CS 395/495-26: Spring 2002

IBMR: Week 9A

The Camera Matrix and World Geometry
Chapters 5, 6, 7 (partial)

Jack Tumblin
jet@cs.northwestern.edu

Reminders

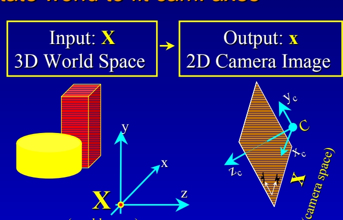
CTEC Online – please add your comments…

- Proj3 Due Thurs May 23
  HW2 posted on website.
- HW2 due Thurs May 30
  Proj4 posted on website.
  HW 3 Assign Thu May 30
- Proj4 Due Tues June 11
- HW3 Due Tues June 11

Camera Matrix P links $P^3 \rightarrow P^2$

- K matrix: "internal camera calib. matrix"
- R·T matrix: "external camera calib. matrix"
  - T matrix: Translate world to cam. origin
  - R matrix: 3D rotate world to fit cam. axes
  - 11 DOF total

Combine: write
$(P_0 \cdot R \cdot T) \cdot X = x$
as
$P \cdot X = x$



Uses for Camera Matrix P

$$P \cdot X = x, \text{ or } \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & P & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \qquad P = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & M & \cdot & p_4 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

- P's Null Space: camera center C in world space

  $P \cdot C = 0$ (solve for C, get) $C = \begin{bmatrix} -M^{-1} \cdot p_4 \\ 1 \end{bmatrix}$
- P's **Columns**: p_1, p_2, p_3, (and p_4)

  world's x_w, y_w, z_w axis vanishing pts.

  (and origin) in image space
- P's **Rows**: p^{1T}, p^{2T}, p^{3T}

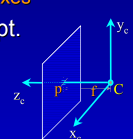
  camera planes in world-space…

Uses for Camera Matrix P

$$P \cdot X = x, \text{ or } \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & P & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \qquad P = \begin{bmatrix} P^1 & P^2 & P^3 & P^4 \end{bmatrix}$$

Columns of P matrix: Points in image-space:

- P^1, P^2, P^3 == image of x,y,z axis vanishing points
  - Proof: let $D = [1\ 0\ 0\ 0]^T$ = point on x axis, at inifinity
  - PD = 1st column of P. Repeat for y and z axes
- P^4 == image of the world-space origin pt.
  - Proof: let **D** = [0 0 0 1]T = world origin
  - **PD** = 4th column of **P** = image of origin pt.



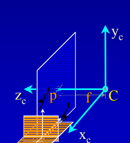
Uses for Camera Matrix P

$$P \cdot X = x, \text{ or } \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & P & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \qquad P = \begin{bmatrix} p^{1T} \\ p^{2T} \\ p^{3T} \end{bmatrix}$$

Rows of P matrix: planes in world space

- row 1 = $\mathbf{p}^1$ = image x-axis plane
- row 2 = $\mathbf{p}^2$ = image y-axis plane
- Careful! Shifting image origin

  shifts the x,y axis planes!



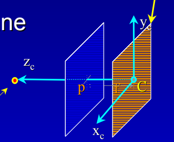
Uses for Camera Matrix P

$P \cdot X = x$, or $\begin{bmatrix} & P & \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ $\quad P = \begin{bmatrix} p^{1T} \\ p^{2T} \\ p^{3T} \end{bmatrix}$

Rows of P matrix: planes in world space

- row 1 = $\mathbf{p}^1$ = image x-axis plane
- row 2 = $\mathbf{p}^2$ = image y-axis plane
- row 3 = $\mathbf{p}^3$ = camera's principal plane
  - princip. plane $\mathbf{p}^3 = [p_{31}\ p_{32}\ p_{33}\ p_{34}]^T$
  - its normal vector: $[p_{31}\ p_{32}\ p_{33}\ 0\ ]^T$
  - Why is it normal? It's the world-space endpoint of z_c axis (at infinity)

Principal plane p^3



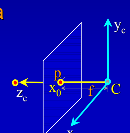
Uses for Camera Matrix P

$P \cdot X = x$, or $\begin{bmatrix} & P & \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ $\quad P = \left[\begin{array}{c|c} m^{1T} & \\ m^{2T} & P^4 \\ m^{3T} & \end{array}\right]$

M

- Principal Axis Vector in **world** space:
  - Normal of principal plane: = $m^3 = [p_{31}\ p_{32}\ p_{33}\ 0\ ]^T$
  - Scaling → Ambiguous direction!! +/- m^3 ?
  - Solution: use $\det(M) \cdot m^3$ as front of camera
- Principal Point p in **image** space:
  - image of (infinity point on z_c axis = m^3)

    $\mathbf{M} \cdot m^3 = p = \mathbf{x_0}$ (book renames p as x_0)



Uses for Camera Matrix P

$P \cdot X = x$, or $\begin{bmatrix} & P & \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ $\quad P = \left[\begin{array}{c|c} M & P^4 \end{array}\right]$

Given image point $\mathbf{x_0}$ and camera matrix $\mathbf{P}$,
Find ray $X(\mu)$ in world space through both:

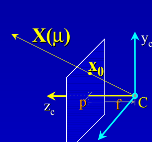
Slow, Obvious way: 'Pseudo-invert' P:

- Define pseudo-inverse P^+ as $= P^T(PP^T)^{-1}$ (note $P \cdot P^+ = I$)
- Find a world-space point on ray: $X_0 = P^+ x_0$
- LIRP with camera: $X(\mu) = C + (X_0 - C)\mu$

Better way:

- Find where ray from x hits infinity in world space:

$$X(\mu) = \mu \mathbf{M}^{-1}\mathbf{x} - \mathbf{C} = \begin{bmatrix} \mathbf{M}^{-1}(\mu\mathbf{x} - \mathbf{p}^4) \\ 1 \end{bmatrix}$$



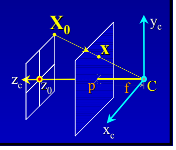
Uses for Camera Matrix P

$\mathbf{P \cdot X = x}$, or $\begin{bmatrix} & & & \\ & \mathbf{P} & & \\ & & & \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ t_w \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$ $\quad \mathbf{P} = \left[\; \mathbf{M} \; \middle| \; \mathbf{P}^4 \right]$

Given world-space point $\mathbf{X_0}$, camera matrix $\mathbf{P}$,
Find camera depth z_0:

- $\mathbf{X_0} = [x_w, y_w, z_w, t_w]^T$ seen thru camera is $\mathbf{X_0 \cdot P = x_0} = [x_c, y_c, 1]^T \cdot w_c$
- Then signed depth z_0 is:

$$z_0 = \frac{w_c \; \text{sign}(\det \mathbf{M})}{t_w \; \| m^3 \|}$$



Skipped:

- P = [K|0]·R·T How can we separate K, R, T?
  - Answer: K is triangular; use QR decomposition
- Cameras at Infinity:
  - Orthographic or 'Parallel Projection' Cameras
  - Transition to Orthographic:
    - Weak Perspective projection cameras
    - the 'zoom' lens (variable f)
  - Moving line-scan or 'pushbroom' cameras
    - Translation Scan: aerial/sattelite cameras
    - Cylindrical Scan: panoramic cameras
    - UNC 'HiBall Tracker': 6 tiny self-locating line-scan cameras

Chapter 6 In Just One Slide:

Given point correspondence sets ($x_i \leftrightarrow X_i$), How do you find camera matrix **P ?** (full 11 DOF)

Surprise! You already know how !

- DLT method:
  - -rewrite H x = x' as Hx × x' = 0
  - -rewrite P X = x as PX × x = 0
  - -vectorize, stack, solve Ah = 0 for h vector
  - -vectorize, stack, solve Ap = 0 for p vector
  - -Normalizing step removes origin dependence
- More data → better results (at least 28 point pairs)
- Algebraic & Geometric Error, Sampson Error…

Chapter 7: More One-Camera Fun

Full 3x4 camera matrix **P** maps $\mathbb{P}^3_{world}$ to $\mathbb{P}^2_{image}$

? What does it do to basic 3D world shapes?

- Plane
  - Given any point $\mathbf{X}_\pi$ on a plane in $\mathbb{P}^3$,
  - Change world's coord. system: let plane be z=0:
  - Matrix **P** reduces to 3x3 matrix **H** in $\mathbb{P}^2$:

$$x_\pi = P \cdot X_\pi = \begin{bmatrix} p_{11} & p_{12} & \not{p_{13}} & p_{14} \\ p_{21} & p_{12} & \not{p_{23}} & p_{24} \\ p_{31} & p_{32} & \not{p_{33}} & p_{34} \end{bmatrix} \begin{bmatrix} x_\pi \\ y_\pi \\ 0 \\ t_\pi \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_\pi \\ y_\pi \\ t_\pi \end{bmatrix}$$

- THUS

$\mathbb{P}^2$ can do all 3D plane transforms

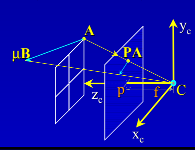
Chapter 7: More One-Camera Fun

Full 3x4 camera matrix **P** maps $\mathbb{P}^3_{world}$ to $\mathbb{P}^2_{image}$

? What does it do to basic 3D world shapes?

Forward Projection:

- Line / Ray in world → Line/Ray in image:
  - Ray in $\mathbb{P}^3$ is $\quad \mathbf{X}(\mu) = A + \mu B$
  - Camera changes to $\mathbb{P}^2$: $\quad x(\mu) = PA + \mu PB$



Chapter 7: More One-Camera Fun

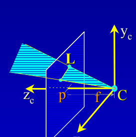
Full 3x4 camera matrix **P** maps $\mathbb{P}^3_{world}$ to $\mathbb{P}^2_{image}$

? What does it do to basic 3D world shapes?

Back Projection:

- Line L in image → Plane π_L in world:
  - Recall: Line L in $\mathbb{P}^2$ (a 3-vector): $L = [x_1 \; x_2 \; x_3]^T$
  - Plane π_L in $\mathbb{P}^3$ (a 4-vector):

$$\pi_L = P^T \cdot L = \begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \\ p_{14} & p_{24} & p_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$



- (SKIP Plücker Matrix lines…)

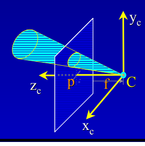
Chapter 7: More One-Camera Fun

Full 3x4 camera matrix **P** maps P^3_{world} to P^2_{image}
? What does it do to basic 3D world shapes?

- Conic C in image → Cone Quadric Q_{co} in world

$$Q_{co} = P^T \cdot C \cdot P$$

- Tip of cone is camera center **C**



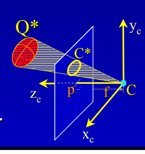
Chapter 7: More One-Camera Fun

Full 3x4 camera matrix **P** maps P^3_{world} to P^2_{image}
? What does it do to basic 3D world shapes?

- Dual Quadric Q* in world →
  Dual Conic C* silhouette in image

$$C^* = P^T \cdot Q^* \cdot P$$

- **Works for ANY quadric!**
  sphere, cylinder, ellipsoid, paraboloid, hyperboloid, line, disk …

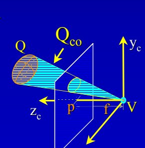


Chapter 7: More One-Camera Fun

Full 3x4 camera matrix **P** maps P^3_{world} to P^2_{image}
? What does it do to basic 3D world shapes?

- World –space cone from camera center V to quadric Q is the degenerate quadric Q_{co}:

$$Q_{co} = (V^T QV)Q - (QV)(QV)^T$$



Chapter 7: More One-Camera Fun

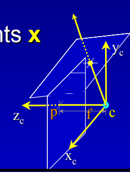
Full 3x4 camera matrix **P** maps P^3_{world} to P^2_{image}

? What does it do to basic 3D world shapes?

'Pure' Rotation:

- Given internal camera calibration **K** → $\begin{bmatrix} \alpha_x f & s & p_x & 0 \\ 0 & \alpha_y f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
- 3D rotate camera P about its center **C** using 3D rotation matrix **R** (3x3)
- Get new points **x'** from old image points **x**

  $K \cdot R \cdot K^{-1} x = x'$



- aka 'conjugate rotation'
- use this to construct planar panoramas

END