High-quality Textured Surface Reconstruction from Registered Images: State-of-the-art methods

(a) Image collection
(b) Structure from Motion
(c) Multi-view Stereo
(d) Surface generation
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection
(b) Structure-from-Motion
(c) Multi-view Stereo
(d) Surface generation
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection

(b) Structure-from-Motion

(c) Multi-view Stereo

(d) Surface generation

(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection

(b) Structure-from-Motion

(c) Multi-view Stereo

(d) Surface generation

(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection
(b) Structure-from-Motion
(c) Multi-view Stereo
(d) Surface generation
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection
(b) Structure-from-Motion
(c) Multi-view Stereo
(d) Surface generation
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection
(b) Structure-from-Motion
(c) Multi-view Stereo
(d) Surface generation
+ Surface refinement
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection
(b) Structure-from-Motion
(c) Multi-view Stereo
(d) Surface generation
(e) Texture mapping
The pipeline of image-based 3D reconstruction

(a) Image collection

(b) Structure-from-Motion

(c) Multi-view Stereo

(d) Surface generation

(e) Texture mapping
Content

• Multi-view Stereo
  • Pairwise Stereo
  • Propagation Stereo

• Surface generation
  • Surface extraction
  • Surface refinement

• Texture mapping
  • View selection
  • Color adjustment and blending
Content

• Multi-view Stereo
  • Pairwise Stereo
  • Propagation Stereo

• Surface generation
  • Surface extraction
  • Surface refinement

• Texture mapping
  • View selection
  • Color adjustment and blending
Multi-view Stereo

- Problem definition: given several registered images of the same object or scene, compute a dense representation of its 3D shape

- “Registered images of same object or scene”
  - Known camera parameter
  - Arbitrary number of images (from two to thousands)

- “Dense representation of 3D shape”
  - Depth maps
  - Point clouds
  - Patch clouds
  - Meshes
  - Voxels
Multi-view Stereo

- Problem definition: given several **registered images** of the same object or scene, compute a **dense representation** of its 3D shape

- “Registered images of same object or scene”
  - Known camera parameter
  - Arbitrary number of images (from two to thousands)

- “Dense representation of 3D shape”
  - Depth maps
  - Point clouds
  - Patch clouds
  - Meshes
  - Voxels
Multi-view Stereo

• Problem definition: given several registered images of the same object or scene, compute a dense representation of its 3D shape

• “Registered images of same object or scene”
  • Known camera parameter
  • Arbitrary number of images (from two to thousands)

• “Dense representation of 3D shape”
  • Depth maps
  • Point clouds
  • Patch clouds
  • Meshes
  • Voxels
Multi-view Stereo: 1D matching problem

Epipolar line (Projected ray)

Compare the similarity between $p$ and $q$, find the best $q$
Multi-view Stereo: Similarity measure

- Convert the patches $p$ and $q$ into vectors $f$ and $g$

SSD (Sum of Squared Differences)
\[
\rho_{SSD}(f, g) = \|f - g\|^2
\]
- Pros: efficient, derivable
- Cons: sensitive to bias/gain

SAD (Sum of Absolute Differences)
\[
\rho_{SAD}(f, g) = \|f - g\|_1
\]
- Pros: efficient, robust to salt/pepper noise
- Cons: non-derivable, sensitive to bias/gain

NCC (Normalized Cross Correlation)
\[
\rho_{NCC}(f, g) = \left\langle \frac{f}{\|f\|}, \frac{g}{\|g\|} \right\rangle = \frac{f \cdot g}{\|f\| \cdot \|g\|}
\]
- Pros: robust to gain, derivable
- Cons: sensitive to bias

ZNCC (Zero-mean Normalized Cross Correlation)
\[
\rho_{ZNCC}(f, g) = \left\langle \frac{f - \hat{f}}{\|f - \hat{f}\|}, \frac{g - \hat{g}}{\|g - \hat{g}\|} \right\rangle = \frac{(f - \hat{f}) \cdot (g - \hat{g})}{\sigma_f \sigma_g}
\]
- Pros: robust to bias/gain, derivable
- Cons: less efficient
Multi-view Stereo: Similarity measure

- Convert the patches \( p \) and \( q \) into vectors \( f \) and \( g \)

SSD (Sum of Squared Differences)

\[
\rho_{SSD}(f, g) = ||f - g||^2
\]
- Pros: efficient, derivable
- Cons: sensitive to bias/gain

SAD (Sum of Absolute Differences)

\[
\rho_{SAD}(f, g) = ||f - g||_1
\]
- Pros: efficient, robust to salt/pepper noise
- Cons: non-derivable, sensitive to bias/gain

NCC (Normalized Cross Correlation)

\[
\rho_{NCC}(f, g) = \left\langle \frac{f}{||f||}, \frac{g}{||g||} \right\rangle = \frac{f \cdot g}{||f|| \cdot ||g||}
\]
- Pros: robust to gain, derivable
- Cons: sensitive to bias

ZNCC (Zero-mean Normalized Cross Correlation)

\[
\rho_{ZNCC}(f, g) = \left\langle \frac{f - \hat{f}}{||f - \hat{f}||}, \frac{g - \hat{g}}{||g - \hat{g}||} \right\rangle = \frac{(f - \hat{f}) \cdot (g - \hat{g})}{\sigma_f \sigma_g}
\]
- Pros: robust to bias/gain, derivable
- Cons: less efficient
Multi-view Stereo: two branches of methods

- **Pairwise depth map reconstruction**
  - Input two images
  - Output per-image depth map
  - Need fusion afterwards

- **Global point cloud propagation**
  - Input $N$ images
  - Output global 3D points
  - Difficult parallelism
Multi-view Stereo: matching strategy

Local matching
Naive winner-take-all
Image segmentation
Gerrits et.al., CRV2006

Global regularization
Graph cuts
Kolmogorov et.al. ICCV2001
Semi-global matching
Hirschmüller, CVPR2006
Belief Propagation
Klaus et.at. ICPR2006

Deep learning
CNN
Žbontar and LeCun. CVPR2015

next……?
Multi-view Stereo: plane sweep stereo

- Winner-take-all
- GPU projective texture mapping
- Highly efficient and parallelable
- Noisy output, need filter afterwards

Space-sweep
(Pioneer)
Collins, 1996

Multi-direction Planesweep
Gallup et.al. CVPR2007

Local Planesweep
Sinha et.al. CVPR2014
Multi-view Stereo: point cloud propagation

Pioneer: Quasi-dense approach
Lhuillier and Quan. PAMI2005

Patch-based MVS (PMVS)
Furukawa and Ponce. PAMI2010

GPU PatchMatch
Galliani et.al. ICCV2015

PatchMatch Stereo
Bleyer et.al. BMVC2011
Multi-view Stereo: PMVS

• Patch model

A 3D patch has
• Position
• Normal
• Scale
• Visibility

Patch-based MVS (PMVS)
Furukawa and Ponce. PAMI2010
Multi-view Stereo: PMVS

Initialization
- Feature Detection
  - Harris and DoG
  - Sparse seed patches

Propagation
- Propagation, then optimize
  - Optimize via Levenberg–Marquardt
    - Position
    - Normal

Filter
- Confidence filter
- Visibility filter
- Small group filter

Patch cloud
- Point cloud with orientations

Poisson Surface Reconstruction
Multi-view Stereo: PMVS
Content

• Multi-view Stereo
  • Pairwise Stereo
  • Propagation Stereo

• Surface generation
  • Surface extraction
  • Surface refinement

• Texture mapping
  • View selection
  • Color adjustment and blending
Surface representation

<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>

Mesh rocks! triangular mesh is more suitable for mesh processing
Surface extraction from point cloud

Definition: Given a set of points $\mathbf{P} = \{P_i\}_{i \in [1,N]} \in \mathbb{R}^3$ sampled from a surface $S$, find a best approximate surface $S'$ to the original $S$.

- Implicit surface (Computer Graphics)
  - Model the surface implicitly with a function $f(x, y, z) = 0$

- Delaunay approach (Computational geometry)
  - Based on Delaunay triangulation/tetrahedra, find the best mesh surface

Bloomenthal, 1997

Cheng et.al. 2012
Implicit Surface

**Distance functions**

- Local tangent plane + MST
  - Hoppe et.al. SIGGRAPH1992
- Line of sight distance weighted
  - Curless & Levoy. SIGGRAPH1996

- Moving Least-squares (MLS)
  - Pioneer of MLS method
    - Lancaster and Salkauskas. 1981
  - Least-MEDIAN-of-squares
    - Fleishman et.al. SIGGRAPH2005
  - Interactive, constrained
    - Shen et.al. SIGGRAPH2004

**Radial basis functions (RBF)**

- RBF approximation
  - Carr et.al. SIGGRAPH2001
- Multi-level partition of unity
  - Ohtake et.al. SIGGRAPH2003

**Indicator functions**

- Poisson Surface
  - Kazhdan et.al. SGP2006
- Signed distance function
  - Hornung & Kobbelt. SGP2006
Implicit Surface: Poisson method

- Reconstruct the surface of the model by solving for the indicator function of the shape.

\[ \chi_M(p) = \begin{cases} 
1 & \text{if } p \in M \\
0 & \text{if } p \notin M 
\end{cases} \]
Implicit Surface: Poisson method

- There is a relationship between the normal field and gradient of indicator function.
Implicit Surface: Poisson method

• Represent the points by a vector field $\vec{V}$
• Find the function $\chi$ whose gradient best approximates $\vec{V}$:

$$\min_{\chi} \left\| \nabla \chi - \vec{V} \right\|$$

• Applying the divergence operator, we can transform this into a Poisson problem:

$$\nabla \cdot (\nabla \chi) = \nabla \cdot \vec{V} \iff \Delta \chi = \nabla \cdot \vec{V}$$

• Discretize from coarse-to-fine over an octree.
Delaunay method

Points -> Delaunay tetrahedron -> Mesh surface

Power Crust
Ameta et.al. SIGGRAPH1998

Robust Cocone
Dey & Goswami. SoSMA2003

Visibility-consistent
Vu et.al. PAMI2012
Delaunay method: visibility-consistent

Point cloud reconstructed by multi-view images

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent

Delaunay Triangulation of Point cloud

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent

Visibility of a vertex, labeling the tetrahedra

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent

Visibility conflicts

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent

Extract a mesh surface from tetrahedron

- A tetrahedron is a graph
  - Every tetrahedral (cell) is a node
  - Linking the source and sink by visibility
  - Smoothness by neighboring relations
  - Additional terms
    - Surface area
    - Photo-consistency

- Energy minimization via Graph Cuts

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent

Mesh surface as the boundary between IN and OUT

(credit to the course material of ETH Computer Vision)
Delaunay method: visibility-consistent
Delaunay method: visibility-consistent
Delaunay method: visibility-consistent EXT

Preserving weakly supported surface

Visibility-consistent surface reconstruction
Vu et.al. PAMI2012

Weakly-supported surface reconstruction
Jancosek & Pajdla. CVPR2013
Surface extraction from point cloud

Weakly-supported surface reconstruction
Jancosek & Pajdla. CVPR2013

Visibility-consistent surface reconstruction
Vu et.al. PAMI2012
Surface refinement: crucial to high accuracy!

Variational surface refinement
Vu et.al. PAMI2012
Surface refinement: formulation

Minimizing the error between the observed image $i$ and reprojection of image $j$:

$$E_{\text{error}}(S) = \sum_{i,j} \int_{\Omega_{ij}^S} h(I_i, I_{ij}^S)(x_i) \, dx_i$$

- Ill-posed, difficult to solve directly
  - Local minima
  - Bad initialization
- Can be modeled via variational methods

Variational surface refinement
Vu et.al. PAMI2012
# Surface refinement: formulation

## Continuous (level-set)

\[
M(S) = \sum_i \sum_{j \neq i} M_{ij}(S), \\
M_{ij}(S) = M|_{\Omega_i \cap \Pi_j(S)} \left( I_i \circ I_j \circ \Pi_j^{-1} \right).
\]

- **Energy formulation**
- **Variations/derivative**
- **Regularization**

## Discretized (triangular mesh)

\[
E_{\text{error}}(S) = \sum_{i,j} \int_{\Omega_{ij}} h(I_i, I_{ij}^S)(x_i) \, dx_i
\]

- **similarity measure**
- **observed image**
- **reprojection image**

\[
E_{\text{error}}(S) = \int_S \phi(x) \sum_{i,j} \nabla M_{ij}(x) \, dx
\]

\[
= \sum_{i,j} \int_{\Omega_{ij}} \phi(x) f_{ij}(x_i) / (N^T d_i) N \, dx_i
\]

- **Umbrella operator on mesh**

---

Pons et al. IJCV2007

Vu et al. PAMI2012
Surface refinement: results

Top to bottom: evolution of iterative refinement

Variational surface refinement
Vu et.al. PAMI2012

Refinement recovers the fine details of the scene
Surface refinement: problems?

- Iterative, repeated computations of
  - visibility
  - image reprojection
  - image similarity
Surface refinement: problems?

• Iterative, repeated computations of
  • visibility
  • image reprojection
  • image similarity

• Not all regions contribute equally
  • Potential regions may gain details

• A flat plane is still a flat plane
Surface refinement: adaptive refinement

1. Evaluate the importance of a vertex movement

\[ g_{CV} = \max_{p \in \text{planes}(v')} \{(p^t v)^2\} \]
\[ g_{C_i} = \frac{1}{3} \sum_v g_{CV}, \]

where \( p = [a \ b \ c \ d]^t \) represents a plane and \( v' = [v'_x \ v'_y \ v'_z \ 1] \).

2. Optimal trade-off between accuracy & efficiency

\[ u(r_\omega, l_\omega) = \max_{(r, l) \in \text{curve}} u(r, l) \]

\[ u(r, l) = u(r) + u(l) \]
\[ = w_r \cdot r + w_l \cdot (1 - l) \]

3. Graph cuts optimization

\[ E(f) = E_{\text{optimality}}(f) + E_{\text{smoothness}}(f) + E_{\text{prior}}(f). \]
Surface refinement: adaptive refinement

- Adaptive refinement
  - \(~5x\) more efficient than uniform refinement
  - much more compact mesh
  - Similar reconstruction details.
Surface refinement: adaptive refinement

**Uniform refinement**
- 2,414,767 vertex, 4,829,450 triangle

**Adaptive refinement**
- 817,254 vertex, 1,592,022 triangle

*Herz-Jesu-P25*
Content

- Multi-view Stereo
  - Pairwise Stereo
  - Propagation Stereo

- Surface generation
  - Surface extraction
  - Surface refinement

- Texture mapping
  - View selection
  - Color adjustment and blending
Texture mapping (for 3D reconstruction)

• Texturing a point cloud
  • Trivial, directly fetch color from image for every 3D point

• Texturing a triangular mesh surface
  • Non-trivial

  • Triangle view selection
    • For each triangle, select its best image for texture (data term)
    • Minimize number of texture seam (smoothness term)

• Color adjustment and blending
  • Alleviate the artifacts at seams between two texture patches (atlas), due to inaccurate camera or imbalance illumination
Texture mapping: Triangle view selection

- For each triangle, select multiple view
  - blending multiple image
    Callieri et.al. CG2008
  - Color interpolation
    Grammatikopoulos et.al. ISPRS2007

- For each triangle, select one best view
  - Seam optimization
    Gal et.al. CG2010
  - MRF+Poisson blending
    Lempitsky & Ivanov. CVPR2007
  - Continuous color optimization
    Velho & Sossai. CVPR2007
Texture mapping: Triangle view selection

- Triangle view selection as a MRF problem

\[ E(l) = \sum_{F_i \in Faces} E_{data}(F_i, l_i) + \sum_{(F_i, F_j) \in Edges} E_{smooth}(F_i, F_j, l_i, l_j). \]

- Each triangle selects a best view
- Neighboring triangles select same view

- Solved by alpha expansion
Texture mapping: Triangle view selection

- **Difficulties:**
  - dynamic objects

  Handle by photo-consistency check and outlier removal

- **out-of-focus image**

  Handle by weighting the gradient magnitude

Large-scale texturing. Waechter et.al. ECCV2014
Texture mapping: Color adjustment and blending

- Let $f$ be the original intensity, $g$ be the adjustment (gain)

$$\arg\min_g \sum_{v_{\text{left}/\text{right}}} (f_{v_{\text{left}}} + g_{v_{\text{left}}} - (f_{v_{\text{right}}} + g_{v_{\text{right}}}))^2 + \frac{1}{\lambda} \sum_{i,j} (g_i - g_j),$$

  - Minimize the difference of neighboring intensity
  - Minimize the imposed adjustment (be as much original as possible)

**Blending**
- Alpha blending causes ghosting effect
- Poisson blending:

Poisson Image Editing
Perez et.al. SIGGRAPH2003
Unsolved problems and future work

- **local camera optimization** for MVS
  - SfM computes globally optimized camera, which is not locally optimized
  - Inaccurate camera is detrimental
  - Related work: (Zhu et.al. CVPR2014)

- Simultaneously surface **refinement + texturing**
  - Repeated computations of depth, visibility, etc.
  - Not and end-to-end optimization
  - Optimize the surface **geometry** and **textures**, for the rendering photo-realism

- **Fine-scale object** reconstruction (such sticks, wire)
  - Due to sparse point cloud at that objects
  - Can be improved by more structural point cloud, such as adding connectivity information