Digital Geometry Processing

Poisson Mesh Reconstruction

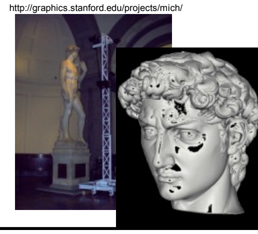
Poisson Surface Reconstruction

Motivation

In many domains, scanners are used to obtain virtual representations of 3D shapes

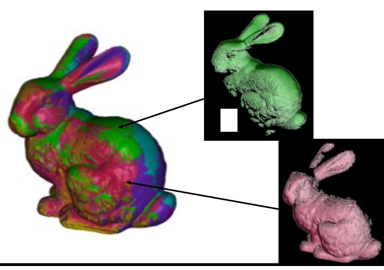
http://www.jhu.edu/digitalhammurabi/

http://graphics.stanford.edu/projects/mich/



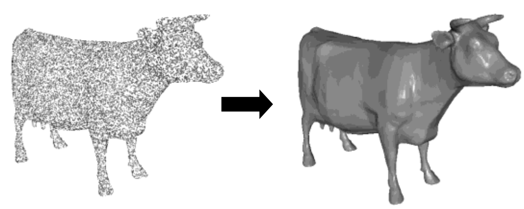
Motivation

Scanning often gives only local connectivity



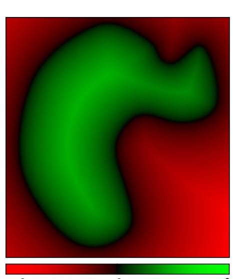
Surface Reconstruction

Generate a mesh from a set of surface samples



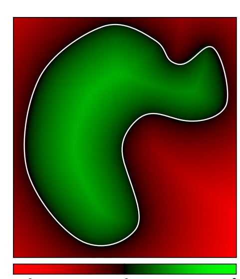
Implicit Function Approach

- Define a function with value less than zero outside the model and greater than zero inside



Implicit Function Approach

- Define a function with value less than zero outside the model and greater than zero inside
- Extract the zero-set



Digital Geometry Processing

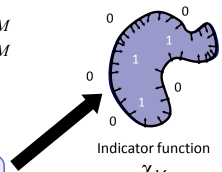
Poisson Mesh Reconstruction

The Indicator Function Approach

Solve for the **indicator function** of the shape *M*

$$\chi_M(p) = \begin{cases} 1 & \text{if } p \in M \\ 0 & \text{if } p \notin M \end{cases}$$

Shape boundary is contour $\chi_M = 0.5$



Indicator function

χ_M

The Poisson Equation

Given *n gradient* data pairs(x_i, g_i), where $x_i, g_i \in R^d$, find $f : R^d \to R$

such that $\nabla f(x_i) = g_i$. $(\nabla \varphi = \left(\frac{\partial \varphi}{\partial x_1}, \ldots, \frac{\partial \varphi}{\partial x_d}\right))$

If the system is overdetermined, solve the *normal equations*:

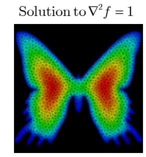
$$\nabla \bullet (\nabla f(x_i)) = \nabla \bullet g_i \quad (\nabla \bullet \vec{\varphi} = \frac{\partial \vec{\varphi}}{\partial x_1} + .. + \frac{\partial \vec{\varphi}}{\partial x_d})$$

or

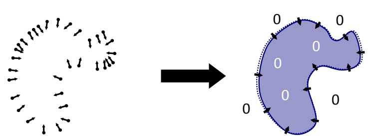
$$\boxed{\nabla^2 f(x_i) = \nabla \bullet g_i} \quad (\nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial x_1^2} + .. + \frac{\partial^2 \varphi}{\partial x_d^2})$$

possibly subject to some *boundary conditions.*

Solution to $\nabla^2 f = 1$



The Key Observation



Input
Oriented 3D points

Indicator gradient
$\nabla \chi_M$

The normal field *n* is **gradient** of "smoothed" χ_M :
$\nabla \hat{\chi}_M = n$

Solve Poisson equation for $\hat{\chi}_M$:
$\nabla^2 \hat{\chi}_M = \nabla \bullet n$

Solving the Poisson Equation

Represent *f* as $f(x) = \sum_{j=1}^{k} \alpha_j B_j(\| x - c_j \|)$

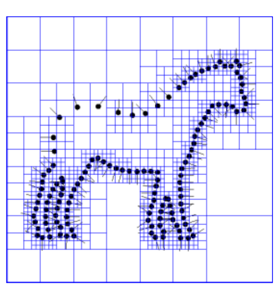
and solve a linear system for the coefficients α_j

Need to compute second derivatives of B_j

Implementation: Adapted Octree

Given the Points:

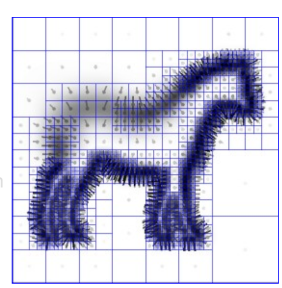
- Set octree
- Compute vector field
- Compute indicator function
- Extract iso-surface



Implementation: Vector Field

Given the Points:

- Set octree
- Compute vector field
  - Define a function space
  - Splat the samples
- Compute indicator function
- Extract iso-surface



Implementation: Indicator Function

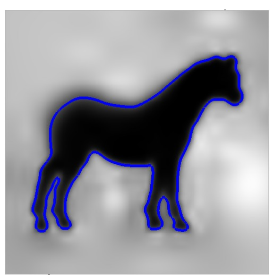
Given the Points:
- Set octree
- Compute vector field
- Compute indicator function
  - Compute divergence
  - Solve Poisson equation
- Extract iso-surface

Implementation: Indicator Function

Given the Points:
- Set octree
- Compute vector field
- Compute indicator function
  - Compute divergence
  - Solve Poisson equation
- Extract iso-surface

Implementation: Surface Extraction

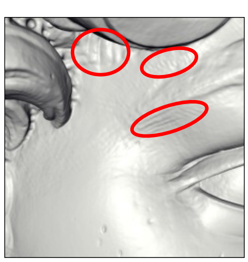
Given the Points:
- Set octree
- Compute vector field
- Compute indicator function
- Extract iso-surface



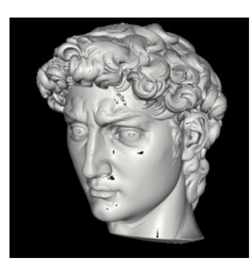
Michelangelo's David

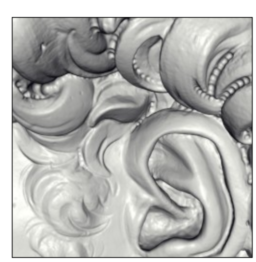
- 215 million data points from 1000 scans
- 22 million triangle reconstruction
- Compute Time: 2.1 hours
- Peak Memory: 6600MB

David – Chisel marks

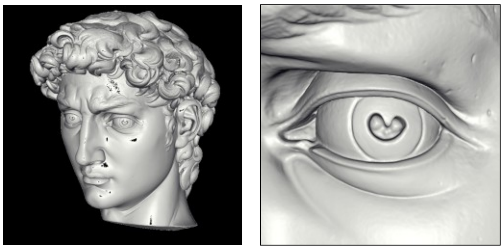


David – Drill Marks

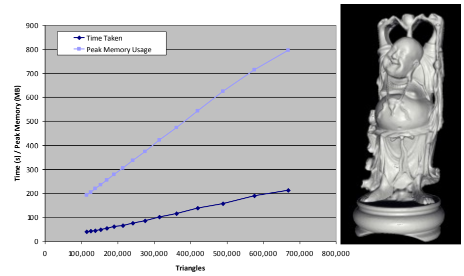




David – Eye



Scalability – Buddha Model



Stanford Bunny