Topics of This Lecture

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis & 3D Reconstruction
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications

Geometric vision

- Goal: Recovery of 3D structure
  - What cues in the image allow us to do this?

Visual Cues

- Shading

Merle Norman Cosmetics, Los Angeles

Visual Cues

- Shading
- Texture

The Visual Cliff, by William Vandiver, 1960
Visual Cues

- Shading
- Texture
- Focus
- Perspective

From The Art of Photography, Canon

Our Goal: Recovery of 3D Structure

- We will focus on perspective and motion
- We need multi-view geometry because recovery of structure from one image is inherently ambiguous

To Illustrate This Point...

- Structure and depth are inherently ambiguous from single views.

Stereo Vision

http://www.well.com/~jimg/stereo/stereo_list.html

Slide credits: Steve Seitz, Kristen Grauman, S. Leibe
What Is Stereo Vision?

• Generic problem formulation: given several images of the same object or scene, compute a representation of its 3D shape

• Narrower formulation: given a calibrated binocular stereo pair, fuse it to produce a depth image.
  - Humans can do it

Stereograms: Invented by Sir Charles Wheatstone, 1838

Autostereograms: [http://www.magiceye.com](http://www.magiceye.com)
**Historic Origin: Random Dot Stereograms**

- Julesz 1960: Do we identify local brightness patterns before fusion (monocular process) or after (binocular)?
- To test: pair of synthetic images obtained by randomly spraying black dots on white objects

**Random Dot Stereograms**

- When viewed monocularly, they appear random; when viewed stereoscopically, see 3d structure.

**Random Dot Stereograms**


**Application of Stereo: Robotic Exploration**

Nomad robot searches for meteorites in Antarctica

Real-time stereo on Mars

**Topics of This Lecture**

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis & 3D Reconstruction
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications

**Depth with Stereo: Basic Idea**

- Basic Principle: Triangulation
  - Gives reconstruction as intersection of two rays
  - Requires
    - Camera pose (calibration)
    - Point correspondence
Camera Calibration

Extrinsic parameters:
- Camera frame ➔ Reference frame

Intrinsic parameters:
- Image coordinates relative to camera ➔ Pixel coordinates

- Parameters
  - **Extrinsic**: rotation matrix and translation vector
  - **Intrinsic**: focal length, pixel sizes (mm), image center point, radial distortion parameters

We'll assume for now that these parameters are given and fixed.

Geometry for a Simple Stereo System

- First, assuming parallel optical axes, known camera parameters (i.e., calibrated cameras):

Depth From Disparity

\[(x', y') = (x + D(x, y), y)\]

General Case With Calibrated Cameras

- The two cameras need not have parallel optical axes.
Stereo Correspondence Constraints

- Given \( p \) in the left image, where can the corresponding point \( p' \) in the right image be?

---

Epipolar Geometry

- **Epipolar Plane**
- **Epipoles**
- **Baseline**
- **Epipolar Lines**
Epipolar Constraint

- Potential matches for $p$ have to lie on the corresponding epipolar line $l'$.
- Potential matches for $p'$ have to lie on the corresponding epipolar line $l$.

http://www.ai.sri.com/~luong/research/My3DViewer/EpipolarGeo.html

Example

Example: Converging Cameras

As position of 3d point varies, epipolar lines "rotate" about the baseline

Example: Motion Parallel With Image Plane

Let’s Formalize This!

- For a given stereo rig, how do we express the epipolar constraints algebraically?
- For this, we will need some linear algebra.
- But don’t worry! We’ll go through it step by step...
If the rig is calibrated, we know:
- How to rotate and translate camera reference frame 1 to get to camera reference frame 2.
  - Rotation: 3 x 3 matrix; translation: 3 vector.

Express 3D rotation as series of rotations around coordinate axes by angles \( \alpha, \beta, \gamma \)

Overall rotation is product of these elementary rotations:
\[
R = R_x R_y R_z
\]

Camera-centered coordinate systems are related by known rotation \( R \) and translation \( T \):
\[
X' = RX + T
\]

Vector cross product takes two vectors and returns a third vector that’s perpendicular to both inputs.
So here, \( c \) is perpendicular to both \( a \) and \( b \), which means the dot product = 0.
Matrix Form of Cross Product

\[ \mathbf{a} \times \mathbf{b} = [a_y, -a_z, a_x] \]

"skew symmetric" matrix

From Geometry to Algebra

\[ \mathbf{X}' = \mathbf{RX} + \mathbf{T} \]
\[ \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) = 0 \]
\[ \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) = 0 \]

Normal to the plane

Essential Matrix

Let \( \mathbf{E} = \mathbf{T} \mathbf{R} \)

\[ \mathbf{X}' \mathbf{E} \mathbf{X} = 0 \]

- This holds for the rays \( \mathbf{p} \) and \( \mathbf{p}' \) that are parallel to the camera-centered position vectors \( \mathbf{X} \) and \( \mathbf{X}' \), so we have: \( \mathbf{p}' \mathbf{E} \mathbf{p} = 0 \)
- \( \mathbf{E} \) is called the essential matrix, which relates corresponding image points [Longuet-Higgins 1981]

Essential Matrix and Epipolar Lines

Epipolar constraint: if we observe point \( \mathbf{p} \) in one image, then its position \( \mathbf{p}' \) in second image must satisfy this equation.

\[ \mathbf{E} \mathbf{p} \]

\( \mathbf{p}' \mathbf{E} \mathbf{l} \) is the coordinate vector representing the epipolar line for point \( \mathbf{p} \)

\( \mathbf{l} = \mathbf{E} \mathbf{p}' \) is the coordinate vector representing the epipolar line for point \( \mathbf{p}' \)

Essential Matrix: Properties

- Relates image of corresponding points in both cameras, given rotation and translation.
- Assuming intrinsic parameters are known

\[ \mathbf{E} = \mathbf{T} \mathbf{R} \]

Essential Matrix Example: Parallel Cameras

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.
Essential Matrix Example: Parallel Cameras

\[
R = I, \quad T = [-d, 0, 0]^T, \quad E = [T^T R]
\]

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

More General Case

\[
(x', y') = (x + D(x, y), y)
\]

What about when cameras' optical axes are not parallel?

Stereo Image Rectification

- In practice, it is convenient if image scanlines are the epipolar lines.

Algorithm
- Reproject image planes onto a common plane parallel to the line between optical centers
- Pixel motion is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection

Topics of This Lecture

- Geometric vision
  - Visual cues
  - Stereo vision
- Epipolar geometry
  - Depth with stereo
  - Geometry for a simple stereo system
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis & 3D Reconstruction
  - Correspondence search
  - Additional correspondence constraints
  - Possible sources of error
  - Applications
- Main Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

Stereo Reconstruction
Correspondence Problem

Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Slide credit: Kristen Grauman

Dense Correspondence Search

- For each pixel in the first image
  - Find corresponding epipolar line in the right image
  - Examine all pixels on the epipolar line and pick the best match (e.g. SSD, correlation)
  - Triangulate the matches to get depth information

- This is easiest when epipolar lines are scanlines
  ⇒ Rectify images first

Example: Window Search

- Data from University of Tsukuba

Example: Window Search

- Data from University of Tsukuba

Effect of Window Size

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.
Alternative: Sparse Correspondence Search

- Idea: Restrict search to sparse set of detected features
- Rather than pixel values (or lists of pixel values) use feature descriptor and an associated feature distance
- Still narrow search further by epipolar geometry

What would make good features?

Dense vs. Sparse

- Sparse
  - Efficiency
  - Can have more reliable feature matches, less sensitive to illumination than raw pixels
  - But...
    - Have to know enough to pick good features
    - Sparse information

- Dense
  - Simple process
  - More depth estimates, can be useful for surface reconstruction
  - But...
    - Breaks down in textureless regions anyway
    - Raw pixel distances can be brittle
    - Not good with very different viewpoints

Difficulties in Similarity Constraint

- Untextured surfaces
- Occlusions

Additional Correspondence Constraints

- Similarity
- Uniqueness
- Ordering
- Disparity gradient

Epipolar lines constrain the search to a line, and these appearance and ordering constraints further reduce the possible matches.

Possible Sources of Error?

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Violations of brightness constancy (e.g., specular reflections)
- Large motions

Application: Depth for Segmentation

Edges in disparity in conjunction with image edges enhances contour found
Application: View Interpolation

Right Image

Disparity

Application: View Interpolation

Left Image

Application: Free-Viewpoint Video

http://www.liberovision.com

Summary: Stereo Reconstruction

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

- So far, we have only considered calibrated cameras...

- Next lecture
  - Uncalibrated cameras
  - Camera parameters
  - Revisiting epipolar geometry
  - Robust fitting
References and Further Reading

- Background information on epipolar geometry and stereopsis can be found in Chapters 10.1-10.2 and 11.1-11.3 of
  D. Forsyth, J. Ponce,
  Computer Vision - A Modern Approach.
  Prentice Hall, 2003

- More detailed information (if you really want to implement 3D reconstruction algorithms) can be found in Chapters 9 and 10 of
  R. Hartley, A. Zisserman
  Multiple View Geometry in Computer Vision
  2nd Ed., Cambridge Univ. Press, 2004

D. Leibe

Computer Vision WS 11/12