Geometry Processing
Dr Cengiz Öztireli
Sources of Geometry

Acquisition from the real world

Modeling applications
Shape Acquisition

• Digitizing real world objects

3D Scanning  Registration  Pre-processing  Reconstruction
Shape Acquisition

• 3D Scanning
  Touch Probes
  + Precise
  - Small objects
  Optical Scanning
  + Fast
  - Glossy objects

Active

Passive
Shape Acquisition

• Optical Scanning – Active Systems

LIDAR

Measures the time it takes the laser beam to hit the object and come back

Triangulation Laser

Projected laser beam is photographed, giving the distance of the pattern
Shape Acquisition

• Optical Scanning – Passive Systems

\[(x, y, z)\]

Multi-view Stereo

Left camera focal point

Right camera focal point

Epipolar line

\[(x_L, y_L)\]

\[(x_R, y_R)\]

Left camera projection plane

Right camera projection plane
Shape Acquisition

• Registration
  – Bringing scans into a common coordinate frame
Shape Acquisition

• Registration

Iterative Closest Point Algorithms

Patches to be aligned

Correspondences

Rigid motion

Alignment

Iterate
Shape Acquisition

• Registration

Feature-based Methods

Patches to be aligned
Compute descriptors
Match descriptors
Alignment
Shape Acquisition

• Pre-processing
  – Cleaning, repairing, resampling
Shape Acquisition

• Pre-processing
  – Sampling for accurate reconstructions
Shape Acquisition

• Reconstruction
  – Mathematical representation for a shape
Shape Acquisition

• Reconstruction

Connect-the-points Methods
- + Theoretical error bounds
- - Expensive
- - Not robust to noise

Approximation-based Methods
- + Efficient to compute
- + Robust to noise
- - No theoretical error bounds
Shape Acquisition

• Approximating an implicit function

\[ f : \mathbb{R}^3 \rightarrow \mathbb{R} \]

with value \( > 0 \) outside the shape and \( < 0 \) inside
Shape Acquisition

• Approximating an implicit function

\[ f : \mathbb{R}^3 \rightarrow \mathbb{R} \]

with value > 0 outside the shape and < 0 inside

\[ \{ \mathbf{x} : f(\mathbf{x}) = 0 \} \]

extract zero set
Texture Mapping

Parametrization
Texture Mapping

Each point \((x, y, z)\) on the surface has mapped coordinates \((u, v)\) in the texture image:

\[
P : M \rightarrow [0, 1] \times [0, 1]
\]

\[
P(x, y, z) = (u, v)
\]
Texture Mapping

Texture itself is a function:

\[ T : [0, 1] \times [0, 1] \rightarrow \text{RGB} \]

\[ T(u, v) = (r, g, b) \]
Texture Mapping

Concatenation of the two functions:

\[ \text{Color}(x, y, z) = T(P(x, y, z)) \]
Parametrization

Texture image
Parametrization

\[ \mathbf{p}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}, \quad (u, v) \in \mathbb{R}^2 \]
Parametrization

\[ p(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}, \quad (u, v) \in \mathbb{R}^2 \]

\[ p_u = \frac{\partial p(u, v)}{\partial u}, \quad p_v = \frac{\partial p(u, v)}{\partial v} \]

\[ I = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} p_u^T p_u & p_u^T p_v \\ p_u^T p_v & p_v^T p_v \end{pmatrix} \]
Parametrization

\[ I = \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} p_u^T p_u & p_u^T p_v \\ p_u^T p_v & p_v^T p_v \end{pmatrix} \]

- **Angle change**
- **Length change**

Area distortion: \[ dA = \sqrt{EG - F^2} \, du \, dv \]
Parametrization

Conformal parametrization (angle preservation)

\[ I = \begin{pmatrix} p_u^T p_u & p_u^T p_v \\ p_v^T p_u & p_v^T p_v \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \]
Editing Geometry

• Modeling tools

Sculpting  CAD/CAM  Procedural
Editing Geometry

• Interactive & sketch-based interfaces
Editing Geometry

- Deformations

Free-form  Elastic  Skeletal  Structure-aware

More structure
Editing Geometry

• Cutting & fracturing
Editing Geometry

• Smoothing & filtering
Editing Geometry

• Compression & Simplification