Image-based modeling

Digital Visual Effects

Yung-Yu Chuang

Outline

- Models from multiple (sparse) images
  - Structure from motion
  - Facade
- Models from single images
  - Tour into pictures
  - Single view metrology
  - Other approaches

Models from multiple images

(Façade, Debevec et. al. 1996)

Facade

- Use a sparse set of images
- Calibrated camera (intrinsic only)
- Designed specifically for modeling architecture
- Use a set of blocks to approximate architecture

- Three components:
  - geometry reconstruction
  - texture mapping
  - model refinement
**Idea**

- **Geometric modeling**
  - A block is a geometric primitive with a *small* set of parameters.
  - Hierarchical modeling for a scene: Rotation and translation could be constrained.

- **Reasons for block modeling**
  - Architectural scenes are well modeled by geometric primitives.
  - Blocks provide a high level abstraction, easier to manage and add constraints.
  - No need to infer surfaces from discrete features; blocks essentially provide prior models for architectures.
  - Hierarchical block modeling effectively reduces the number of parameters for robustness and efficiency.
Reconstruction

\[ \minimize \mathcal{O} = \sum \text{Err}_i \]

\[ \mathbf{m} = R_j (\mathbf{v} \times (\mathbf{d} - t_j)) \]

\[ m_x x + m_y y - m_z f = 0 \]

\[ \text{Err}_i = \int_0^l h^2(s) ds \]

[Diagram of camera and model nonlinear w.r.t.]

Results

3 of 12 photographs

\[ h_1 = \frac{m_x x_1 + m_y y_1 + m_z}{\sqrt{m_x^2 + m_y^2}} \]

\[ h_2 = \frac{m_x x_2 + m_y y_2 + m_z}{\sqrt{m_x^2 + m_y^2}} \]

\[ h(s) = h_1 + s \frac{h_2 - h_1}{l} \]

\[ \text{Err}_i = \int_0^l h^2(s) ds \]

\[ = \frac{l}{3} (h_1^2 + h_1 h_2 + h_2^2) \]

\[ \mathbf{m} = (m_x, m_y, m_z)^T \]

\[ A = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} \]

\[ B = \frac{l}{3(m_x^2 + m_y^2)} \begin{pmatrix} 1 & 0.5 \end{pmatrix} \]

nonlinear w.r.t. camera and model
Texture mapping

Texture mapping in real world

Demo movie
Michael Naimark, San Francisco Museum of Modern Art, 1984
Texture mapping

View-dependent texture mapping
View-dependent texture mapping

Model-based stereo
- Use stereo to refine the geometry

Stereo
- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - calibration
    - point correspondence

Stereo
- known camera viewpoints

Stereo
- scene point
- image plane
- optical center
Stereo correspondence

- Determine Pixel Correspondence
  - Pairs of points that correspond to same scene point

- Epipolar Constraint
  - Reduces correspondence problem to 1D search along conjugate epipolar lines

Finding correspondences

- apply feature matching criterion (e.g., correlation or Lucas-Kanade) at all pixels simultaneously
- search only over epipolar lines (much fewer candidate positions)

Image registration (revisited)

- How do we determine correspondences?
  - block matching or SSD (sum squared differences)

\[ E(x, y; d) = \sum_{(x', y') \in N(x, y)} [I_L(x' + d, y') - I_R(x', y')]^2 \]

\[ d \] is the disparity (horizontal motion)

- How big should the neighborhood be?

Neighborhood size

- Smaller neighborhood: more details
- Larger neighborhood: fewer isolated mistakes
Depth from disparity

\[ \text{disparity} = x - x' = \frac{\text{baseline} \cdot f}{z} \]

Stereo reconstruction pipeline

- Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

- What will cause errors?
  - Camera calibration errors
  - Poor image resolution
  - Occlusions
  - Violations of brightness constancy (specular reflections)
  - Large motions
  - Low-contrast image regions

Model-based stereo

Results
Comparisons

- single texture, flat
- VDTM, flat
- VDTM, model-based stereo

Final results

Kite photography

Final results
Results

Commercial packages

- Autodesk REALVIZ ImageModeler

The Matrix

Cinefex #79, October 1999.

Since the bullet-time rig would be visible in shots featuring a 360-degree sweep of the characters, it was employed only for the shooting of the foreground subject—namely, the actors or their stunt doubles—necessitating a different approach for the backgrounds. Shot separately, the backgrounds used a virtual cinematography process that allowed a 360-degree environment to be constructed in the computer from stills taken on set. This approach for generating the backgrounds was based on the Berkeley Tower flyover, a novel image-based rendering technique presented at Siggraph '97 by George Borshukov and Paul Debevec, a researcher at UC Berkeley. The technique employed twenty stills of that town's college campus to create a virtual environment through which the camera could travel. "Instead of reinventing the background in traditional CG fashion—painting textures, shooting orthographic views of the set, and then proceeding to texture replication—we generated a completely free, high-resolution camera move that would have been impossible to achieve using traditional CG," Borshukov said, "and we did it working from just a handful of stills."
The Matrix

- **Academy Awards for Scientific and Technical achievement for 2000**
  
  To George Borshukov, Kim Libreri and Dan Piponi for the development of a system for image-based rendering allowing choreographed camera movements through computer graphic reconstructed sets.

This was used in The Matrix and Mission Impossible II; See The Matrix Disc #2 for more details

---

**Models from single images**

---

**Vanishing points**

- **Vanishing point**
  - projection of a point at infinity
Vanishing points

• Properties
  - Any two parallel lines have the same vanishing point \( v \)
  - The ray from \( C \) through \( v \) is parallel to the lines
  - An image may have more than one vanishing point

Vanishing lines

• Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the horizon line
    • also called vanishing line
  - Note that different planes define different vanishing lines

Computing vanishing points

\[
P = P_0 + tD
\]

• Properties \( v = \Pi P_0 \)
  - \( P_0 \) is a point at infinity, \( v \) is its projection
  - They depend only on line direction
  - Parallel lines \( P_0 + tD, P_1 + tD \) intersect at \( P_v \)

Tour into pictures

• Create a 3D “theatre stage” of five billboards

• Specify foreground objects through bounding polygons

• Use camera transformations to navigate through the scene
The idea

- Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)
- Key assumptions:
  - All walls of volume are orthogonal
  - Camera view plane is parallel to back of volume
  - Camera up is normal to volume bottom
  - Volume bottom is y=0
- Can use the vanishing point to fit the box to the particular Scene!

Fitting the box volume

- User controls the inner box and the vanishing point placement (6 DOF)

Foreground Objects

- Use separate billboard for each

  - For this to work, three separate images used:
    - Original image.
    - Mask to isolate desired foreground images.
    - Background with objects removed
Foreground Objects

- Add vertical rectangles for each foreground object.
- Can compute 3D coordinates P0, P1 since they are on known plane.
- P2, P3 can be computed as before (similar triangles).

Example

1. Find world coordinates (X,Y,Z) for a few points
2. Connect the points with planes to model geometry
   - Texture map the planes

Measurements on planes

Approach: unwarped then measure
What kind of warp is this?

Image rectification

To unwarped (rectify) an image
- solve for homography $H$ given $p$ and $p'$
- solve equations of the form: $wp' = Hp$
  - linear in unknowns: $w$ and coefficients of $H$
  - $H$ is defined up to an arbitrary scale factor
  - how many points are necessary to solve for $H$?
Solving for homographies

\[
\begin{bmatrix}
x'_i \\
y'_i \\
1
\end{bmatrix} =
\begin{bmatrix}
h_{00} & h_{01} & h_{02} \\
h_{10} & h_{11} & h_{12} \\
h_{20} & h_{21} & h_{22}
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i \\
1
\end{bmatrix}
\]

\[
x'_i = \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}}
\]

\[
y'_i = \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}}
\]

\[
x'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{00}x_i + h_{01}y_i + h_{02}
\]

\[
y'_i(h_{20}x_i + h_{21}y_i + h_{22}) = h_{10}x_i + h_{11}y_i + h_{12}
\]

\[
\begin{bmatrix}
x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\
0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i
\end{bmatrix}
= \begin{bmatrix}
h_{00} \\
h_{01} \\
h_{02} \\
h_{10} \\
h_{11} \\
h_{12} \\
h_{20} \\
h_{21} \\
h_{22}
\end{bmatrix}
\]

- Defines a least squares problem:

\[
\text{minimize } \|Ah - 0\|^2
\]

- Since \( h \) is only defined up to scale, solve for unit vector \( \hat{h} \)

- Works with 4 or more points

Finding world coordinates \((X,Y,Z)\)

1. Define the ground plane \((Z=0)\)
2. Compute points \((X,Y,0)\) on that plane
3. Compute the heights \(Z\) of all other points

Measuring height

- Camera height

- Height at 1 meter

- Height at 2 meters

- Height at 3 meters
Computing vanishing points

- Intersect $p_1q_1$ with $p_2q_2$

- Least squares version
  - Better to use more than two lines and compute the "closest" point of intersection
  - See notes by Bob Collins for one good way of doing this:

Criminisi et al., ICCV 99

- Load in an image
- Click on lines parallel to X axis
  - repeat for Y, Z axes
- Compute vanishing points

Vanishing point (at infinity)
Oh et. al. SIGGRAPH 2001

Automatic popup

Input Geometric Labels Cut’n’Fold 3D Model

Image Ground

Learned Models Sky

Geometric cues

Color Texture

Location Perspective

Feature Descriptions Value
Color
C1. RGB values: mean 3 3
C2. RGB values: standard deviation 3 3
C3. Hsv: histogram (3 bins) and entropy 6 6
C4. Summation: histogram (3 bins) and entropy 3 3
Texture
T1. DOG filters: mean dyresponse 1 7
T2. DOG filters: mean of variables in T1 1 0
T3. DOG filters: sum of variables in T1 1 1
T4. DOG filters: mean - median of variables in T1 1 1
T5. texture: mean of response 1 7
T6. texture: mean of variables in T5 1 0
T7. texture: mean - median of variables in T5 1 1
Location and Shape
L1. Location: normalized x and y, mean 2 2
L2. Location: normalized x and y, 10th and 90th percentile 4 4
L3. Location: raw x and y, 10th and 90th percentile 2 2
L4. Shape: number of toppegs in centration 1 1
L5. Shape: number of edges of convex hull 1 1
L6. Shape: ratio of convex hull
L7. Shape: whether the centration region is contiguous 1 1
3D Geometry
G1. Long Lines: total number in centration region 1 1
G2. Long Lines: % of nearly parallel part of lines 1 1
G3. Line Intersection: list of 15 selections, entropy 13 11
G4. Line Intersection: % right of center 1 1
G5. Line Intersection: % above center 1 1
G6. Line Intersection: % far from center at 90° rotations 8 8
G7. Line Intersection: % very far from center at 90° rotations 8 5
G8. Ternary gradient: % and y "zigzag" (T2) center 2 2
This approach works roughly for 35% of images.
3-sweep

References


The Avengers

- video