Volumetric Modeling
<table>
<thead>
<tr>
<th>Date</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 20</td>
<td>Introduction</td>
</tr>
<tr>
<td>Feb 27</td>
<td>Lecture: Geometry, Camera Model, Calibration</td>
</tr>
<tr>
<td>Mar 5</td>
<td>Lecture: Features, Tracking/Matching</td>
</tr>
<tr>
<td>Mar 12</td>
<td>Project Proposals by Students</td>
</tr>
<tr>
<td>Mar 19</td>
<td>Lecture: Epipolar Geometry</td>
</tr>
<tr>
<td>Mar 26</td>
<td>Lecture: Stereo</td>
</tr>
<tr>
<td>Apr 2</td>
<td>Short lecture “Active Ranging, Structured Light, ICP” + 2 papers</td>
</tr>
<tr>
<td>Apr 9</td>
<td>Easter</td>
</tr>
<tr>
<td>Apr 16</td>
<td>Project Updates (Sechseläuten in afternoon)</td>
</tr>
<tr>
<td>Apr 23</td>
<td>Short lecture “Volumetric Modeling” + 2 papers</td>
</tr>
<tr>
<td>Apr 30</td>
<td>Short lecture “Structure from Motion, SLAM” + 2 papers</td>
</tr>
<tr>
<td>May 7</td>
<td>Short lecture “SfM++: Photo Collections, Bundle Adjust” + 2 papers</td>
</tr>
<tr>
<td>May 14</td>
<td>Short lecture “Shape-from-X: Single View” + 2 papers</td>
</tr>
<tr>
<td>May 21</td>
<td>Short lecture “Shape-from-X: Reflection/…” + 2 papers</td>
</tr>
<tr>
<td>May 28</td>
<td>Pentecost / White Monday</td>
</tr>
<tr>
<td>May 29-June 1</td>
<td>Final Demos</td>
</tr>
</tbody>
</table>

Schedule (tentative)
Today’s class

Modeling 3D surfaces by means of a discretized volume grid. In particular:

• extracting a triangular mesh from an implicit volume representation
• volumetric range image integration
• convex 3D shape modeling
Volumetric Representation

- Sample a volume containing the surface of interest uniformly

- Label each grid point as lying inside or outside the surface, e.g. by defining a signed distance function (SDF) with positive values inside and negative values outside

- The modeled surface is represented as an isosurface of the labeling (implicit) function
Volumetric Representation

• Why volumetric modeling?
  
  • Gives a flexible and robust surface representation
  • Handles complex surface topologies effortlessly
  • Allows to sample the entire volume of interest by storing information about space opacity
  • Offers possibilities for parallel computing
From volume to mesh: Marching Cubes


- Basic idea: March through the volume and process each voxel by determining all potential intersection points of its edges with the desired isosurface; precise localization via interpolation

- Obtained intersection points serve as vertices of triangles which make up a triangulation of the constructed isosurface
From volume to mesh: Marching Cubes

Example: “Marching Squares” in 2D
From volume to mesh: Marching Cubes

By summarizing symmetric configurations, all possible $2^8 = 256$ cases reduce to:

The 15 Cube Combinations
From volume to mesh: Marching Cubes

- The accuracy of the computed surface depends on the volume resolution

- Precise normal specification at each vertex possible by means of the implicit function
From volume to mesh: Marching Cubes

- Benefits of Marching Cubes
  - Always generates a manifold surface
  - The desired sampling density can easily be controlled
  - Trivial merging or overlapping of different surfaces based on the corresponding implicit functions: minimum of the values for merging and averaging for overlapping
From volume to mesh: Marching Cubes

- Limitations of Marching Cubes
  - Maintains a 3D entry rather than 2D which entails considerable computational and memory requirements
  - Generates consistent topology, but not always the topology you wanted
  - Problems with very thin surfaces
Range Image Integration


- Generate a signed distance function (or something close to it) for each scan
- Computed a weighted average
- Extract isosurface
Range Image Integration

- use voxel space
- new surface as zero-crossing (find using marching cubes)
- least-squares estimate (zero derivative=minimum)
It turns out that the least-squares estimate leads to the following simple fusion scheme:

\[
D_{i+1}(x) = \frac{W_i(x)D_i(x) + w_{i+1}(x)d_{i+1}(x)}{W_i(x) + w_{i+1}(x)}
\]

\[
W_{i+1}(x) = W_i(x) + w_{i+1}(x),
\]

where \(d_i\) and \(w_i\) denote the depth map and weighting function to range image \(i\), and \(D_i\) and \(W_i\) denote the accumulated measures.
Range Image Integration

Depth maps and weighting functions are computed with a ray casting procedure and linear interpolation within intersected triangles.
TV-L1 Range Image Integration


given:
\[
V \subset \mathbb{R}^3 \quad \text{volume}
\]
\[
f_i : V \rightarrow [-1,1] \quad \text{range images}
\]
\[
w_i : V \rightarrow \{0,1\} \quad \text{weighting functions}
\]

find:
\[
u : V \rightarrow [-1,1] \quad \text{such that its 0-isosurface represents the imaged shape}
\]

\[\delta > 0 \quad \text{uncertainty of the depth values}\]
\[\eta > 0 \quad \text{width of the occluded region behind the surface}\]
TV-L1 Range Image Integration

The method is based on minimizing the following energy functional:

\[
E(u) = \int_V \left\{ |\nabla u| + \lambda \sum_{i \in I(x)} |u - f_i| \right\} \, dx,
\]

where \( I(x) = \{ i \mid w_i(x) > 0 \} \) is the set of relevant indices and \( \lambda \in \mathbb{R} \) is a parameter.

The functional consists of two terms:

- a total variation (TV) smoothness term that penalizes the perimeter of the level sets of \( u \), which is in this case exactly the surface area
- a data fidelity term measuring the distance of \( u \) to all \( f_i \) by means of the robust \( L^1 \) norm.
TV-L1 Range Image Integration

Although the proposed functional

$$E(u) = \int \left\{ |\nabla u| + \lambda \sum_{i \in I(x)} |u - f_i| \right\} dx$$

is convex, it entails some computational difficulties, as both regularization term and data term are generally not continuously differentiable. For that reason, we can consider the following approximation:

$$E_\theta(u, v) = \int \left\{ |\nabla u| + \frac{1}{2\theta} (u - v)^2 + \lambda \sum_{i \in I(x)} |v - f_i| \right\} dx,$$

where $\theta$ is a small constant such that $v$ is a close approximation of $u$. The above functional is minimized in an alternating manner with respect to $u$ and $v$. 
TV-L1 Range Image Integration

(a) One source view  (b) Depth image  (c) Front view  (d) Back view

(a) View #1  (b) Depth #1  (c) View #2  (d) Depth #2  (e) Mesh view (688280 triangles)
Convex 3D Modeling


• Multiview stereo allows to compute entities of the type:
  • $\rho : V \rightarrow [0,1]$ photoconsistency map reflecting the agreement of corresponding image projections
  • $f : V \rightarrow [0,1]$ potential function representing the costs for a voxel for lying inside the surface

• How can these measures be integrated in a consistent and robust manner?
Convex 3D Modeling

- Photoconsistency is usually computed by matching corresponding image projections in different views.
- Instead of comparing only the pixel colors, image patches are considered around each point to reach better robustness.
- Challenges:
  - Many real-world objects do not satisfy the underlying Lambertian assumption, i.e. the reflected radiance for each point on the surface is the same in all viewing directions.
  - Matching is ill-posed, as there are usually a lot of different potential matches among multiple views.
  - Visibility.
Convex 3D Modeling

• A potential function $f : V \rightarrow [0,1]$ can be obtained by fusing multiple depth maps or with a direct 3D approach.

• Depth map estimation is fast but leads to a two-step method (disadvantage: errors could propagate and degrade the final result).

• The direct approach is generally computationally more intense but offers more robustness and flexibility (occlusion handling, projective patch distortion etc.).

• Proposed propagation scheme entails additional advantages by the possibility to define a sharp deblurred photoconsistency map via a voting strategy.
Convex 3D Modeling

Example: Middlebury “dino” data set
Convex 3D Modeling

By introducing a binary labeling function $u : V \to \{0,1\}$ reflecting the indicator function of the interior region, the modeling problem can be cast as minimizing

$$E(u) = \int_V \rho |\nabla u| \, dx + \lambda \int_V f \, u \, dx$$

over the set $C_{bin} = \{ u \mid u : V \to \{0,1\} \}$. One can observe that the above functional is convex, but it is optimized over a non-convex domain. A constrained convex optimization problem can be derived by relaxing the binary condition to $C_{rel} = \{ u \mid u : V \to [0,1] \}$.

**Theorem:** A global minimum of $E$ over $C_{bin}$ can be obtained by minimizing over $C_{rel}$ and thresholding the solution at some $thr \in (0,1)$. 
Convex 3D Modeling

input images (2/28)

input images (2/38)
Convex 3D Modeling

• Benefits of the model

  • High-quality 3D reconstructions of sufficiently textured objects possible
  • Allows global optimizability
  • Simple construction without multiple processing stages and heuristic parameters
  • Computational time depends only on the volume resolution and not on the resolution of the input images
  • Perfectly parallelizable
• Limitations of the model

• Computationally intense: computational time could exceed two hours on a single-core CPU depending on the utilized volume resolution

• It may not be possible to compute an accurate potential function $f$ for objects strongly violating the Lambertian assumption
Convex 3D Modeling

“Integration of Multiview Stereo and Silhouettes via Convex Functionals on Convex Domains”, Kalin Kolev and Daniel Cremers, European Conference on Computer Vision (ECCV ’08).

- Idea: Extract the silhouettes of the imaged object and use them as constraints to restrict the domain of feasible shapes
Consider the following functional:

\[ E(u) = \int_{V} \rho |\nabla u| dx \]

s.t. \( u(x) \in \{0,1\} \ \forall \ x \in V \)
\[ \sum_{x \in R_{ij}} u(x) \geq 1 \ \text{if} \ j \in Sil_i \]
\[ \sum_{x \in R_{ij}} u(x) = 0 \ \text{if} \ j \notin Sil_i, \]

where \( Sil_i \subset \Omega_i \) denotes the silhouette in image \( i \), i.e. a binary subdivision of the image domain.

**Proposition:** A solution of the above optimization problem within a certain bound from the globally optimal one can be obtained via relaxation and subsequent thresholding of the computed result with an appropriate threshold.
Convex 3D Modeling

input images (2/24)

input images (2/27)
Convex 3D Modeling

- Benefits of the model
  - Allows to impose exact silhouette consistency
  - Highly effective in suppressing noise due to the underlying weighted minimal surface model

- Limitations of the model
  - Presumes precise object silhouettes which are not always easy to obtain
  - The utilized minimal surface model entails a shrinking bias and tends to oversmooth surface details
Convex 3D Modeling


- Idea: Exploit additionally surface normal information to counteract the shrinking bias of the weighted minimal surface model
Consider the following generalization of the previous model:

\[ E(u) = \int_{V} \sqrt{\nabla u^T D \nabla u} \, dx \]

s.t. \( u(x) \in \{0,1\} \ \forall \ x \in V \)

\[ \sum_{x \in R_{ij}} u(x) \geq 1 \quad \text{if } j \in Sil_i \]

\[ \sum_{x \in R_{ij}} u(x) = 0 \quad \text{if } j \notin Sil_i, \]

where the matrix mapping \( D \) is defined as

\[ D = \rho^2 (\tau F F^T + \frac{3-\tau}{2} (I - F F^T)). \]

\( F \) is the given normal field and \( \tau \in [0,1] \) is a parameter reflecting the confidence in the surface normals.
Convex 3D Modeling

input images (4/21)
Presentations

[Newcombe/Davison ‘10]: “Live Dense Reconstruction with a Single Moving Camera”

[Izadi et al. ‘11]: “KinectFusion: Real-time 3D Reconstruction and Interaction using a Moving Depth Camera”