CS4670/5670: Computer Vision
Kavita Bala

Lec 25: Stereo
Announcements

• Prelim grades out
  – Solutions on CMS
  – Regrades on Problem 4 and 6
    • Please drop off prelim in hand back room or in office hours with TAs
Calibration using a reference object

- Place a known object in the scene
  - identify correspondence between image 2D and scene 3D
  - compute mapping from scene to image

Issues
- must know geometry very accurately
- must know 3D->2D correspondence
Camera parameters

A camera is described by several parameters

- Translation $T$ of the optical center from the origin of world coords
- Rotation $R$ of the image plane
- focal length $f$, principal point $(x'_c, y'_c)$, pixel size $(s_x, s_y)$
- blue parameters are called “extrinsics,” red are “intrinsics”

Projection equation

$$\mathbf{X} = \begin{bmatrix} s_x & * & * & * \\ s_y & * & * & * \\ s & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi} \mathbf{X}$$

- The projection matrix models the cumulative effect of all parameters
- Useful to decompose into a series of operations

$$\mathbf{\Pi} = \begin{bmatrix} -f s_x & 0 & x'_c & 1 \\ 0 & -f s_y & y'_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{T} \end{bmatrix}$$

- The definitions of these parameters are not completely standardized
  - especially intrinsics—varies from one book to another
Calibration from vanishing points
Calibration from 3 vanishing points

3 orthogonal world space axes

\[ e_i = [1, 0, 0, 0]^T, e_j = [0, 1, 0, 0]^T, e_k = [0, 0, 1, 0]^T \]

Vanishing points and relation to axes

\[ v_i = KR e_i, v_j = KR e_j, v_k = KR e_k \]

\[ e_i^T e_j = 0 \]

\[ v_i^T K^{-T} RR^T K^{-1} v_j = v_i^T K^{-T} K^{-1} v_j = 0 \]
\[
K = \begin{bmatrix}
 f & 0 & u_0 \\
0 & f & v_0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
K^{-1} = \begin{bmatrix}
 1/f & 0 & -u_0/f \\
0 & 1/f & -v_0/f \\
0 & 0 & 1
\end{bmatrix}
\]

\[
v_i^T K^{-T} K^{-1} v_j = 0
\]

\[
v_j^T K^{-T} K^{-1} v_k = 0
\]

\[
v_i^T K^{-T} K^{-1} v_k = 0
\]

- 3 finite vanishing points: get f, u0, v0
Alternative: multi-plane calibration

Images courtesy Jean-Yves Bouguet, Intel Corp.

Advantage

• Only requires a plane
• Don’t have to know positions/orientations
• Good code available online! (including in OpenCV)
  – Matlab version by Jean-Yves Bouget:
    http://www.vision.caltech.edu/bouguetj/calib_doc/index.html
  – Zhengyou Zhang’s web site: http://research.microsoft.com/~zhang/Calib/
Stereo

Single image stereogram, by Niklas Een
Announcements

• Readings
  – Szeliski, Chapter 7.1 and 7.2
  – Fundamental matrix song
Next two lectures

• Intro to stereo vision
• Epipolar geometry
• Fundamental matrix
Recovering 3D from images

Real 3D world

Point of observation

2D image
Projective Geometry

• Recovery of structure from one image is inherently ambiguous
Visual Cues for 3D

- Shading
- Texture
- Shadows
- Focus
- Motion

- Shape from X: X is shading, texture, motion, ....
Stereo Reconstruction

- Shape from 2 (or more) images
- Motivated by human visual system

Human performance: up to 6-8 feet
Stereo head

Camera on a mobile vehicle
Stereograms

• Invented by Sir Charles Wheatstone, 1838
Stereo

- Given two images from different viewpoints
  - How can we compute the depth of each point in the image?
  - Based on *how much each pixel moves* between the two images
This is not the web page you are looking for.

- https://github.com/404
Stereo

Two images captured by a purely horizontal translating camera
(rectified stereo pair)
Epipolar geometry

Two images captured by a purely horizontal translating camera (rectified stereo pair)

\[ x_2 - x_1 = \text{the disparity of pixel } (x_1, y_1) \]

Disparity is inversely proportional to depth
Disparity and Depth
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
    • Compute disparity
Stereo matching algorithms

• Match Pixels in conjugate epipolar Lines
  – Assume brightness constancy
  – This is a tough problem
  – Numerous approaches
    • A good survey and evaluation: http://www.middlebury.edu/stereo/
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 \textit{windows}
Effect of window size

- Smaller window
  - more detail
  - more noise
- Larger window
  - less noise
  - less detail

Better results with *adaptive window*

- T. Kanade and M. Okutomi,
- D. Scharstein and R. Szeliski.
Stereo results

- Data from University of Tsukuba
- Similar results on other images without ground truth
Results with window search

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

State of the art method
Boykov et al., Fast Approximate Energy Minimization via Graph Cuts,
International Conference on Computer Vision, September 1999.

Ground truth

For the latest and greatest: http://www.middlebury.edu/stereo/
Back to stereo

• Where do epipolar lines come from?
• Have a good break!