Geometric Image Formation

Introduction to Computer Vision
CSE 152
Lecture 2

Announcements

- HW0 is assigned, due next Thursday
- Read Szeliski, Chapter 1
- Read Szeliski, Pages 27-52
- Lecture notes on web page

- Let’s look at course web page
Image Formation: Outline

- Geometric Image Formation (this lecture)
  - Factors in producing images
  - Projection
  - Vanishing points
  - Homogeneous coordinates and projective geometry
  - Rigid Transformation and SO(3)
  - Intrinsic & Extrinsic parameters
- Photometric Image Formation (Lecture 10)
  - Lenses
  - Sensors
  - Quantization/Resolution
  - Illumination
  - Reflectance and Radiometry
  - Shadows

Earliest Surviving Photograph

- First photograph on record, “la table service” by Nicephore Niepce, 1822
- Note: Niepce first photograph was taken in 1816
How Digital Cameras Produce Images

• Basic process:
  – photons hit a detector
  – the detector becomes charged
  – the charge is read out as brightness

• Sensor types:
  – CCD (charge-coupled device)
    • high sensitivity
    • high power
    • cannot be individually addressed
    • blooming
  – CMOS
    • simple to fabricate (cheap)
    • lower sensitivity, lower power
    • can be individually addressed

Photos are two-dimensional patterns of brightness values

They are formed by the projection of 3D objects
Lighting Affects Appearance: Monet

Haystack at Chailly at Sunrise (1865)

Viewpoint Affects Appearance: Monet
Weather Affects Appearance: Monet

Pinhole Camera: Perspective projection

• Abstract camera model - box with a small hole in it
Camera Obscura

"When images of illuminated objects ... penetrate through a small hole into a very dark room ... you will see [on the opposite wall] these objects in their proper form and color, reduced in size ... in a reversed position, owing to the intersection of the rays". --- Leonardo Da Vinci

http://www.acmi.net.au/AIC/CAMERA_OBSCURA.html (Russell Naughton)

Camera Obscura

Used to observe eclipses
(e.g., Bacon, 1214-1294)

And by Artists: Vermeer, ‘The Music Lesson’, 1662
Camera Obscura

Jetty at Margate England, 1898

http://brightbytes.com/cosite/collection2.html (Jack and Beverly Wilgus)

Purely Geometric View of Perspective

The projection of the point $P$ onto the image plane $\Pi'$ is given by the point of intersection $P'$ of the ray defined by $PO$ with the plane $\Pi'$.
Establish a 3D Camera Coordinate System

- Origin at pinhole
- $k$: vector toward closest point on image plane $C'$
- Sometimes called optical axis
- $i, j$: vectors span the image plane, orthogonal to $k$
- $(x, y, z)$: coordinates of $P$ in camera coordinate system

Let $f' > 0$ be the distance between $O$ and $\Pi'$
By similar triangles, the intersection of $OP$ with $\Pi'$ is $(f' \frac{x}{z}, f' \frac{y}{z}, f')$
Establish an image plane coordinate system at $C'$ aligned with $i$ and $j$
Image coordinates of $P'$ are $(f' \frac{x}{z}, f' \frac{y}{z})$
Often Draw Camera with Virtual Image Plane

- Virtual image plane in front of optical center.
- Image is ‘upright’

\[(x, y, z) \rightarrow (-f \frac{x}{z}, -f \frac{y}{z})\]
A Digression

Projective Geometry

and

Homogenous Coordinates

What is the intersection of two lines in a plane?

A Point
Do two lines in the plane always intersect at a point?

No, Parallel lines don’t meet at a point.

---

Can the perspective projection of parallel lines in 3D meet at a point in an image?

YES
Projective geometry provides an elegant means for handling these different situations in a unified way.

Homogeneous coordinates are a way to represent entities (points & lines) in projective spaces.

Homogeneous Coordinates in 2-D

- Normally, we write the coordinates of a point in a plane with 2 numbers \((x, y)\) and are called Euclidean coordinates.
- In homogeneous coordinates, we use 3 numbers: \((x, y, w)\)

\[
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} \Rightarrow
\begin{bmatrix}
  x/w \\
  y/w
\end{bmatrix}
\]

Euclidean to Homogeneous

Homogeneous to Euclidean

- But if we multiply homogeneous coordinates \([x, y, w]\) by some nonzero number \(\lambda\), the Euclidean coordinates are the same:

\[
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix} \lambda
\begin{bmatrix}
  \lambda x \\
  \lambda y \\
  \lambda w
\end{bmatrix} \Rightarrow
\begin{bmatrix}
  \lambda x/\lambda w \\
  \lambda y/\lambda w
\end{bmatrix} =
\begin{bmatrix}
  x/w \\
  y/w
\end{bmatrix}
\]

CSE152, Winter 2020
Homogeneous Coordinates in 2-D

More generally, conversion is given by:

\[
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \Rightarrow \lambda \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}, \quad \lambda \neq 0
\]

Euclidean to Homogeneous

\[
\begin{bmatrix}
x \\
y \\
w
\end{bmatrix} \Rightarrow \begin{bmatrix}
x / w \\
y / w
\end{bmatrix}
\]

Homogeneous to Euclidean

Homogeneous coordinates are only unique up to non-zero scale factor \( \lambda \):

\[
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}
\lambda x \\
\lambda y \\
\lambda
\end{bmatrix}
\]

are homogeneous coordinates of the same point

Point at Infinity

\[
\begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \Rightarrow \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

Euclidean to Homogeneous

\[
\begin{bmatrix}
x \\
y \\
w
\end{bmatrix} \Rightarrow \begin{bmatrix}
x / w \\
y / w
\end{bmatrix}
\]

Homogeneous to Euclidean

When isn’t homogeneous to Euclidean conversion possible?

When \( w = 0 \)

A point with homogenous coordinates \((x, y, 0)\) is called a

point at infinity.
Homogeneous Coordinates in 2-D & 3-D

- 2-D

\[
\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad \lambda \neq 0
\]

Euclidean to Homogeneous

\[
\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \frac{x}{w} \begin{bmatrix} 1 \\ y/w \end{bmatrix}
\]

Homogeneous to Euclidean

- 3-D

\[
\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \lambda \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad \lambda \neq 0
\]

Euclidean to Homogeneous

\[
\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow \frac{x}{w} \begin{bmatrix} 1 \\ y/w \\ z/w \end{bmatrix}
\]

Homogeneous to Euclidean

Homogenous coordinates

A way to represent points in a projective space

Use three numbers to represent a point on a projective plane

Add an extra coordinate
e.g., (x,y) -> (x,y,1)

Impose equivalence relation
(x,y,w) = \lambda*(x,y,w)
such that (\lambda not 0)
i.e., (x,y,1) \sim (\lambda x, \lambda y, \lambda)
Points at infinity

Point at infinity – last coordinate is zero (x,y,0) and equivalence relation
(x,y,0) = \lambda\cdot(x,y,0)

No corresponding Euclidean point (you’d divide by zero).

Perspective projection in Homogeneous Coordinates

Cartesian coordinates:
(x,y,z) \rightarrow (f\frac{x}{z}, f\frac{y}{z})

Homogenous coordinates and camera matrix
\begin{pmatrix}
U \\
V \\
W
\end{pmatrix} = \begin{pmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix} \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
Something to try at home

- Confirm that given a point \((x,y,z)\), after converting to homogenous coordinates, multiplying by camera matrix, and converting result back to Euclidean coordinates, the result is \((f \frac{x}{z}, f \frac{y}{z})\)

End of the Digression
In a perspective image, lines that are parallel in 3D meet at a point, called the vanishing point.

Doesn’t need to be near the center of the image.

Parallel lines meet in the image

- A single line $L$ can have a vanishing point.
- Vanishing point location: Intersection of image plane with a 3-D line $L^*$ through optical center $O$ parallel to $L$.
Vanishing points

- A scene can have more than one vanishing point
- Different 3-D directions correspond different vanishing points

Vanishing Points
Vanishing Point and Homogeneous Coordinates

- When lines are parallel Euclidean 3-D, they meet at a point at infinity \((x, y, z, 0)\) in homogeneous coordinates.

- The vanishing point is the perspective projection of that point at infinity \((x, y, z, 0)\) and is computed by multiplying by the camera matrix.

What if camera coordinate system differs from an object’s coordinate system?

If we know the coordinates of \(P\) in the bird frame \(\{b\}\), what are the coordinates of \(P\) in the camera frame \(\{c\}\)?
What if camera coordinate system differs from an object’s coordinate system?

Let $^bP$ denote coordinates of P in the bird frame $\{b\}$
Let $^cP$ denote coordinates of P in the camera frame $\{c\}$

$^cP = ^bR^bP + ^cO_b$

- Where $^cO_b$ is the origin of the bird frame in the world frame.
- $^bR$ is the rotation between the frames.
A convenient notation

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

- Points: e.g., \( \mathbf{A} \mathbf{P}, \mathbf{A} \mathbf{P}_1 \)
  - Leading superscript indicates the coordinate system that the coordinates are with respect to
  - Subscript – an identifier
- To add or subtract vectors, coordinate systems must agree
- Rotation Matrices \( \mathbf{b} \mathbf{R} \)
  - Lower left (Going from b system)
  - Upper left (Going to c system)
- To rotate a vector or point, the point’s coordinate system must agree with lower left of rotation matrix \( \mathbf{b} \mathbf{P} = \mathbf{b} \mathbf{R} \mathbf{a} \mathbf{P} \)

Properties of Rotation Matrices

- Rotation matrices are members of the Special Orthogonal Group of matrices called SO(n)
- \( \text{SO}(n) = \{ \mathbf{R} \in \mathbb{R}^{n \times n} : \mathbf{R}^T \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1 \} \)
  - \( \text{SO}(2) \): rotation matrices in plane
  - \( \text{SO}(3) \): rotation matrices in 3D
- Bounded \( R_{i,j} \in [-1, +1] \)
- Does not form a vector space – Don’t add rotation matrices!!
- Inverse \( \mathbf{R}^{-1} = \mathbf{R}^T \)
- Not commutative.
Parameterizing Rotation Matrices

- 3D Rotation matrices are 3x3 and have 9 numbers.
- They’re not all independent because of constraint $R^T R = I$
- 6 independent constraints -> 3 degrees of freedom
- Many ways to parameterize 3D rotation matrices using 3 numbers: Euler angles, roll-pitch-yaw, angle-axis, quaternions, etc.

Rotation in 2D

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

At home, confirm that $R^T R = I$ and $\det(R) = 1.$
3-D Rotation about the Z axis

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

3-D Rotation about x, y axis

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

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

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

Euler Angles

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

Angle-Axis Rotation

- About an arbitrary axis \((k_x, k_y, k_z)\) by angle \(\theta\) (Rodrigues Formula)

\[
\begin{bmatrix}
    x' \\
    y' \\
    z' \\
    1
\end{bmatrix} =
\begin{bmatrix}
    k_xk_z(1-c)+c & k_xk_y(1-c)-k_zs & k_xk_z(1-c)+k_zs & 0 \\
    k_yk_z(1-c)+k_zs & k_yk_y(1-c)+c & k_yk_z(1-c)-k_zs & 0 \\
    k_zk_z(1-c)-k_zs & k_zk_y(1-c)+k_zs & k_zk_z(1-c)+c & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z \\
    1
\end{bmatrix}
\]

where \(c = \cos \theta\) & \(s = \sin \theta\)
Rigid Transformation in Euclidean and Homogeneous Coordinates

- Euclidean Coordinates
  \[ {^B}_P = {^B}R {^A}_P + {^B}O_A \]

- Homogeneous coordinates
  \[
  \begin{bmatrix}
  {^B}_P \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  {^B}R {^A}_P + {^B}O_A \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  {^B}R & {^B}O_A \\
  0^T & 1
  \end{bmatrix}
  \begin{bmatrix}
  {^A}_P \\
  1
  \end{bmatrix}
  =
  \begin{bmatrix}
  {^A}_P \\
  1
  \end{bmatrix}
  \]

So, what are the image coordinates of P?

1. Rigid Transformation
   \[ ^c_P = ^bT^bP \]
   where \(^cP\) and \(^bP\) are homogeneous

2. Perspective projective
   \[
   \begin{pmatrix}
   U \\
   V \\
   W
   \end{pmatrix}
   =
   \begin{pmatrix}
   f & 0 & 0 & 0 \\
   0 & f & 0 & 0 \\
   0 & 0 & 1 & 0
   \end{pmatrix}
   \begin{pmatrix}
   ^cX \\
   ^cY \\
   ^cZ \\
   1
   \end{pmatrix}
   \]

3. Combine them
   \[ q = \Pi_p ^bT^bP \]

Note: \( q \) is in same units as \( P \), not pixels
From image plane to pixel coordinates

\[
\Pi_p = \begin{bmatrix}
    f & s & c_x \\
    0 & \alpha f & c_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\]

K is an upper triangular matrix parameterized by

- \( f \) : focal length in units of pixels/mm when world coordinates in mm
- \( \alpha \) : aspect ratio (1 unless pixels are not square)
- \( S \) : skew (0 unless pixels are shaped like rhombi/parallelograms)
- \((c_x, c_y)\) : principal point ((0,0) unless optical axis doesn’t intersect projection plane at origin)

Full Camera Matrix

\[
M = K \Pi_w^T = \begin{bmatrix}
    f & s & c_x \\
    0 & \alpha f & c_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    \mathbf{R} & \mathbf{O}_w \\
    \mathbf{0}^T & 1
\end{bmatrix}
\]

- What is the shape of M? \( 3 \times 4 \)
- Mapping from point in homogenous world coordinates \( wP \) to homogenous pixel coordinates \( q \)

\[
q = M wP
\]

- Can then map \( q \) to Euclidean coordinates.
• Given $n$ points $P_1, \ldots, P_n$ with known 3-D position and their pixel coordinates $q_1, \ldots, q_n$, estimate intrinsic $K$ and extrinsic camera parameters $w^T$.

• See textbook for details.

• Camera Calibration Toolbox for Matlab (Bouguet)

http://wwwvision.caltechedu/bouguetj/calib_doc/