Surface Approximation
with
Triangle Meshes

Outline

- Classification of surfaces
- Approximating surfaces with triangle meshes
- Encoding triangle meshes
- Compressed mesh representations
What is a surface?

- A surface $S$ embedded in space is a subset of $\mathbb{R}^3$ that is intrinsically two-dimensional.

- For any neighborhood $u_p$ of a point $P$ on $S$:
  - $u_p$ contains at least half of an open disk (i.e., no part of $S$ is less than two-dimensional).
  - $u_p$ does not contain any open ball (i.e., no part of $S$ is solid).

Surface Modeling

- Surfaces defined in the continuum:
  - Sources: mathematics, CAD
  - Problem: a finite (digital) representation is necessary for surface analysis and rendering.

- Surfaces known at a finite set of points:
  - Source: sampling (range scanners, photogrammetry, medical data), simulation (finite element methods)
  - Problem: a surface in the continuum must be defined through a reconstruction process.
Topological Characterization of Surfaces

- Manifold without boundary

A surface $S$ in $\mathbb{R}^3$ such that any point on $S$ has an open neighborhood homeomorphic to an open disk in $\mathbb{R}^2$.

- Manifold with boundary

A surface $S$ in $\mathbb{R}^3$ such that any point on $S$ has an open neighborhood homeomorphic to an open disk or to half an open disk in $\mathbb{R}^2$. 
...Topological Characterization of Surfaces...

- An example of a non-manifold situation

Geometric Representation of Surfaces

- Implicit form:
  
  An implicit surface is the locus of solutions of an equation
  
  \[ F(x,y,z) = 0 \]

  where \( F \) is a mathematical expression of three variables

  Problems:
  
  - definition is too general: some expressions give objects that are not intrinsically two-dimensional
  - we might not know expression \( F(x,y,z) \)
  - even if we know \( F \), we might not be able to solve the equation

  Remark: surfaces which cannot be described in an analytic form are called free-form surfaces
Parametric form: a parametric patch is the image of a continuous function

\[ \psi: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3 \]

- \( \Omega \) is called parametric space
- \( \mathbb{R}^3 \) is called physical space
- boundaries of \( \Omega \) and of \( \psi(\Omega) \) are formed by trimming curves

Parametric surface: a collection of parametric patches properly abutting
...Geometric Representation of Surfaces...

- Explicit surfaces:
  - Special case of a parametric surface
  - A surface can be represented as a bivariate function when it is
    the image of a scalar field

\[ \phi: \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R} \]

- Example: topographic surfaces

![Topographic Surface Diagram]

Hypersurfaces

- Generalization of explicit surfaces to higher dimensions
- Image of a scalar field

\[ \phi: \Omega \rightarrow \mathbb{R} \]

where \( \Omega \) is a compact domain in \( \mathbb{R}^k \)

- Example: volume data (for \( k=3 \))
Approximating surfaces with triangle meshes

- Surface representation:
  mesh of triangles (i.e., a set of triangles such that any two of them either do not intersect or share a common edge or vertex)
- Each triangle approximates a surface patch within a given accuracy
- Triangle meshes are easy to represent, manipulate, visualize
- Triangle meshes can be constructed from irregularly sampled data

Approximating a 2-dimensional scalar field with a triangle mesh

$K=2$: 

- A 2-dimensional scalar field is described as a function $z = \phi(x,y)$
- A triangle meshes in 3D is obtained by triangulating the domain of $\phi$ and lifting it to three-dimensional space
Approximating a 3-dimensional scalar field with a tetrahedral mesh

\( k=3:\)

- A 3-dimensional scalar field is defined by a function \( z = f(x_1, x_2, x_3) \)
- An approximation is obtained by discretizing the domain of \( f \) with a tetrahedral mesh
- A \textit{tetrahedral mesh} is a set of tetrahedra such that any two of them either do not intersect or share a common face, edge or vertex

Approximating a \( k \)-dimensional scalar field with a simplicial mesh

- Discretization of the domain of a \( k \)-dimensional field \( z = f(x_1, x_2, \ldots, x_k) \) with a simplicial mesh
- Defines a linear approximation of \( f \) in \((k+1)\)-dimensional Euclidean space
How to compute the approximation?

- How well does a mesh approximate a given surface?

- We are not given surfaces but:
  - mesh of triangles for free-form surfaces
  - set of points at which the field is known for scalar fields (hypersurfaces)

The Error Metric

- Euclidean distance between a point $p$ and a set $Q \subseteq \mathbb{R}^n$:

  \[ d(p,Q) = \inf \{ d(p,q) \mid q \in Q \} \]

  where $d(p,q)$ is the Euclidean distance between point $p$ and $q$

- "Distance" from a set $P$ to a set $Q$

  \[ d_E(P,Q) = \sup \{ d(p,Q) \mid p \in P \} \]

  However $d_E(P,Q) \not< d_E(Q,P)$
Hausdorff distance defined as:

\[ d_H(P, Q) = \max \{ d_E(P, Q), d_E(Q, P) \} \]

It follows that \( d_H(P, Q) = 0 \) iff \( P = Q \)

Thus, we can express the distance between a surface \( S \) and its approximating triangle mesh \( T \) as \( d_H(S, T) \)

Discrete case

- Free-form surfaces:
  - Surface \( S \) given as a fine mesh of triangles \( T_s \)
  - \( \Rightarrow \) we measure distance between two triangle meshes

- Hypersurfaces:
  - Scalar field \( \phi \) is known at a finite set of points \( Q \)
  - \( \Rightarrow \) we measure the distance of the points of \( Q \) from the triangle mesh \( T \) approximating the hypersurface
The *Metro tool* [Cignoni et al. 98]

Given two triangular meshes $T_1$ and $T_2$, *Metro*:

- scan converts each triangle $t$ of $T_1$ with a user-selected scan step, or, alternatively, chooses a set $P$ of points distributed randomly on $t$
- for each point $p$ in $P$, computes $d(p,T_2)$ (distances are computed efficiently using a bucketing data structure)

and switches meshes to be symmetric.

- precision of the evaluation depends on *sampling resolution*!
- with a sufficiently fine sampling step, almost equal results in both directions (e.g., 0.01% of mesh bounding box diagonal)

Metro returns

- accurate **numerical** distance estimation
- a **visual** representation of the approximation error

- tool runs on SGI ws (OpenInventor)
- available in public domain
...The Error Metric...

Approximating the error on scalar fields

- Error of a point $p$ of set $Q$ defined as

$$ e(p) = | \phi(p) - \phi_T(p) | $$

where

- $\phi(p)$ is the known value of the field at $p$
- $\phi_T(p)$ is the approximated value of the field at $p$ computed on the basis of simplicial mesh $T$

---

Error function defined by a discrete norm:

- $E(T,Q) = \| e(p) \|$
- $E(T,Q) = \max \{ e(p), p \in Q \}$

- Example: two-dimensional scalar field (terrain)
Remarks

- More accurate representation ⇒ more triangles
- More triangles ⇒ higher storage and processing time

Trade-off between accuracy and space/time:

- adapting the accuracy to the needs of an application can improve efficiency
- accuracy might be variable over different portions of the object