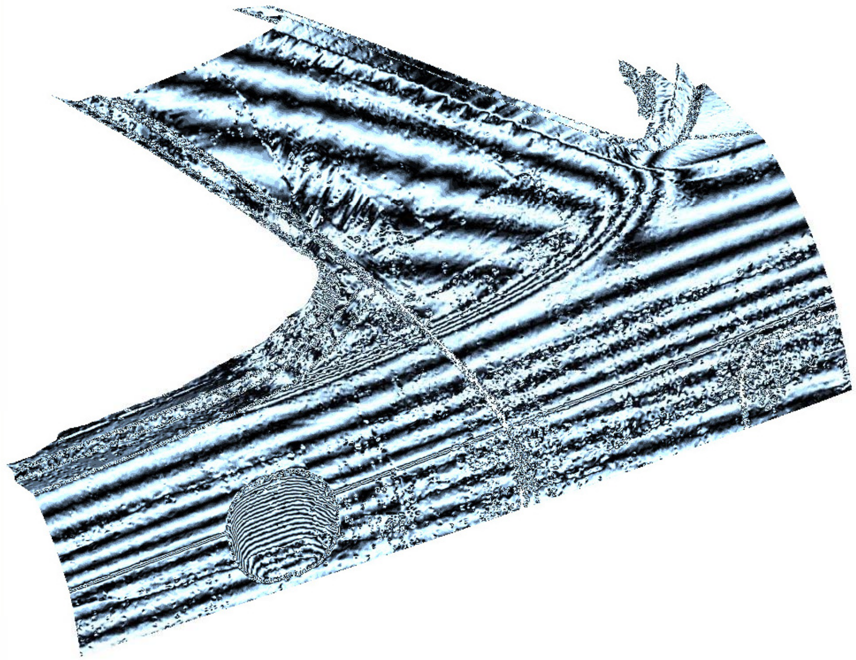
Surface Fitting

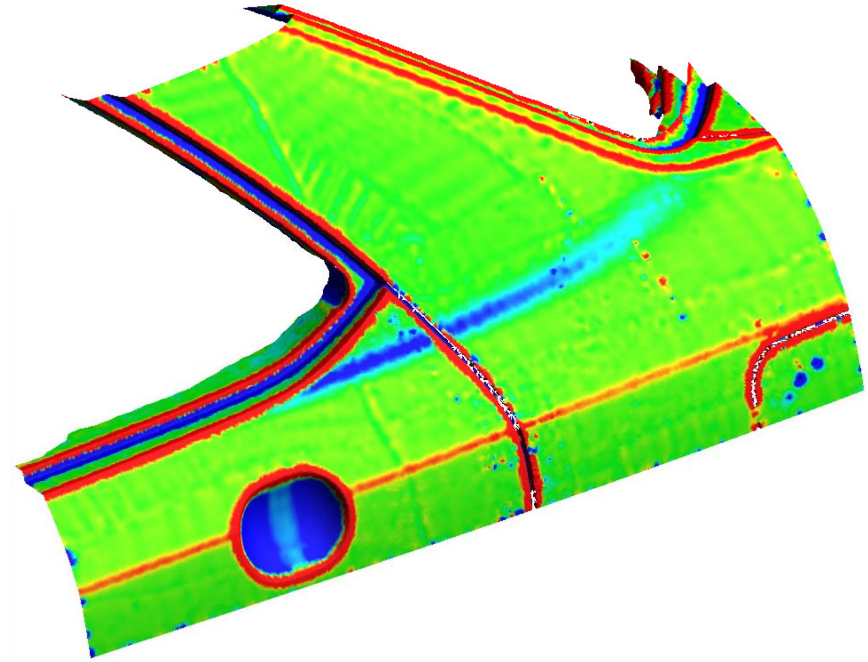
- The region-growing algorithm can be used with any class of surfaces that allows
  - approximation (surface fitting)
  - differentiation
  - point projection (testing geometric compatibility)
- We use bicubic Bézier surfaces

$$\mathbf{b}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{p}_{i,j} B_i^3(u) B_j^3(v), \quad 0 \le u,v \le 1$$

- First we get an initial parameterization for the region
- Then we iterate between surface fitting and parameterization to get a good fit

Results

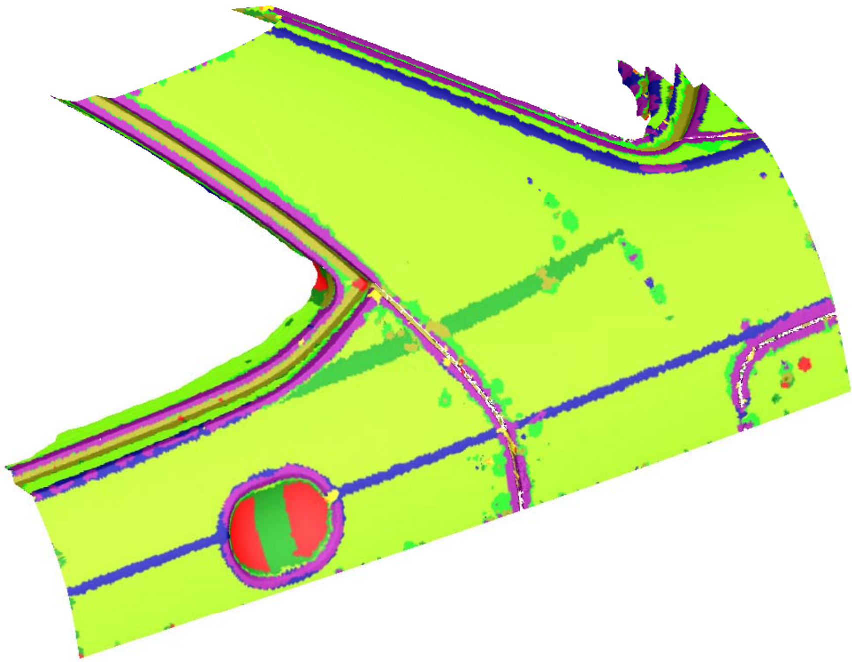


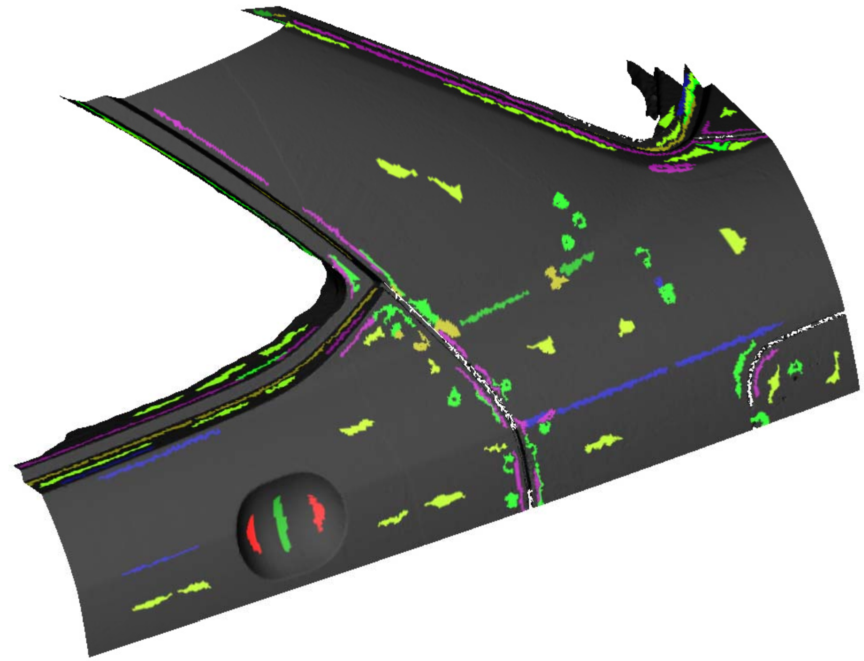


- Simulated reflection lines on the raw mesh show how noisy the geometry is
- The estimated curvatures are shown on the right

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

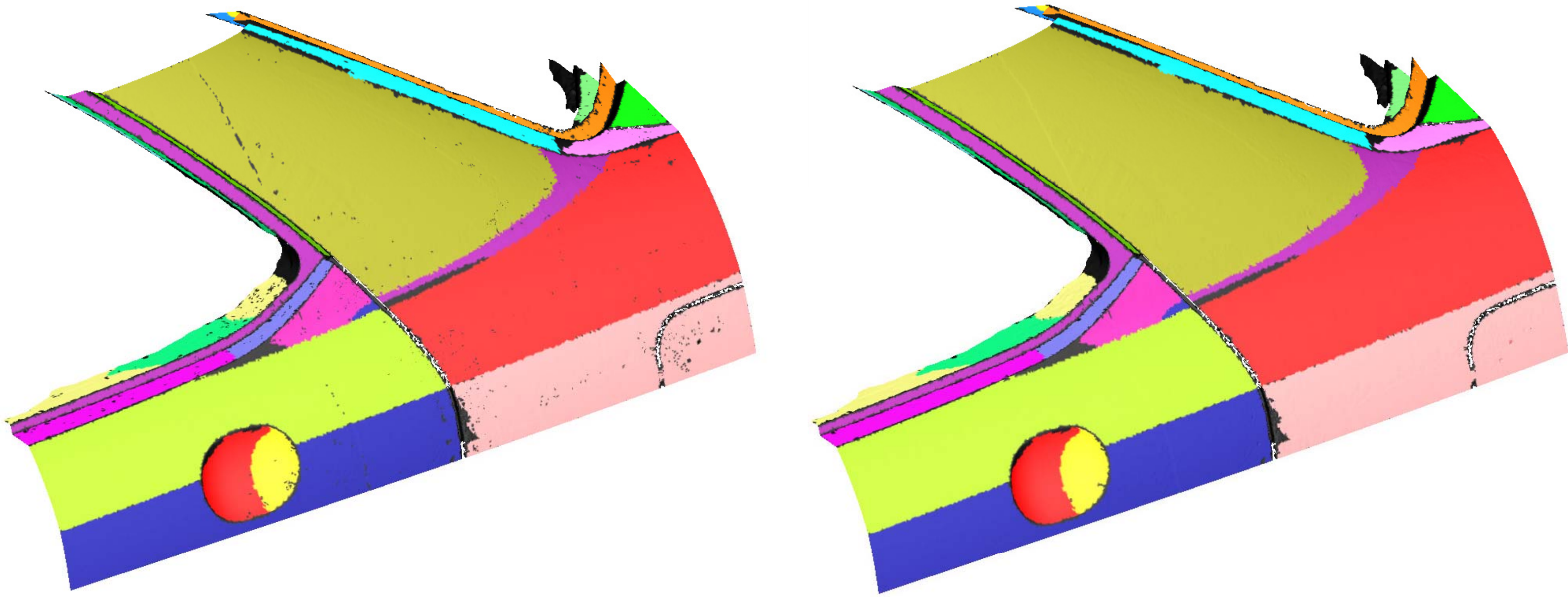
Results





- We label vertices using the signs of the Gaussian and mean curvatures
- Then we contract the labeled regions to form seed regions

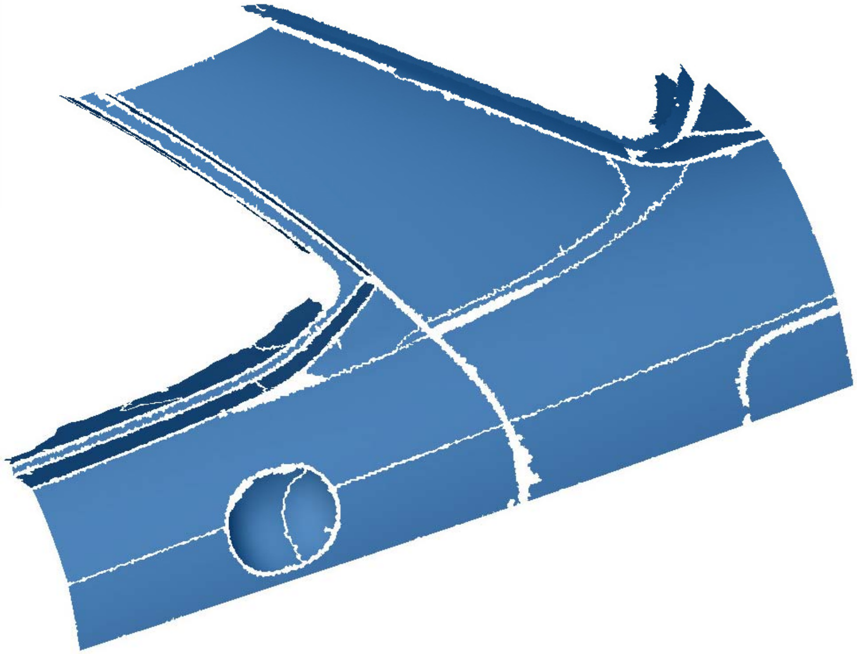
Results

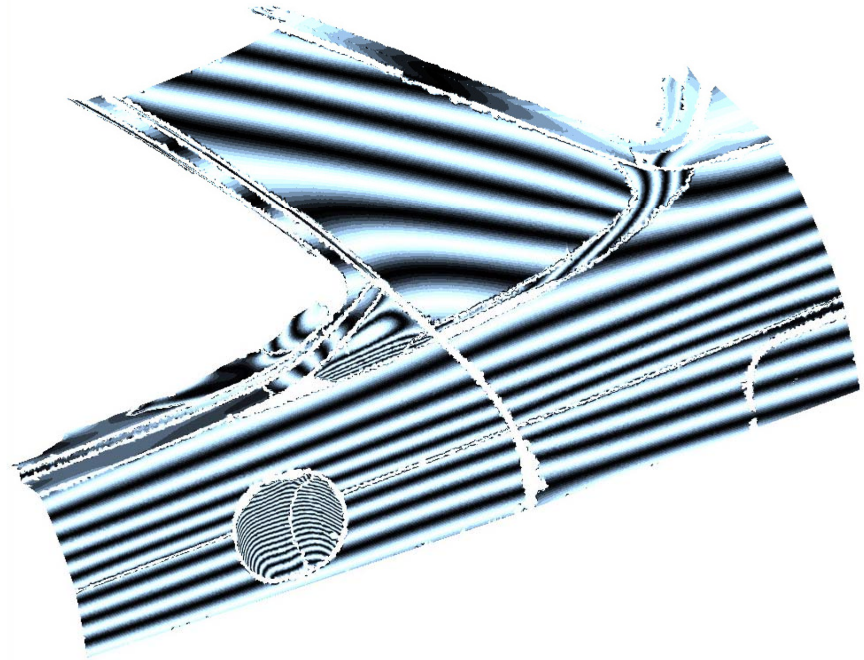


- Each color corresponds to a region of vertices approximated by a single surface
- As a last step, we remove holes in the segmentation caused by outlier noise

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Results





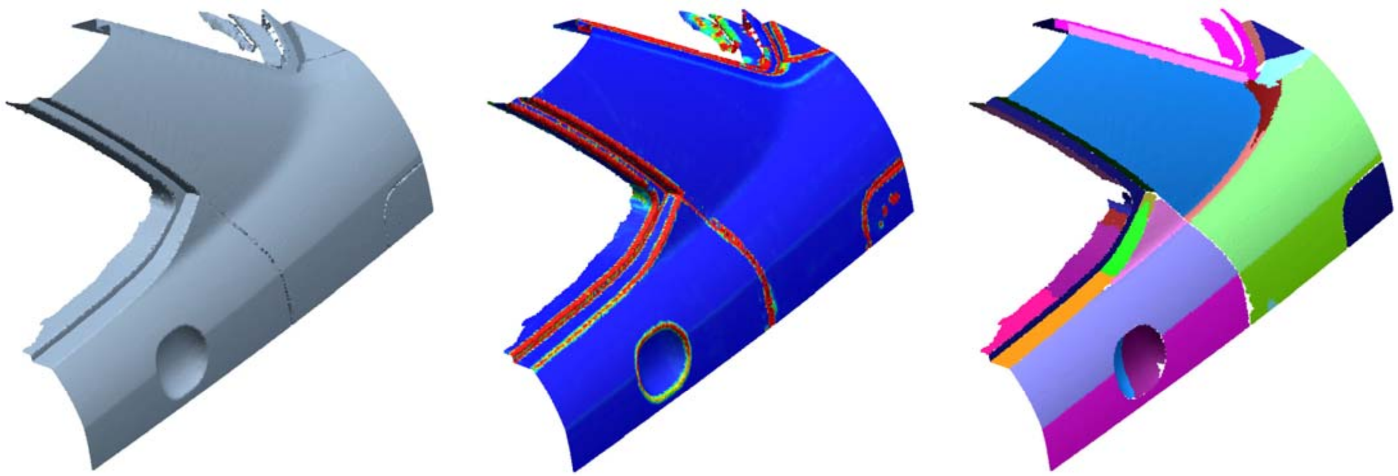
- Here we projected the segmented vertices onto the extracted surfaces
- Note the smooth reflection lines on the extracted surfaces

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Algorithms

- Mesh Smoothing
- Surface Extraction from Meshes
- Surface Extraction from Point-Sampled Data
- Surface Reconstruction Proposal
- Summary

Motivation



- Using a mesh for surface extraction requires too much computer memory
- Can we modify the algorithm and reduce the data representation so we can process very large models?
- When the data is very dense, we can replace topology with spatial proximity!

.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Algorithm

- Grow regions from seed points in in order of increasing surface variation
- Region Growing
  - Use points near the seed point to fit an initial surface approximation for region growing
  - Grow the region by adding points that are geometrically compatible with the underlying surface
  - Once all compatible points have been added, fit a new surface to the region to improve the approximation
  - Use the new surface to re-grow the region
- Repeat until region size stops increasing

Surface Variation

- Estimates local surface properties on point clouds
- Like finding the mean and variance of a 1D distribution
- Let $\mathbf{N}(\mathbf{x}_i)$ be the *k*-nearest neighbors of a point $\mathbf{x}_i$
- Then the covariance matrix of the points is

$$\mathbf{C} = \sum_{\mathbf{y} \in \mathbf{N}(\mathbf{x}_i)} (\mathbf{y} - \overline{\mathbf{x}}) \cdot (\mathbf{y} - \overline{\mathbf{x}})^T$$

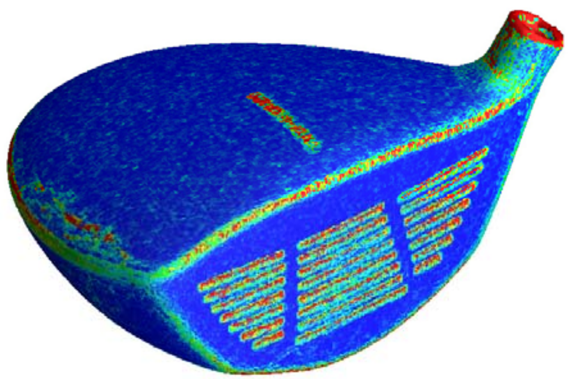
- The eigenvalues of this matrix measure the variance of $\mathbf{N}(\mathbf{x}_i)$ in the directions of the eigenvectors
- The surface variation is

$$\sigma_k(\mathbf{x}_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$$

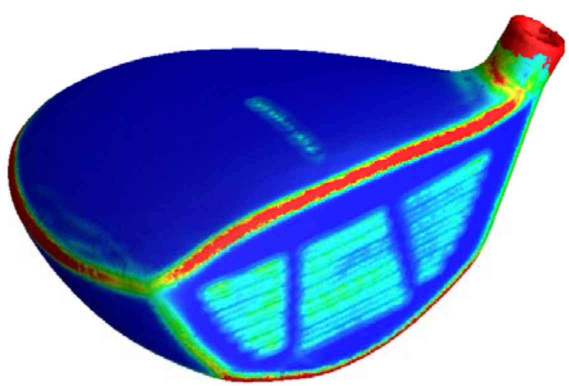
.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Surface Variation

Input point cloud (200,000 points)



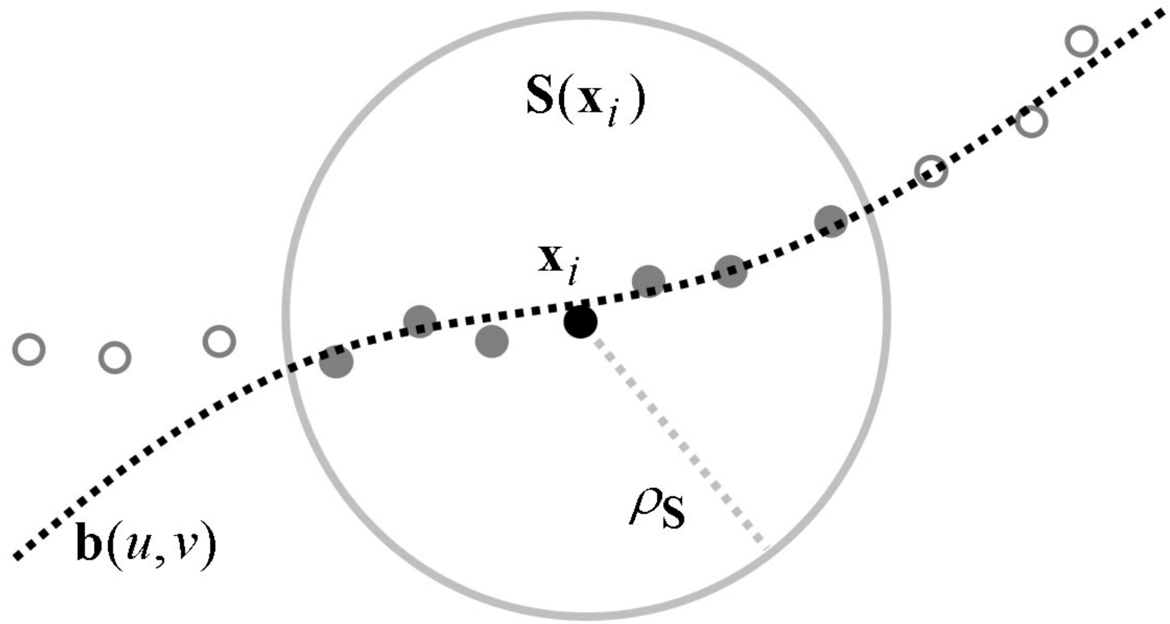
Surface variation using few nearest neighbors



Surface variation using many nearest neighbors

.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Seed Regions



- If $\mathbf{x}_i$ has already been assigned to a region, skip it
- Otherwise, construct a seed region $\mathbf{S}(\mathbf{x}_i)$ of all the unlabeled points within a radius ρ_S of $\mathbf{x}_i$

$$\mathbf{S}(\mathbf{x}_i) = \left\{ \mathbf{x} \in \mathbf{X} \;\middle|\; \|\mathbf{x} - \mathbf{x}_i\| < \rho_S \;\; and \;\; \mathbf{x} \; is \; unlabeled \right\}$$

.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

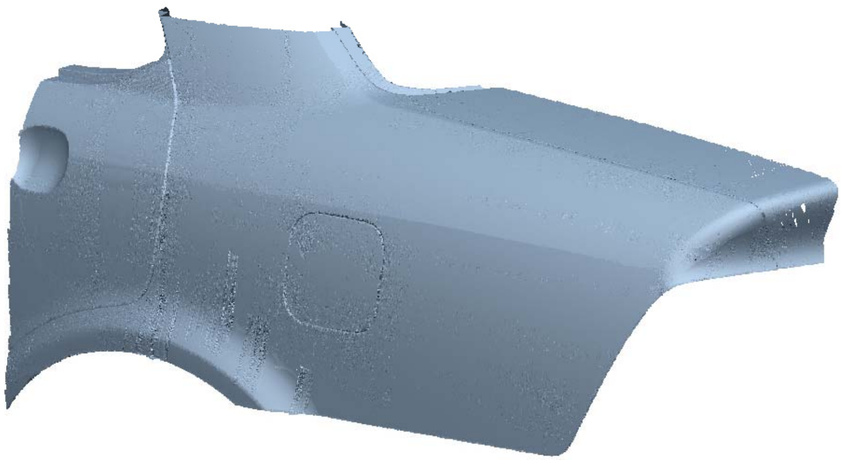
Results

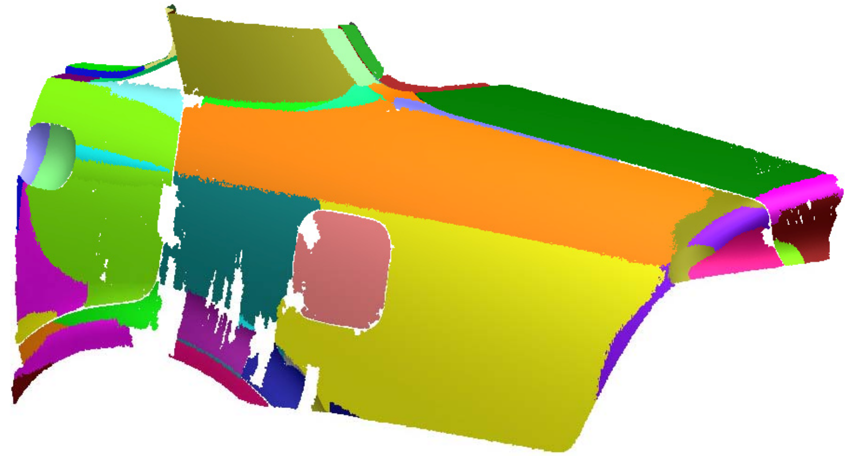
- We used a threshold distance of 0.3 mm
- Uses about 80 bytes per point, a fraction of the RAM required for processing a mesh
- Processes very large models in a few minutes

Model	Num. Points	Thresh. Angle	Peak RAM	Time (secs)
C-pillar	99,790	5°	25 MB	24
Golf Club	209,779	5°	33 MB	25
Rear Fender	1,065,886	6°	90 MB	255
Front Door	1,497,459	5°	125 MB	288

.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Results – Rear Fender

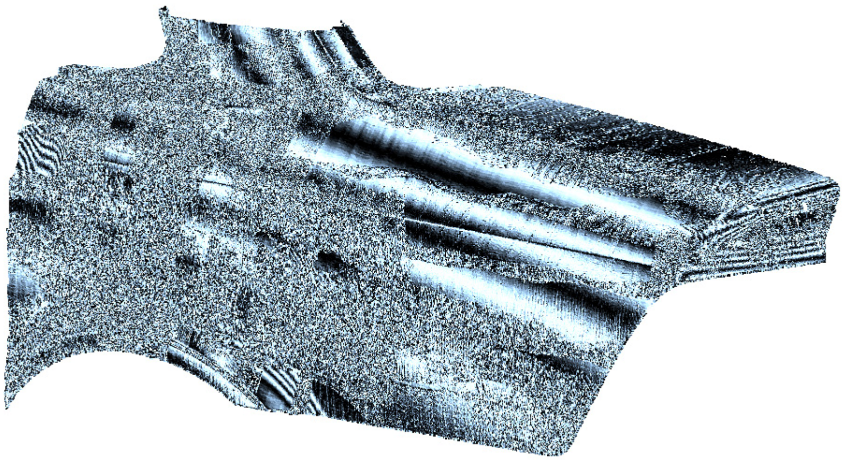


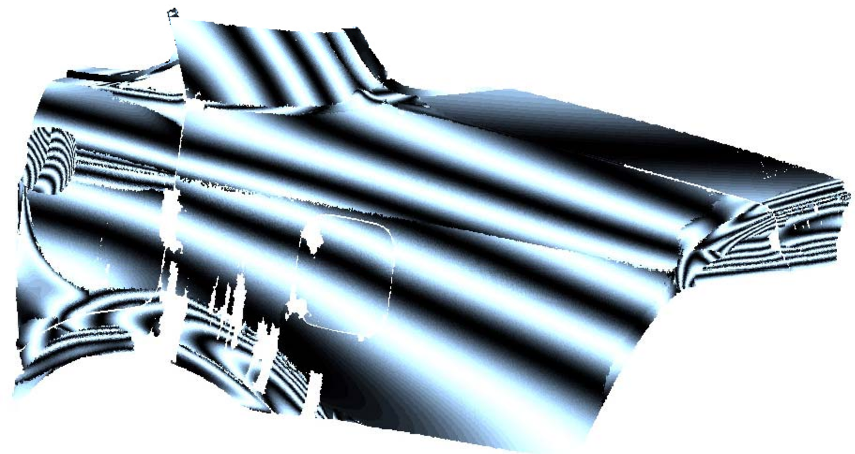


- 1,065,886 points
- 90 MB peak RAM
- 255 secs

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Results – Rear Fender





- Simulated reflection lines on raw data and extracted surfaces

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Algorithms

- Mesh Smoothing
- Surface Extraction from Meshes
- Surface Extraction from Point-Sampled Data
- Surface Reconstruction Proposal
- Summary

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Motivation

- Can we blend and intersect the extracted surfaces?
- Can we create a piecewise-smooth reconstruction?
- The reconstructed model must:
  - Approximate the data to engineering tolerances
  - Be free of unnecessary undulations (principle of simplest shape)
  - Preserve sharp edges and character lines
- Reconstruction useful for: aesthetic interrogation, engineering analysis, manufacturing

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Proposed Work Outline

- Extract surfaces from point data
- Subdivide the object's bounding box into cells
- Find surface approximations in each cell
- Convert surfaces into an implicit representation
- Intersect surfaces at sharp edges
- Use the **Partition of Unity Method** to blend the approximations in each cell
- Reconstructed surface is the solution of a single implicit equation
- Established methods to polygonize the surface

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Related Work

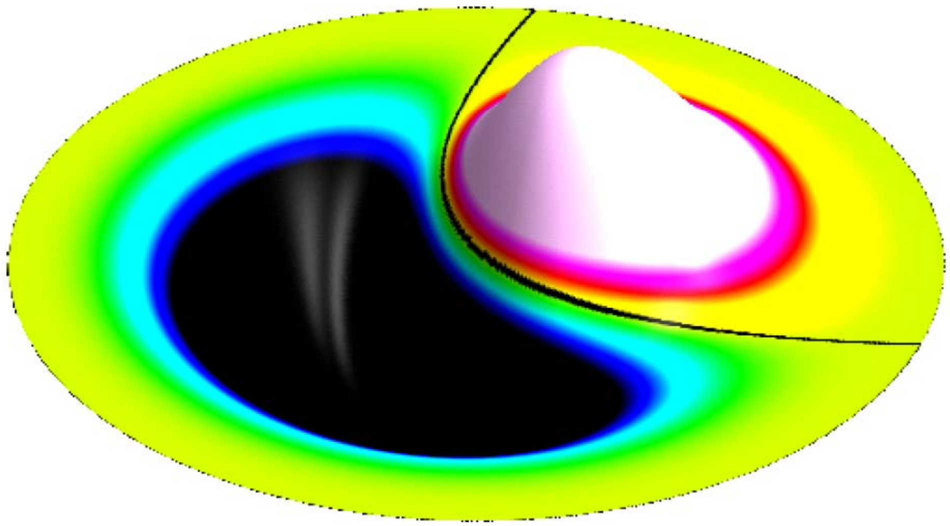
- An implicit surface is the locus of 3D points such that $F(\mathbf{x})=0$ for some scalar-valued function $F(\mathbf{x})$
- Blending and intersecting surfaces becomes trivial [1]
- Used to create meshes from point clouds [2,3]:
  - Compute a signed distance function as the distance from each point to the tangent plane of the closest point in the point cloud
  - Extract the surface by meshing the zero level set of the signed distance function
- Implicit representations have also been used to create C^1-smooth reconstructions with B-splines [4]

[1] Bloomenthal. *Introduction to Implicit Surfaces.* Morgan Kaufmann Publishers, 1997.
[2] Hoppe, et al. *Piecewise Smooth Surface Reconstruction, SIGGRAPH 94.*
[3] Curless & Levoy. *A Volumetric Method for Building Complex Models from Range Images, SIGGRAPH 96.*
[4] Bajaj, et al. *Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans, SIGGRAPH 95.*

Related Work - RBFs

- Approximate the points around various centers near the point cloud with a polynomial $h(u,v)$ and set $g_i(\mathbf{x})=w-h(u,v)$
- Sum the products of these local approximations with radial basis functions

$$f(\mathbf{x}) = \sum_i [g_i(\mathbf{x}) + \lambda_i]\phi_\sigma(\|\mathbf{x} - \mathbf{p}_i\|)$$



Ohtake, Y., Belyaev, A., and Seidel, H.-P. *A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions. Shape Modeling International 2003, pp. 153-161.*

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Related Work - PUM

- Partition of unity methods integrate locally defined approximations into a global approximation
  - Partition the global domain into small subdomains
  - Approximate the data in each subdomain
  - Blend the local solutions together with smooth weights that sum to one everywhere on the domain
- Originally developed to solve PDEs
- Used for computer graphics [1,2]
- Uses local representations that can be created and evaluated quickly

[1] Ohtake, et al. *Multi-Level Partition of Unity Implicits, SIGGRAPH 2003.*
[2] Tobor, et al. *Efficient Reconstruction of Large Scattered Geometric Datasets using the Partition of Unity and Radial Basis Functions, WSCG 2004.*

Related Work Summary

- We need to blend and intersect extracted surfaces
  - Difficult to automate with explicit representation
  - Impossible with RBFs
- Methods developed for computer graphics ignore constraints of automotive styling design
- Can we use partition of unity methods to satisfy accuracy or geometric continuity constraints?
- How can we ensure that the surfaces and edges of the reconstructed object satisfy the principle of simplest shape?

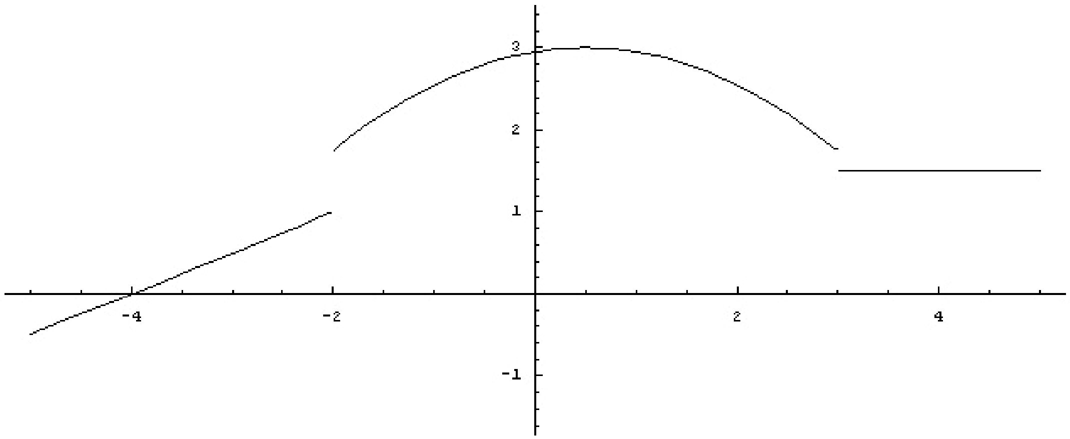
Partition of Unity Method

- On a bounded domain Ω, we want a set of nonnegative functions $\{\varphi_i\}$ with compact support such that
$$\sum_i \varphi_i = 1 \text{ on } \Omega$$
- Given a local approximation $V_i(x)$ in each subdomain, the global approximation is
$$f(x) = \sum_i \varphi_i(x) V_i(x)$$
- Given a set of nonnegative compactly supported functions $\{w_i(x)\}$ that cover the domain
$$\Omega \subset \bigcup_i \operatorname{supp}(w_i)$$
- This equation generates the partition of unity functions
$$\varphi_i(x) = \frac{w_i(x)}{\sum_j w_j(x)}$$

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

PUM – 1D Example

- Consider three $V_i(x)$ on the domain Ω=(5,5)

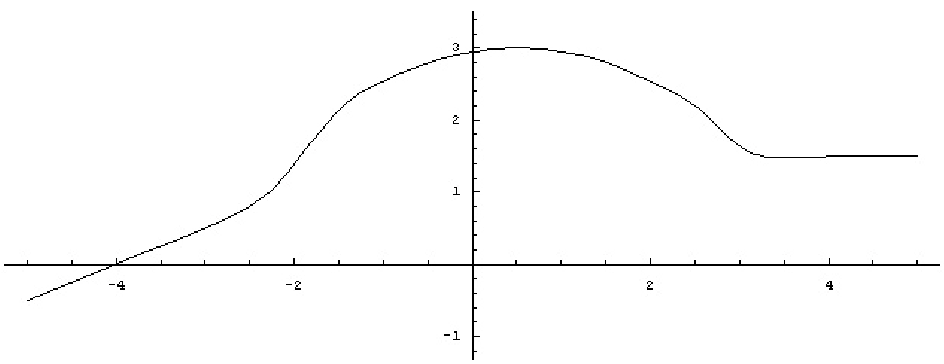


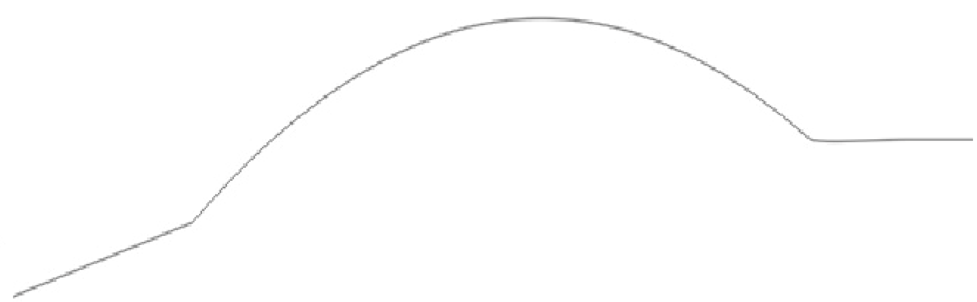
$$V_1(x) = \frac{x}{2} + 2 \qquad \text{for} \quad -5 < x < -2$$

$$V_2(x) = -\frac{1}{5}\left(x - \frac{1}{2}\right)^2 + 3 \qquad \text{for} \quad -2 < x < 3$$

$$V_3(x) = \frac{3}{2} \qquad \text{for} \quad 3 < x < 5,$$

- How can we easily blend and intersect them?

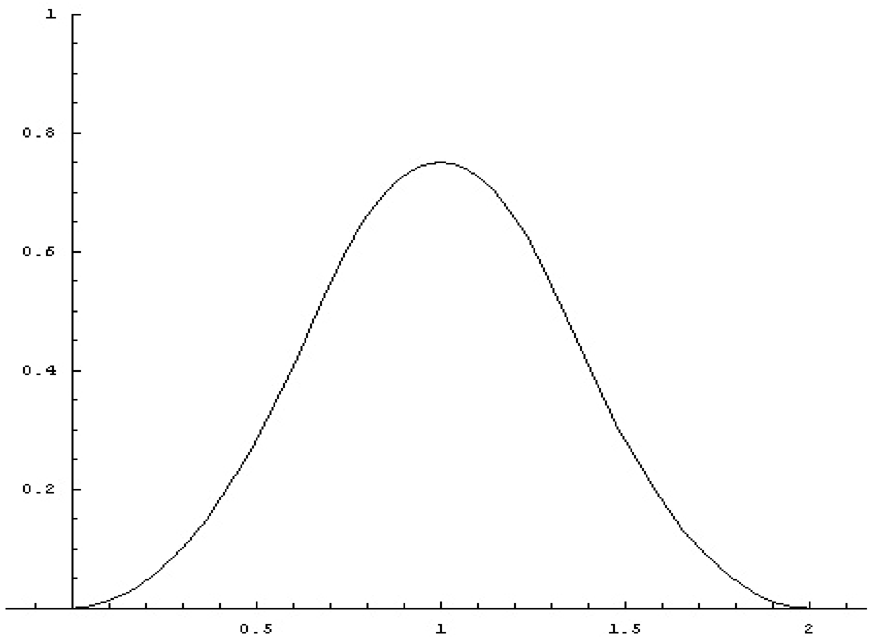




.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

PUM – 1D Example

- The partition of unity functions $\{\varphi_i\}$ are generated from the C^1 function



$$w(x,R)=\begin{cases}\frac{9}{8}\frac{x^2}{R^2}+\frac{9}{4}\frac{x}{R}+\frac{9}{8} & \text{for } -R\le x<-\frac{R}{3}\\ -\frac{9}{4}\frac{x^2}{R^2}+\frac{3}{4} & \text{for } -\frac{R}{3}\le x<\frac{R}{3}\\ \frac{9}{8}\frac{x^2}{R^2}-\frac{9}{4}\frac{x}{R}+\frac{9}{8} & \text{for } \frac{R}{3}\le x<R.\end{cases}$$

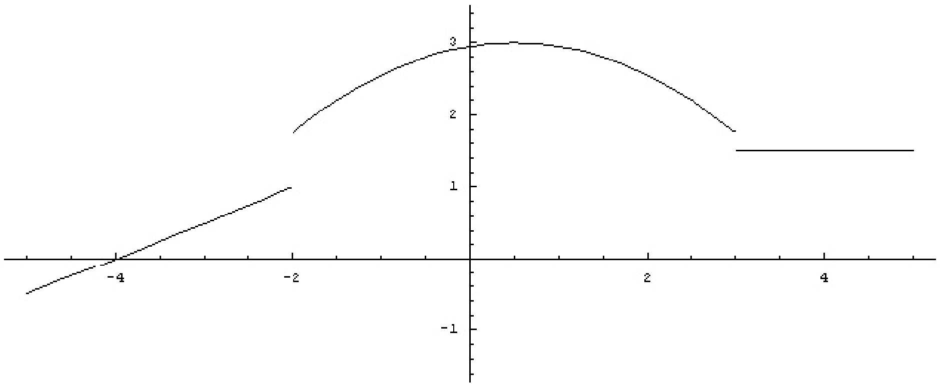
- We can use any positive functions with compact support
- The reconstruction inherits the continuity of the $\{\varphi_i\}$

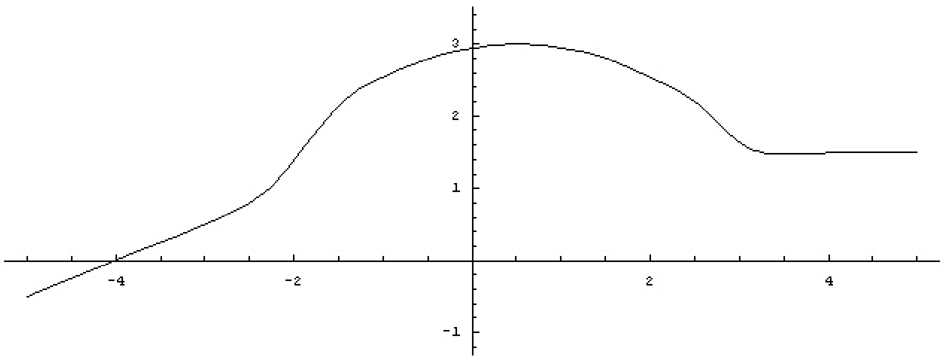
.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

PUM – 1D Example

- The blended global approximation is simply

$$f(x) = \frac{w_1 V_1 + w_2 V_2 + w_3 V_3}{w_1 + w_2 + w_3}$$



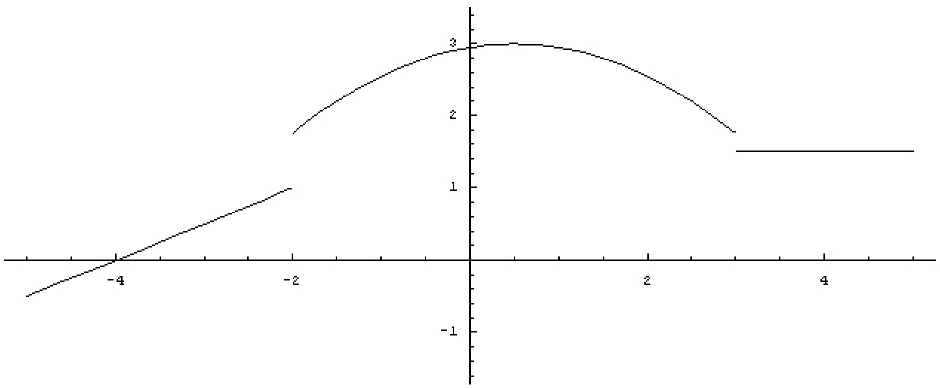


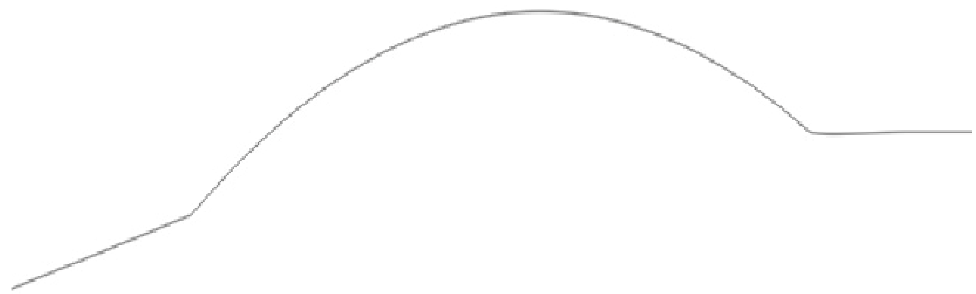
- We can control the strength of the blending by changing the parameter R of the $w_i(x,R)$

PUM – Implicit 1D Example

- Replace the $V_i(x)$ with y-$V_i(x)$ to define the blended function implicitly
- Now it's trivial to intersect the functions

$$\frac{\min[w_1(y-V_1),\ w_1(y-V_2)]+w_2(y-V_2)+\min[w_3(y-V_2),\ w_3(y-V_3)]}{w_1+w_2+w_3}=0$$

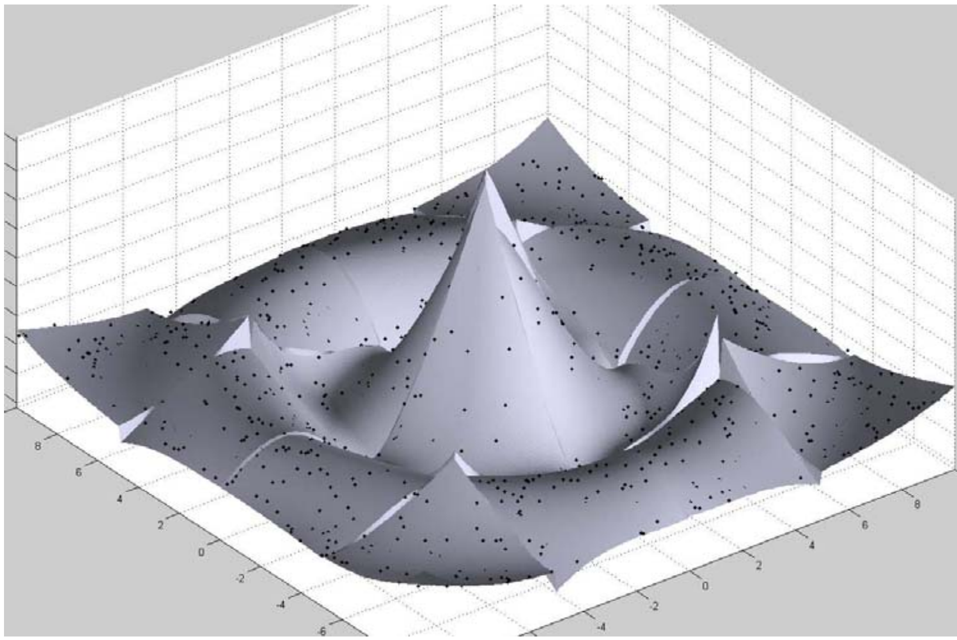


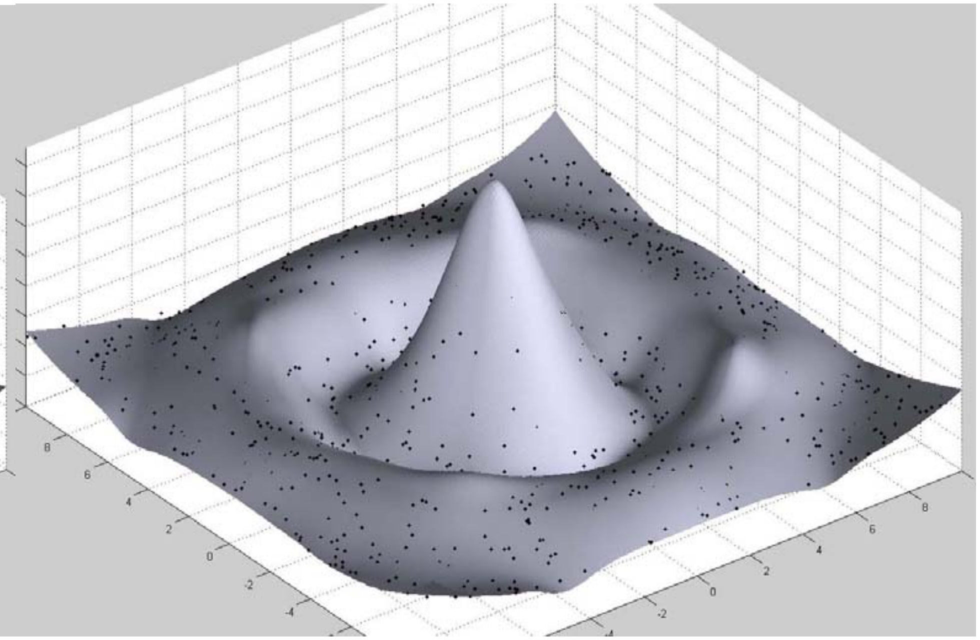


.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

PUM – 2D Example

- These ideas extend to 2- and 3-dimensions



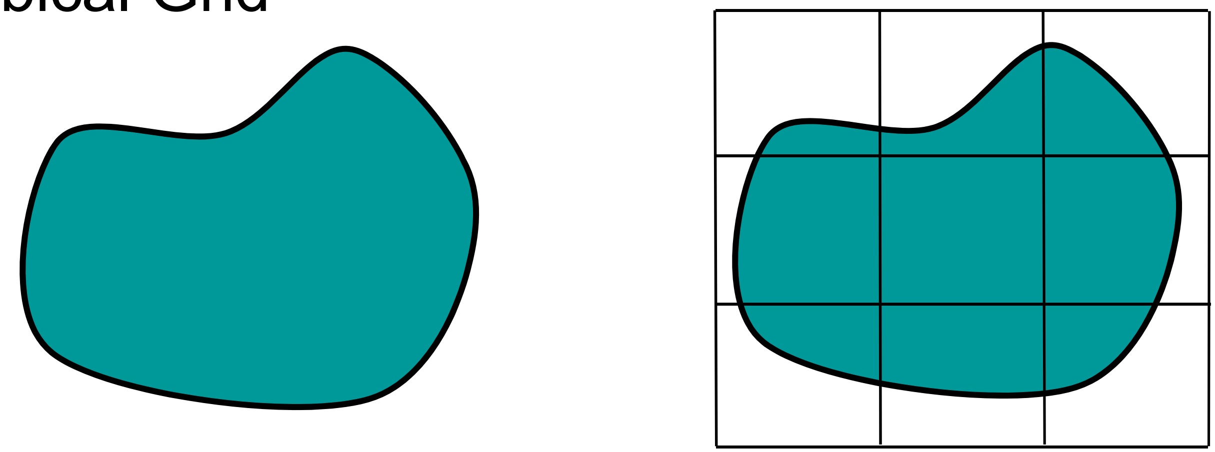


- 16 cubic surfaces approximating synthetic noisy data

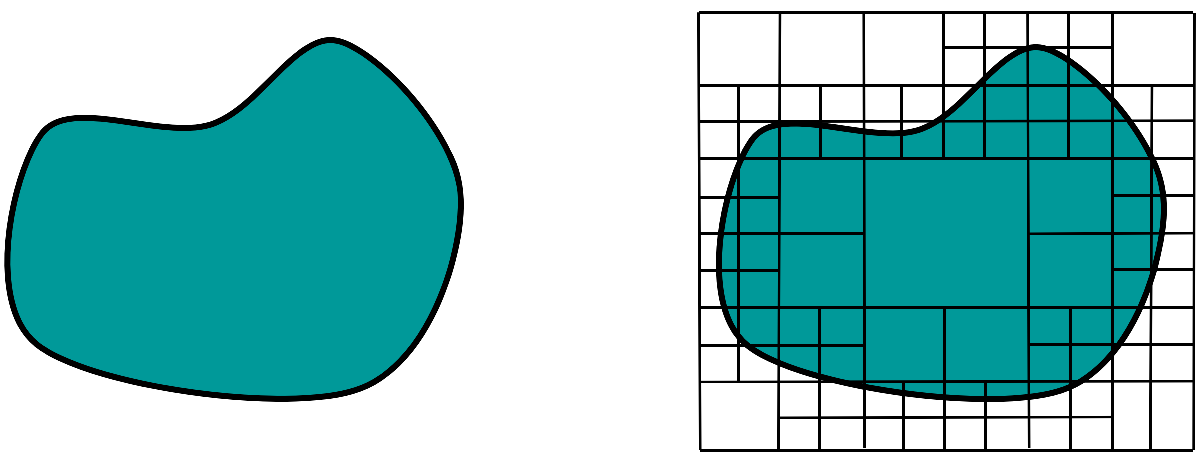
- C^2 blend of the 16 surfaces

Proposed Research – Domain Subdivision

- First step is to subdivide the domain into small cells
- Cubical Grid



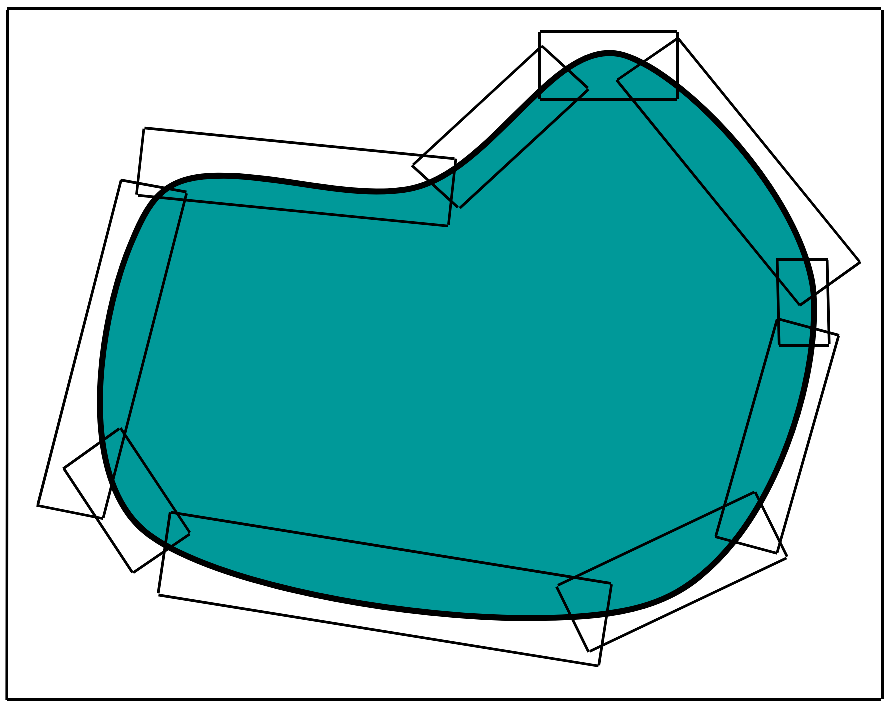
- Octree Subdivision



.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Proposed Research – Domain Subdivision

- Geometry-adaptive cells
  - Quality of the reconstructed surface varies with the size and orientation of the subdivision cells
  - We can use local geometry information to determine the size and orientation of the cells on the point cloud



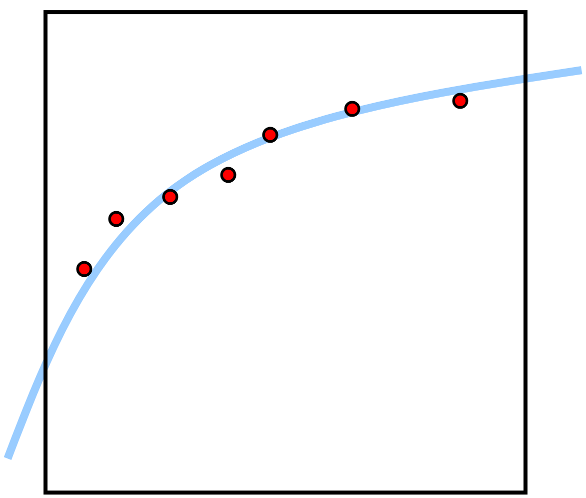
.CIELAB.
Computer Integrated Engineering Laboratory
Carnegie Mellon

Proposed Research – Domain Subdivision

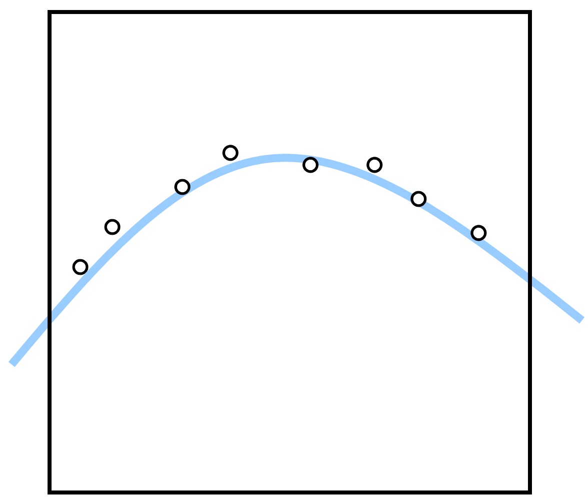
- Questions we will address:
  - Can we make cells that vary in size and orientation according to the object geometry?
  - How do we ensure the cells overlap?
  - How do we integrate them with the rest of the domain?
  - Will the computational cost and difficulty of finding the geometry-adaptive cells by justified by the improved quality of the reconstruction?

Proposed Research – Local Approximations

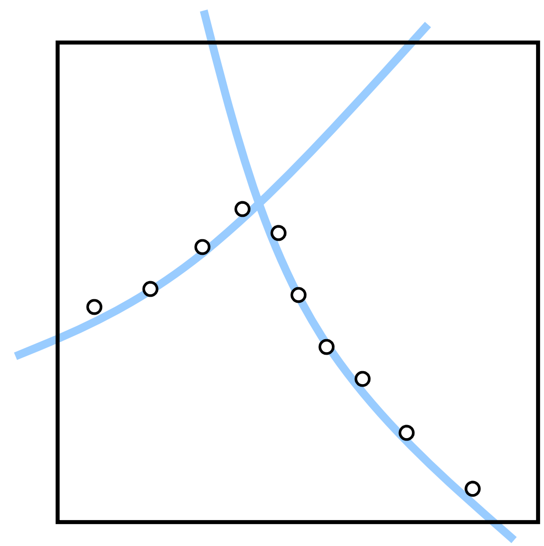
- Then we use the following information to determine the local approximations in each cell
  - Does it contain less than a certain number of points?
  - How many extracted surfaces are defined inside it?
  - Does it contain a sharp edge or corner?



Points already segmented



Points not segmented



Sharp edge detected

Proposed Research – Local Approximations

- Questions we will address:
  - How can we integrate geometry-adaptive cells with this approach?
  - What recursive subdivision rules will give the best results?
  - How will we deal with outlier noise?
  - How reliably can we detect sharp edges or corners and fit separate surfaces to the points around them?

Proposed Research – Isosurface Extraction

- The reconstructed surface is the zero level set of a signed distance function
- Extract isosurfaces from implicit functions
- Convert the approximation in each cell to implicit forms
- Multiply the approximations by the partition of unity functions and sum them over the domain
- For cells that do not contain points, we can estimate the signed distance function by finding the nearest point to the cell and examining its normal

Algorithms

- Mesh Smoothing
- Surface Extraction from Meshes
- Surface Extraction from Point-Sampled Data
- Surface Reconstruction Proposal
- Summary

Expected Contributions

- Mesh Smoothing
  - Creates an aesthetically pleasing surface
  - Minimizes mesh deformation
  - Preserves sharp edges
- Surface Extraction
  - Estimates curvatures and noise levels on the data
  - Detects sharp edges
  - Segments data into regions approximated by single surfaces
  - Fits surfaces to a specified tolerance
- Surface Reconstruction (proposed)
  - Will reconstruct objects with large, smooth surfaces, smooth blends between surfaces, and sharp edges
  - Will bound the accuracy and geometric continuity of the reconstructed object

.CIELAB.
Computer Integrated
Engineering Laboratory
Carnegie Mellon

Publications

- Mesh Smoothing
  - M. Vieira, K. Shimada, and T. Furuhata. *Local Least Squares Fitting for Surface Mesh Fairing in Automobile Panel Design, ASME DETC / DAC*. 2002, Montreal, Canada.
  - M. Vieira, K. Shimada, and T. Furuhata. *Smoothing of Noisy Laser Scanner Generated Meshes Using Polynomial Fitting and Neighborhood Erosion, ASME Journal of Mechanical Design*, May 2004.
- Surface Extraction
  - M. Vieira, K. Shimada. *Segmentation of Noisy Laser-Scanner Generated Meshes With Piecewise Polynomial Approximations, ASME DETC / DAC*. 2004, Salt Lake City, Utah. (Winner DAC Best Paper Award)
  - M. Vieira, K. Shimada. *Surface Mesh Segmentation and Smooth Surface Extraction Through Region Growing, Computer Aided Geometric Design Journal*, (accepted; submitted July 2004).
  - M. Vieira, K. Shimada. *Surface Extraction from Point-Sampled Data through Region Growing, International Journal of CAD/CAM*, (submitted).