A 3-Tier Model from 2D Video

Martin Jagersand

joint work with Neil Birkbeck, Dana Cobzas, Adam Rachmielowski, Keith Yerex

University of Alberta
Computing Science
1. Overview of Research Interests & Projects

• Mathematical imaging models
• Computer vision
• Medical imaging
• Robotics
• Visual Servoing
Human-in-the-loop teleoperation is a current mission bottleneck

- Current ground-based tele-manipulation inefficient
  - Transmission delays
  - Non-anthophomomorphic arms

- Space craft don’t fit enough operators

Shuttle flight trainer, Johnson Space Ctr
Predictive Display for Tele-robotics

Problem: Even small delays (~¼ s) degrade operator performance
Solution: Predict and synthesize immediate visual feedback

Local operator
Model renders new views synchronously

Remote site
Model is captured by remote camera and transmitted asynchronously
Types of Predictive Display

• **What type of model?**
  - CAD line model
  - Video image warping
  - Textured graphics model

• **How is it acquired?**
  - A-priori
  - Sensed from scene once
  - Updated on-line
Segmentation = **surface/curve evolution** such that an **energy functional** is minimized

Energy: defined using data + [shape/atlas priors] + geometric priors (regularizers such that it has minimum at the desired segmentation

Surface/curve evolution: calculus of variation/PDE’s
Where we are? Western Canada!
Winter in the Rockies

Martin Jagersand
U of Alberta
And summer ...
Low budget 3D from video (Main talk)

- Inexpensive
- Quick and convenient for the user
- Integrates with existing SW e.g. Blender, Maya
Low budget 3D from video

• Inexpensive

$100: Webcams, Digital Cams  $100,000 Laser scanners etc.
Low budget 3D from video

- Inexpensive
- Quick and convenient for the user
- Integrates with existing SW e.g. Blender, Maya

Modeling geom primitives into scenes: >>Hours

Capturing 3D from 2D video: minutes
Low budget 3D from video

- Inexpensive
- Quick and convenient for the user
- Integrates with existing SW e.g. Blender, Maya
Application Case Study
Modeling Inuit Artifacts

- New acquisition at the UofA: A group of 8 sculptures depicting Inuit seal hunt
- Acquired from sculptor by Hudson Bay Company
Results:

1. A collection of 3D models of each component

2. Assembly of the individual models into animations and Internet web study material.
Preliminaries: Capturing Macro geometry:

- **Shape From Silhouette**
  - Works for objects
  - Robust
  - Visual hull not true object surface

- **Structure From Motion**
  - Works for Scenes
  - Typically sparse
  - Sometimes fragile (no salient points in scene)

- **(Dense “Stereo” -- Later)**
  - Use as second refinement step
Multi-Tiered Models:

- Commonly:
  - Two tiers: 3D Geometry and appearance (* texture mapping)
  - Used in graphics applications, recovered in Vision applications
- Three-Tier
  - Macro scale: describes scene geometry (triangulated mesh)
  - Meso scale: fine scale geometric detail (displacement map)
  - Micro: fine scale geometry and reflectance (Texture basis)
- Captured by sequential refinement
Geometry alone does not solve modeling!
Need: Multi-Scale Model

Multi-Scale model: **Macro** geometry, **Meso** depth, **Micro** texture
Three scales map naturally to CPU and GPU hardware layers

Key issue: Efficient memory access and processing

1. Macro: Conventional geometry processing
2. Meso: Pixel shader
   - Fixed code, variable data access
3. Micro: Shader or Register comb.
   - Fixed code, fixed data access

Speedup
10x
10x
2. **Meso Structure:**
Depth with respect to a plane

- Base geometry
- Displacement map
- Displacement mapped geometry

**Flat texture**

**Displacement mapped**
Per-point cost function

\[ \Phi(X, n) = \sum_i h(X, P_i) \| I_i(P_i(X)) - R(X, n, L_i) \| \]

Visibility + sampling \quad reflectance

\[ \frac{\partial S}{\partial t} = (2\Phi k - \langle \nabla \Phi, n \rangle) n \]

Deformable mesh \quad Depth from Base
1. Sample \(d\) and ray at \(N\) (say15) points.
2. Find point location \(j\) of intersection
3. Approximate \(d\) with line, calculate intersection
4. Potentially iterate if needed for accuracy
Results:

Over 100 fps on consumer graphics cards
3. Micro structure: Spatial texture basis

- Modulated texture
- Traditional texture

- Fixed execution and data access pattern

=> very fast implementation in graphics hardware
How/why do dynamic textures work?

3D geometry and texture warp map between views and texture images

View

Re-projected geometry

Texture

Problem:
Texture images different
Sources of errors:

3D geometry and texture warp map between views and texture images

1: Planar error: Incorrect texture coordinates

2: Out of plane error: Object surface /= texture plane
1. Moving sine wave can be modeled:

\[ I(t) = \sin(u + at) \]
\[ = \sin(u) \cos(at) + \cos(u) \sin(at) \]
\[ = \sin(u)y_1(t) + \cos(u)y_2(t) \]

Spatially fixed basis

2. Small image motion

\[ I = I_0 + \frac{\partial I}{\partial u} \Delta u + \frac{\partial I}{\partial v} \Delta v \]

Spatially fixed basis
On the object/texture plane:

- Variation resulting from small warp perturbations
- Taylor expansion:

\[
T(\text{view}) = T_0 + \frac{\partial}{\partial \mu} T_0 \Delta \mu + h.o.t.
\]

Small if \( \Delta \mu \) small and \( T_0 \) smooth

Similarly: Can derive linear basis for out of plane and light variation!
Image “warp”

\[ T(x) = I(W(x, \mu)) \]

Image variability caused by an imperfect warp

\[ \Delta T = I(W(x, \mu + \Delta \mu)) - T_w \]

First order approximation

\[ \Delta T = I(W(x, \mu)) + \nabla T \frac{\partial W}{\partial \mu} - T_w = \nabla T \frac{\partial W}{\partial \mu} \]

Concrete examples

- Image plane
- Out of plane
Variability due to a planar projective warp (homography)

**Homography warp**

\[
\begin{bmatrix}
  u' \\
  v'
\end{bmatrix} = \mathcal{W}_h(x_h, h) = \frac{1}{1+h_7u+h_8v}
\begin{bmatrix}
  h_1u & h_3v & h_5 \\
  h_2u & h_4v & h_6
\end{bmatrix}
\]

**Projective variability:**

\[
\Delta T_h = \frac{1}{c_1} \left[ \frac{\partial T}{\partial u}, \frac{\partial T}{\partial v} \right]
\begin{bmatrix}
  u & 0 & v & 0 & 1 & 0 & -\frac{uc_2}{c_1} & -\frac{vc_2}{c_1} \\
  0 & u & 0 & v & 0 & 1 & -\frac{uc_3}{c_1} & -\frac{vc_3}{c_1}
\end{bmatrix}
\begin{bmatrix}
  \Delta h_1 \\
  \vdots \\
  \Delta h_8
\end{bmatrix}
\]

\[
= [B_1 \ldots B_8][y_1, \ldots, y_8]^T = B_hy_h
\]

**Where**

\[c_1 = 1 + h_7u + h_8v\ , \quad c_2 = h_1u + h_3v + h_5\]

and

\[c_3 = h_2u + h_4v + h_6\]
Variability due to a planar projective warp (homography)

- **Homography warp**

\[
\begin{bmatrix}
  u' \\
  v'
\end{bmatrix} = \mathcal{W}_h(x_h, h) = \frac{1}{1+h_7u+h_8v} \begin{bmatrix}
  h_1u & h_3v & h_5 \\
  h_2u & h_4v & h_6
\end{bmatrix}
\]

- **Projective variability:**

\[
\Delta T_h = \frac{1}{c_1} \left[ \frac{\partial T}{\partial u}, \frac{\partial T}{\partial v} \right] = \begin{bmatrix}
  u & 0 & v & 0 & 1 & 0 & -\frac{uc_2}{c_1} & -\frac{vc_2}{c_1} \\
  0 & u & 0 & v & 0 & 1 & \frac{uc_3}{c_1} & \frac{vc_3}{c_1}
\end{bmatrix} \begin{bmatrix}
  \Delta h_1 \\
  \vdots \\
  \Delta h_8
\end{bmatrix}
\]
Out-of-plane variability

• Let $r = [\alpha, \beta]$ angle for ray to scene point

• Pre-warp texture plane rearrangement:

\[
\begin{bmatrix}
d\alpha \\
d\beta
\end{bmatrix} = \mathcal{W}_p(x, d) = d(u, v) \begin{bmatrix}
tan \alpha \\
tan \beta
\end{bmatrix}
\]

• Texture basis

\[
\Delta T_p = d(u, v) \left[ \frac{\partial T}{\partial u}, \frac{\partial T}{\partial v} \right] \begin{bmatrix}
\frac{1}{\cos^2 \alpha} & 0 \\
0 & \frac{1}{\cos^2 \beta}
\end{bmatrix} \begin{bmatrix}
\Delta \alpha \\
\Delta \beta
\end{bmatrix} = B_p y_p
\]
Photometric variation

Analytic formula for irradiance for a convex Lambertian object under distant illumination (with attached shadows) - spherical harmonics

[Barsi and Jacobs, Ramamoorthi and Hanrahan 2001]

\[
T(\alpha, \beta, \theta, \phi) \approx \sum_{l=0}^{2} \sum_{k=-l}^{l} L_{lk}(\alpha, \beta) A_l Y_{lk}(\theta, \phi)
\]

\[
T = [B_1 \cdots B_9][L_1 \cdots L_9]^T
\]
Example of photometric variation

Light basis images

Rendered combination
Similarly, composite texture intensity variability

\[ \Delta T = \Delta T_s + \Delta T_d + \Delta T_l + \Delta T_e \]

Can be modeled as sum of basis

\[ \Delta T = B_s y_s + B_d y_d + B_l y_1 + \Delta T_e \]

\[ = B y + \Delta T_e \]
How to compute?

From a 3D graphics model:
1. Texture intensity derivatives
2. Jacobian of warp or displacement function

• Results in about 20 components:
  – $T_0$
  – 8 for planar,
  – 2 out-of plane (parallax),
  – 3-9 light

From video:
• We can expect an approximately 20dim variation in the space of all input texture images.

=> Extract this subspace
1. Take input video sequence, use SFS/SFM geometry to warp into texture space

Input Images

Geometry

Texture warp

... → ... → ...

PCA

2. Extract a 20-dim subspace through PCA

TexDemo
Are analytic image derivatives and PCA basis the same?

• Same up to a linear transform!

• Experimental verification: planar homog

99% agreement
Example renderings from 3D models
Recap: hierarchical model scale levels

1. **Macro:**
   - SFM, SFS can generate coarse geometry but not detailed enough for realistic rendering
   - Integrate tracking and structure computation
   
   **Scale:** dozen pixels and up

2. **Meso:**
   - Refine coarse geometry and acquire reflectance–variational surface evolution
   
   **Scale:** 1–dozen pixels

3. **Micro spatial basis:**
   - Represents appearance and corrects for small geometric texture errors limited by linearity of image
   
   **Scale:** 0-5 pixels
Comparison

1. **Static texturing**: (Many, e.g. Baumgartner et al. 3DSOM)
   - Average color projected to point.
   - Better: Pick color minimizing reprojection error over all input images
   Works when model geometry is close to ground truth and light simple

2. **Viewdependent texture**: (Debevec et al)
   - Pick color from closest input photograph (or interpolate from nearest 3)
   Works when possible to store large numbers of images

3. **Lumigraph / Surface light field**: (Buehler et al / Wood et al)
   - Store all ray colors (plenoptic function) intersecting a proxy surface
   Works if proxy surface close to true geometry

4. **Dynamic texture**: (Ours: Jagersand ’97/ Matusik / Ikeuchi99 /Vasilescu04...)
   - Derive a Taylor expansion and represent derivatives of view dependency
   Works for light and small (1-5 pixel) geometric displacements.
1. **Simple Geom**: SFS alone ok

2. **General Geom**: SFS + Variational Shape and Reflectance fitting (+View dep texture)

3. **Complex Light**: Dynamic Texture / Lumigraph

4. **Challenge for Computer Vision**
1. **Simple Geom**: SFS alone ok
2. **General Geom**: SFS + Variational Shape and Reflectance fitting (+View dependence on texture)
3. **Complex Light**: Dynamic Texture / Lumigraph
4. **Challenge for Computer Vision**

<table>
<thead>
<tr>
<th>err (var)</th>
<th>temple</th>
<th>house</th>
<th>eleph.</th>
<th>wreath</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>10.8(1.5)</td>
<td>11.8(1.2)</td>
<td>19.0(1.4)</td>
<td>28.4(2.8)</td>
</tr>
<tr>
<td>VDTM</td>
<td>8.3(1.9)</td>
<td>9.8(1.3)</td>
<td>10.1(1.9)</td>
<td>21.4(3.5)</td>
</tr>
<tr>
<td>Lumigr</td>
<td>10.8(2.5)</td>
<td>9.8(1.2)</td>
<td>5.9(0.7)</td>
<td>14.3(1.3)</td>
</tr>
<tr>
<td>DynTex</td>
<td>7.3(1.0)</td>
<td>9.4(1.0)</td>
<td>6.6(0.7)</td>
<td>13.4(1.2)</td>
</tr>
</tbody>
</table>

Table 1. Numerical texture errors and variance. %-scale.
Example of render differences

• **Jade Elephant**
  - Complex Reflectance (specularities and scattering)

<table>
<thead>
<tr>
<th></th>
<th>Input</th>
<th>Static</th>
<th>ViewDep</th>
<th>Lumigr.</th>
<th>DynTex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specular highlight</td>
<td><img src="image" alt="Specular highlight" /></td>
<td><img src="image" alt="Specular highlight" /></td>
<td><img src="image" alt="Specular highlight" /></td>
<td><img src="image" alt="Specular highlight" /></td>
<td><img src="image" alt="Specular highlight" /></td>
</tr>
</tbody>
</table>
Capturing non-rigid animatable models

Current PhD project, Neil Birkbeck
Questions?

More information:
• Downloadable renderer+models
  www.cs.ualberta.ca/~vis/ibmr

• Capturing software + IEEE VR tutorial text
  www.cs.ualberta.ca/~vis/VR2003tut

• Main references for this talk:
  Jagersand et al “Three Tier Model” 3DPVT 2008 …. 
  Jagersand “Image-based Animation…” CVPR 1997

• More papers: www.cs.ualberta.ca/~jag
Video: see web page:
www.cs.ualberta.ca/~vis/ibmr/movies/capsys_1min.avi