An Interactive Approach to Point Could Triangulation

Chakrit Watcharopas

Outline

- Introduction
- Virtual Range Scanning
- Post Processing
- Masking
- Stitching
- Demo

Introduction

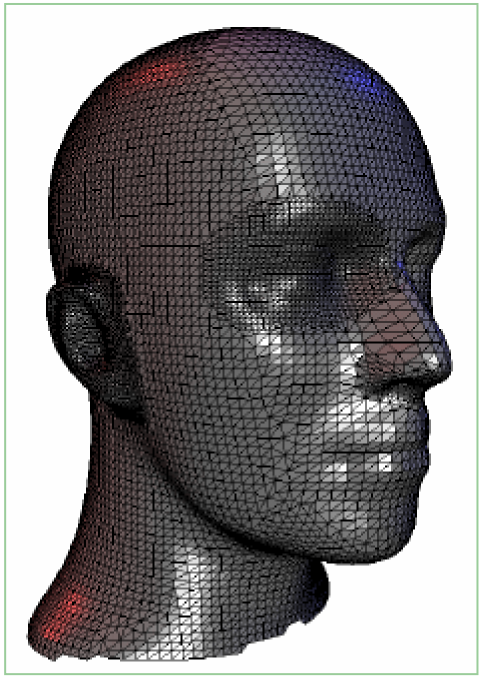
- 3D scanners are becoming the standard source for geometric input data
- Two well-known approaches
  - The distance field approach [Hoppe]
  - Start by finding the tangent plane for every sample point
  - Then sample the distance field on a regular spatial grid by the Marching Cubes algorithm
  - It's very important to find a consistent orientation of the normal vectors of the tangent planes

Introduction

- Two well-known approaches (Cont.)
  - The Voronoi-based approach [Amenta]
  - Use the shape of individual Voronoi cells to determine the surface normal direction at every surface point
  - The requirements in terms of computation time and memory are quite high such that massive data set with millions of data points cannot be processed with reasonable effort
- These two are off-line algorithms

Introduction

- An interactive approach incorporates user interaction during the surface generation
- Resolution and orientation of the triangles can be adapted manually to varying detail levels and quality requirements in different regions of the object



Introduction

- The CPU and storage requirements are much lower than for the other approaches since no additional data structure has to be generated
- The goal is identify operations that can be performed by the graphics hardware to exploit its superior computing performance
- One iteration consists of
  - Placing, Scaling (→Resolution), and Orientation (→ alignment) the object on the screen
  - Determining the valid region of interest
  - Extracting the patch and automatically stitching it to the already existing mesh

Virtual Range Scanning

- The concept behind the user interface is to simulate a virtual 3D scanning device (much more flexible than a real 3D scanner)
- This device allows the user to rotate the given geometry on the screen in order to adjust the optimal viewing direction
- With OpenGL, we render the geometry with enabled z-buffer option
- If the screen resolution exceeds the sampling density such that the number of holes increases, samples can be drawn as large points (glPointSize ())
- A 3D point is usually transformed by the Modelview and the Projection matrix and eventually mapped onto the screen by the Viewport transformation

Virtual Range Scanning

- Viewport Transformation

$$\begin{pmatrix} i \\ j \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{w}{2} & 0 & 0 & \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & \frac{h}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

- The difficulty may arise from the rounding step which is implicitly introduced by assigning real coordinate values to integer indexed pixel locations, therefore we use the modified inverse

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{w} & 0 & 0 & \frac{1}{w}-1 \\ 0 & \frac{2}{h} & 0 & \frac{1}{h}-1 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ z' \\ 1 \end{pmatrix}$$

to un-project $[i+\frac{1}{2}, j+\frac{1}{2}, z']$ instead of $[i, j, z']$

Virtual Range Scanning

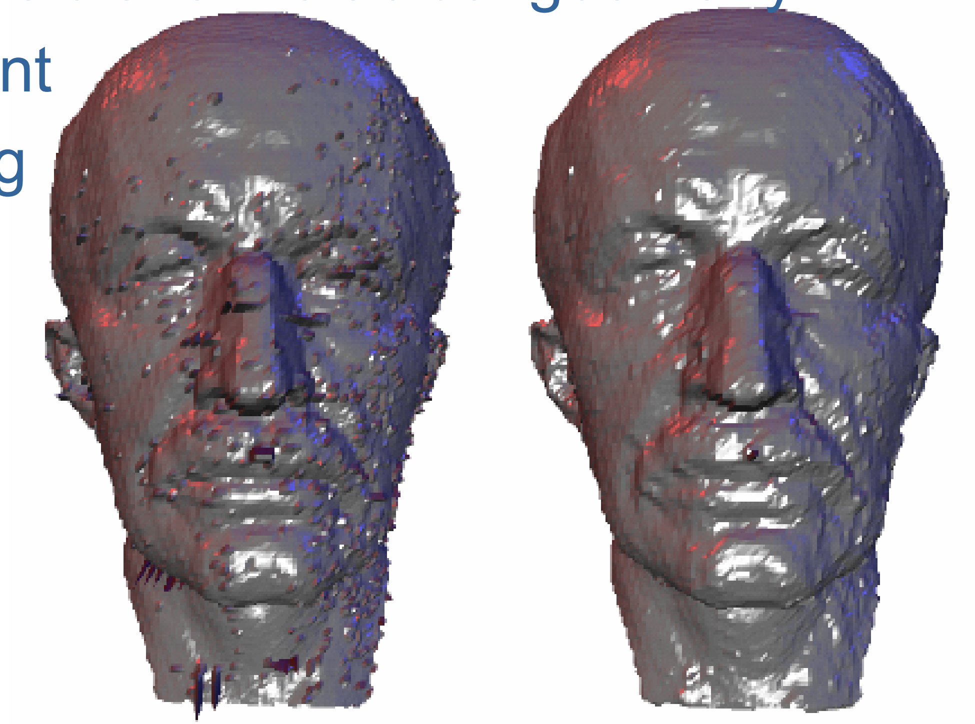
```
glReadPixels (0,0,width,height,GL_DEPTH_COMPONENT,GL_FLOAT,depth_buffer);
glGetFloatv (GL_PROJECTION_MATRIX,P);
TransposeMatrix (P);
glGetFloatv (GL_MODELVIEW_MATRIX,M);
TransposeMatrix (M);
MxM (P,M,PM);
MxM (V,PM,VPM);
```

$(VPM)^{-1}$ = InverseMatrix (VPM);

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = (\mathbf{VPM})^{-1} \begin{pmatrix} i \\ j \\ z' \\ 1 \end{pmatrix}$$

Post Processing

- Besides the noise in the original point data, we can observe additional jitter which is due to discretization errors in the z-buffer
- We minimize this effect by placing the near and far planes as close as possible to the actual geometry
- Additional enhancement is achieved by applying low-pass filter to the depth buffer

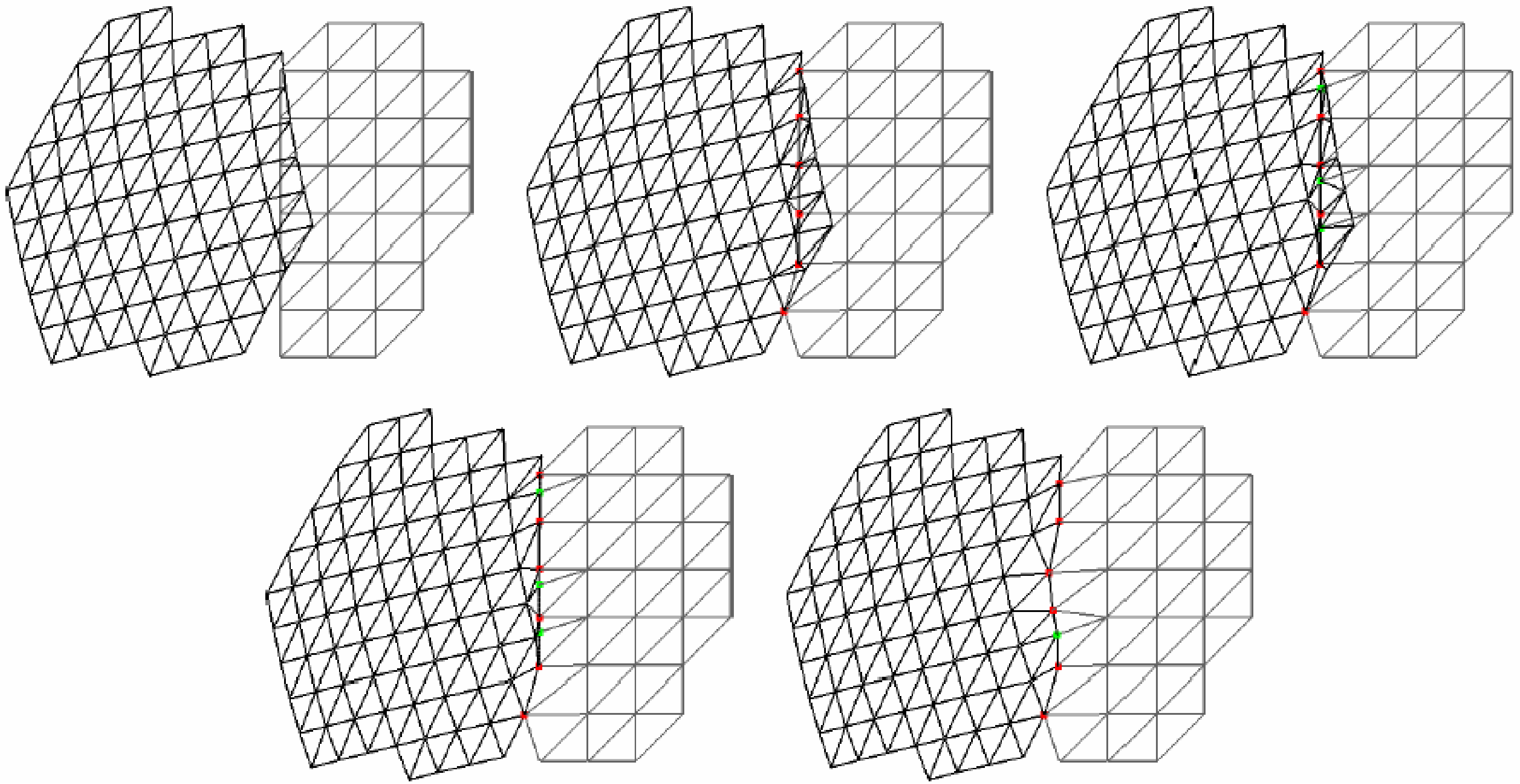


Masking

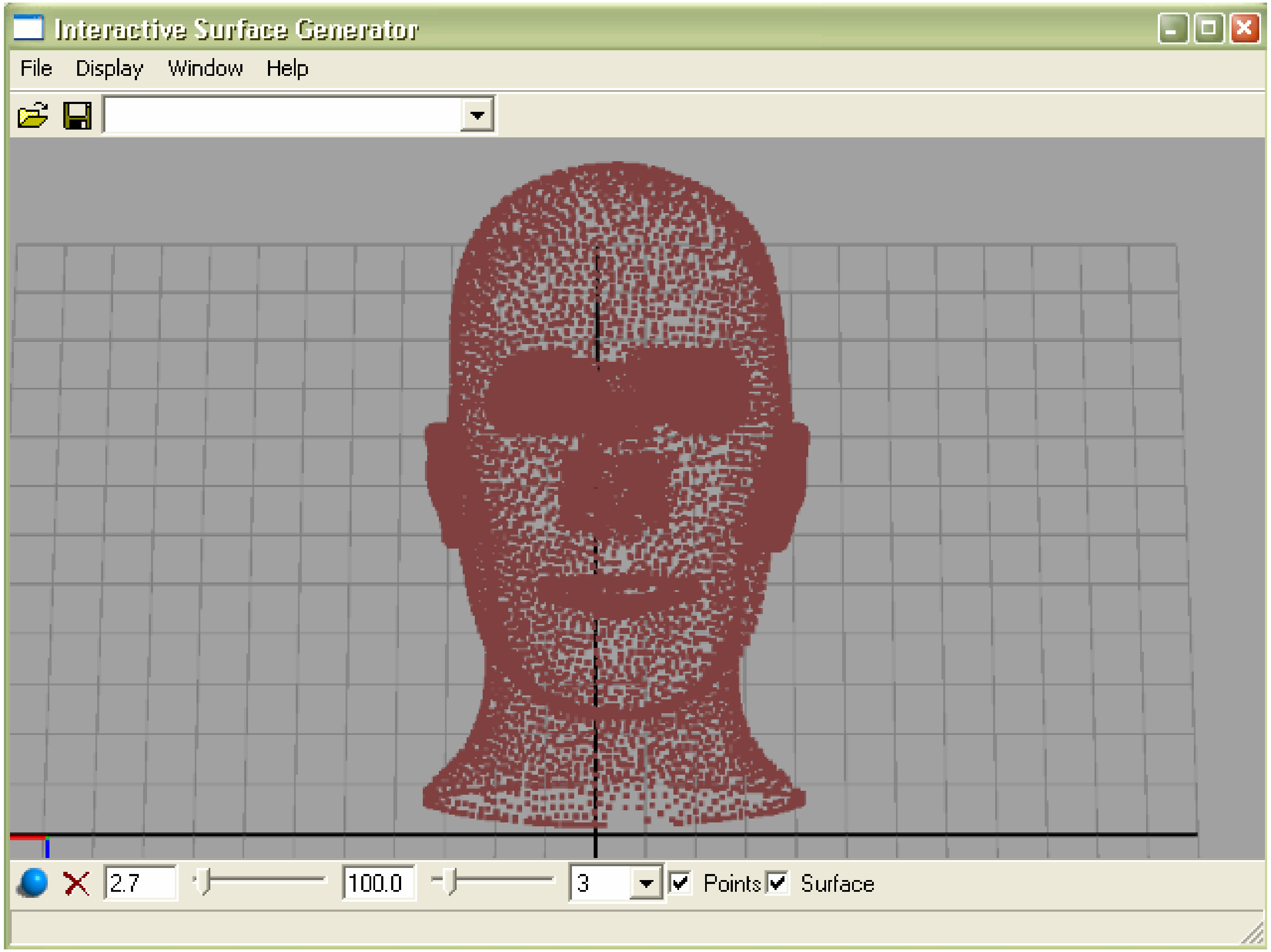
- The masking phase selects the relevant part of the sampled points and discards the rest
- One mechanism for the masking is the user defined region of interest
- The region of interest is a portion of the screen defined by the user
- We use the stencil buffer to block all the pixels outside this area

Stitching

- The final step in the interactive loop is to join the newly acquired geometry with the already existing one (using the modified *zippering algorithm*)



Demo


Interactive Surface Generator
File Display Window Help
2.7
100.0
3
Points
Surface