Vision Review: Image Formation

Course web page:

www.cis.udel.edu/~cer/arv

University of Delaware

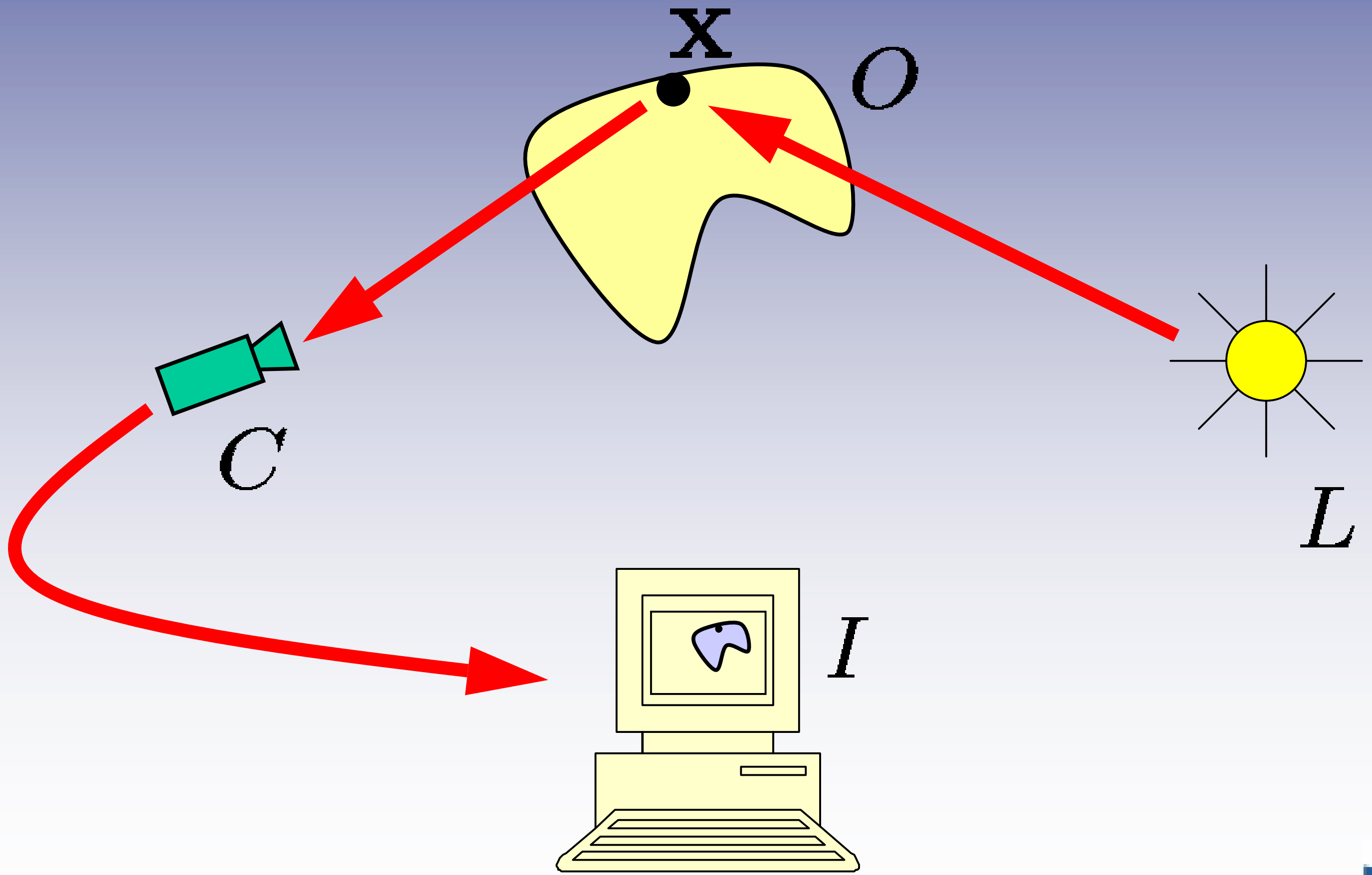
Announcements

- Lecture on Thursday will be about Matlab; next Tuesday will be "Image Processing"
- The dates some early papers will be presented are posted. Who's doing what is not, yet (except that I'm the first two).
- In particular, read "Video Mosaics" paper up to "Projective Depth Recovery" section
- Supporting readings: Chapters 1, 3 (through 3.3.2 "Hue, Saturation, and Value" subsection), 5 (through 5.3.2), and 7.4 of Forsyth & Ponce

Computer Vision Review Outline

- **Image formation**
- Image processing
- Motion & Estimation
- Classification

The Image Formation Pipeline

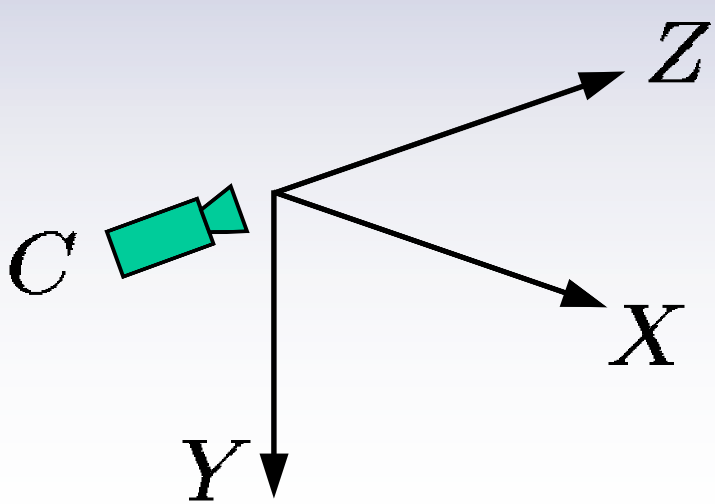


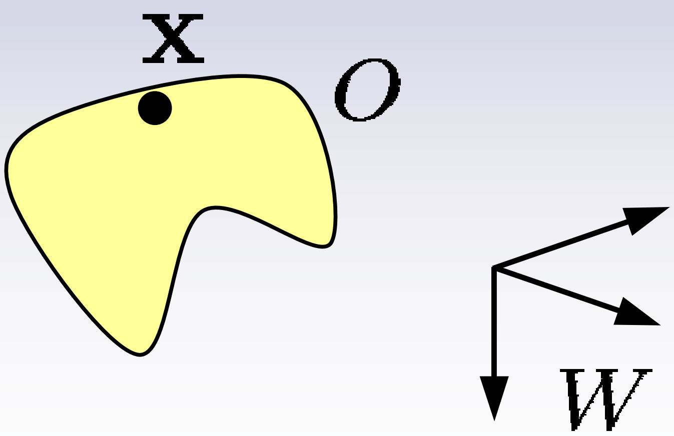
Outline: Image Formation

- Geometry
  - Coordinate systems, transformations
  - Perspective projection
  - Lenses
- Radiometry
  - Light emission, interaction with surfaces
- Analog $\rightarrow$ Digital
  - Spatial sampling
  - Dynamic range
  - Temporal integration

Coordinate System Conventions

- $\mathbf{i}, \mathbf{j}, \mathbf{k}$ unit vectors along positive X, Y, Z axes, respectively; $\mathbf{k} = \mathbf{i} \times \mathbf{j}$
- Right- vs. left-handed coordinates
- Local coordinate systems: camera, world, etc.





Homogeneous Coordinates (Projective Space)

- Let $\mathbf{x} = (x_1, \ldots, x_n)^T$ be a point in Euclidean space
- Change to homogeneous coordinates:

$$\mathbf{x} \rightarrow (\mathbf{x}^T, 1)^T$$

- Defined up to scale:

$$(\mathbf{x}^T, 1)^T \equiv (\lambda \mathbf{x}^T, \lambda)^T$$

- Can go back to non-homogeneous representation as follows:

$$(\mathbf{x}^T, \lambda)^T \rightarrow \mathbf{x}/\lambda$$

3-D Transformations: Translation

- Ordinarily, a translation between points is expressed as a vector addition $\mathbf{t}$
- Homogeneous coordinates allow it to be written as a matrix multiplication:

$$\mathbf{x}' = \begin{pmatrix} \mathbf{Id} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{x}$$

3-D Rotations: Euler Angles

- Can decompose rotation $\mathbf{R}$ of ρ about arbitrary 3-D axis into rotations $(\mathbf{Y}_\phi, \mathbf{Z}_\psi, \mathbf{X}_\theta)$ about the coordinate axes ("yaw-roll-pitch")
- $\mathbf{R} = \mathbf{X}_\theta \mathbf{Z}_\psi \mathbf{Y}_\phi$, where:

$$\mathbf{X}_\theta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix} \quad \mathbf{Z}_\psi = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Y}_\phi = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}$$

(Clockwise when looking toward the origin)

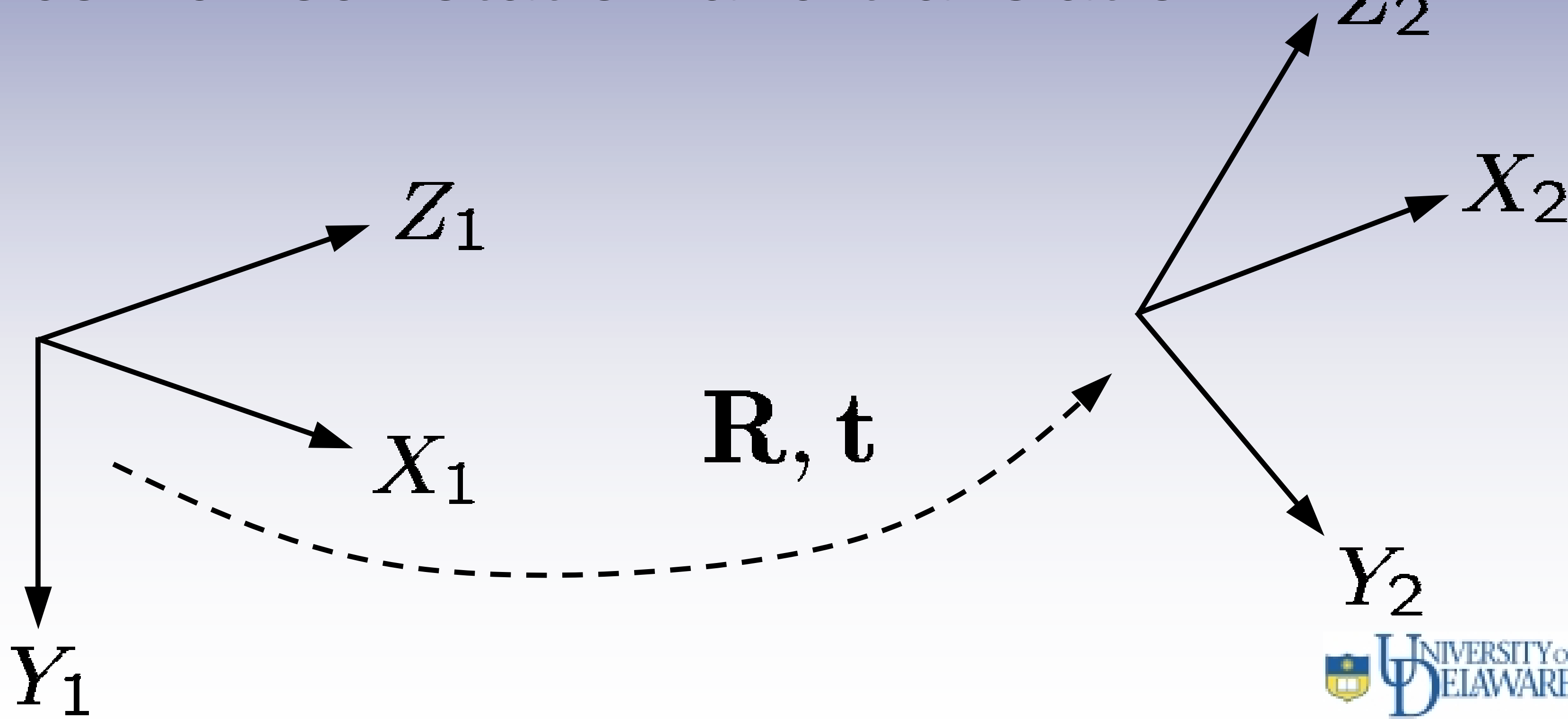
3-D Transformations: Rotation

- A rotation of a point about an arbitrary axis normally expressed as a multiplication by the rotation matrix $\mathbf{R}$ is written with homogeneous coordinates as follows:

$$\mathbf{x}' = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{1} \end{pmatrix} \mathbf{x}$$

3-D Transformations: Change of Coordinates

- Any rigid transformation can be written as a combined rotation and translation:



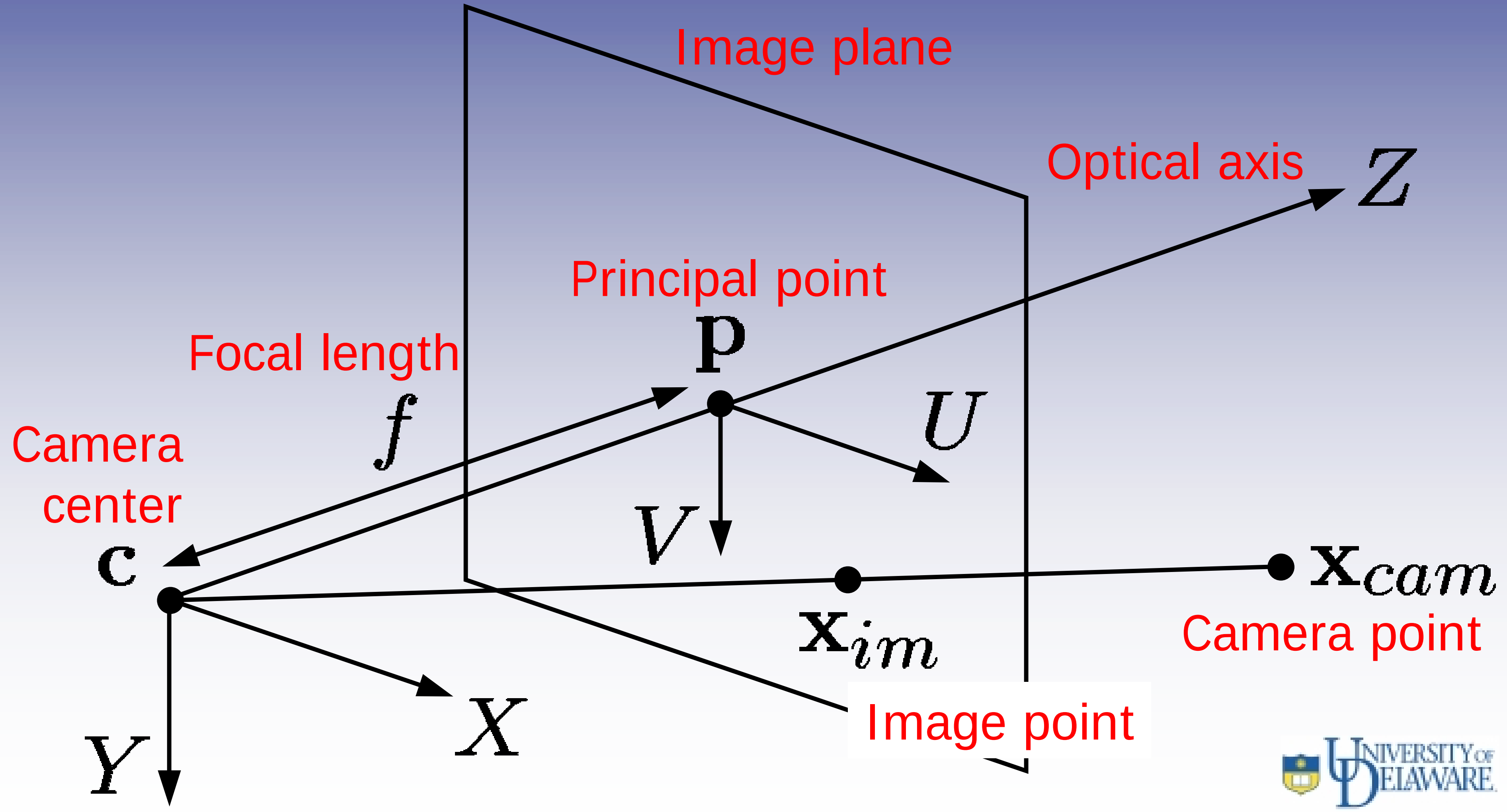
3-D Transformations: Change of Coordinates

- Points in one coordinate sytem are transformed to the other as follows:

$$\mathbf{x}_2 = \mathbf{T}\mathbf{x}_1 = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{x}_1$$

- $\mathbf{T} : \mathbf{x}_{world} \rightarrow \mathbf{x}_{cam}$ (taking the camera to the world origin) represents the camera's *extrinsic parameters*

Pinhole Camera Model

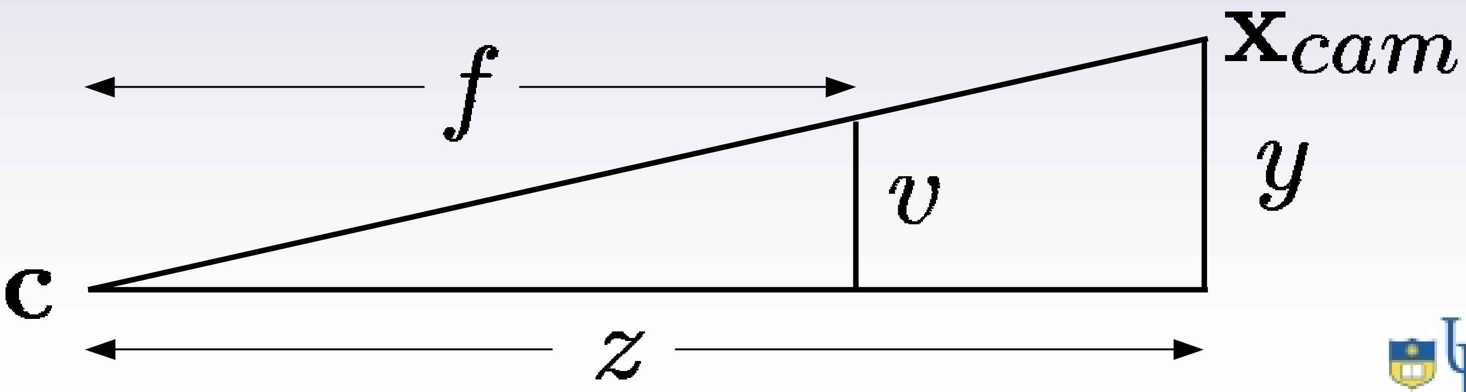


UNIVERSITY OF DELAWARE

Pinhole Perspective Projection

- Letting the camera coordinates of the projected point be $\mathbf{x}_{cam} = (x, y, z)^T$ leads by similar triangles to:

$$\mathbf{x}_{im} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} fx/z \\ fy/z \end{pmatrix}$$



Projection Matrix

- Using homogeneous coordinates, we can describe perspective projection with a linear equation:

$$\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \end{pmatrix}$$

(by the rule for converting between homogeneous and regular coordinates)

Camera Calibration Matrix

- More general projection matrix allows:
  - Image coordinates with an offset origin (e.g., convention of upper left corner)
  - Non-square pixels
  - Skewed coordinate axes

$$\mathbf{K} = \begin{pmatrix} f_u & \gamma & u_0 & 0 \\ 0 & f_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

- These five variables are known as the camera's *intrinsic parameters*

Combining Intrinsic & Extrinsic Parameters

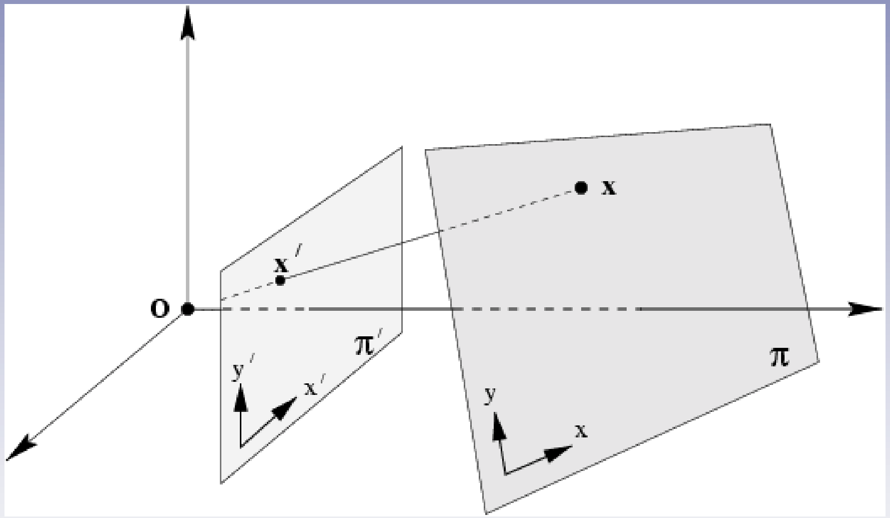
- The transformation performed by a pinhole camera on an arbitrary point can thus be written as:

$$\mathbf{x}_{im} = (\mathbf{KT})\mathbf{x}_{world} = \mathbf{P}\mathbf{x}_{world}$$

- $\mathbf{P}$ is called the *camera matrix*

Homographies

- 2-D to 2-D projective transformation mapping points from plane to plane (e.g., image of a plane)



from Hartley & Zisserman

- 3 x 3 homogeneous matrix $\mathbf{H}$ defines homo-graphy such that for any pair of corresponding points $\mathbf{x}_i$ and $\mathbf{x}'_i$, $\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i$

Computing the Homography

- 8 degrees of freedom in $\mathbf{H}$, so 4 pairs of 2-D points are sufficient to determine it
  - Other combinations of points and lines also work
- 3 collinear points in either image are a *degenerate configuration* preventing a unique solution
- Direct Linear Transformation (DLT) algorithm: Least-squares method for estimating $\mathbf{H}$

DLT Homography Estimation: Each of *n* Correspondences

- Since vectors are homogeneous, $\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i$ are parallel, so $\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \mathbf{0}$
- Let $\mathbf{h}_j^T$ be row *j* of $\mathbf{H}$, $\mathbf{h}$ be stacked $\mathbf{h}_j$'s
- Expanding and rearranging cross product above, we obtain $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where

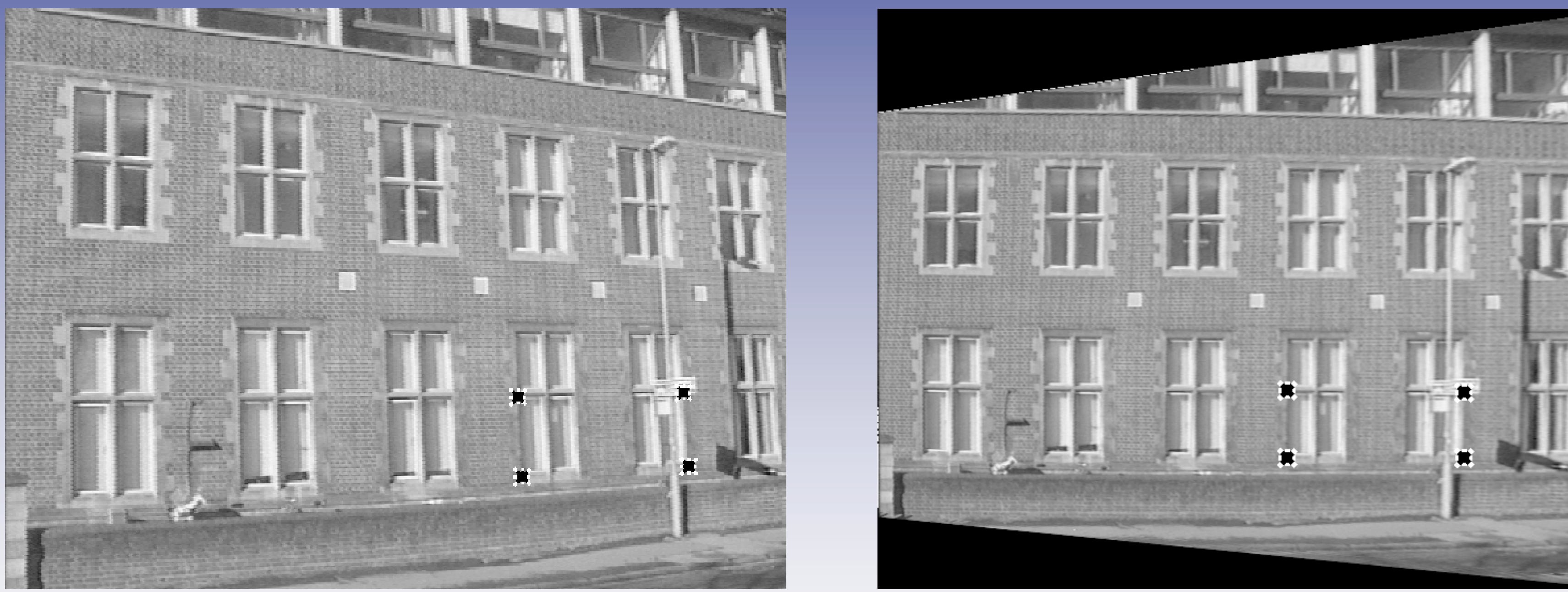
$$\mathbf{A}_i = \begin{bmatrix} \mathbf{0}^T & -w'_i\mathbf{x}_i^T & y'_i\mathbf{x}_i^T \\ w'_i\mathbf{x}_i^T & \mathbf{0}^T & -x'_i\mathbf{x}_i^T \\ -y'_i\mathbf{x}_i^T & x'_i\mathbf{x}_i^T & \mathbf{0}^T \end{bmatrix}$$

University of Delaware

DLT Homography Estimation: Solve System

- Only 2 linearly independent equations in each $\mathbf{A}_i$, so leave out 3rd to make it 2 x 9
- Stack every $\mathbf{A}_i$ to get 2*n* x 9 $\mathbf{A}$
- Solve $\mathbf{A}\mathbf{h} = \mathbf{0}$ by computing singular value decomposition (SVD) $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T$; $\mathbf{h}$ is last column of $\mathbf{V}$
- Solution is improved by normalizing image coordinates before applying DLT

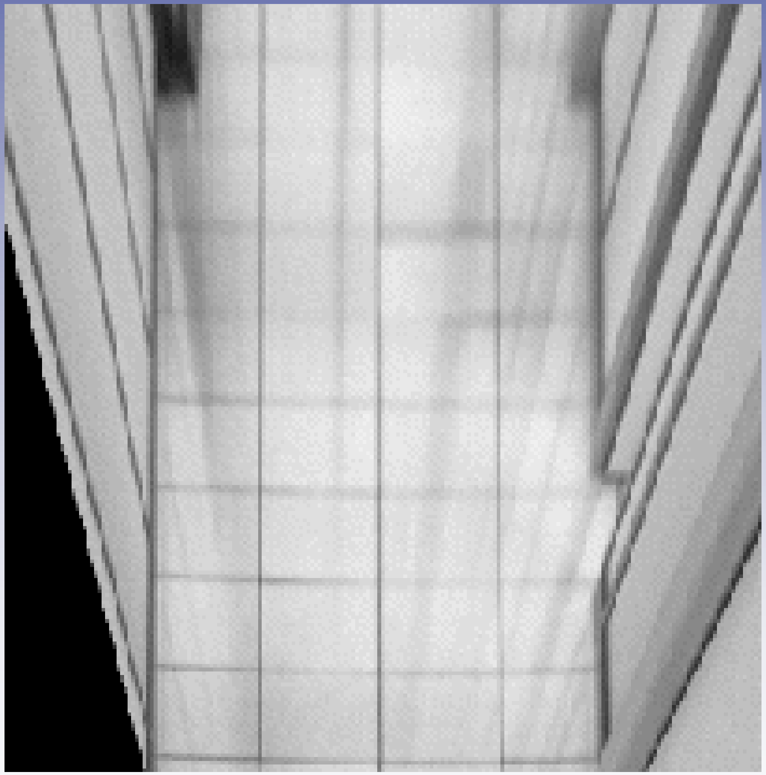
Applying Homographies to Remove Perspective Distortion



from Hartley & Zisserman

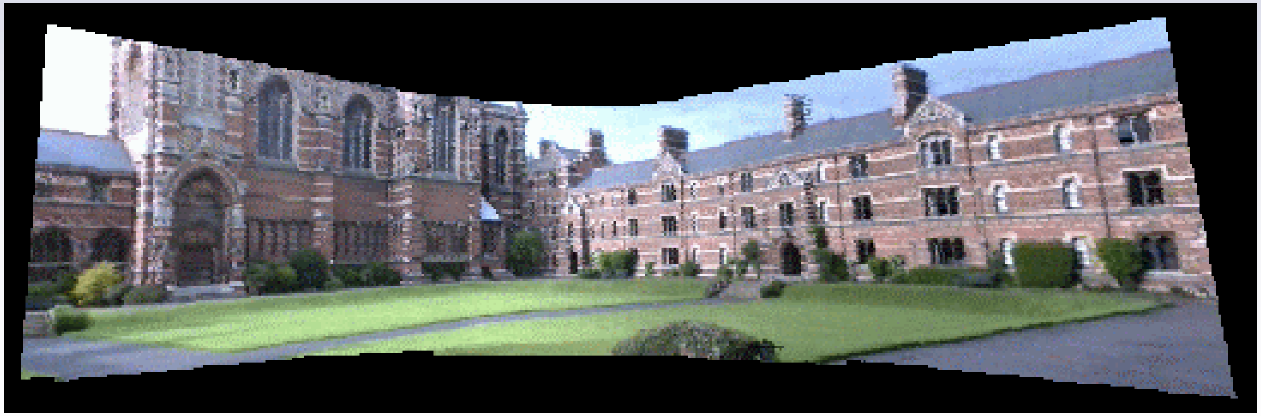
4 point correspondences suffice for the planar building facade

Homographies for Bird's-eye Views



from Hartley & Zisserman

Homographies for Mosaicing



from Hartley & Zisserman

University of Delaware

Camera Calibration

- For a given camera, how to deduce $\mathbf{K}$ so we'll be able to predict the image locations of known points in the world accurately?
- Basic idea: take images of measured 3-D objects, estimate camera parameters that minimize difference between observations and predictions

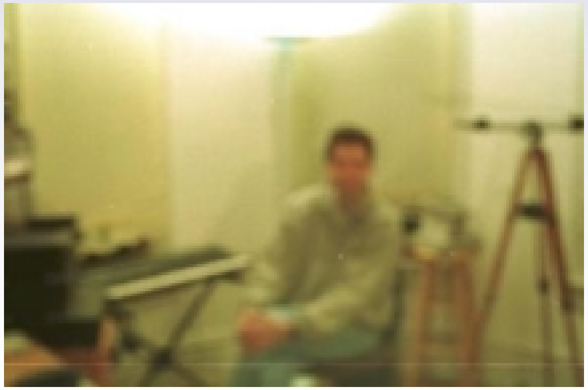
Estimating **K**

- Now we have a 3-D to 2-D projective transformation described by $\mathbf{x}' = \mathbf{P}\mathbf{x}$
- Follow approach of DLT used for homography estimation, except now:
  - $\mathbf{P}$ is 3 x 4, so need 5 1/2 point correspondences
  - Degeneracy occurs when 3-D points are coplanar or on a twisted cubic (space curve) with camera
  - Use RQ decomposition to separate estimated $\mathbf{P}$ into $\mathbf{K}$ and $\mathbf{T}$

Real Pinhole Cameras

- Actual pinhole cameras place the camera center between the image plane and the scene, reversing the image
- Problem: Size of hole leads to sharpness vs. dimness trade-off
- A really small hole introduces diffraction effects
- Solution: Light-gathering lens

A (fuzzy) pinhole image



courtesy of Paul Debevec

University of Delaware

Lenses

- Benefits: Increase light-gathering power by focusing bundles of rays from scene points onto image points
- Complications
  - Limited depth of field
  - Radial, tangential distortion: Straight lines curved
  - Vignetting: Image darker at edges
  - Chromatic aberration: Focal length function of wavelength

Correcting Radial Distortion

Distorted

courtesy of Shawn Becker

After correction

Modeling Radial Distortion

- Function of distance to camera center

$$\begin{pmatrix} u_d \\ v_d \end{pmatrix} = L(r) \begin{pmatrix} u \\ v \end{pmatrix}$$

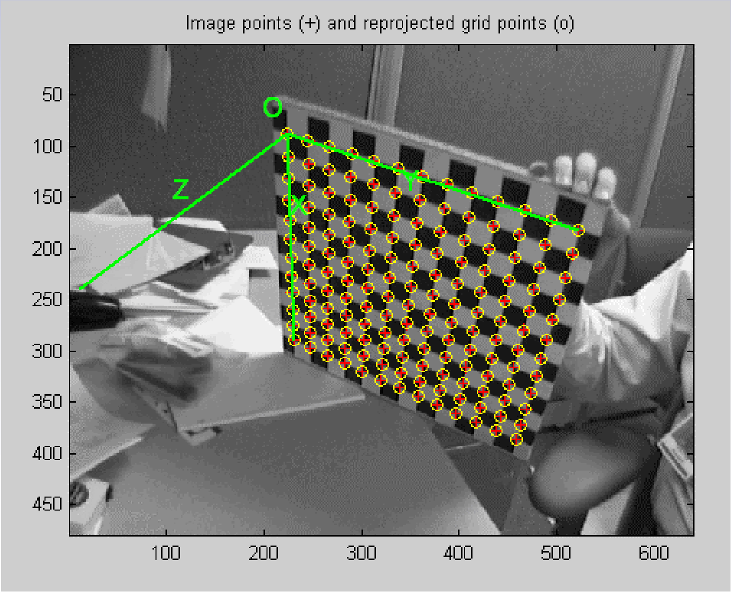
- Approximate with polynomial

$$L(r) = 1 + \kappa_1 r^2 + \kappa_2 r^4 + \ldots$$

- Necessitates nonlinear camera calibration

Camera Calibration Software

- Examples of camera calibration software in Links section of course web page
- Will discuss Bouguet's Matlab toolbox on Thursday



courtesy of
Jean-Yves Bouguet

University of Delaware