Mesh-Based Modeling

Amaël Delaunoy
3D Photography,
ETH Zürich, May 2013
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 18</td>
<td>Introduction</td>
</tr>
<tr>
<td>Feb 25</td>
<td>Lecture: Geometry, Camera Model, Calibration</td>
</tr>
<tr>
<td>Mar 4</td>
<td>Lecture: Features, Tracking/Matching</td>
</tr>
<tr>
<td>Mar 11</td>
<td>Project Proposals by Students</td>
</tr>
<tr>
<td>Mar 18</td>
<td>Lecture: Epipolar Geometry</td>
</tr>
<tr>
<td>Mar 25</td>
<td>Lecture: Stereo Vision</td>
</tr>
<tr>
<td>Apr 1</td>
<td>Easter</td>
</tr>
<tr>
<td>Apr 8</td>
<td>Short lecture “Stereo Vision (2)” + 2 papers</td>
</tr>
<tr>
<td>Apr 15</td>
<td>Project Updates</td>
</tr>
<tr>
<td>Apr 22</td>
<td>Short lecture “Active Ranging, Structured Light” + 2 papers</td>
</tr>
<tr>
<td>Apr 29</td>
<td>Short lecture “Volumetric Modeling” + 1 paper</td>
</tr>
<tr>
<td>May 6</td>
<td>Short lecture “Mesh-based Modeling” + 2 papers</td>
</tr>
<tr>
<td>May 13</td>
<td>Lecture: Structure from Motion, SLAM</td>
</tr>
<tr>
<td>May 20</td>
<td>Pentecost / White Monday</td>
</tr>
<tr>
<td>May 27</td>
<td>Final Demos</td>
</tr>
</tbody>
</table>
Last Time’s Class

Volumetric 3D reconstruction
Convex Modeling

Marching cubes
Today’s Class

Modeling 3D surfaces by polyhedral surfaces. In particular:

- Mesh representations
- Extract a mesh from silhouettes → Exact visual hull
- Extract a mesh from visual data (point clouds)
- Mesh optimization and refinement
Using Meshes

- Intensively used in Computer Graphics
- GPU Friendly and a lot of available post-processing tools
- Triangular meshes / Polyhedral surfaces
- Modeling the problem directly with the final representation
Mesh Representation

- Explicit representation / Compact representation
  - Non uniform sampling → Meshes do not rely on space discretization (like a volume grid)
  - The modeled surface can be directly represented as a triangular (or polyhedral) mesh or as an isosurface of the labeling (implicit) function defined on a tetrahedrization of the space.
  - No need for additional conversion (like Marching Cubes)
  - Represented as a set of connected vertices (a graph)
# Mesh vs Volumetric

<table>
<thead>
<tr>
<th></th>
<th>Meshes</th>
<th>Volumetric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Discretization</td>
<td>Adaptive</td>
<td>Yes</td>
</tr>
<tr>
<td>Topology Handling</td>
<td>Difficult (Self intersections,...)</td>
<td>Naturally handled</td>
</tr>
<tr>
<td>Memory</td>
<td>Compact, Limited</td>
<td>Large</td>
</tr>
<tr>
<td>Parallelization</td>
<td>Sometimes</td>
<td>Very good</td>
</tr>
<tr>
<td>Scalability</td>
<td>Very good</td>
<td>Difficult</td>
</tr>
<tr>
<td>Adaptive Resolution</td>
<td>Very good</td>
<td>Difficult (Octree, Narrow band)</td>
</tr>
<tr>
<td>Surface extraction</td>
<td>Natural</td>
<td>Precision Loss (Marching cubes)</td>
</tr>
</tbody>
</table>

**ETH**
Visual Hull from Silhouettes

Slides on graph-cut from Edmond Boyer
http://morpheo.inrialpes.fr/people/Boyer/
Using volumetric approaches

Criteria per voxel

Precision VS complexity
Exact Visual Hull Reconstruction

Volumetric Approaches
- voxels

Surface based Approaches

Image based Approaches
Exact Visual Hull Reconstruction from Silhouettes

- Franco & Boyer 2003

Viewing segments → The mesh connecting viewing segments → Facets by going along the oriented mesh

- Discrete visual hull
- Viewing cone
- Cone intersection edges
- Viewing edges
- Triple point
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Exact Visual Hull Reconstruction
Results - Examples


- Allows real-time performances
Application Example

INRIA Grenoble

4D View Solutions – http://www.4dviews.com
Multi-view Stereo Problem

- Given a set of observed calibrated views, how to find the 3D shape that best fits the input images.
- Inverse problem of image rendering.

Middlebury Data - [Seitz et al. 2006]
3D Reconstruction using Multi-view Stereo

- Structure from Motion
- Sparse Bundle Adjustment
- Dense 3D Points
- Surface Extraction
- Mesh Refinement
- Post-processing

Features, Matching, Camera calibration,…
Calibration refinement, Non-linear optimization,…
Plane Sweep, Dense Stereo, etc…
Structured mesh, Initial surface,…
High quality mesh reconstruction,…
Texturing, Mesh processing, simplification, relighting, rendering, etc…
Surface Extraction from Point Clouds

- Techniques based on the Delaunay Triangulation. Idea:
  - Build a Delaunay triangulation of the Point Set
  - Label each tetrahedron as inside / outside
  - Extract the boundary → Obtain a 3D mesh
2D Example: Points / Cameras

Camera 1

C 2

C 3

C 4

C 5
Delaunay Triangulation
Delaunay Tetrahedrization

Delaunay Triangulation complexity: $n \log(n)$ in 2D and $n^2$ in 3D, but tends to $n \log(n)$ if points are distributed on a surface.

Advantages:

- Do not rely on an implicit representation → keep the original reconstructed points, no discretization problem
- Compact representation → memory efficient
Camera Visibility
Labeling Tetrahedra
Labeling Tetrahedra
Labeling Tetrahedra
Visibility Conflicts
Surface Extraction
Surface Extraction Examples
Surface Extraction Examples

This is the joint video of paper: Incremental Reconstruction of Manifold Surface from Sparse Visual Mapping, Shuda Yu and Maxime Lhuillier, 3DIMPVT'2012
Extract a Mesh from the Triangulation

• Visibility
• Energy Minimization via Graph Cut
  • A Mesh IS a Graph
  • Efficient
  • Add smoothness constraints
    • Surface area
    • Photo-consistency
Visibility Reasoning

OUTSIDE

line of sight

sensor / emitter

i_1

i_2

i_3

i_4

i_5

i_6

INSIDE

vertex / sample

oriented facet weights

S

V_i_1 → V_i_2 → V_i_3 → V_i_4

V_i_5 → V_i_6

t
Labeling Tetrahedra

S (outside)

T (inside)

S (outside)
Additional Constraints

- Smoothing terms
- Surface Area
- Photo-consistency

\[ E_{\text{photo}}(S) = \int_S \rho \, dS = \sum_{T \in S} \rho(T) A(T) \]
Surface Extraction Results
Surface Extraction Results
Surface Extraction Results
Surface Extraction Results
Mesh Optimization

- Refine the geometry of the mesh with optimization based on the photo-consistency
The Reprojection Error

- Error to measure the photo-consistency, or error between an *observed* image and a *generated* image.
- The mesh is optimized such that the reprojection error is minimized.

- It is a ill-posed problem, difficult to solve directly.
- Can be modeled using *variational methods*, and *gradient descent* techniques.

The Reprojection Error

The energy functional – (error between input images and generated images)

\[ E(S) = \frac{1}{2} \sum_i \int_{\mathcal{I}_i} (I_i(u) - R_i(u))^2 \, du \]
A Simple Generative Model

\[ E(S) = \frac{1}{2} \sum_i \int_{\mathcal{I}_i} \frac{(I_i(u) - R(u))^2}{g(u)} \, du \]

- \( R \) can be the mean color seen from all visible cameras.
- For example one can choose: \( R_i = T(x) \), with

\[ T(x) = \frac{\sum_i I_i(\pi_i(x)) \, w_i(x) \nu_{i,S}(x)}{\sum_i w_i(x) \nu_{i,S}(x)} \]
Minimizing the Reprojection Error

- Need to rewrite the error functional to an energy over the *surface*.

\[
E(S) = \int_{\Pi(S)} g(u) du = \int_{S \cap \mathcal{V}} g(x) \frac{x \cdot n(x)}{x_z^3} d\sigma
\]
The Energy on the Visible Volume

Visible volume: $V_S(x) = 1$

Occluded volume: $V_S(x) = 0$

Visibility interface

$$E(S) = - \int_{\mathbb{R}^3} \nabla \cdot \left( g(x) \frac{x}{x^3} \right) V_S(x) dx$$
Mesh Optimization

- Initial mesh $S_0$ (Visual hull, a bounding sphere, Stereoscopic segmentation)
- While have not converged, do:
  - Compute visibility
  - Estimate color $T(x)$ of $S$ (mean color from images)
  - Update the shape:
  $$\frac{\partial S(t)}{\partial t} = -\nabla E(S)$$
  - Check convergence
Gradient Descent

- Gradient of E with respect to S (See Gargallo et al. [2007])
  - We need to define the energy with respect to S
  - We want to write the derivative of the energy for any vector deformation field V on S as

\[
\frac{d}{dt} E(S(t)) \bigg|_{t=0} = -\langle G, V \rangle, \quad G \text{ is the gradient of } E(S)
\]

\[
\frac{\partial S(t)}{\partial t} = -\nabla E(S), \quad S(t) = S + t \, V
\]
Discretize then Minimize

\[ E(S) \quad \rightarrow \quad E(X) \]

\[ \nabla E(S) \quad \rightarrow \quad \nabla E(X) \]

- Allows vertex displacements coherent with the surface representation.

\( S \) : Continuous surface
\( X \) : Discrete Mesh
Discrete representation & Gradient

- For triangular meshes parametrized by

  \[ x_k(t) = x_k + t \mathbf{V}_k \]

→ We want to write the variation of the energy for a small deformation as a linear combination with respect to \( \mathbf{V} \).
Problems: Remeshing...

- Dealing with Self-Intersections
- Adding/Removing points

Meshes: CGAL + Topology-adaptive meshes [Pons & Boissonnat 2007] [Zaharescu et al. 2007]

But: easily allows adaptive remeshing
A Classical Multi-view Stereo Benchmark

• Recovered shape for the temple sparse ring data (16 images) [Seitz et al. 2006]

• http://vision.middlebury.edu/mview/
3D Mesh Details

16 input images
3D Surface Reconstruction

11 input images
Other 3D Reconstruction Problems

- One can change the cost function of the generative model in order to apply it to:
  - Multi-view photometric stereo
  - Multi-view Shape-from-shading
  - Multi-view Range Integration (Depth Map fusion)
  - …

- The problem has to be modeled as a reprojection error, where I (the measurement) and R (the generative model) could mean different things (normals, depth, …)

\[ E(S) = \frac{1}{2} \sum_i \int_{\mathcal{I}_i} (I_i(u) - R_i(u))^2 \, du \]
Some References on Mesh Refinement for 3D Reconstruction

- ...
Global Image-based Matching Score for Multi-view Stereo

• Minimize error between image pairs \((i,j)\)
• Error between an observed image \(i\), and the reprojection of the back-projection of an image \(j\) onto the surface.

\[
E_{\text{error}}(S) = \sum_{i,j} \int_{\Omega_{ij}^S} h(I_i, I_{ij}^S)(x_i) \, dx_i
\]
Towards a complete Multi-View Stereo pipeline

Structure from Motion

Bundle Adjustment

Dense 3D Points

Mesh Extraction

Mesh Refinement

The final results quality and accuracy will depend on which algorithm, or which energy functional, is used in each of those steps.

Results - from Acute3D

Automatically generate true-to-life 3D models from 2D photographs

www.acute3D.com
Some Videos Links on Mesh-based Modeling

- **ProForma**: [http://www.youtube.com/watch?v=vEOmzjImsvC](http://www.youtube.com/watch?v=vEOmzjImsvC)
- **Acute3D**: [http://www.youtube.com/watch?v=ADVQso0KZzo](http://www.youtube.com/watch?v=ADVQso0KZzo)  
  [http://www.youtube.com/watch?v=Fu3HoRPRU9Q](http://www.youtube.com/watch?v=Fu3HoRPRU9Q)
- **4DViews**: [http://www.youtube.com/watch?v=uVbYi-wr0Y](http://www.youtube.com/watch?v=uVbYi-wr0Y)  
  [http://www.youtube.com/watch?v=AJw1omc3bTk](http://www.youtube.com/watch?v=AJw1omc3bTk)
- **Incremental Delaunay Reconstruction**: [http://www.youtube.com/watch?v=4QZFgfMeG4E](http://www.youtube.com/watch?v=4QZFgfMeG4E)
- ...
Conclusion

• 3D Modeling Using Meshes requires:
  • Non trivial optimization
    • Gradient descent (possible local minima)
  • A good initialization
    • Delaunay + Graph Cut + visibility
    • Any other techniques
  • Remeshing tools
  • A reprojection error which depends on your problem
  • Compact: memory efficient (allows large scale)
Conclusion

• 3D Modeling Using Meshes offers:
  • High quality refinement
  • Scalability
  • Flexibility
  • Reconstruction Accuracy

• With some Maths and a lot of programming… and it works!!
Results - from Acute3D