# CS231A PS1 Review

CS231A

Computer Vision: From 3D Reconstruction to Recognition

Spring 2017

# Problem Outline

- Q1: Projective Geometry
- Q2: Affine Camera Calibration
- Q3: Single View Geometry

# Problem Outline

- Q1: Projective Geometry
- Q2: Affine Camera Calibration
- Q3: Single View Geometry

# Cross Products

- If lines *k* and *l* are parallel, and $k_1$ and $k_2$ are any 2 points on *k*, and $l_1$ and $l_2$ are any 2 points on *l*, by definition of parallel lines:

$$\boxed{(k_1 - k_2) \times (l_1 - l_2) = 0}$$

- Given a square *pqrs,*
  - Area = $\boxed{\|(q - p) \times (s - p)\|}$

# Sample Problem

Prove that the angle between intersecting lines in the world reference system is the same as in the camera reference system.

# Problem Outline

- Q1: Projective Geometry
- Q2: Affine Camera Calibration
- Q3: Single View Geometry

# Least Squares

$Ax = b$

$x = (A^T A)^{-1} A^T b$

Useful: numpy.linalg.lstsq

# Problem Outline

- Q1: Projective Geometry
- Q2: Affine Camera Calibration
- Q3: Single View Geometry

# Vanishing Points

*Courtesy of last year's slides

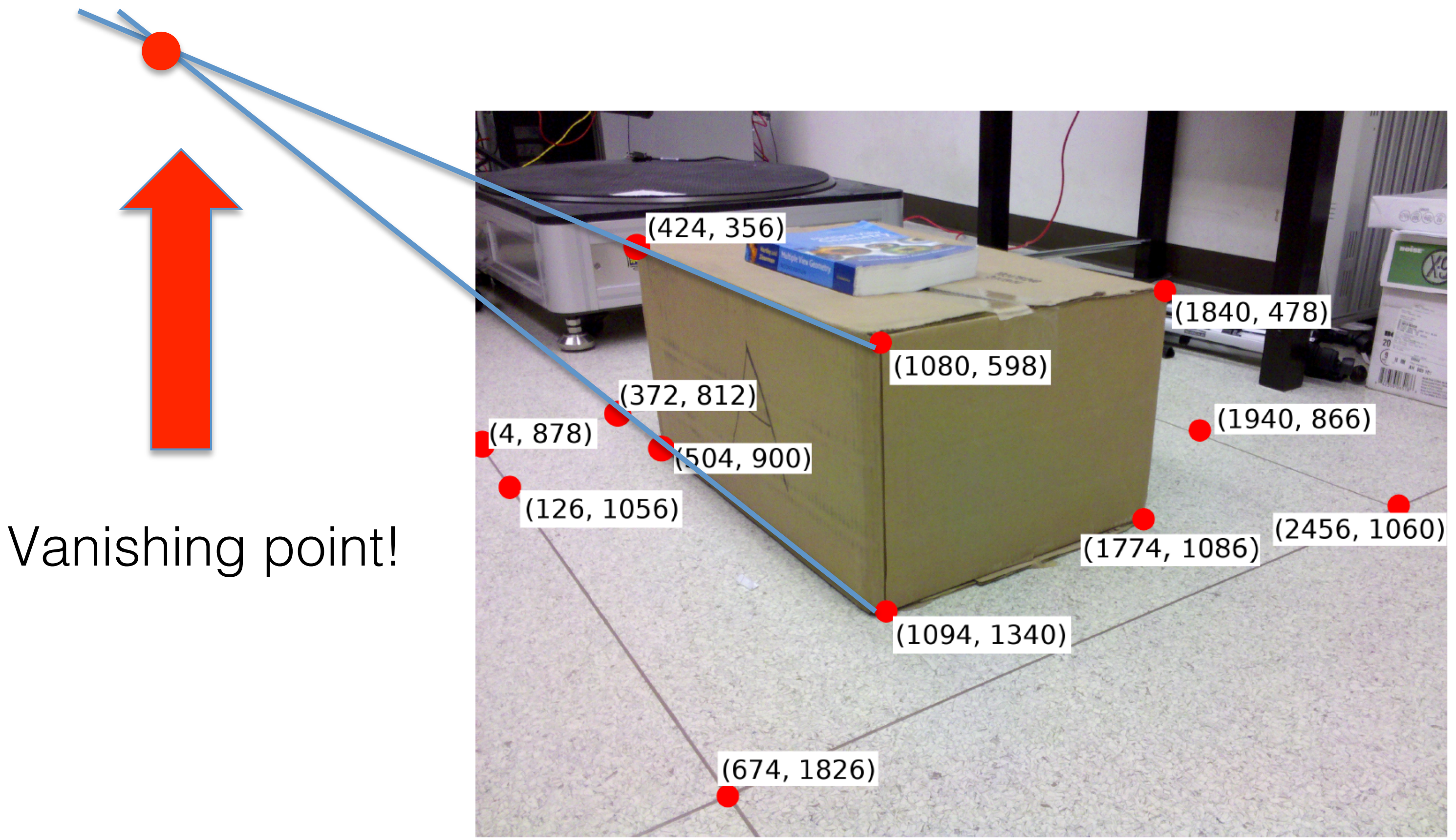
# Vanishing Points

- Under perspective projection, lines that are parallel in the world frame are no longer parallel in the image frame
  - Exception: Lines parallel to the image plane remain parallel

- In the image plane, parallel lines meet at the vanishing point

*Courtesy of last year's slides

# Calculating Vanishing Point




(a) Image 1 (1.jpg) with marked pixels

# Calculating Vanishing Point

- Points in $L_1$: $(x_1, y_1)$, $(x_2, y_2)$ → $m_1 = (y_2 - y_1)/(x_2 - x_1)$
- Points in $L_2$: $(x_3, y_3)$, $(x_4, y_4)$ → $m_2 = (y_4 - y_3)/(x_4 - x_3)$

- **Equation of a line**: $y = mx + b$ !
  - $b_1 = y_2 - m_1x_2$ ; $b_2 = y_4 - m_2x_4$
  - $L_1$: $y = m_1x + b_1$; $L_2$: $y = m_2x + b_2$

- Intersection of $L_1$ and $L_2$: $(x, y)$
  - $m_1x + b_1 = m_2x + b_2$
    - $x = (b_2 - b_1)/(m_1 - m_2)$
    - $y = m_1x + b_1 = m_1[(b_2 - b_1)/(m_1 - m_2)] + b_1$

# Vanishing Points to Compute K

- Lecture 4 slides are very useful!
- Only need 3 vanishing points
- $\omega = (K\,K^T)^{-1}$
  - Matrix $\omega$ is the projective transformation in the image plane of an absolute conic in 3D

$$\boxed{\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}}$$

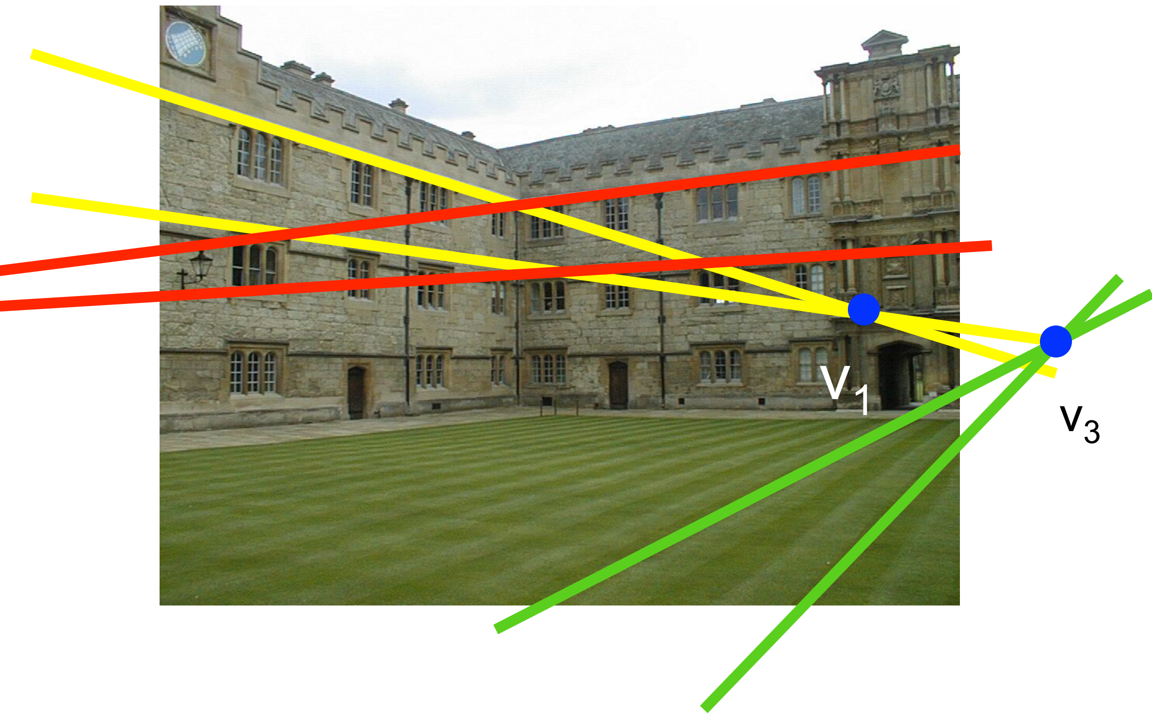
# Vanishing Points to Compute K

- We assume the camera has zero skew and square pixels
  - Zero skew: $\omega_2 = 0$
  - Square pixels: $\omega_1 = \omega_3$

$$\boxed{\omega = \begin{bmatrix} \omega_1 & 0 & \omega_4 \\ 0 & \omega_1 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}}$$

# Single view calibration - example

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_4 \\ 0 & \omega_1 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$$




- Square pixels
- No skew

$\rightarrow$ $\boldsymbol{\omega}_2 = 0$, $\boldsymbol{\omega}_1 = \boldsymbol{\omega}_3$

[Eqs. 31]

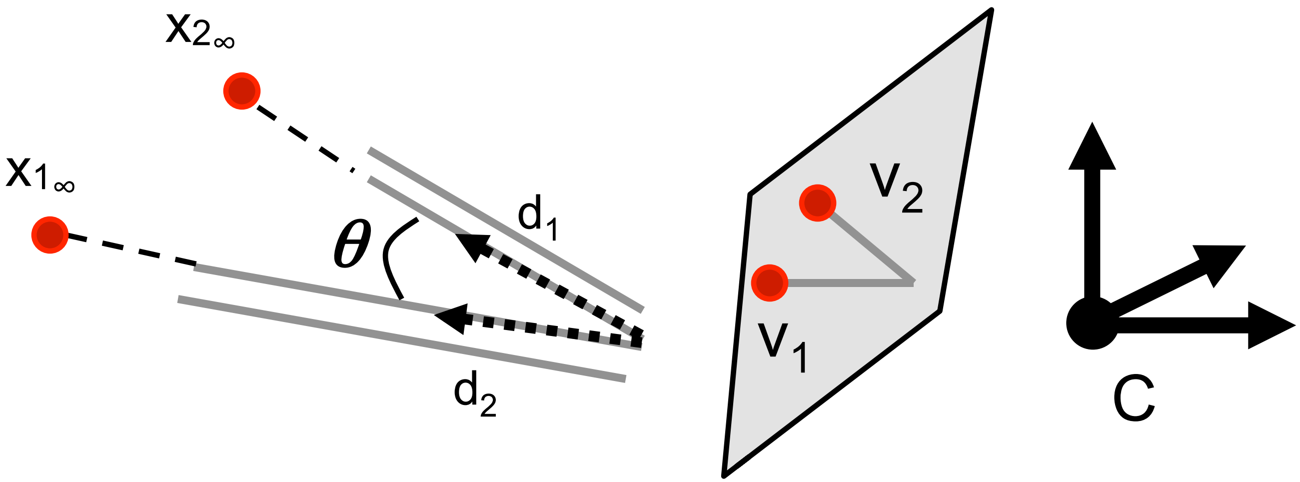
$$\begin{cases} \mathrm{v}_1^T \boldsymbol{\omega}\, \mathrm{v}_2 = 0 \\ \mathrm{v}_1^T \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \\ \mathrm{v}_2^T \boldsymbol{\omega}\, \mathrm{v}_3 = 0 \end{cases}$$

Once $\omega$ is calculated, we get K:

$$\omega = (K\,K^T)^{-1} \longrightarrow \mathrm{K}$$

(Cholesky factorization; HZ pag 582)

# Angle between 2 vanishing points




$$\cos\theta = \frac{v_1^T \omega \, v_2}{\sqrt{v_1^T \omega \, v_1}\sqrt{v_2^T \omega \, v_2}}$$

[Eq. 28]

$$\omega = (K\,K^T)^{-1}$$

If $\theta = 90 \quad \rightarrow \quad \boxed{v_1^T \omega \, v_2 = 0}$ [Eq. 29]

Scalar equation

# Compute Angle Between Planes

- Similar to the previous slide!
- Vanishing lines $L_1$ and $L_2$
- $L_1 = v_1 \times v_2$ ; $L_2 = v_3 \times v_4$
  - $v_1$ *and* $v_2$ = vanishing points corresponding to one plane
  - $v_3$ *and* $v_4$ for the other plane

$$\boxed{\cos\theta = \frac{\ell_1^T \omega^{-1} \ell_2}{\sqrt{\ell_1 \omega^{-1} \ell_1}\sqrt{\ell_2 \omega^{-1} \ell_2}}}$$

# Rotation Matrix using Vanishing Points

- Find corresponding vanishing points from both images ($v_1$, $v_2$, $v_3$) and ($v_1$', $v_2$', $v_3$')
- Calculate directions of vanishing points:
  - v = K d → $\boxed{\mathbf{d} = \frac{K^{-1}\,\mathbf{v}}{\left\|K^{-1}\,\mathbf{v}\right\|}}$ where d = direction of line
- $d_i$' = R $d_i$ , where
  - $d_i$' = direction of the $i^{th}$ vanishing point in second image
  - $d_i$ = direction of the $i^{th}$ vanishing point in first image