CS 532: 3D Computer Vision
1st Set of Notes

Instructor: Philippos Mordohai
Webpage: www.cs.stevens.edu/~mordohai
E-mail: Philippos.Mordohai@stevens.edu
Office: Lieb 215

Objectives

- Approach Computer Vision from a geometric, 3D perspective
  - Negligible overlap with traditional Computer Vision course (CS 558)
  - Explain image formation, single and multi-view geometry, structure from motion
- Introduce Computational Geometry concepts
  - Point clouds, meshes, Delaunay triangulation

Important Points

- This is an elective course. You chose to be here.
- Expect to work and to be challenged.
- Exams won't be based on recall. They will be open book and you will be expected to solve new problems.

Logistics

- Office hours: Tuesday 5-6 and by email
- Evaluation:
  - 7 homework sets (70%)
  - Quizzes and participation (15%)
  - Final exam (15%)

Textbooks

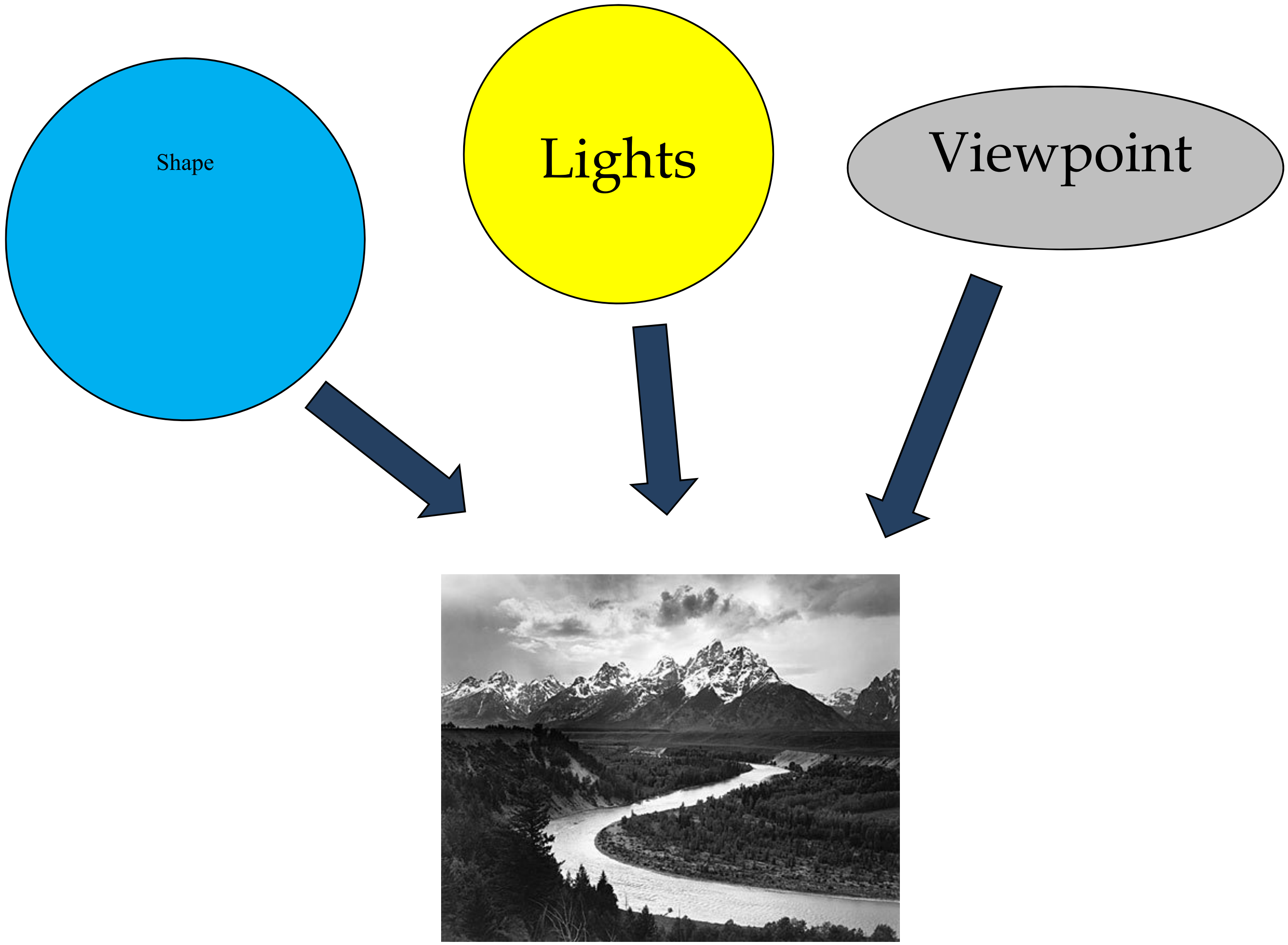
- Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010
- David M. Mount, CMSC 754: Computational Geometry lecture notes, Department of Computer Science, University of Maryland, Spring 2012

- Both available online

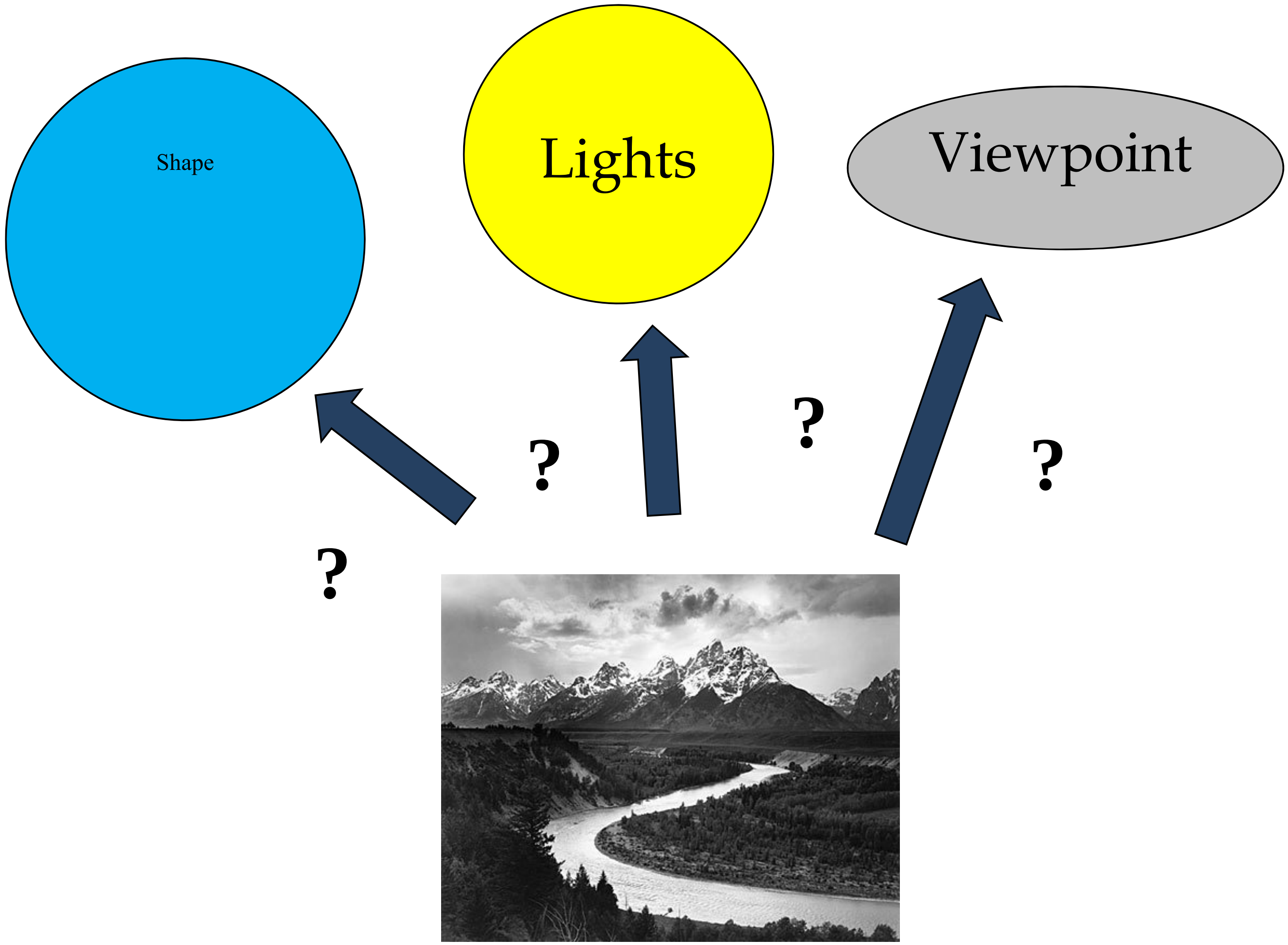
What is Computer Vision

- Why is it not image processing?

Graphics vs. Vision



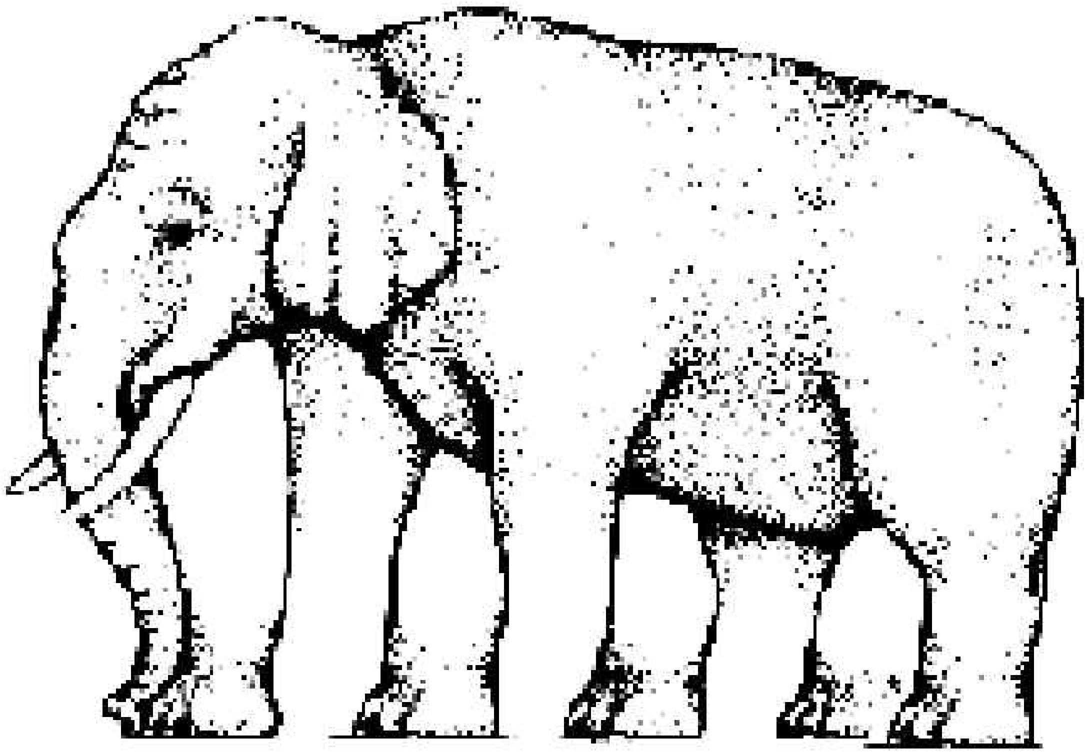
Graphics vs. Vision



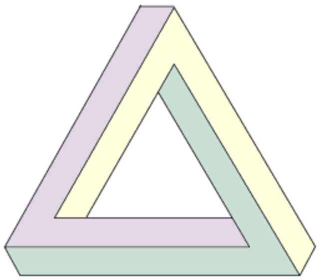
Vision is Hard

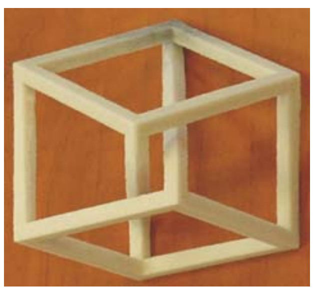
Vision is Hard

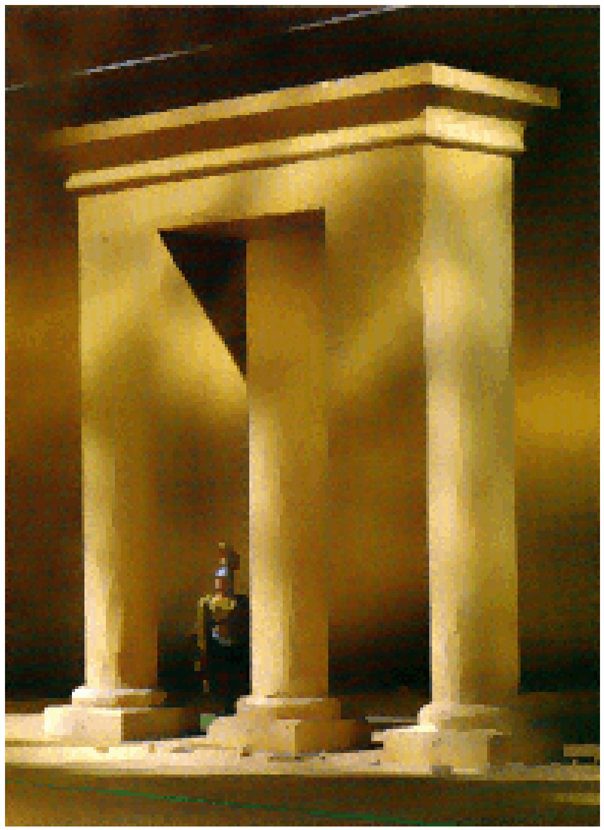
Vision is Hard

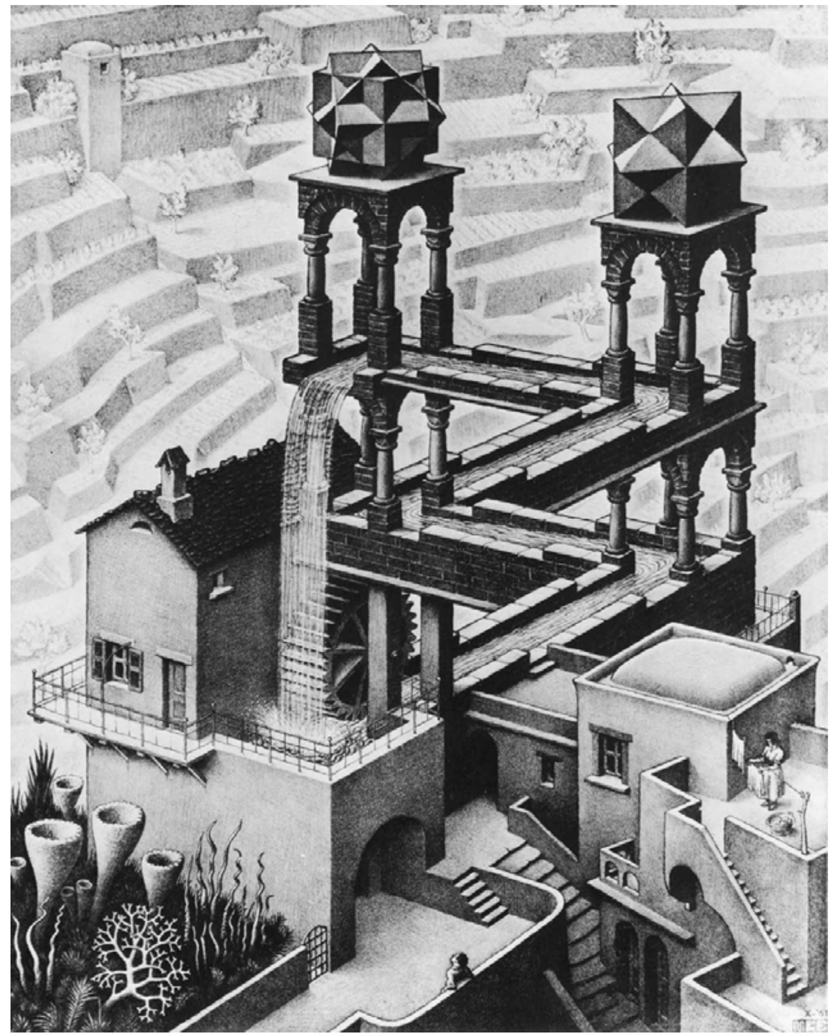


Vision is Hard

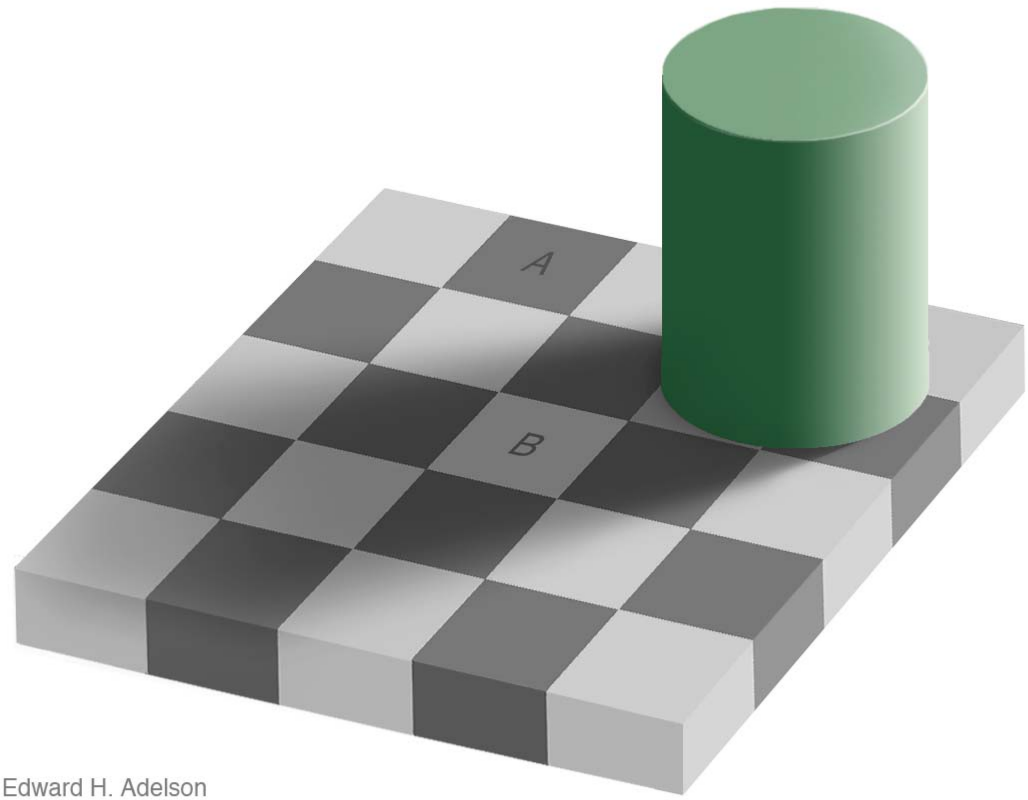






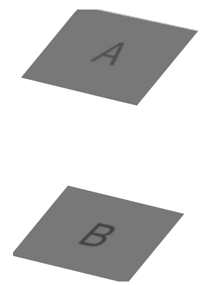


Vision is Hard

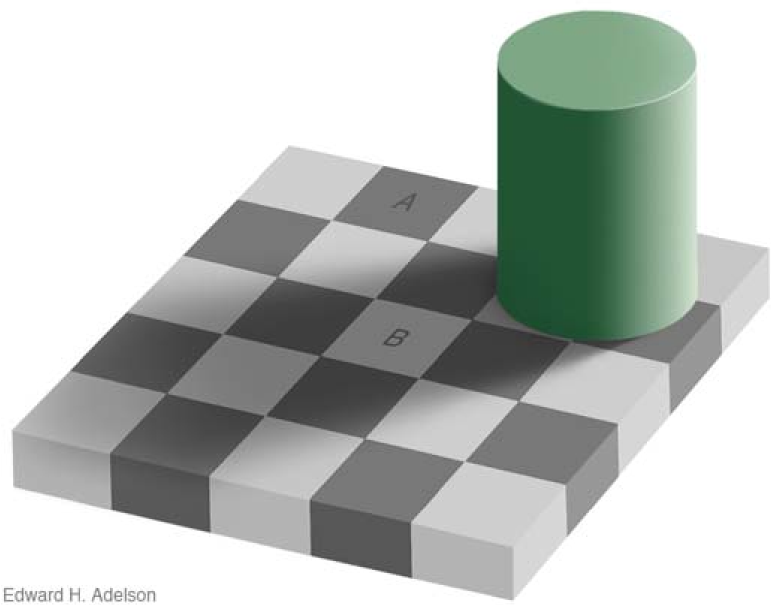


Edward H. Adelson

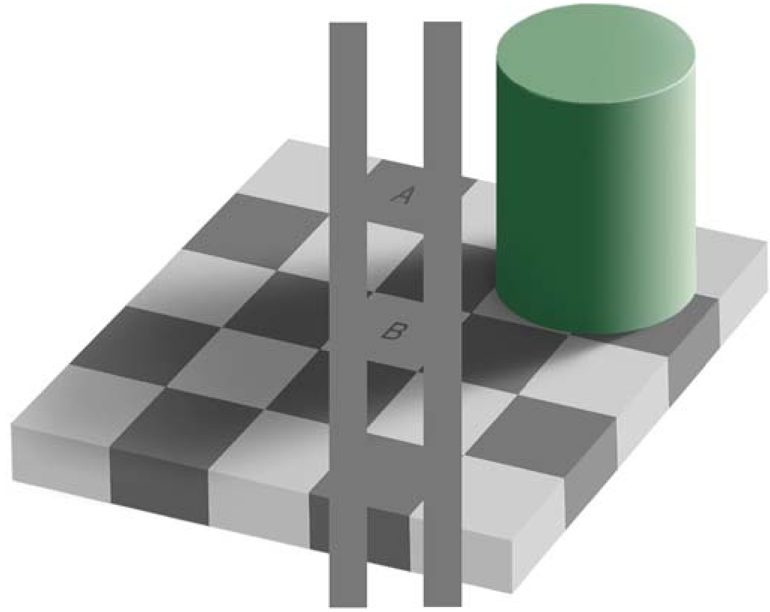
Vision is Hard



Vision is Hard



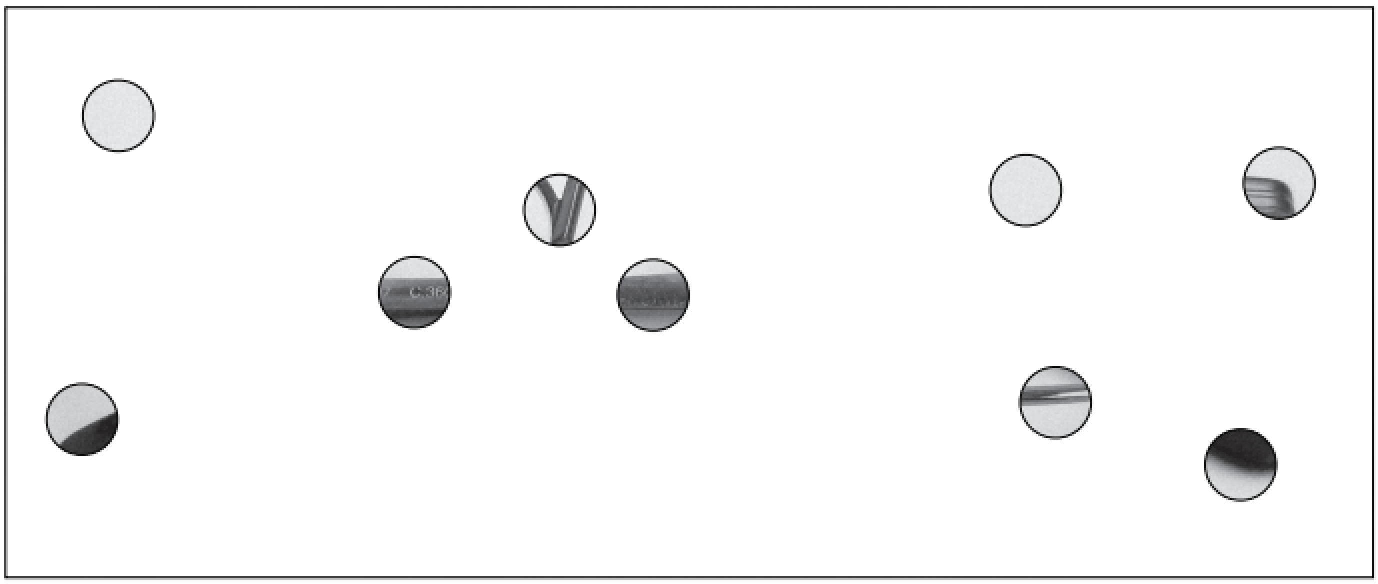
Edward H. Adelson



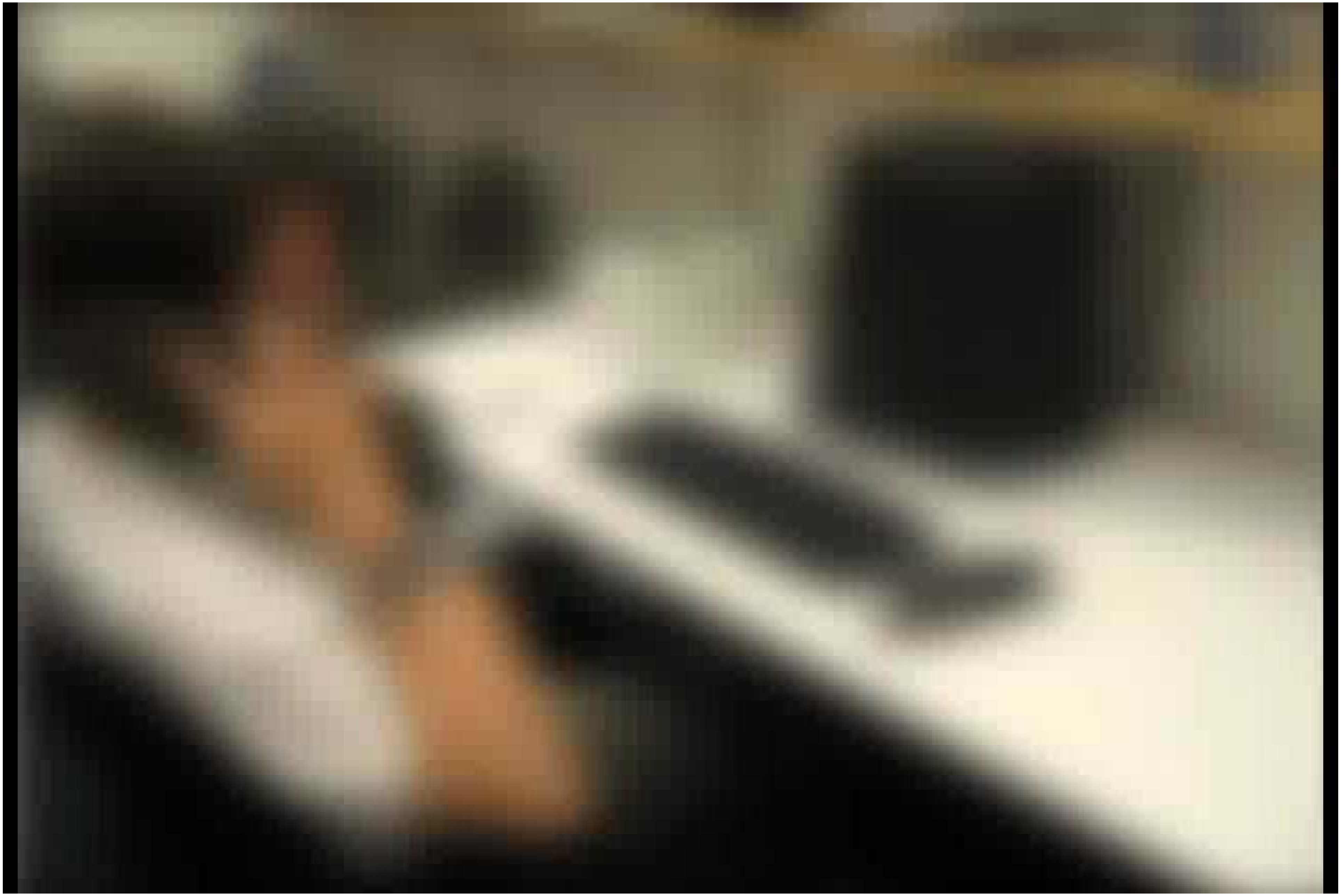
Vision is Hard

- A 2D picture may be produced by many different 3D scenes

Vision is Hard



Vision is Hard



Why is Vision Hard?

- Loss of information due to projection from 3D to 2D
  - Infinite scenes could have generated a given image
- Image colors depend on surface properties, illumination, camera response function and interactions such as shadows
  - HVS very good at ignoring distractors
- Noise
  - sensor noise and nonlinearities, quantization
- Lots of data
- Conflicts among local and global cues
  - Illusions

The Horizon

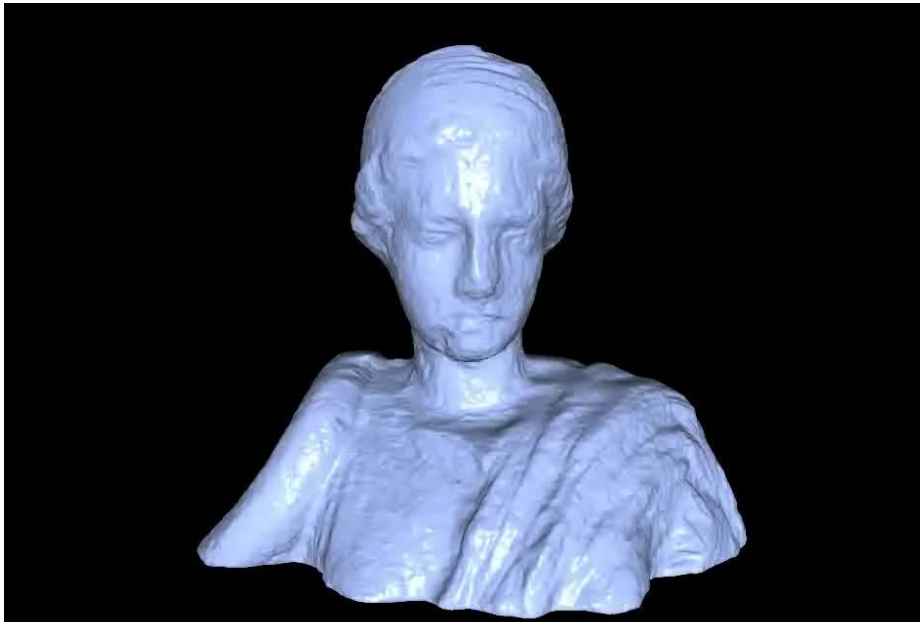
- Not all hard to explain phenomena are unusual…

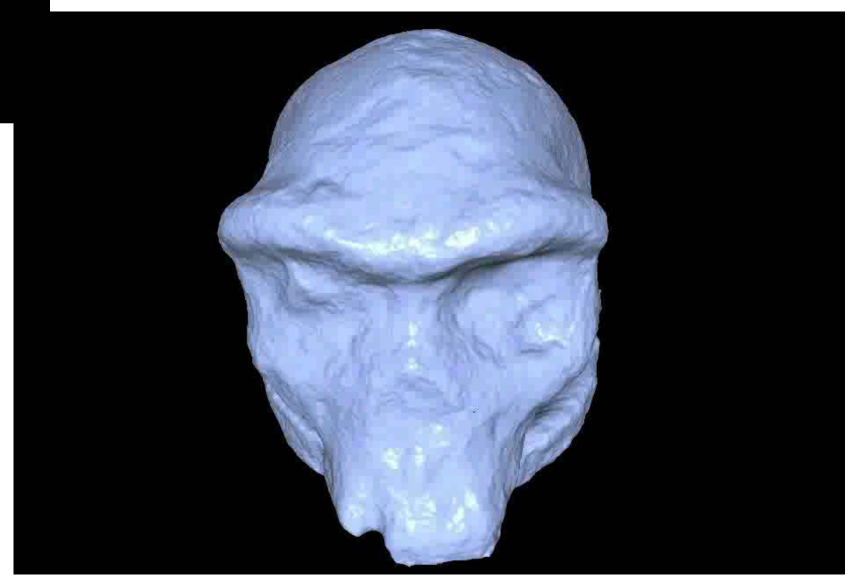
Vanishing Points

Why 3D Vision?

- Structure from Motion
  – Simultaneous Localization and Mapping
- 3D reconstruction
  – Dense mapping …
- 3D motion capture
- Medical applications
- Robotics and autonomous driving
  – Driver assistance

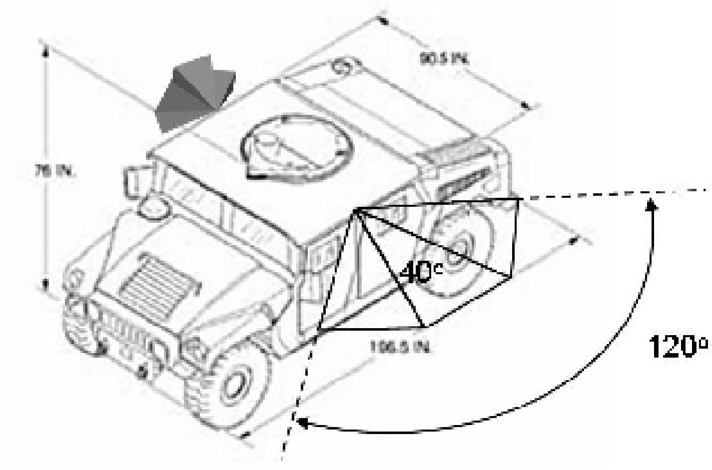
3D Models





Real-Time Video-based 3D Reconstruction

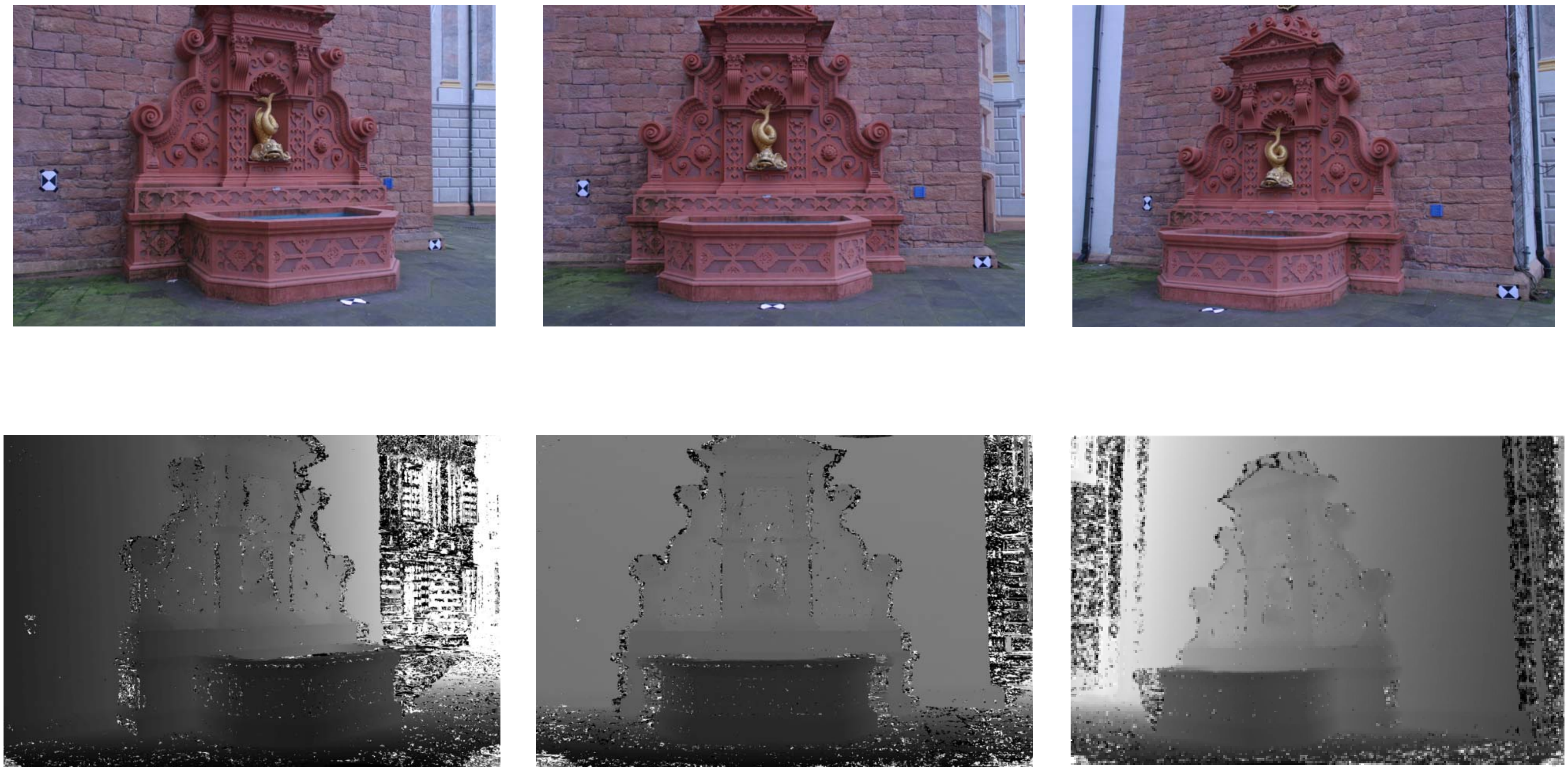
- Goal: real-time reconstruction of urban environments for visualization and training
- Platform:
  - 8 *non-overlapping* cameras
  - Differential GPS
  - Inertial Navigation System



Data Collection

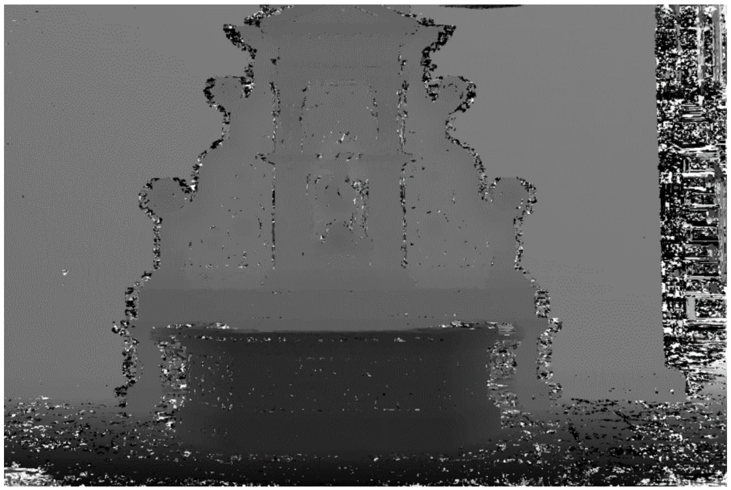
Results: Chapel Hill

Depth Map Estimation



3 of 11 images and corresponding depth maps

Depth Map Fusion



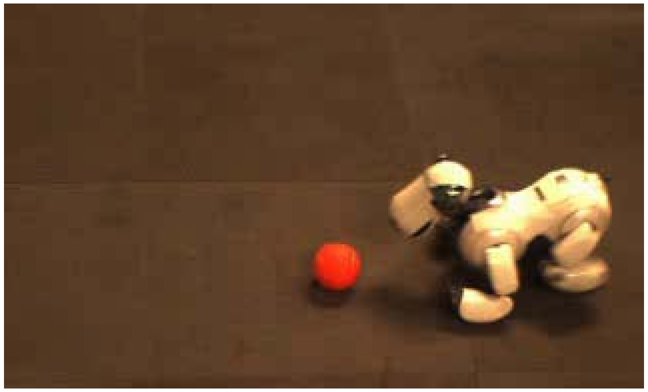
Raw Depth Map

Fused Depth Map

Colored Point Clouds

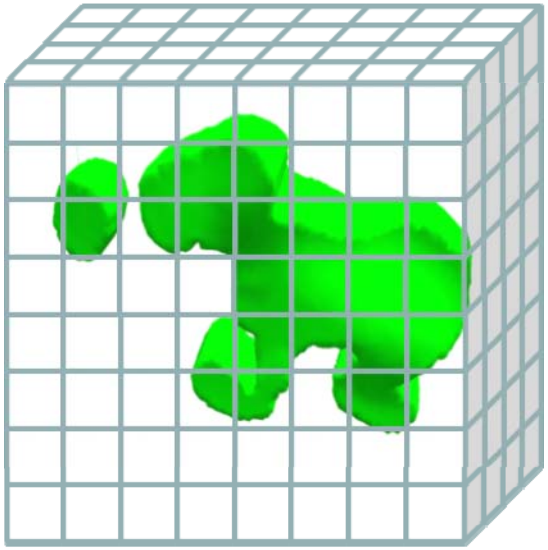
Fluid-in-Video

- Insert non-rigid objects in real video
- Stereo generates visible surfaces only
- Need:
  - Plausible completion of invisible surfaces
  - 3D velocities
  - Temporal consistency for fluid simulation stability

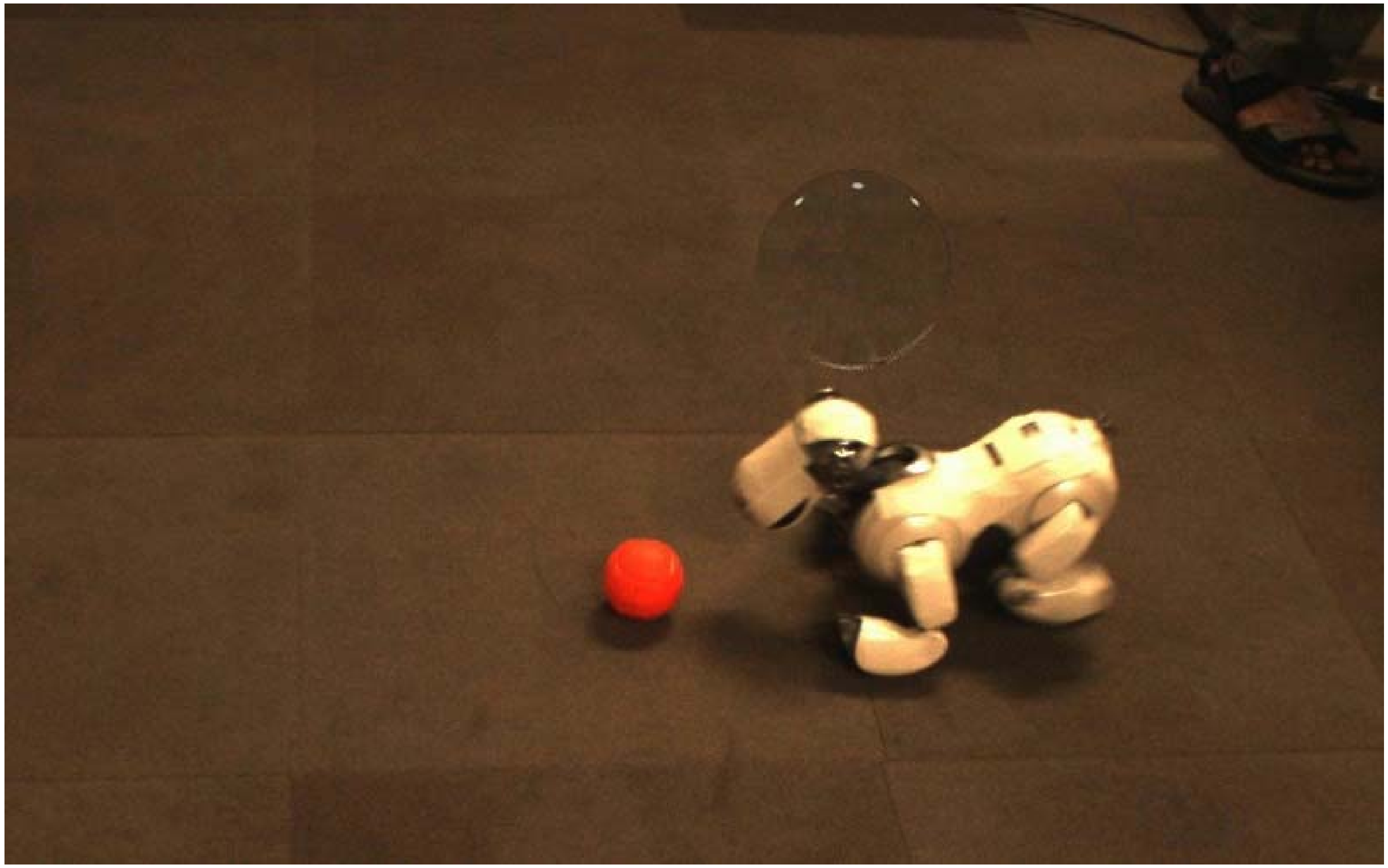


Fluid Simulation

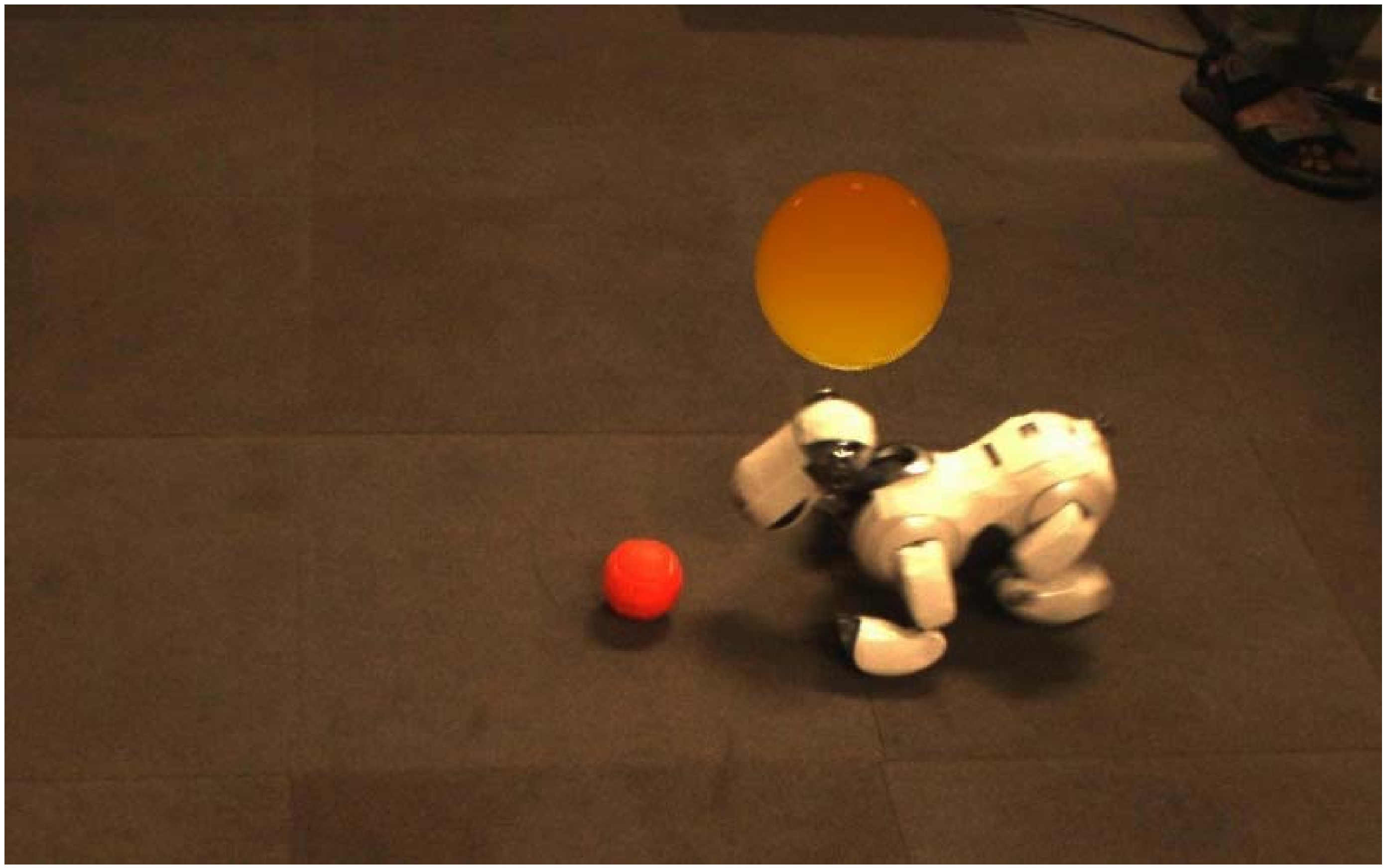
- Discretize reconstructed objects onto grid
  - Signed Distance Function representation
- Foreground discretized for every simulation time-step
- Background discretized just once



AIBO and Water



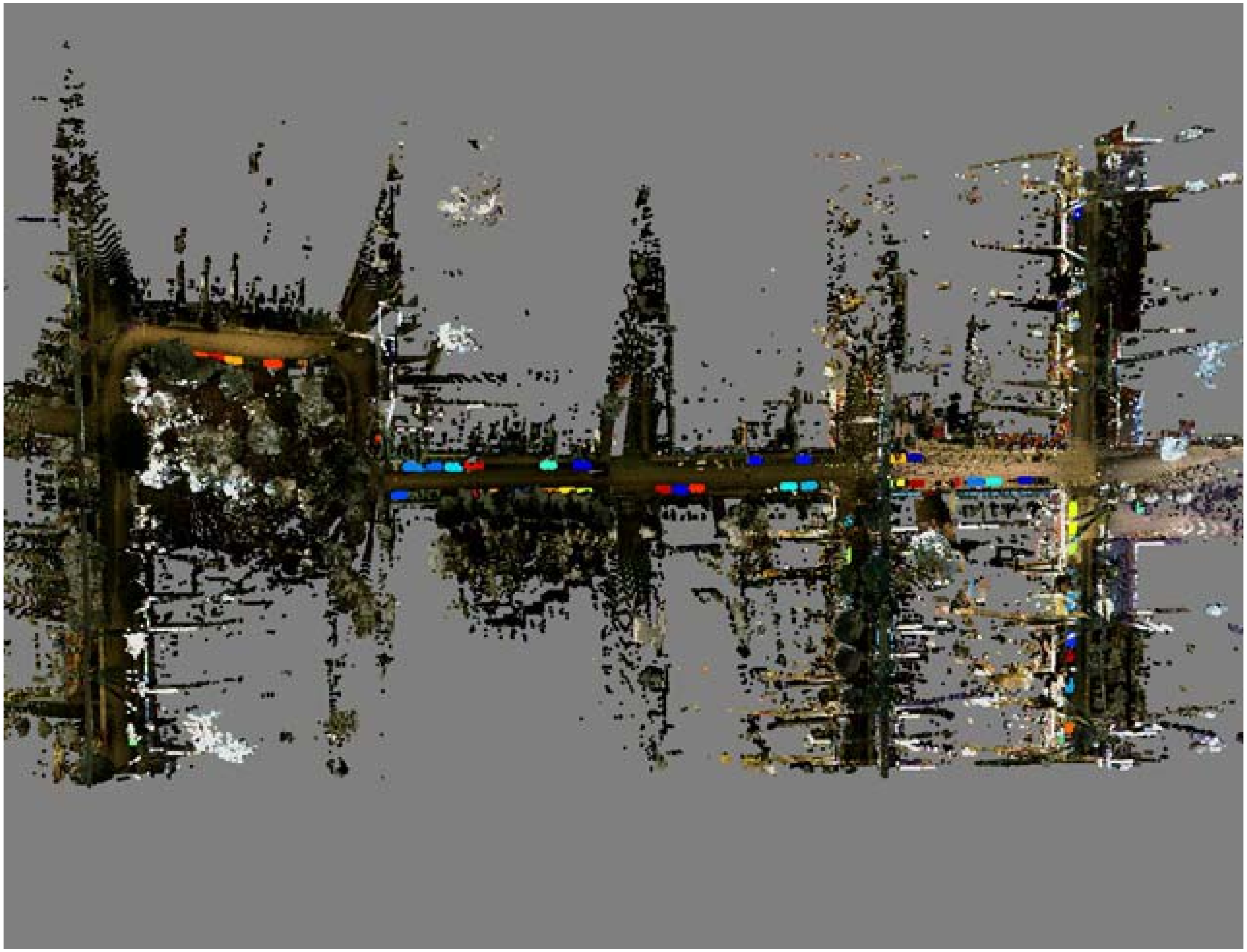
AIBO and Honey



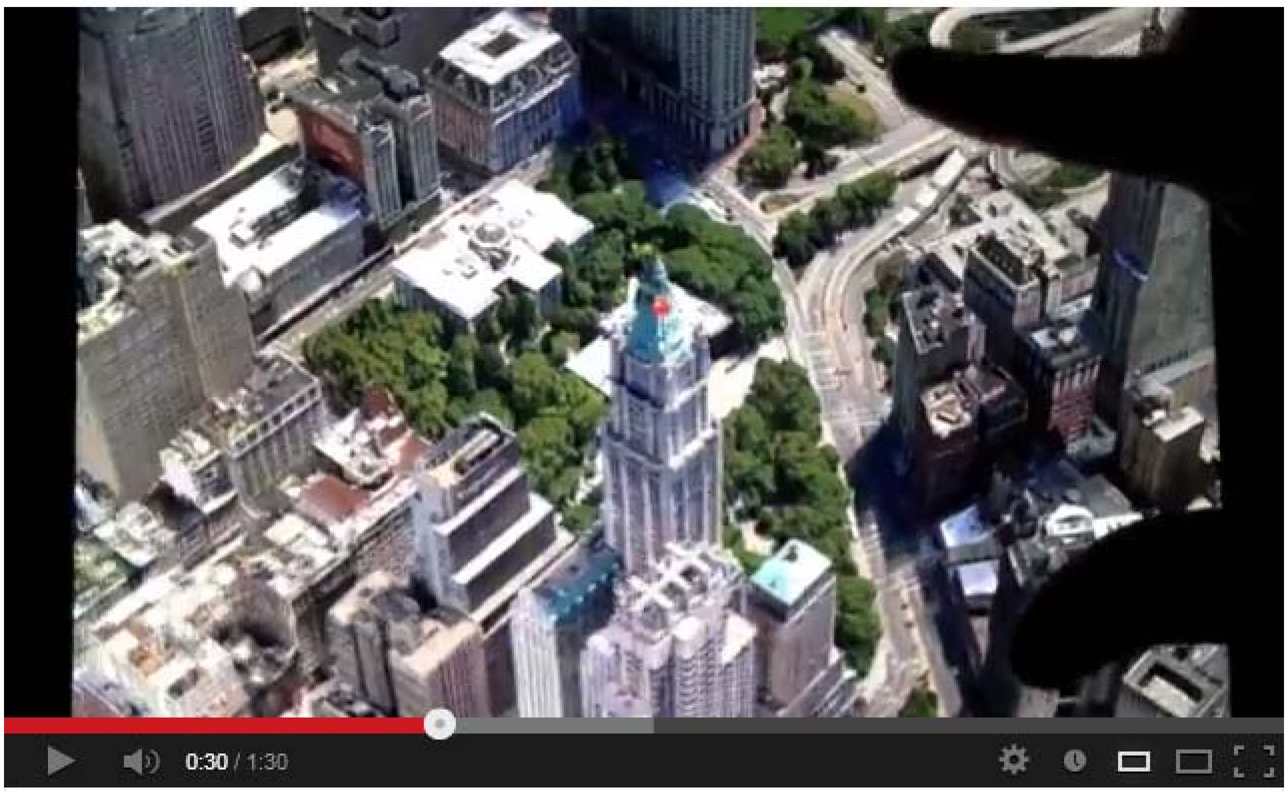
Baby and Water

Baby and Milk

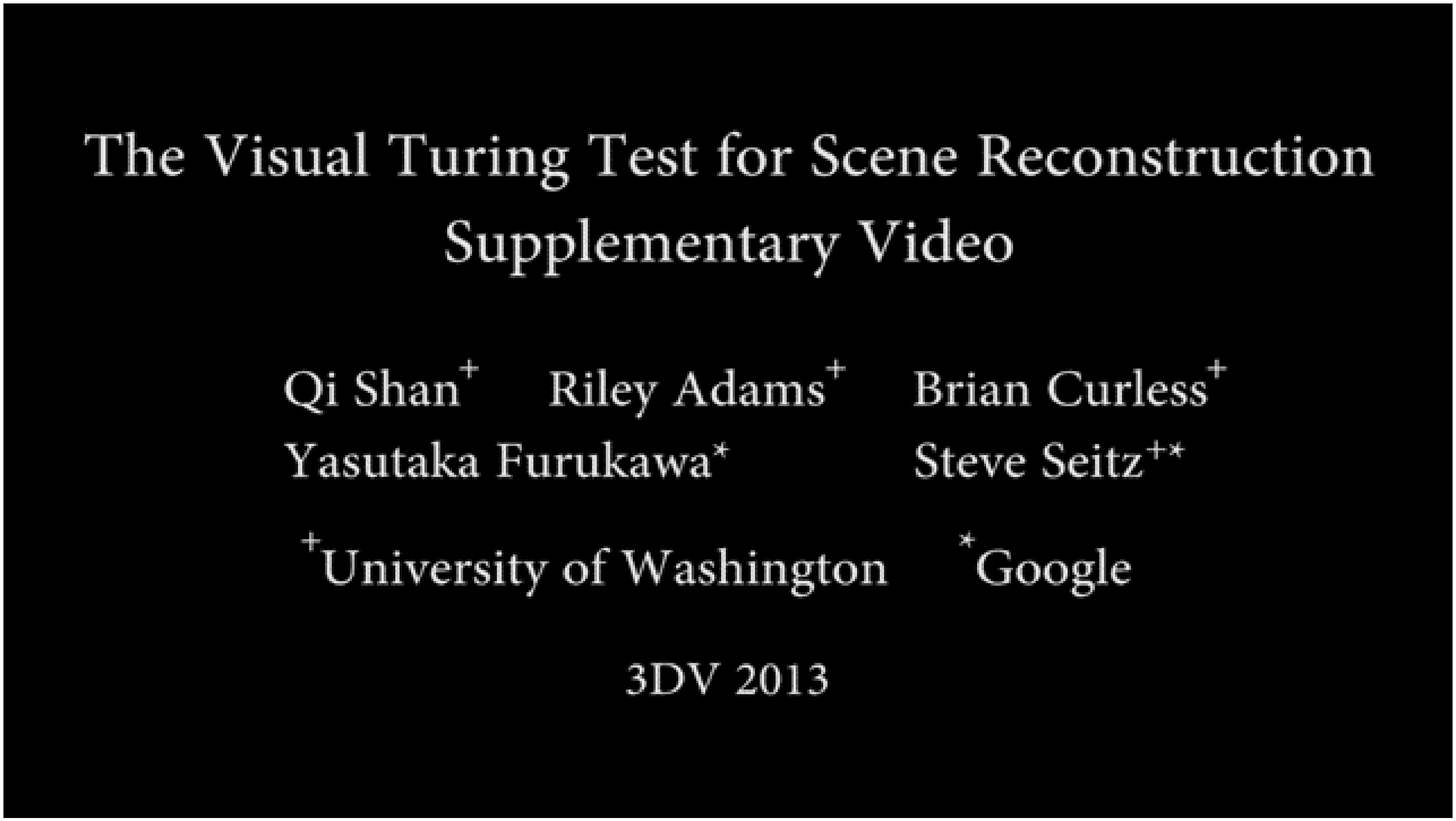
Car Detection



Apple Maps Flyover – New York



Visual Turing Test (UW)



Shan, Adams, Curless, Furukawa and Seitz (2013)

Introduction to Geometry

Based on slides by M. Pollefeys (ETH)
and D. Cappelleri (Purdue)

Points and Lines in 2D

- A point (x, y) lies on a line (a, b, c) when:
  - $ax+by+c = 0$ or $(a, b, c)(x, y, 1)^T = 0$
- Use homogeneous coordinates to represent points => add an extra coordinate
  - Note that scale is unimportant for determining incidence: $k(x, y, 1)$ is also on the line
  - Homogeneous coordinates (x_1, x_2, x_3), but only two degrees of freedom
  - Equivalent to inhomogeneous coordinates (x, y)

Points from Lines and Vice Versa

- The intersection of two lines l and l' is given by: $l \times l'$
- The line connecting two points x and x' is given by: $x \times x'$

Ideal Points and the Line at Infinity

- Intersection of two parallel lines:
  - l = (a, b, c) and l' = (a, b, c')
  - l×l' = (b, -a, 0)
- Ideal points: $(x_1, x_2, 0)$
- Belong to the line at infinity l = (0, 0, 1)
- $P^2 = R^3$-(0, 0, 0) (projective space)
  - In P^2 there is no distinction between regular and ideal points

Rotation in 2D

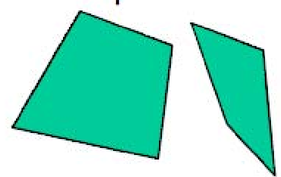
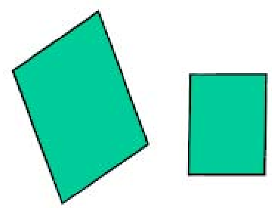
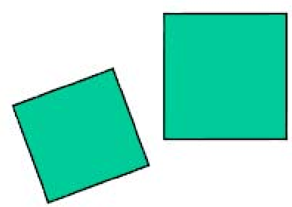
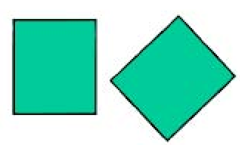
- Matrices are operators that transform vectors
  - 2D rotation matrix R= $\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$

- In homogeneous coordinates $\begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}$

Hands-on: 2D Transformations

- How to translate a point in homogeneous and inhomogeneous coordinates?
- How to rotate a point around the origin?
- How to rotate a point around a center other than the origin?

Hierarchy of 2D Transformations

		transformed squares	invariants
Projective 8dof	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6dof	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Parallellism, ratio of areas, ratio of lengths on parallel lines (e.g midpoints), linear combinations of vectors (centroids). **The line at infinity l_∞**
Similarity 4dof	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		Ratios of lengths, angles. **The circular points I,J**
Euclidean 3dof	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		lengths, areas.

Transformation of Points and Lines

Point transformation

$$\mathrm{x}' = \mathbf{H}\,\mathrm{x}$$

Line transformation

$$\mathrm{l}' = \mathbf{H}^{-\mathsf{T}}\,\mathrm{l}$$

Why?

3D points

3D point

$(X, Y, Z)^{\mathsf{T}}$ in $\mathbf{R}^3$

$\mathrm{X} = (X_1, X_2, X_3, X_4)^{\mathsf{T}}$ in $\mathbf{P}^3$

$$\mathrm{X} = \left(\frac{X_1}{X_4}, \frac{X_2}{X_4}, \frac{X_3}{X_4}, 1 \right)^{\mathsf{T}} = (X, Y, Z, 1)^{\mathsf{T}} \qquad (X_4 \neq 0)$$

projective transformation

$$\mathrm{X}' = \mathbf{H}\,\mathrm{X} \quad \text{(4x4-1=15 dof)}$$

Planes

3D plane

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0$$

$$\pi_1 X_1 + \pi_2 X_2 + \pi_3 X_3 + \pi_4 X_4 = 0$$

$$\pi^{\mathsf{T}} \mathrm{X} = 0$$

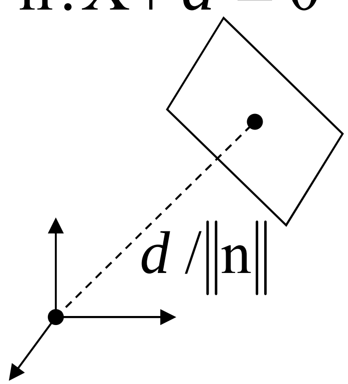
Transformation

$$\mathrm{X}' = \mathbf{H}\,\mathrm{X}$$
$$\pi' = \mathbf{H}^{-\mathsf{T}}\,\pi$$

Euclidean representation

$$\mathrm{n}.\tilde{\mathrm{X}} + d = 0 \qquad \mathrm{n} = (\pi_1, \pi_2, \pi_3)^{\mathsf{T}} \qquad \tilde{\mathrm{X}} = (X, Y, Z)^{\mathsf{T}}$$

$$\pi_4 = d \qquad X_4 = 1$$



Planes from points

Solve π from $X_1^T\pi = 0, X_2^T\pi = 0$ and $X_3^T\pi = 0$

$$\begin{bmatrix} X_1^T \\ X_2^T \\ X_3^T \end{bmatrix}\pi = 0$$ (solve as right nullspace of π)

$$\begin{bmatrix} X_1^T \\ X_2^T \\ X_3^T \end{bmatrix}$$

Or implicitly from coplanarity condition

$$\det\begin{bmatrix} X_1 & (X_1)_1 & (X_2)_1 & (X_3)_1 \\ X_2 & (X_1)_2 & (X_2)_2 & (X_3)_2 \\ X_3 & (X_1)_3 & (X_2)_3 & (X_3)_3 \\ X_4 & (X_1)_4 & (X_2)_4 & (X_3)_4 \end{bmatrix} = 0$$

$$X_1 D_{234} - X_2 D_{134} + X_3 D_{124} - X_4 D_{123} = 0$$

$$\pi = \left(D_{234}, -D_{134}, D_{124}, -D_{123}\right)^T$$

Planes from points

$$\pi = \left(D_{234}, -D_{134}, D_{124}, -D_{123}\right)^{\mathsf{T}}$$

$$\det\begin{bmatrix} X_1 & (X_1)_1 & (X_2)_1 & (X_3)_1 \\ X_2 & (X_1)_2 & (X_2)_2 & (X_3)_2 \\ X_3 & (X_1)_3 & (X_2)_3 & (X_3)_3 \\ X_4 & (X_1)_4 & (X_2)_4 & (X_3)_4 \end{bmatrix} = 0$$

$$D_{234} = \begin{vmatrix} Y_1 & Y_2 & Y_3 \\ Z_1 & Z_2 & Z_3 \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} Y_1 - Y_3 & Y_2 - Y_3 & Y_3 \\ Z_1 - Z_3 & Z_2 - Z_3 & Z_3 \\ 0 & 0 & 1 \end{vmatrix} = \left(\left(\tilde{\mathrm{X}}_1 - \tilde{\mathrm{X}}_3\right) \times \left(\tilde{\mathrm{X}}_2 - \tilde{\mathrm{X}}_3\right)\right)_1$$

(~Euclidean)

Points from planes

Solve X from $\pi_1^{\mathsf{T}}\mathrm{X} = 0$, $\pi_2^{\mathsf{T}}\mathrm{X} = 0$ and $\pi_3^{\mathsf{T}}\mathrm{X} = 0$

$$\begin{bmatrix} \pi_1^{\mathsf{T}} \\ \pi_2^{\mathsf{T}} \\ \pi_3^{\mathsf{T}} \end{bmatrix} \mathrm{X} = 0$$

(solve as right nullspace of $\begin{bmatrix} \pi_1^{\mathsf{T}} \\ \pi_2^{\mathsf{T}} \\ \pi_3^{\mathsf{T}} \end{bmatrix}$)

Lines are complicated…

Rotations

- Rotation matrices around the 3 axes

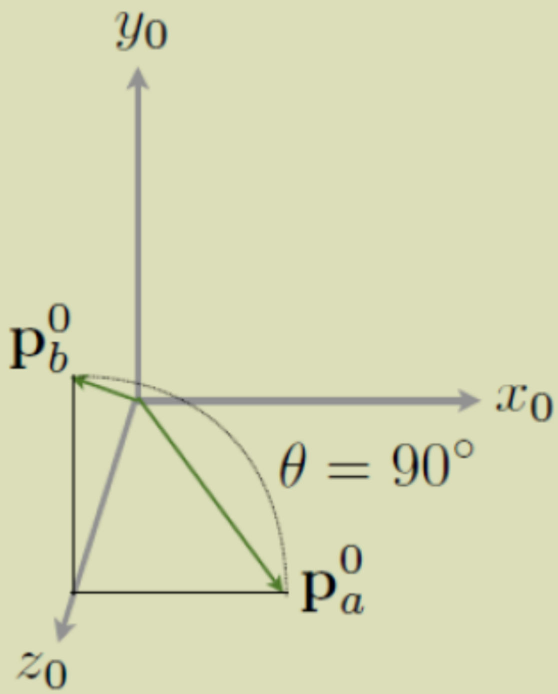
=> What is the inverse of a rotation matrix?

$$\mathbf{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation Example



The rotation matrix can be used to perform arbitrary rotations on vectors

$$\mathbf{v}_p^0 = \mathbf{R}_1^0 \; \mathbf{v}_p^1$$

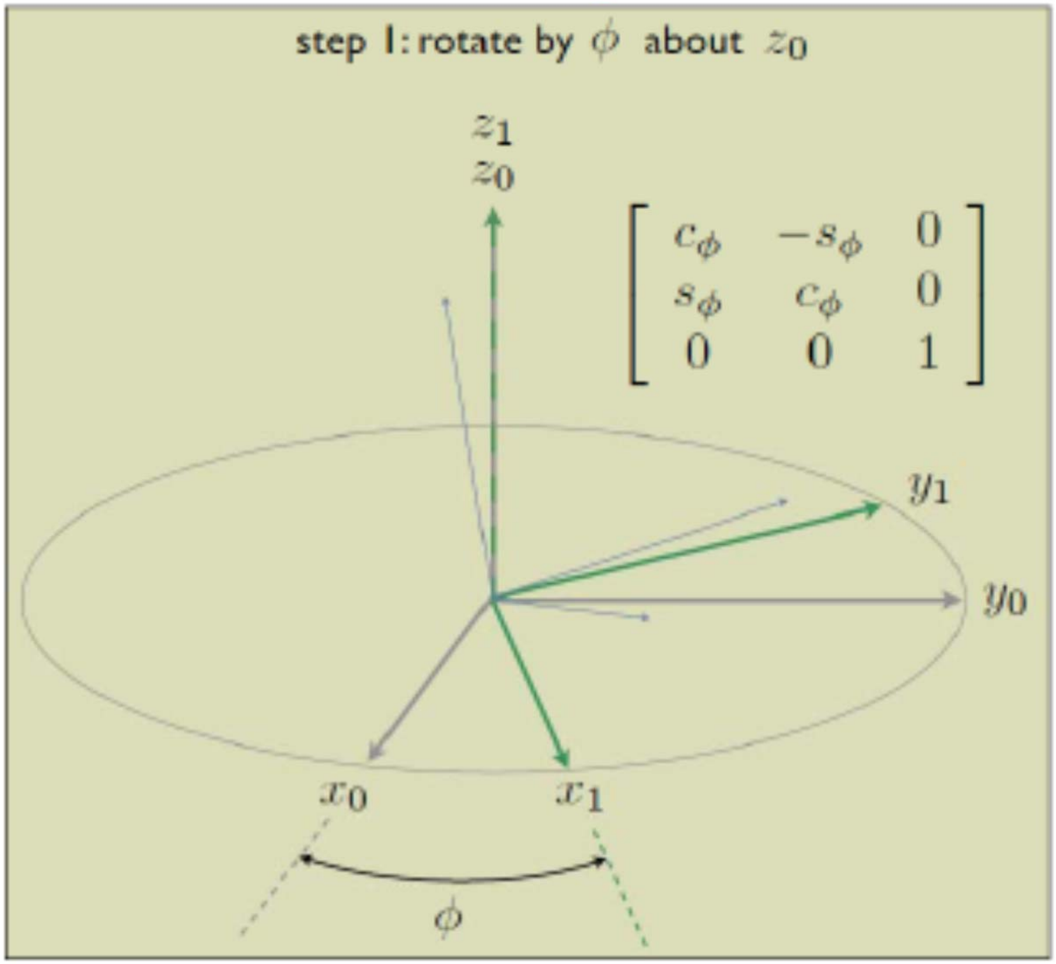
$$\mathbf{p}_a^0 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{p}_b^0 = \mathbf{R}_{z,\theta} \; \mathbf{p}_a^0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

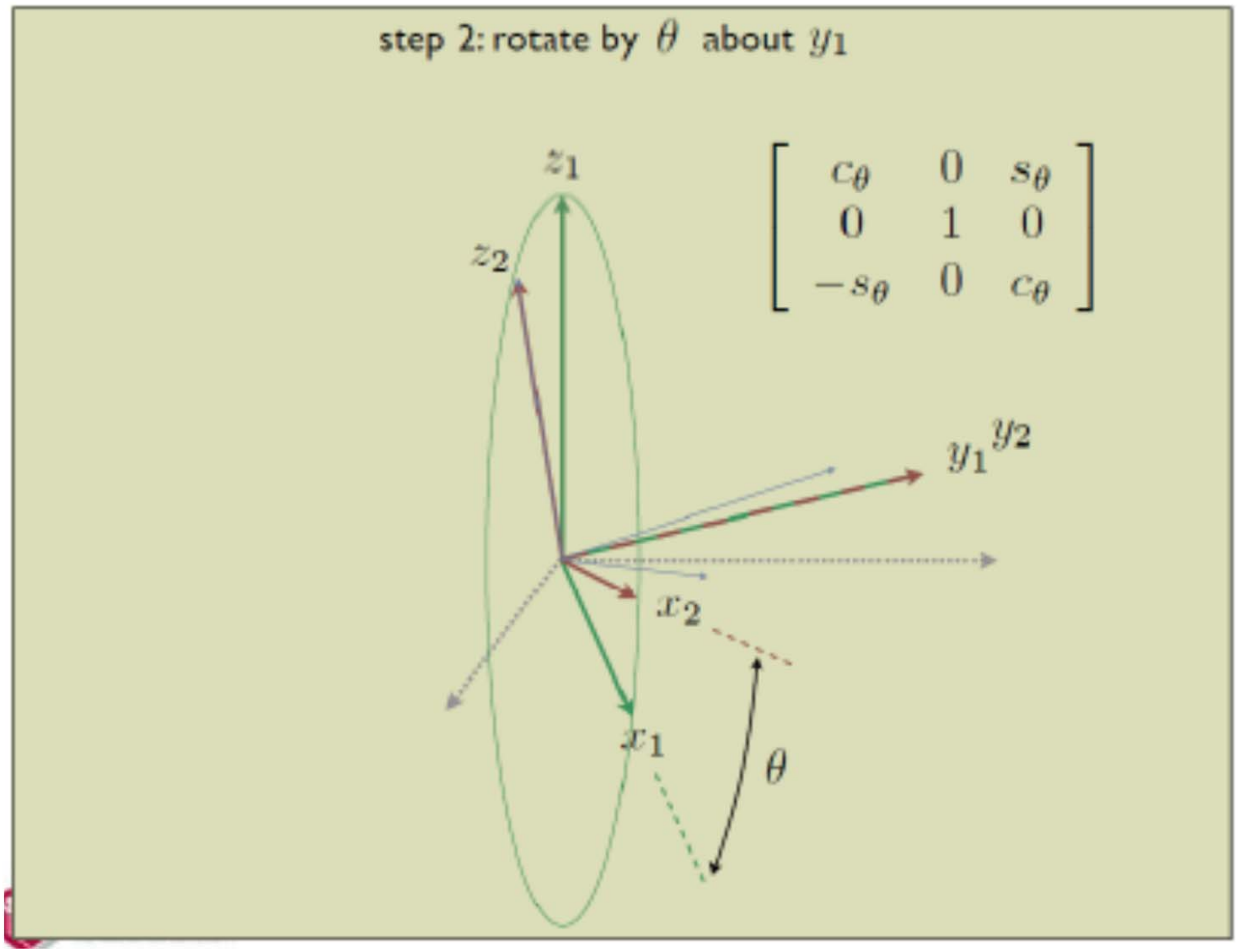
Parameterization of Rotations

- In 3D, the 9-element rotation matrix has 3 DOF
- Several methods exist for representing a 3D rotation
  - Euler angles
  - Pitch, Roll, Yaw angles
  - Axis/Angle representation
  - Quaternions

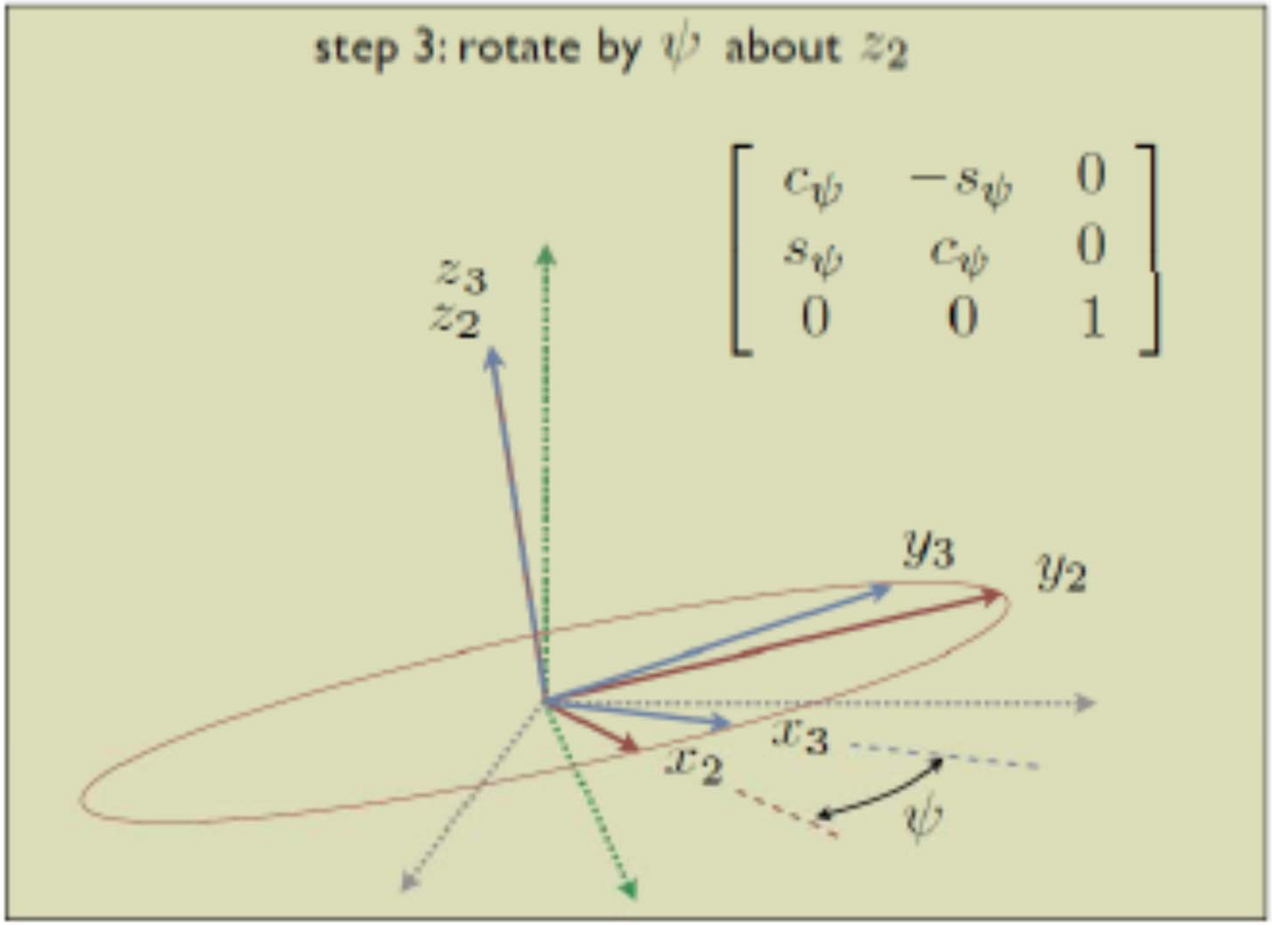
Euler Angles



Euler Angles



Euler Angles



Euler Angles to Rotation Matrix

(**post**-multiply using the **basic rotation matrices**)

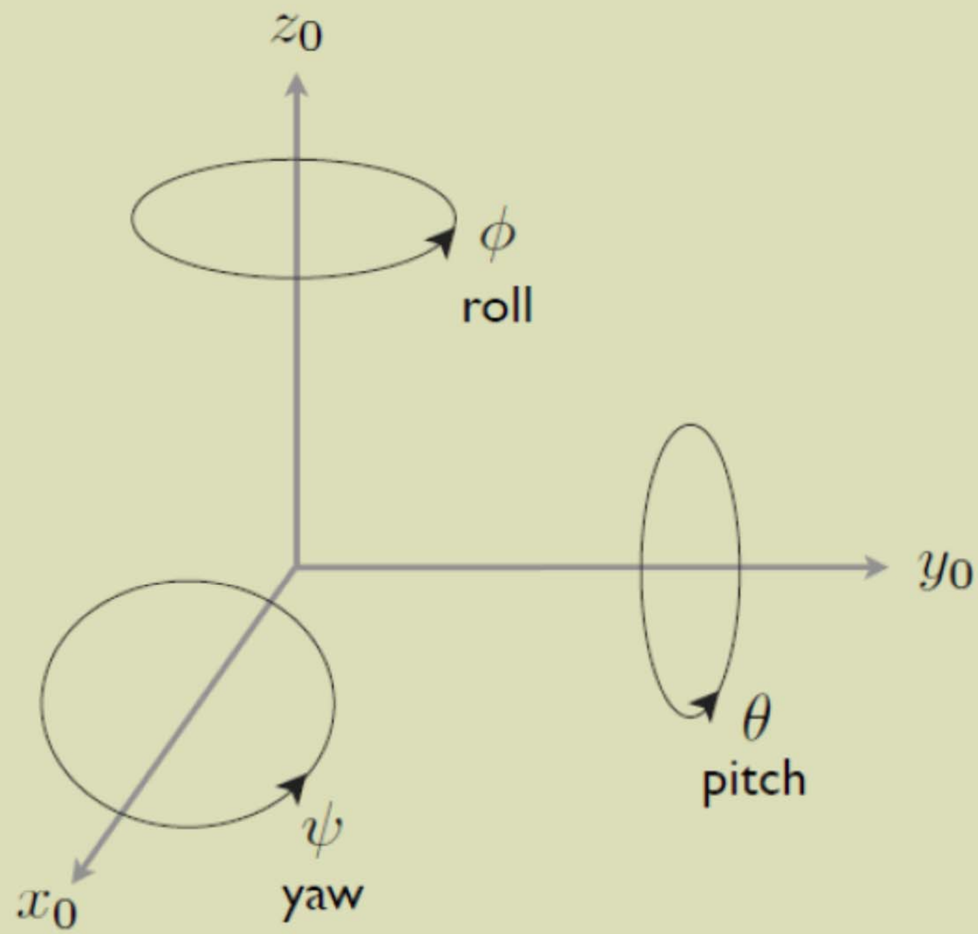
$$\mathbf{R} = \mathbf{R}_{z,\phi}\,\mathbf{R}_{y,\theta}\,\mathbf{R}_{z,\psi}$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi + c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\psi & c_\theta \end{bmatrix}$$

Roll, Pitch, Yaw Angles

defined as a set of three angles about a **fixed** reference



Roll, Pitch, Yaw Angles to Rotation Matrix

(**pre**-multiply using the **basic rotation matrices**)

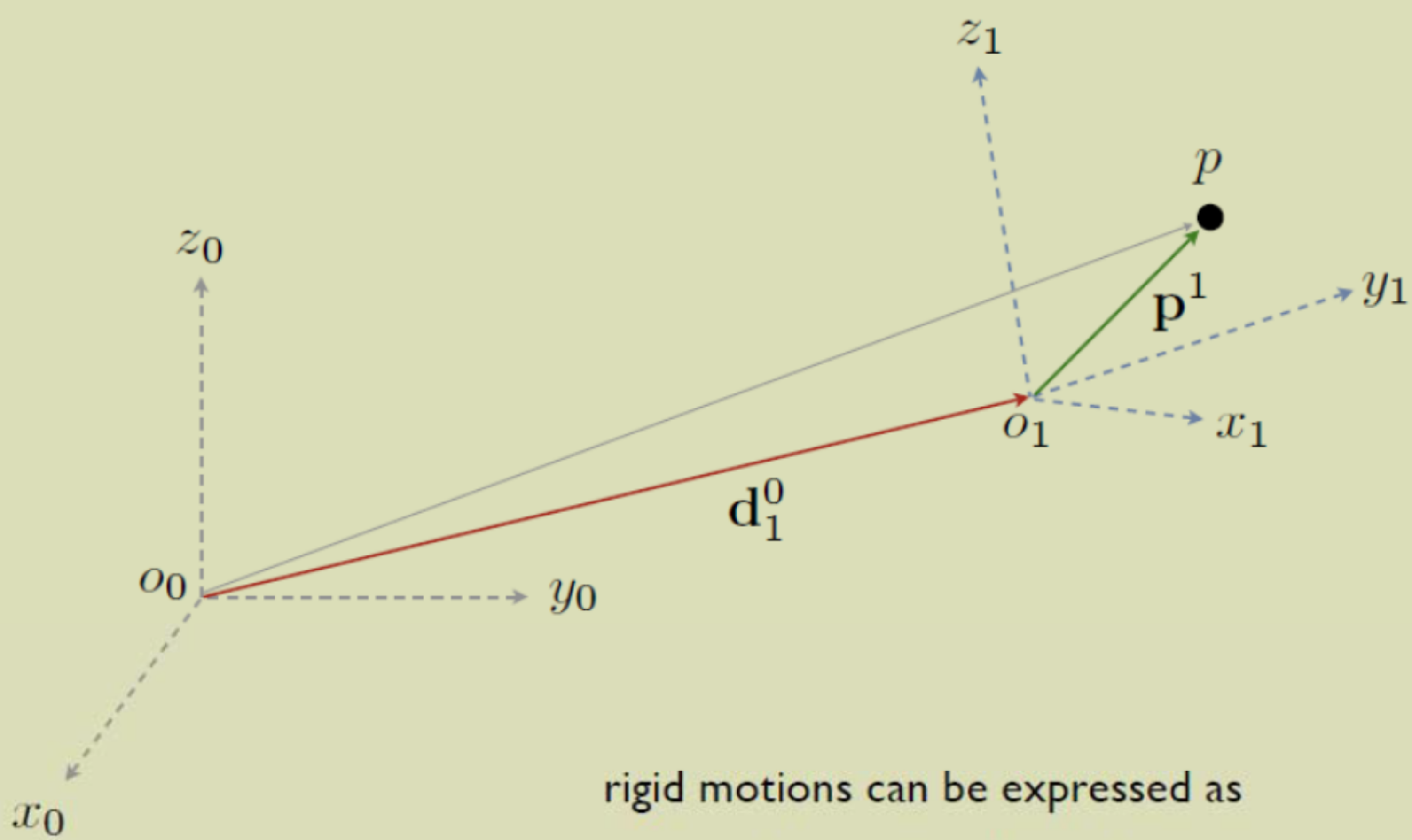
$$\mathbf{R} = \mathbf{R}_{z,\phi}\, \mathbf{R}_{y,\theta}\, \mathbf{R}_{x,\psi}$$

$$= \begin{bmatrix} c_\phi & -s_\phi & 0 \\ s_\phi & c_\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\psi & -s_\psi \\ 0 & s_\psi & c_\psi \end{bmatrix}$$

$$= \begin{bmatrix} c_\phi c_\theta & c_\phi s_\theta s_\psi - s_\phi c_\psi & s_\phi s_\psi + c_\phi s_\theta c_\psi \\ s_\phi c_\theta & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi \\ -s_\theta & c_\theta s_\psi & c_\theta c_\psi \end{bmatrix}$$

Rigid Motion

a **rigid motion** couples pure translation with pure rotation



rigid motions can be expressed as

$$\mathbf{p}^0 = \mathbf{R}_1^0\, \mathbf{p}^1 + \mathbf{d}_1^0$$

Homogeneous Transformation

a **homogeneous transform** is a matrix representation of rigid motion, defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{R}$ is the 3x3 rotation matrix, and $\mathbf{d}$ is the 1x3 translation vector

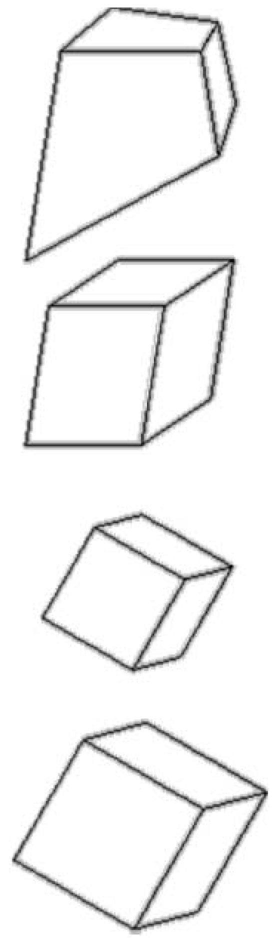
$$\mathbf{H} = \begin{bmatrix} n_x & s_x & a_x & d_x \\ n_y & s_y & a_y & d_y \\ n_z & s_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

the **inverse** of a homogeneous transform can be expressed as

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top d \\ 0 & 1 \end{bmatrix}$$

Hierarchy of 3D Transformations

Projective 15dof	$\begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$	Intersection and tangency
Affine 12dof	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$	Parallellism of planes, Volume ratios, centroids, **The plane at infinity π_∞**
Similarity 7dof	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$	Angles, ratios of length **The absolute conic Ω_∞**
Euclidean 6dof	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$	Volume



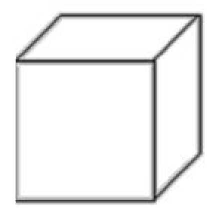
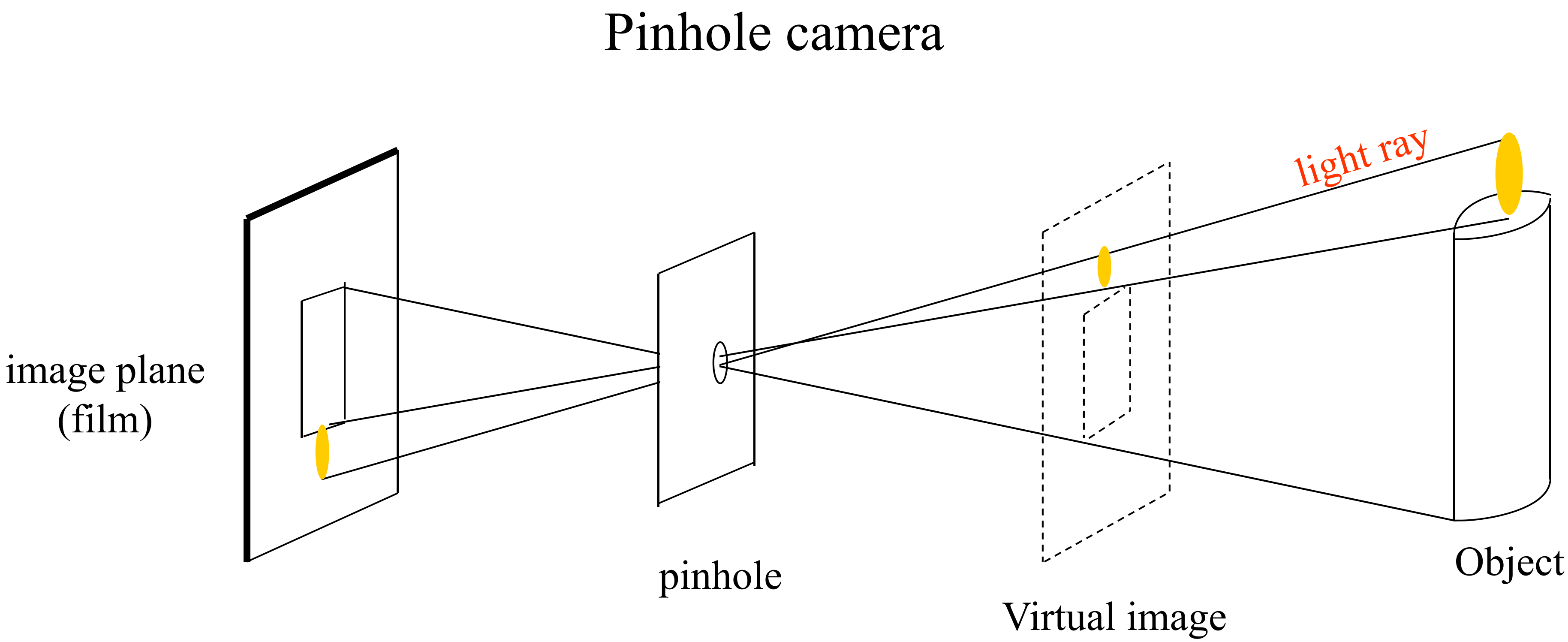


Image Formation

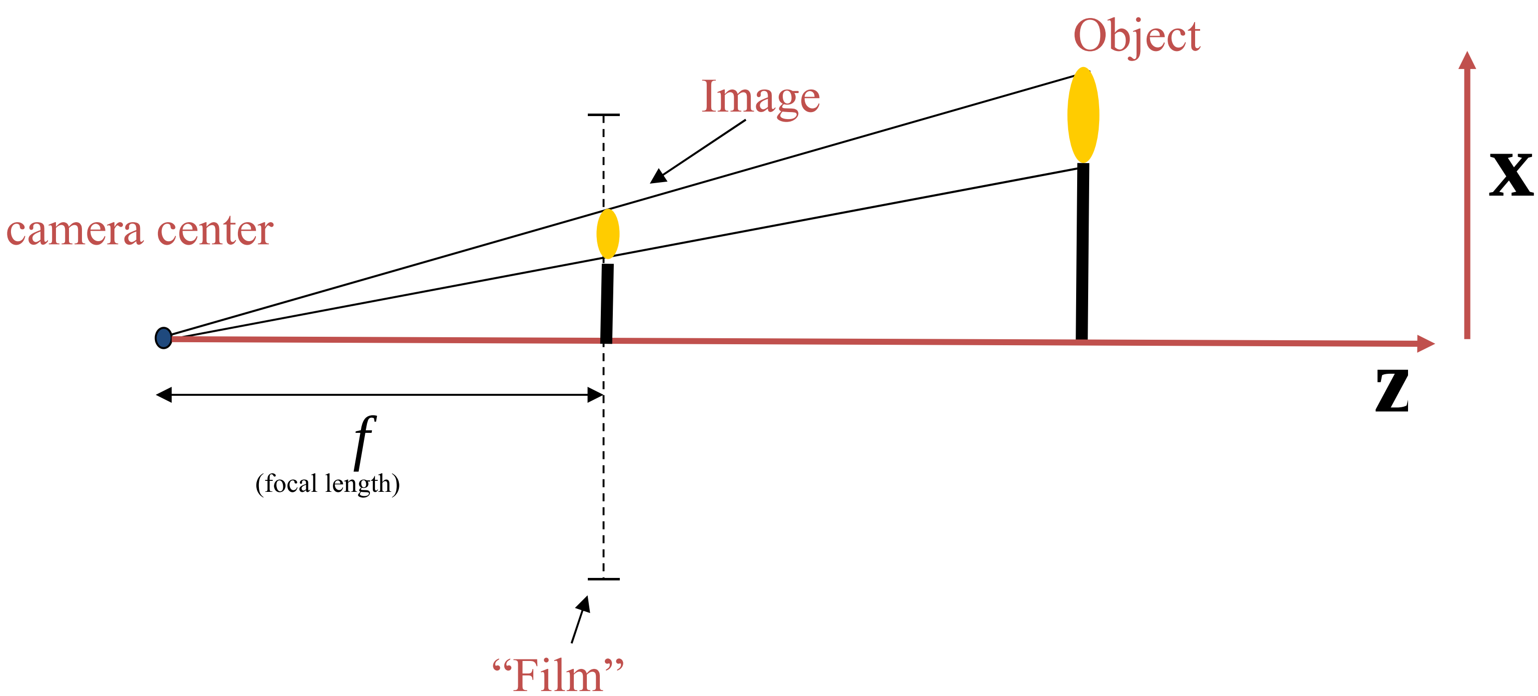
Based on slides by John Oliensis

Image Formation

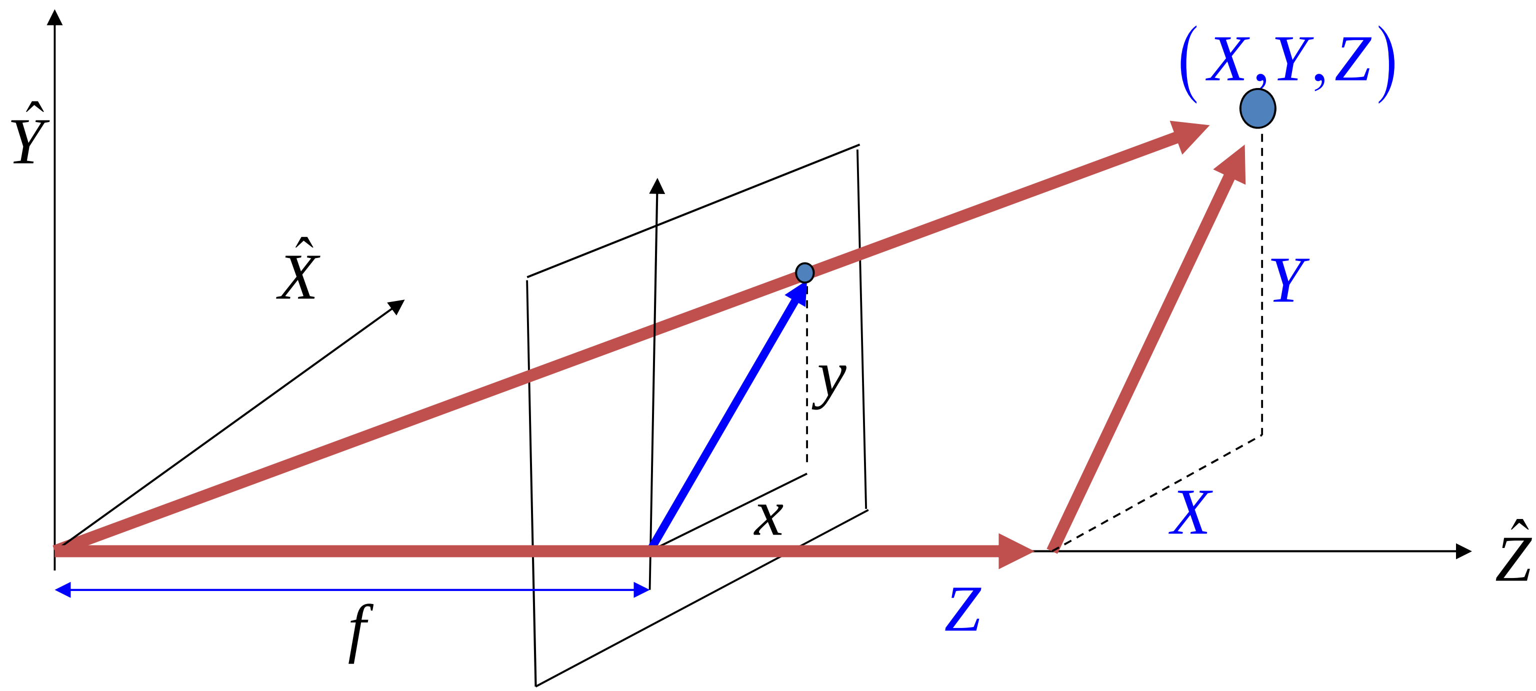


Projection Equation

- 2D world ➔ 1D image



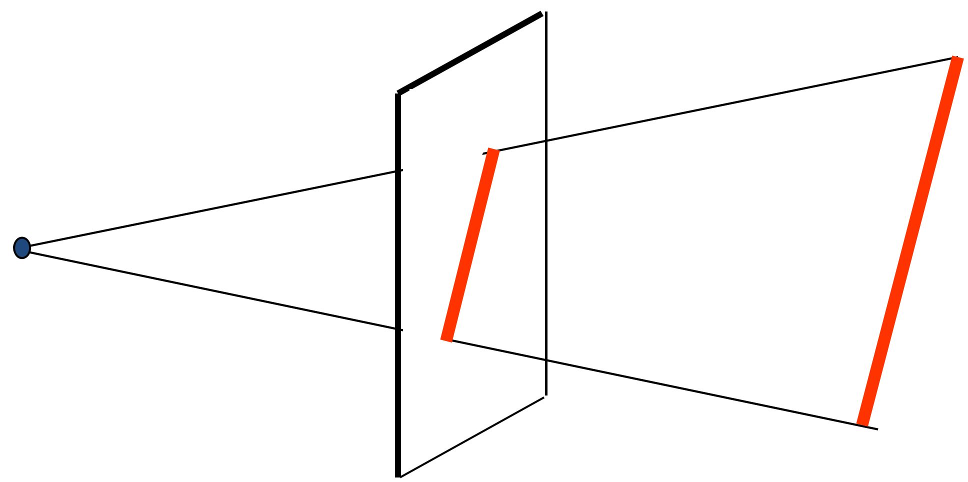
Projection Equation: 3D



Similar triangles: $\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z}$ $\longrightarrow$ $\boxed{(x, y) = \frac{f}{Z}(X, Y)}$

Perspective Projection: Properties

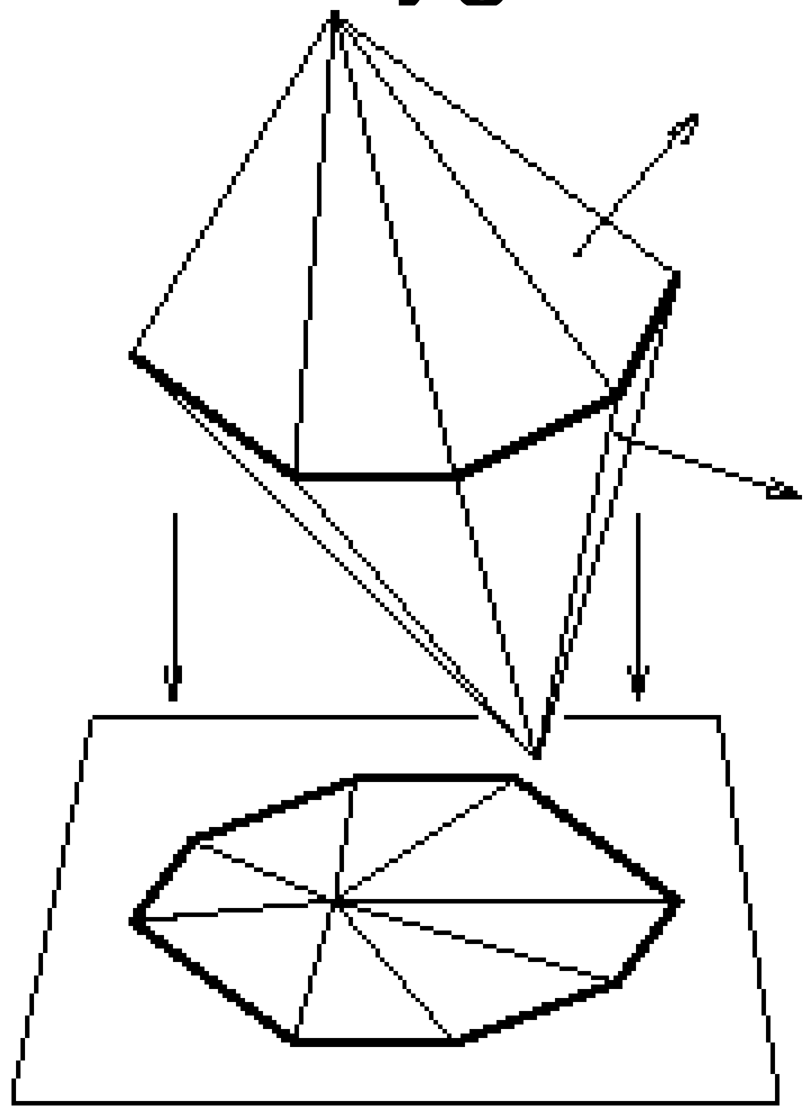
- 3D points ➔ image points
- 3D straight lines ➔ image straight lines



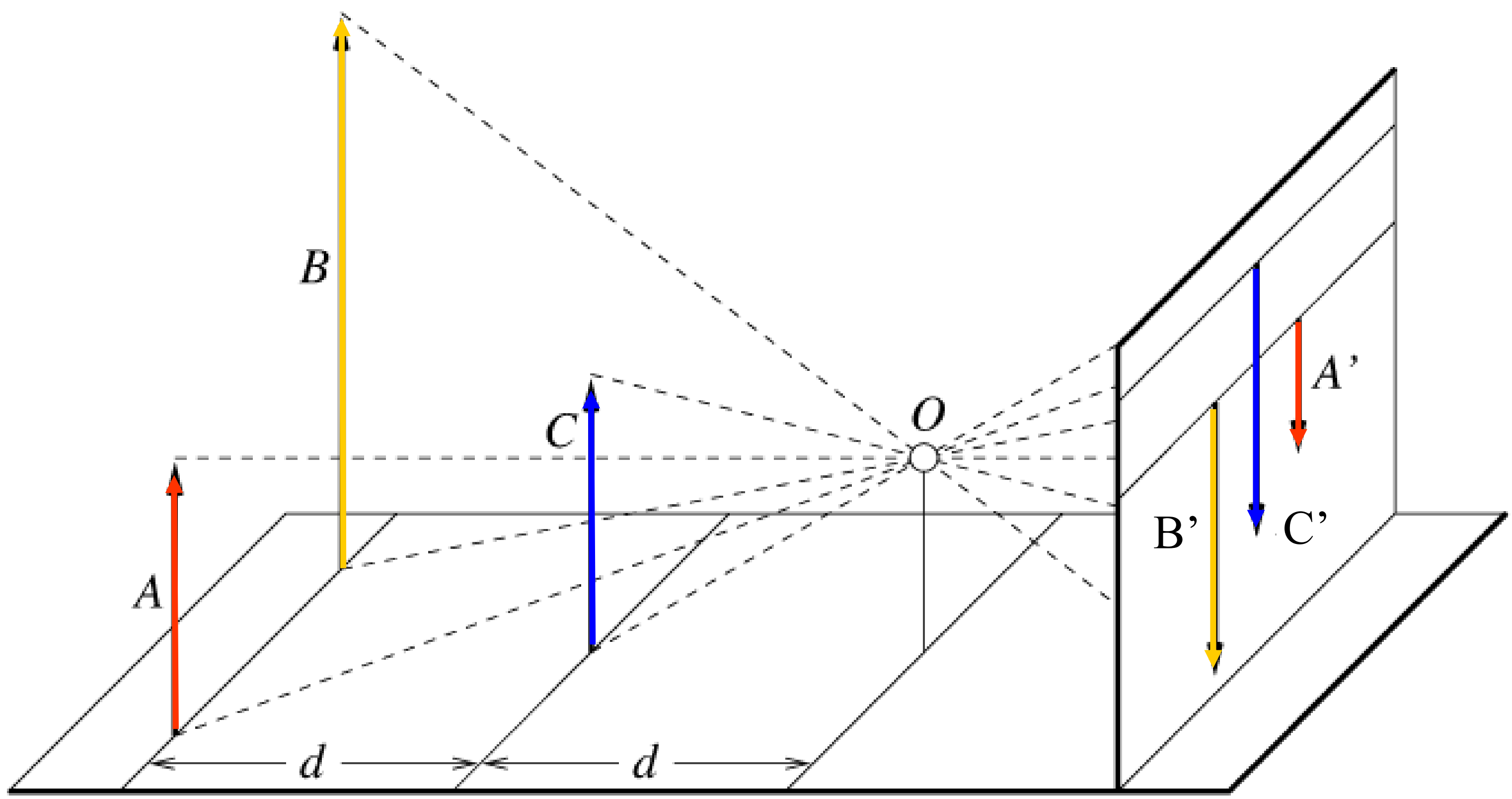
- 3D Polygons ➔ image polygons

Polyhedra Project to Polygons

(since lines project to lines)



Properties: Distant objects are smaller

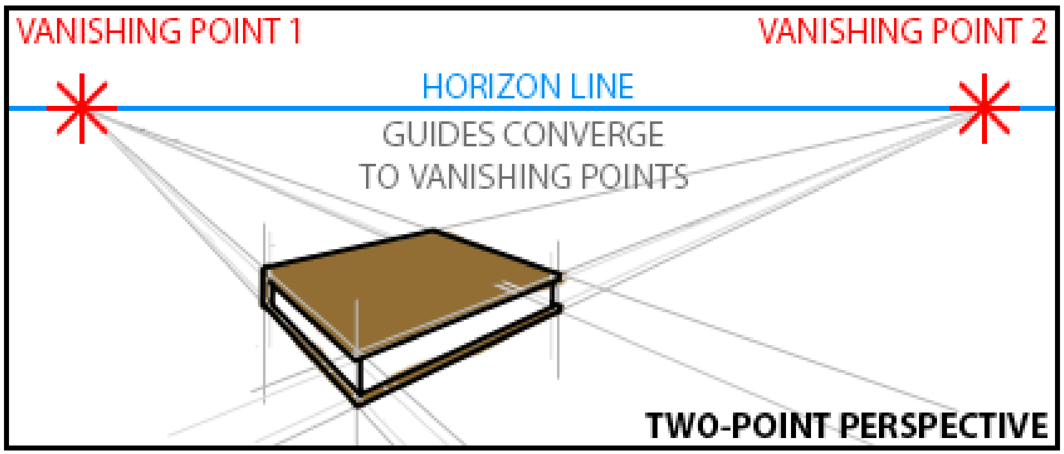


Properties: Vanishing Points

- Image of an infinitely distant 3D point

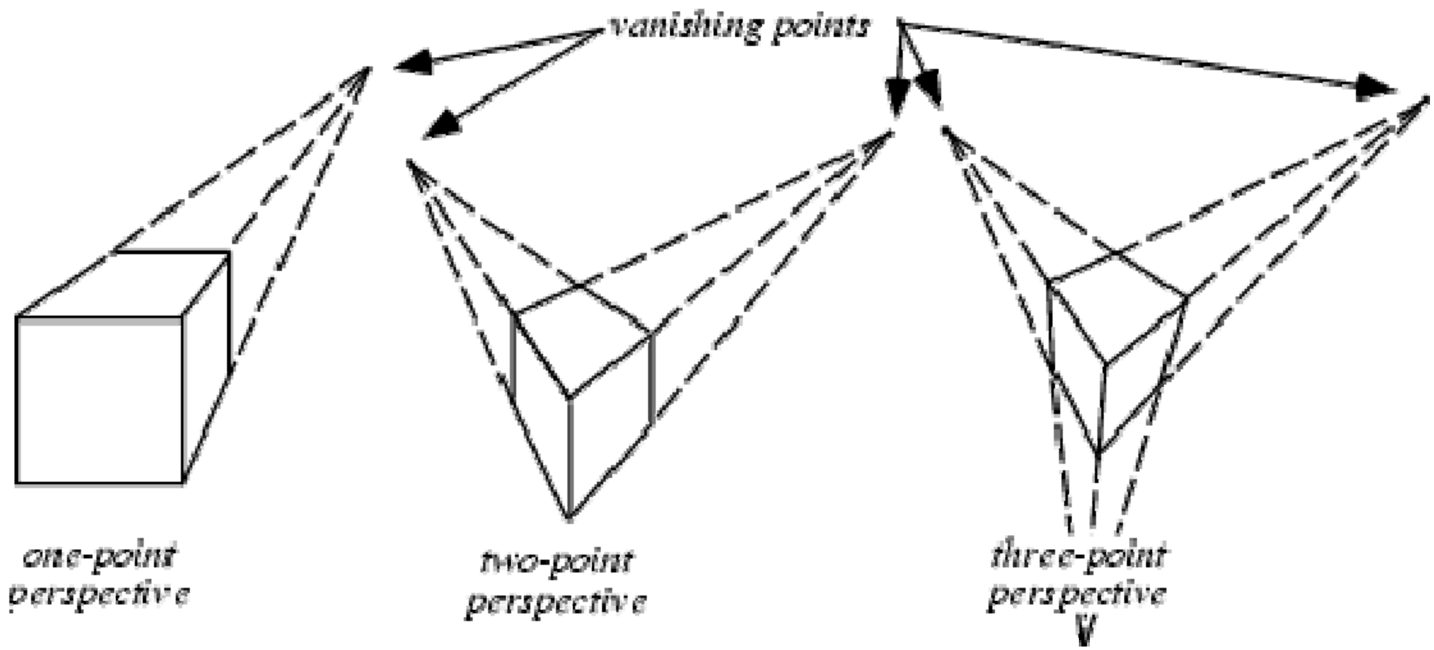
Vanishing Points + Horizon

- Vanishing point
  - Vanishing ray parallel to World Line
    ➔ gives *World Line's direction*



- Horizon: all vanishing points for World Lines in (or parallel to) plane.

Properties: Vanishing Points



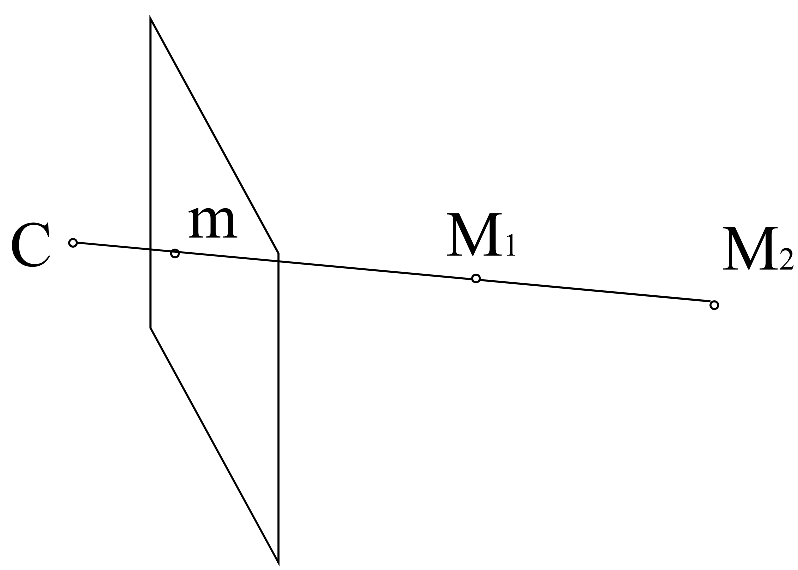
Single View Geometry

Richard Hartley and Andrew Zisserman
Marc Pollefeys

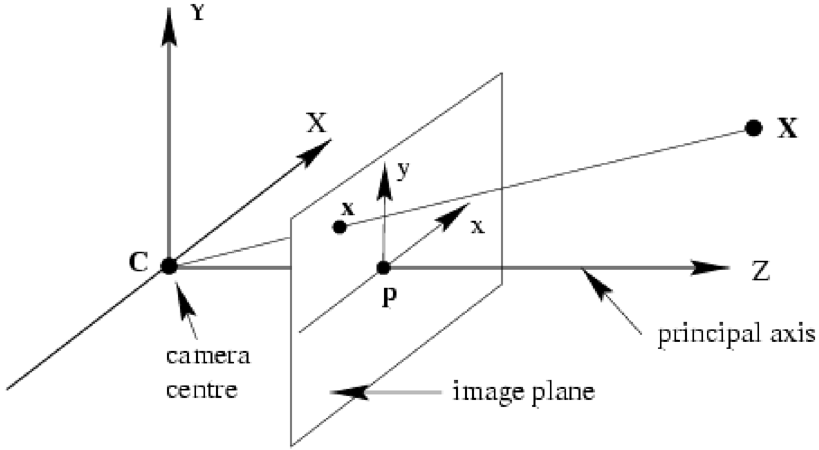
Modified by Philippos Mordohai

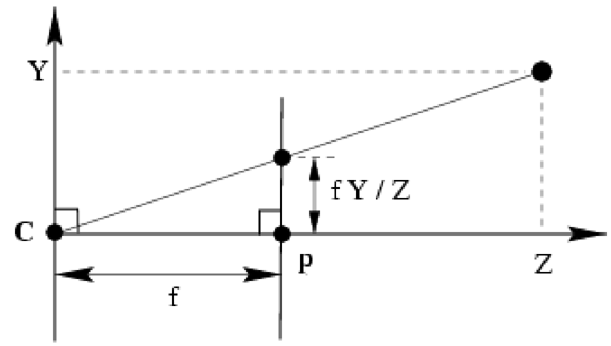
Homogeneous Coordinates

- 3-D points represented as 4-D vectors $(X\ Y\ Z\ 1)^T$
- Equality defined up to scale
  - $(X\ Y\ Z\ 1)^T \sim (WX\ WY\ WZ\ W)^T$
- Useful for perspective projection → makes equations linear



Pinhole camera model



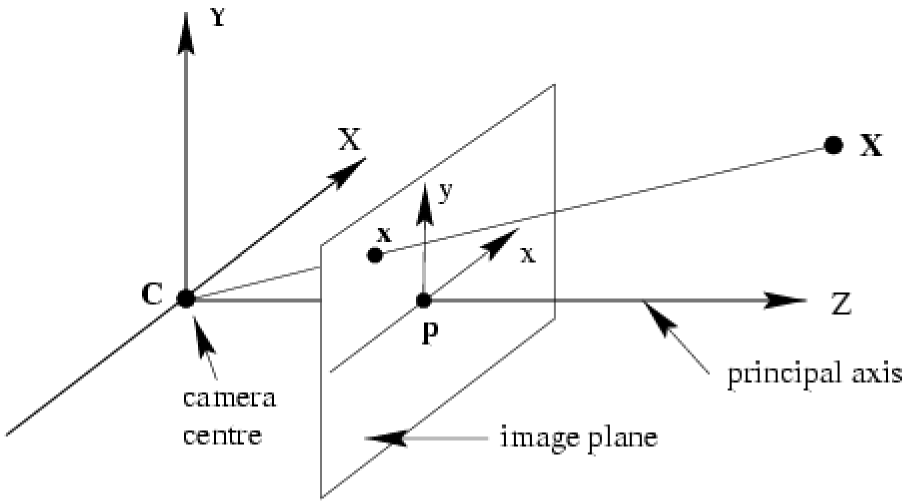


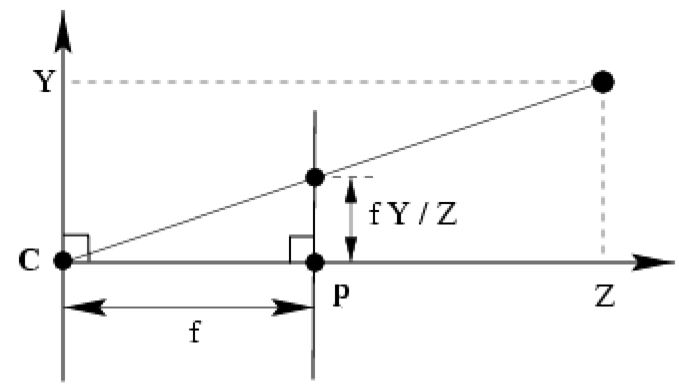
$$(X, Y, Z)^T \mapsto (fX / Z, fY / Z)^T$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

linear projection in homogeneous coordinates!

The Pinhole Camera



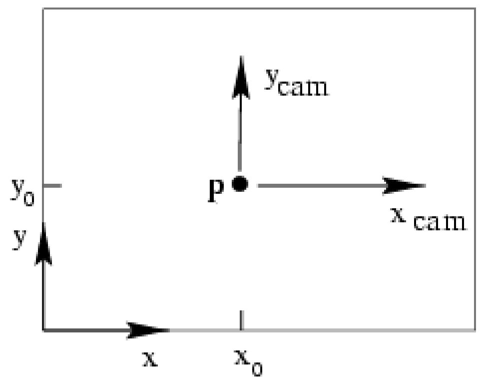


$$x = \frac{fX}{Z}$$

$$y = \frac{fY}{Z}$$

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Principal Point Offset

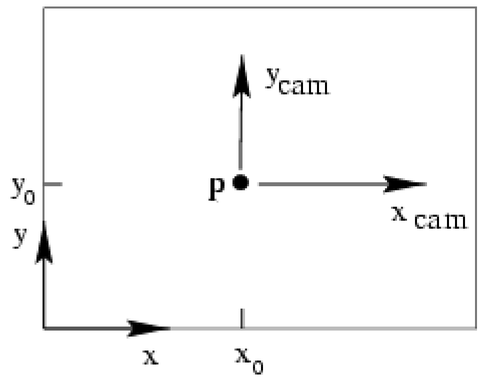


$$(X, Y, Z)^T \mapsto (fX/Z + p_x, fY/Z + p_y)^T$$

$(p_x, p_y)^T$ principal point

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal Point Offset



$$\mathrm{x} = \mathrm{K}\left[\mathrm{I} \mid 0\right]\mathrm{X}_{\mathrm{cam}}$$

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

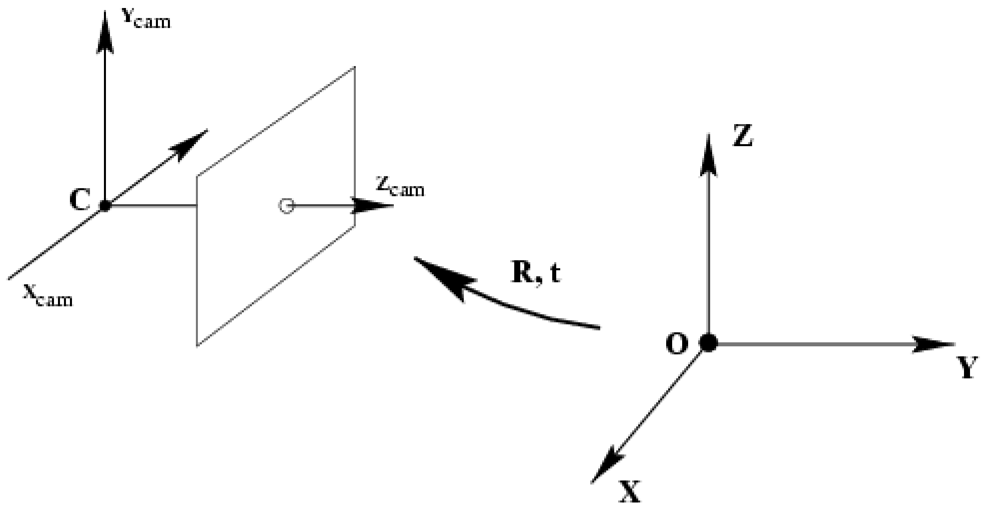
$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}$$ calibration matrix

Hands On: Image Formation

- For a 640 by 480 image with focal length equal to 640 pixels, find 3D points that are marginally visible at the four borders of the image

- Increase and decrease the focal length. What happens?

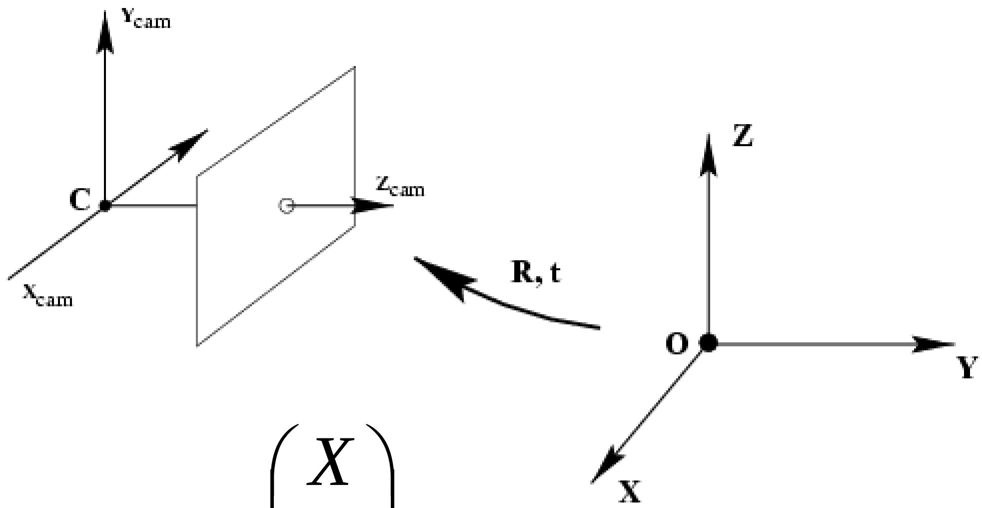
Camera Rotation and Translation



$$\tilde{\mathrm{X}}_{\mathrm{cam}} = \mathrm{R}\left(\tilde{\mathrm{X}} - \tilde{\mathrm{C}}\right)$$

$$\mathrm{X}_{\mathrm{cam}} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\tilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} \mathrm{R} & -\mathrm{R}\tilde{\mathrm{C}} \\ 0 & 1 \end{bmatrix} \mathrm{X}$$

Camera Rotation and Translation



$$X_{cam} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\widetilde{C} \\ 0 & 1 \end{bmatrix} X$$

$$x = PX$$

$$P = K[R \mid t]$$

$$x = K[I \mid 0] X_{cam}$$

$$t = -R\widetilde{C}$$

$$x = KR[I \mid -\widetilde{C}] X$$

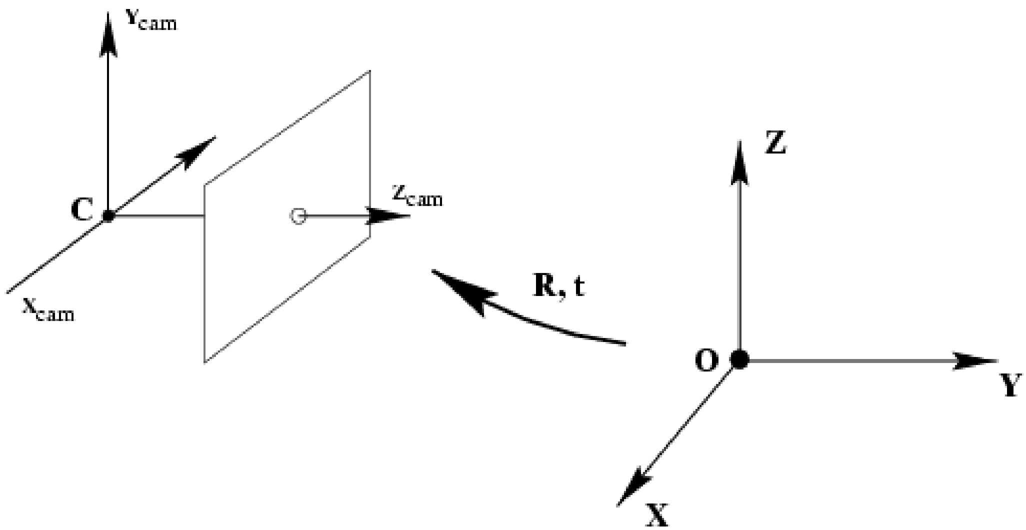
Intrinsic Parameters

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \quad \text{or} \quad \mathbf{K} = \begin{bmatrix} af & f\cos(s) & u_o \\ & f & v_o \\ & & 1 \end{bmatrix}$$

- Camera deviates from pinhole
  - *s*: skew
  - $f_x \neq f_y$: different magnification in *x* and *y*
  - $(c_x\, c_y)$: optical axis does not pierce image plane exactly at the center
- Usually:
  - rectangular pixels: $s = 0$
  - square pixels: $f_x = f_y$
  - principal point known: $\left(c_x, c_y\right) = \left(\frac{w}{2}, \frac{h}{2}\right)$

$$\mathbf{K} = \begin{bmatrix} \gamma f & sf & x_0 \\ & f & y_0 \\ & & 1 \end{bmatrix}$$

Extrinsic Parameters



Scene motion

$$M = \begin{bmatrix} R_{(3x3)} & t_{(3x1)} \\ 0_{(1x3)} & 1 \end{bmatrix}$$

Camera motion

$$M' = \begin{bmatrix} R^T{}_{(3x3)} & -(R^T t)_{3x1} \\ 0_{(1x3)} & 1 \end{bmatrix}$$

Projection matrix

- Includes coordinate transformation and camera intrinsic parameters

$$\lambda\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- Everything we need to know about a pinhole camera
- Unambiguous
- Can be decomposed into parameters

Projection matrix

- Mapping from 2-D to 3-D is a function of internal and external parameters

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left[R^\top \mid -R^\top t \right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\lambda x = K \left[R^\top \mid -R^\top t \right] X$$

$$\lambda x = PX$$

Hands On: Camera Motion

- Choose a few 3D points visible to a camera at the origin. (f=500, w=500, h=500)
- Now, move the camera by 2 units of length on the z axis. What happens to the images of the points?
- Rotate the points by 45 degrees about the z axis of the camera and then translate them by 5 units on the z axis away from the camera. What are the new images of the points?