Image-based Rendering

Dr. Abdelaziz Djelouah
Aziz.Djelouah@disneyresearch.com
Computer graphics

[Szeliski and Cohen]
Motivation

Some things are just really hard to render … and model
Image-based rendering and light fields

[Szeliski and Cohen]
Rendering & light fields

Our result with complex light transport and geometry
3D scene representations

Geometry Based

- 3D model and reflectance information
- View-dependent texture mapping

Image Based

- (Multi-view) Image plus depth
- Light field
Overview

- Geometry and texture maps
- Depth based rendering and view interpolation
- View-dependent texture mapping
- Scene representation trade-offs
3D RECONSTRUCTION AND TEXTURE MAPPING
3D reconstruction pipeline

- Images
- Key point detection and matching
- Structure from motion (SFM)
- Point cloud & surface reconstruction
- Multi-view stereopsis
3D reconstruction from images

[Images of photographs and 3D model of a rabbit]

Computer Graphics AS 2019
Texture mapping – recall

- Mapping between the surface and the image
Texture mapping – recall

• Mapping between the surface and the image

Texture itself is a function:

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

\[ T(u, v) = (r, g, b) \]
Texture mapping – recall

- Mapping between the surface and the image
Problem - Texture from images
Problem - Texture from images

$T_1$ $T_2$

$I_1$ $I_2$
Optimization problem

\[
T \quad \text{min} \sum_{i} ||T(u, v) - T_{i}(u, v)||
\]

\(T_{1} \quad T_{2} \quad \ldots\)
Are all the views equal?
Optimization problem

\[
min \sum_i \lambda \ ||T(u, v) - T_i(u, v)||
\]
Optimization problem

\[ T \]

\[
\min \sum \lambda \left| |T(u, v) - T_i(u, v)| \right| + |\nabla T|
\]
Error sources in the imaging pipeline

[Image of human models and scenes with labels]

[Tsiminaki et al.]

[High resolution projected image] [Warped high resolution image] [Blurred high resolution image] [Noiseless low resolution image] [Input: noisy low res. image]
MAP estimation of the texture

Observations
Issues

• Lambertian surface assumptions

• Sensitive to geometric errors

• To handle more interesting scenarios, we need to account for
  – View dependent effects
  – Motion in the scene
DEPTH BASED RENDERING AND VIEW INTERPOLATION
Image + depth

[Zitnick et al.]
How to get depth?

- Billboard proxies
- Stereo
- Depth cameras (e.g. Kinect)
Virtual view synthesis

Original camera

Depth Map

Virtual camera

[Zitnick et al.]
Forward mapping
Forward mapping
Forward mapping
Problems of forward mapping

Non-surjective map:

- Occlusions (fundamental)
- Perspective changes (troublesome because of target discretization)

Image filtering (or point splatting) does not fix it

[Zitnick et al.]
Dense virtual depth maps

- If we had a depth map at the virtual camera:
  - Could backtrack every pixel
  - Discretization would become a non-issue (interpolation)
Backward mapping

Pixels in novel viewpoint

Input camera

$w_q$
Bilinear interpolation

Contribution of the pixels from the input view
Dense virtual depth maps

• If we had a depth map at the virtual camera:
  – Could backtrack every pixel
  – Discretization would become a non-issue (interpolation)

• We can have that!
  – Forward project depth
  – Merge several cameras
  – Filtering does make sense
Backward mapping

Original camera

Virtual camera

Iterate over

Filter depth
Forward vs. backward mapping

Forward mapping

Backward mapping

[Zitnick et al.]
Wide range navigation

[Zitnick et al.]
Wide range navigation

[Zitnick et al.]
Artifacts along depth discontinuities

[Zitnick et al.]
Detection of Depth Discontinuities

[Zitnick et al.]
Processing along depth discontinuities

[Zitnick et al.]
VIEW-DEPENDENT TEXTURE MAPPING
View-dependent texture mapping

1. Determine visible cameras for each surface element
2. Blend textures (images) depending on distance between original camera and novel viewpoint

[Debevec et al. 1996]
Geometry modeling

[Debevec et al. 1996]
Resulting rendering

[Debevec et al. 1996]
"Bullet time effects"
Unstructured Lumigraph

Objectives

• Link depth based IBR with light field rendering (lumigraph)

• Take into account several constraints
  – Continuous reconstruction
  – Viewing angles
  – Resolution sensitivity
Blending Fields

\[ \text{color}_{\text{desired}} = \sum_i w_i \text{color}_i \]

Desired Camera

Slide credit: C. Buehler
Blending Fields

\[ color_{\text{desired}} = \sum_i w(c_i) color_i \]

Desired Camera

Slide credit: C. Buehler
Unstructured Lumigraph Rendering

• Explicitly construct blending field
  – Computed using penalties
  – Sample and interpolate over desired image

• Rendering
  – Projective texture mapping and alpha blending

Slide credit: C. Buehler
Angle Penalty

\[ \text{penalty}_{\text{ang}}(C_i) = \theta_i \]
Resolution Penalty

Geometric Proxy

\[ \text{penalty}_{\text{res}}(C_i) = \max(0, \text{dist}(C_i) - \text{dist}(C_{\text{desired}})) \]
Field-Of-View Penalty

\[ \text{penalty}_{\text{FOV}} \]

\[ \text{angle} \]
Total Penalty

\[
\text{penalty}(C_i) = \alpha \text{penalty}_{\text{ang}}(i) + \beta \text{penalty}_{\text{res}}(i) + \gamma \text{penalty}_{\text{fov}}(i)
\]

Slide credit: C. Buehler
K-Nearest Continuous Blending

• Only use cameras with K smallest penalties

• $C^0$ Continuity: contribution drops to zero as camera leaves $K$-nearest set

• Normalization of the weights before blending
Example of ULR based rendering

[Hedman et al. 2016]
Depth Based IBR: pros and cons

• Advantages
  – Easy to capture images: photorealistic by definition
  – Few images
  – Simple, universal representation

• Disadvantages
  – WYSIWyG but also WYSIAyG
  – Explosion of data as flexibility increased
What you see is all what you get

[Hedman et al. 2016]
Transparency and reflection?

[Sinha et al. 2012]
Transparency and reflection?

- Standard IBR
- [Sinha et al. 2012]
- [Kopf et al. 2013]
## Scene representation trade-offs

<table>
<thead>
<tr>
<th>Method</th>
<th>Acquisition</th>
<th>Reconstruction</th>
<th>Rendering</th>
<th>FVV range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model-based</strong></td>
<td>Few cameras</td>
<td>Full 3D geometry and reflectance reconstruction, difficult for complete scenes, often applied for objects of interest, exploiting a-priori knowledge or interactive operation</td>
<td>Classical computer graphics</td>
<td>Wide</td>
</tr>
<tr>
<td><strong>Depth-based</strong></td>
<td>Few cameras</td>
<td>Depth estimation, error prone</td>
<td>Depth-based view interpolation</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Image-based</strong></td>
<td>Dense sampling of the scene, many cameras necessary to enable navigation</td>
<td>Data size / sampling issues</td>
<td>View interpolation, light field rendering</td>
<td>Limited</td>
</tr>
</tbody>
</table>
Thanks
Talk announcement

Disney Tech Talk
Workflow and technology challenges for Animated Films by Rajesh Sharma

- 17:00 on Tuesday, December 10th (tomorrow)

- CAB G 11