Multiple View Geometry

© Martin Quinn

...with a lot of slides stolen from Steve Seitz and Jianbo Shi

15-463: Computational Photography
Alexei Efros, CMU, Fall 2007
Our Goal
The Plenoptic Function

\[ P(\theta, \phi, \lambda, t, V_x, V_y, V_z) \]

How can we compress this into something manageable?
Stereo Reconstruction

The Stereo Problem

- Shape from two (or more) images
- Biological motivation

known camera viewpoints
Why do we have two eyes?

Cyclope vs. Odysseus
1. Two is better than one

"Just checking."
2. Depth from Convergence

\[ d = \frac{c}{2\tan\left(\frac{a}{2}\right)} \]

Human performance: up to 6-8 feet
3. Depth from binocular disparity

Sign and magnitude of disparity

P: converging point

C: object nearer projects to the outside of the P, disparity = +

F: object farther projects to the inside of the P, disparity = -
Stereo

Scene point

Image plane

Optical center
Stereo

Basic Principle: Triangulation

- Gives reconstruction as intersection of two rays
- Requires
  - calibration
  - *point correspondence*
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*
Stereo image rectification
Stereo image rectification

Image Reprojection
- reproject image planes onto common plane parallel to line between optical centers
- a homography (3x3 transform) applied to both input images
- pixel motion is horizontal after this transformation
Stereo Rectification
Your basic stereo algorithm

For each epipolar line
  For each pixel in the left image
    • compare with every pixel on same epipolar line in right image
    • pick pixel with minimum match cost

Improvement: match *windows*
  • This should look familiar...
  • Can use Lukas-Kanade or discrete search (latter more common)
Window size

Effect of window size

- Smaller window
  +
  -

- Larger window
  +
  -
Stereo results

- Data from University of Tsukuba
- Similar results on other images without ground truth

Scene

Ground truth
Results with window search

Window-based matching (best window size)  Ground truth
Better methods exist...  

State of the art method

Ground truth
Depth from disparity

Input image (1 of 2)

Depth map
[Slentski & Kang '95]

3D rendering

\[
\text{disparity} = x - x' = \frac{\text{baseline} \times f}{z}
\]
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
Stereo matching

Need texture for matching

Julesz-style Random Dot Stereogram
Active stereo with structured light

Project “structured” light patterns onto the object
- simplifies the correspondence problem
Active stereo with structured light
Laser scanning

Optical triangulation

- Project a single stripe of laser light
- Scan it across the surface of the object
- This is a very precise version of structured light scanning

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
Portable 3D laser scanner (this one by Minolta)
Real-time stereo

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

Used for robot navigation (and other tasks)
  • Several software-based real-time stereo techniques have been developed (most based on simple discrete search)
Volumetric Stereo

Goal: Determine transparency, radiance of points in V

Scene Volume $V$

Input Images (Calibrated)
Discrete Formulation: Voxel Coloring

Goal: Assign RGBA values to voxels in V photo-consistent with images
Complexity and Computability

Discretized Scene Volume

$N^3$ voxels
$C$ colors

All Scenes ($C^{N^3}$)

True Scene

Photo-Consistent Scenes
Issues

Theoretical Questions
  • Identify class of *all* photo-consistent scenes

Practical Questions
  • How do we compute photo-consistent models?
Voxel Coloring Solutions

1. C=2 (silhouettes)
   - Volume intersection [Martin 81, Szeliski 93]

2. C unconstrained, viewpoint constraints
   - Voxel coloring algorithm [Seitz & Dyer 97]

3. General Case
   - Space carving [Kutulakos & Seitz 98]
Reconstruction from Silhouettes

Approach:

• *Backproject* each silhouette
• Intersect backprojected volumes
Volume Intersection

Reconstruction Contains the True Scene
  • But is generally not the same
  • In the limit get *visual hull*
    > Complement of all lines that don’t intersect S
Voxel Algorithm for Volume Intersection

Color voxel black if on silhouette in every image

- $O(MN^3)$, for $M$ images, $N^3$ voxels
- Don’t have to search $2^{N^3}$ possible scenes!
Properties of Volume Intersection

Pros
- Easy to implement, fast
- Accelerated via octrees [Szeliski 1993]

Cons
- No concavities
- Reconstruction is not photo-consistent
- Requires identification of silhouettes
Voxel Coloring Solutions

1. C=2 (silhouettes)
   • Volume intersection [Martin 81, Szeliski 93]

2. C unconstrained, viewpoint constraints
   • Voxel coloring algorithm [Seitz & Dyer 97]

3. General Case
   • Space carving [Kutulakos & Seitz 98]
Voxel Coloring Approach

1. Choose voxel
2. Project and correlate
3. Color if consistent
   (standard deviation of pixel colors below threshold)

Visibility Problem: in which images is each voxel visible?
The Global Visibility Problem

Which points are visible in which images?

Forward Visibility

Inverse Visibility
Depth Ordering: visit occluders first!

Condition: depth order is view-independent
Calibrated Image Acquisition

Calibrated Turntable

Selected Dinosaur Images

Selected Flower Images
Voxel Coloring Results (Video)

Dinosaur Reconstruction
72 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI

Flower Reconstruction
70 K voxels colored
7.6 M voxels tested
7 min. to compute
on a 250MHz SGI
Limitations of Depth Ordering

A view-independent depth order may not exist

Need more powerful general-case algorithms
- Unconstrained camera positions
- Unconstrained scene geometry/topology
Voxel Coloring Solutions

1. C=2 (silhouettes)
   • Volume intersection [Martin 81, Szeliski 93]

2. C unconstrained, viewpoint constraints
   • Voxel coloring algorithm [Seitz & Dyer 97]

3. General Case
   • Space carving [Kutulakos & Seitz 98]
Space Carving Algorithm

- Initialize to a volume $V$ containing the true scene
- Choose a voxel on the current surface
- Project to visible input images
- Carve if not photo-consistent
- Repeat until convergence
Convergence

Consistency Property

- The resulting shape is photo-consistent
  > all inconsistent points are removed

Convergence Property

- Carving converges to a non-empty shape
  > a point on the true scene is *never* removed
Structure from Motion

Reconstruct

- Scene geometry
- Camera motion

Unknown camera viewpoints
Three approaches

(a) Geometry-Based
- User input
- Texture maps
  - Modeling Program
  - Model
  - Rendering Algorithm
  - Renderings

(b) Hybrid Approach
- Images
- User input
  - Photogrammetric Modeling Program
  - Basic model
  - Model-Based Stereo
  - Depth maps
  - Image Warping
  - Renderings

(c) Image-Based
- Images
- User input (depth maps)
  - Stereo Correspondence
  - Depth maps
  - Image Warping
  - Renderings
Outline of a simple algorithm (1)

- Based on constraints
- Input to the algorithm (1): two images
Outline of a simple algorithm (2)

- Input to the algorithm (2):
  User select edges and corners
Outline of a simple algorithm (3)

• Camera Position and Orientation
  
  Determine the position and orientation of camera
Outline of a simple algorithm (4)

- Computing projection matrix and Reconstruction
Outline of a simple algorithm (5)

- Compute 3D textured triangles
Facade

*SFMOMA (San Francisco Museum of Modern Art) by Yizhou Yu,*
Façade (Debevec et al) inputs
Façade (Debevec et al)