Announcements

- Read Trucco & Verri: pp. 22-40
- 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

Equation of Perspective Projection

Cartesian coordinates:
- We have, by similar triangles, that $(x, y, z) \rightarrow (f'x/z, f'y/z, f')$
- Establishing an image plane coordinate system at $C'$ aligned with $i$ and $j$, we get $(x, y, z) \rightarrow (x/z, y/z, z)$

Euclidean -> Homogenous-> Euclidean

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

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

Turn

\[(x, y, z) \rightarrow (x/z, y/z, 1)\]

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.

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

- 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

\[
\begin{bmatrix}
  u \\
  v \\
  w
\end{bmatrix} =
\begin{bmatrix}
  f/z_0 & 0 & 0 & 0 \\
  0 & f/z_0 & 0 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  X \\
  Y \\
  Z
\end{bmatrix}
\]

Three projection models

- Perspective – all depths
- Affine – points near point of Taylor series expansion. E.g., when depth and size variation is small, and camera is far from point relative to size.
- Orthographic – when object is near optical axis, and depth variation of object is small compared to distance to the object.

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

Depth (z) is lost

```
http://www.youtube.com/watch?v=Yd9gaOU6zFY
```

What if camera coordinate system differs from object coordinate system

```
http://www.youtube.com/watch?v=Yd9gaOU6zFY
```

- Trombone
  - http://www.youtube.com/watch?v=Yd9gaOU6zFY
  - Trombone Stabalized
  - http://www.youtube.com/watch?v=Yd9gaOU6zFY
Euclidean Coordinate Systems

\[ \begin{align*}
  x &= \overline{OP}i \\
  y &= \overline{OP}j \\
  z &= \overline{OP}k
\end{align*} \]
\[ \Rightarrow \overline{OP} = xi + yj + zk \Rightarrow P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]

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

\[ \overline{O_B}P = \overline{O_B}O_A + \overline{O_A}P, \quad \overline{O_B}P = \overline{A}P + \overline{B}O_A \]

A convenient notation
\[ \overline{O_B}P = \overline{A}R \overline{A}P + \overline{BO_A} \]

- Points: \( \overline{A}P \)
  - Leading superscript indicates the coordinate system that the coordinates are with respect to
  - Subscript – an identifier
- Rotation Matrices \( \overline{A}R \)
  - 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

Rotation Matrix
\[ \overline{A}R = \begin{bmatrix} \overline{i_A} \cdot \overline{j_A} & \overline{j_A} \cdot \overline{k_A} & \overline{k_A} \cdot \overline{i_A} \\
 \overline{i_A} \cdot \overline{k_A} & \overline{j_A} \cdot \overline{i_A} & \overline{k_A} \cdot \overline{j_A} \\
 \overline{i_A} \cdot \overline{j_A} & \overline{j_A} \cdot \overline{k_A} & \overline{k_A} \cdot \overline{i_A} \end{bmatrix} = \begin{bmatrix} \overline{i_A} \cdot \overline{i_A} & \overline{i_A} \cdot \overline{j_A} & \overline{i_A} \cdot \overline{k_A} \\
 \overline{j_A} \cdot \overline{i_A} & \overline{j_A} \cdot \overline{j_A} & \overline{j_A} \cdot \overline{k_A} \\
 \overline{k_A} \cdot \overline{i_A} & \overline{k_A} \cdot \overline{j_A} & \overline{k_A} \cdot \overline{k_A} \end{bmatrix} \]

Coordinate Changes: Rigid Transformations
Rotation + Translation

\[ \overline{B}P = \overline{A}R \overline{A}P + \overline{B}O_A \]
More about rotations matrices

A rotation matrix $R$ has the following properties:

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

Or equivalently:
- Rows (or columns) of $R$ 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

- About $z$ axis:
  \[
  \begin{pmatrix}
  x' \\
  y' \\
  z'
  \end{pmatrix} = \begin{pmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
  \end{pmatrix} \begin{pmatrix}
  x \\
  y \\
  z
  \end{pmatrix}
  \]
  Note: $z$ coordinate doesn’t change after 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:
  \[
  \begin{pmatrix}
  x' \\
  y' \\
  z'
  \end{pmatrix} = \begin{pmatrix}
  \cos \theta & 0 & \sin \theta \\
  0 & 1 & 0 \\
  -\sin \theta & 0 & \cos \theta
  \end{pmatrix} \begin{pmatrix}
  x \\
  y \\
  z
  \end{pmatrix}
  \]

Roll-Pitch-Yaw

Euler Angles

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

Homogeneous Representation of Rigid Transformations

\[
\begin{pmatrix}
^{t}p' \\
1
\end{pmatrix} = \begin{pmatrix}
^{t}R & ^{t}pO \end{pmatrix} = \begin{pmatrix}
^{t}R \\
1
\end{pmatrix} \begin{pmatrix}
^{t}p \\
1
\end{pmatrix} = \begin{pmatrix}
^{t}R \\
1
\end{pmatrix} \begin{pmatrix}
^{t}p \\
1
\end{pmatrix}
\]

Transformation represented by 4 by 4 Matrix

Block Matrix Multiplication

Given
\[
A = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix} \quad B = \begin{bmatrix}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{bmatrix}
\]

What is $AB$?

\[
AB = \begin{bmatrix}
A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\
A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22}
\end{bmatrix}
\]
What if camera coordinate system differs from object coordinate system

\[
\begin{bmatrix}
U \\
V \\
W
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
R \\
O
\end{bmatrix}
\]

\[
{T} = \begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
\]

Intrinsic parameters

- 3x3 homogenous matrix
- Focal length:
- Principal Point: C'
- Units (e.g. pixels)
- Orientation and position of image coordinate system
- Pixel Aspect ratio

Camera parameters

- Extrinsic Parameters: Since camera may not be at the origin, there is a rigid transformation between the world coordinates and the camera coordinates
- Intrinsic parameters: Since scene units (e.g., cm) differ image units (e.g., pixels) and coordinate system may not be centered in image, we capture that with a 3x3 transformation comprised of focal length, principal point, pixel aspect ratio, angle between axes, etc.

\[
\begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix} =
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix}
\begin{pmatrix}
0 & 0 & 1
\end{pmatrix}
\]

Camera Calibration

- Given \( n \) points \( P_1, \ldots, P_n \) with known positions and their images \( p_1, \ldots, p_n \), 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

Thin Lens: Center

Thin Lens: Focus

The reason for lenses

Thin Lens

Thin Lens: Center

Thin Lens: Focus
Thin Lens: Image of Point

All rays passing through lens and starting at \( P \) converge upon \( P' \).

Thin Lens: Image of Point

\[
\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}
\]

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