Image Formation and Cameras

Introduction to Computer Vision
CSE 152
Lecture 4

Announcements

- Read Trucco & Verri: pp. 15-40
- Irfanview: http://www.irfanview.com/ is a good Windows utility for manipulating images. Try xv for linux.
- Assignment 0: due today

Pinhole Camera: Perspective projection

- Abstract camera model - box with a small hole in it

Geometric Aspects of Perspective Projection

- Points project to points
- Lines project to lines
- Angles & distances (or ratios) are NOT preserved under perspective
- Vanishing point

The equation of projection

Cartesian coordinates:
- We have, by similar triangles, that (x, y, z) -> (f x/z, f y/z, -f)
- Ignore the third coordinate, and get

Euclidean -> Homogenous-> Euclidean

In 2-D
- Euclidean -> Homogenous: (x, y) -> λ (x,y,1) (can just take λ =1)
- Homogenous -> Euclidean: (x, y, z) -> (x/z, y/z)

In 3-D
- Euclidean -> Homogenous: (x, y, z) -> λ(x,y,z,1) (can just take λ =1)
- Homogenous -> Euclidean: (x, y, z, w) -> (x/w, y/w, z/w)
The camera matrix

Turn \( (x, y, z) \rightarrow (f \frac{x}{z}, f \frac{y}{z}) \)
into homogenous coordinates

- HC’s for 3D point are \((X, Y, Z, 1)\)
- HC’s for point in image are \((U, V, W)\)

Affine Camera Model

- Take Perspective projection equation, and perform Taylor Series Expansion about some point \((x_0, y_0, z_0)\).
- Drop terms of higher order than linear.
- Resulting expression is called affine camera model.

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = \frac{1}{z_0} \begin{bmatrix}
  x_0 & 0 & \frac{-x_0 z_0}{z} \\
  y_0 & 0 & \frac{-y_0 z_0}{z} \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x' \\
  y' \\
  z'
\end{bmatrix} = Ap + b
\]

- Properties
  - Points map to points
  - Lines map to lines
  - Parallel lines map to parallel lines (no vanishing point – at infinity)
  - Ratios of distance/angles preserved

Orthographic projection

Start with affine camera model, and take Taylor series about \((x_0, y_0, z_0) = (0, 0, z_0)\) – a point on optical axis

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  f/z_0 & 0 & 0 \\
  0 & f/z_0 & 0
\end{bmatrix} \begin{bmatrix}
  X \\
  Y \\
  Z
\end{bmatrix}
\]

Depth \((z)\) is lost

What if camera coordinate system differs from object coordinate system

Euclidean Coordinate Systems

Coordinate Changes: Pure Translations
No rotation (e.g., \(i_k = i_0\) etc)

\[
\overrightarrow{O_B P} = \overrightarrow{O_B O_A} + \overrightarrow{O_A P} \quad \overrightarrow{B P} = \overrightarrow{A P} + \overrightarrow{A O_A}
\]
A convenient notation

\[ \mathbf{b} P = \mathbf{a} R \mathbf{a} P + \mathbf{b} O \]

- Points: \( \mathbf{a} P \)
  - Leading superscript indicates the coordinate system that the coordinates are with respect to
  - Subscript – an identifier
- Rotation Matrices \( \mathbf{b} R \)
  - Lower left (Going from this system)
  - Upper left (Going to this system)
- To add vectors, coordinate systems (leading superscript) must agree
- To rotate a vector, points coordinate system must agree with lower left of rotation matrix

\[ \mathbf{B} P = \mathbf{A} R \]

Rotation Matrix

\[ \mathbf{b} R = [\mathbf{i}_b \cdot \mathbf{i}_a \cdot \mathbf{j}_b \cdot \mathbf{j}_a \cdot \mathbf{k}_b \cdot \mathbf{k}_a] = [\mathbf{i}_a \cdot \mathbf{i}_b \cdot \mathbf{j}_a \cdot \mathbf{j}_b \cdot \mathbf{k}_a \cdot \mathbf{k}_b] \]

Coordinate Changes: Rigid Transformations
Rotation + Translation

\[ \mathbf{B} P = \mathbf{A} R \mathbf{a} P + \mathbf{b} O \]

More about rotations matrices

A rotation matrix \( \mathbf{R} \) has the following properties:

- Its inverse is equal to its transpose \( \mathbf{R}^{-1} = \mathbf{R}^T \) or \( \mathbf{R}^T \mathbf{R} = \mathbf{I} \)
- Its determinant is equal to 1: \( \det(\mathbf{R}) = 1 \)

Or equivalently:

- Rows (or columns) of \( \mathbf{R} \) form a right-handed orthonormal coordinate system.

Even though a rotation matrix is 3x3 with nine numbers, it only has three degrees of freedom can be parameterized with three numbers. There are many parameterization.

Rotation: Homogenous Coordinates

- About z axis

\[ \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]

Note: z coordinate doesn’t change after rotation
Rotation

- About x axis:
  \[
  \begin{pmatrix}
  x' \\
  y' \\
  z'
  \end{pmatrix} =
  \begin{pmatrix}
  1 & 0 & 0 \\
  0 & \cos \theta & -\sin \theta \\
  0 & \sin \theta & \cos \theta
  \end{pmatrix}
  \begin{pmatrix}
  x \\
  y \\
  z
  \end{pmatrix}
  \]

- About y axis:
  \[
  \begin{pmatrix}
  x' \\
  y' \\
  z'
  \end{pmatrix} =
  \begin{pmatrix}
  \cos \theta & 0 & \sin \theta \\
  0 & 1 & 0 \\
  -\sin \theta & 0 & \cos \theta
  \end{pmatrix}
  \begin{pmatrix}
  x \\
  y \\
  z
  \end{pmatrix}
  \]

where \( c = \cos \theta \) & \( s = \sin \theta \)

Roll-Pitch-Yaw

\[
R = \text{rot}(\hat{\imath}, \alpha)\text{rot}(\hat{j}, \beta)\text{rot}(\hat{k}, \varphi)
\]

Euler Angles

\[
R = \text{rot}(\hat{k}', \alpha)\text{rot}(\hat{j}', \beta)\text{rot}(\hat{k}, \varphi)
\]

Camera parameters

- Issue
  - camera may not be at the origin, looking down the z-axis
  - extrinsic parameters (Rigid Transformation)
    - one unit in camera coordinates may not be the same as one unit in world coordinates
    - intrinsic parameters - focal length, principal point, aspect ratio, angle between axes, etc.

Camera Calibration

Given \( n \) points \( P_0, \ldots, P_n \) with known positions and their images \( p_0, \ldots, p_n \), estimate intrinsic and extrinsic camera parameters

- See Text book for how to do it.
- Camera Calibration Toolbox for Matlab (Bouguet)
  - http://www.vision.caltech.edu/bouguetj/calib_doc/
What about light?

Getting more light – Bigger Aperture

Limits for pinhole cameras

Pinhole Camera Images with Variable Aperture

The reason for lenses

Thin Lens

- Rotationally symmetric about optical axis.
- Spherical interfaces.
Thin Lens: Center

• All rays that enter lens along line pointing at O emerge in same direction.

Thin Lens: Focus

Incoming light rays parallel to the optical axis pass through the focus, F

Thin Lens: Image of Point

All rays passing through lens and starting at P converge upon P'

Thin Lens: Image Plane

A price: Whereas the image of P is in focus, the image of Q isn’t.

Thin Lens: Aperture

• Smaller Aperture -> Less Blur
• Pinhole -> No Blur
Light Field Camera

Lytro.com
Post-acquisition refocussing