Machine Learning for Computer Vision

Geometry of Vision

- Geometric model of a pinhole camera
- Geometric relation between 3D point coordinates in real world and 2D coordinates in images
- Homogeneous coordinate system
- Planar image transformations
  - parameters, invariants and matrix formulations
- Stereo vision

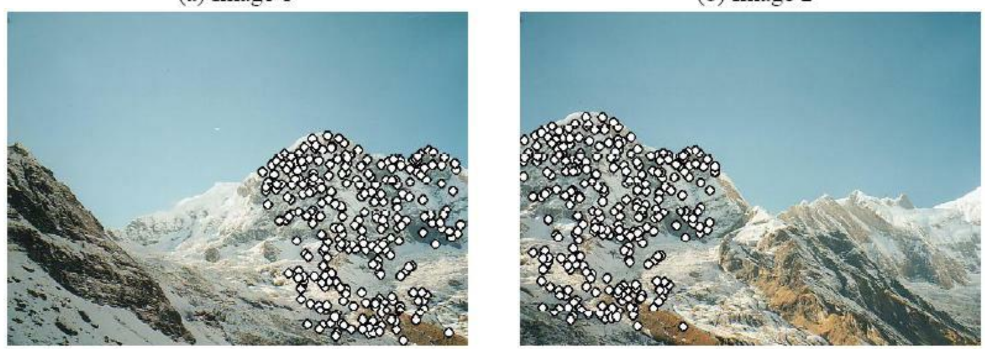
Applications: Building image panoramas

(a) Image 1

(b) Image 2

Applications: Building image panoramas

(a) Image 1 (b) Image 2



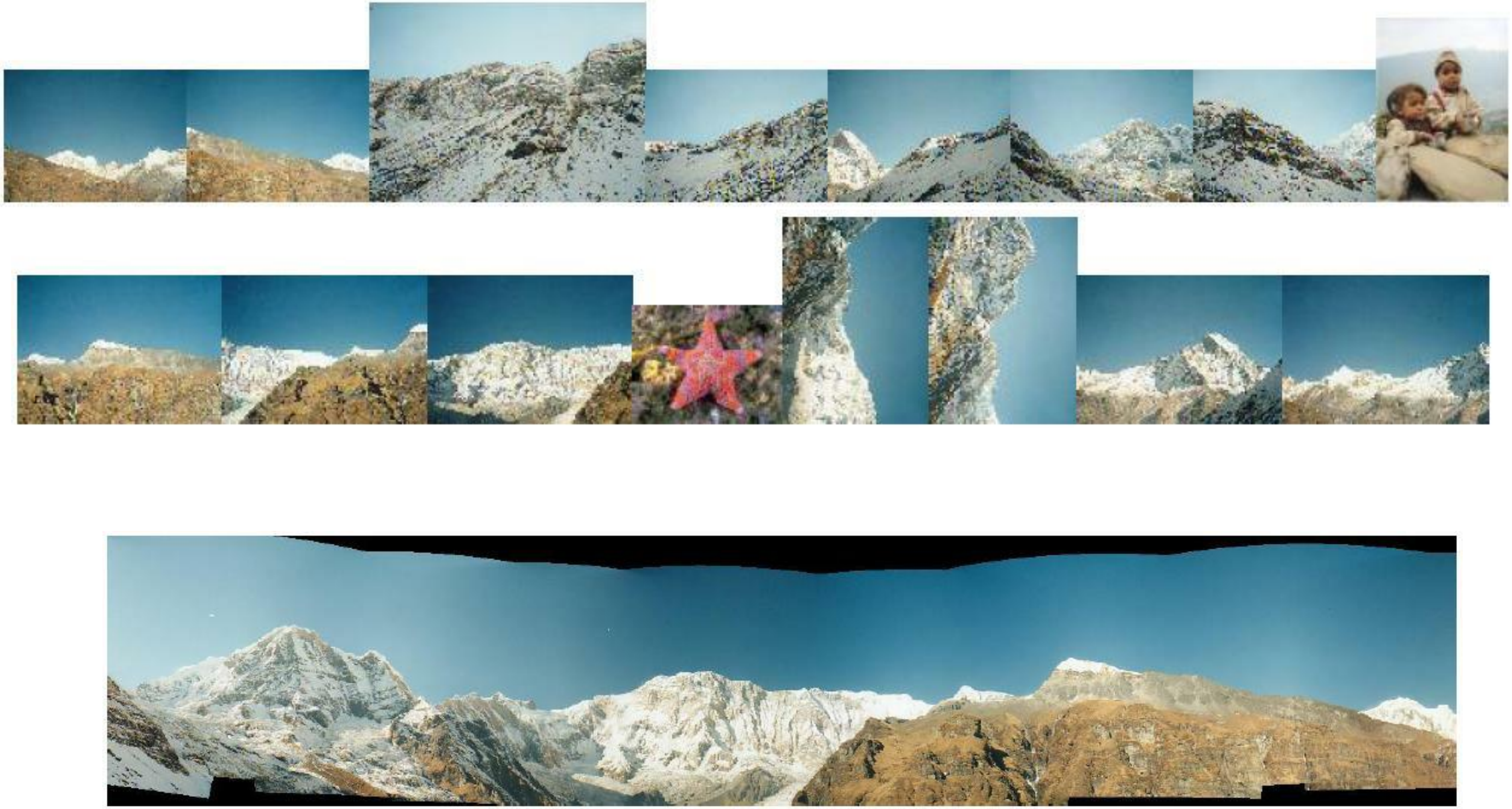
Applications: Building image panoramas

(a) Image 1

(b) Image 2

Matching points and finding transformation between images

Applications: Building image panoramas



Applications: Building image panoramas

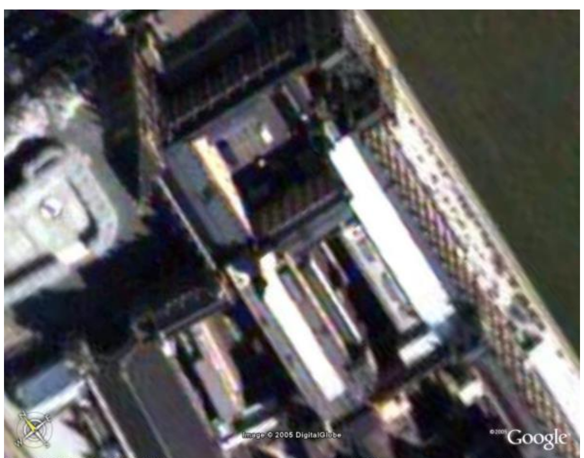
(registering many images into a common frame)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Applications: map building

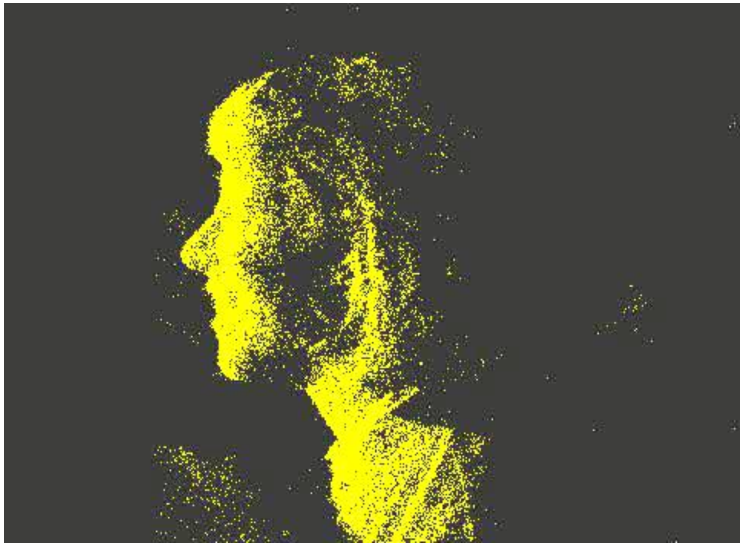
View 1

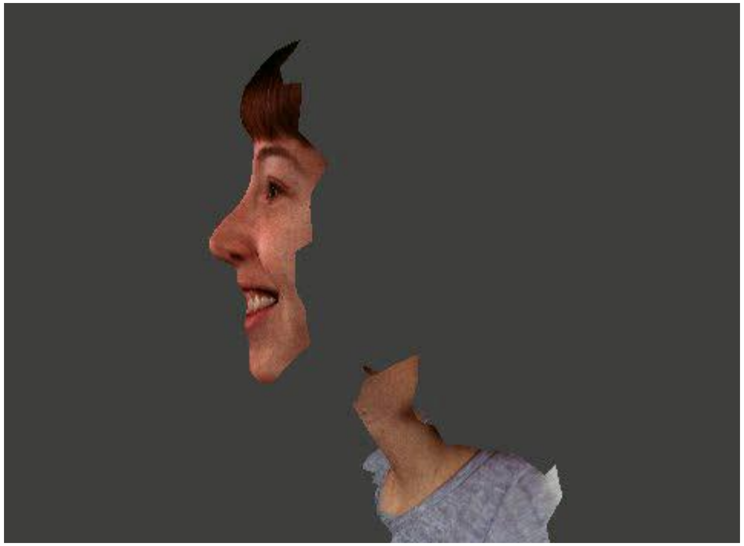
View 2



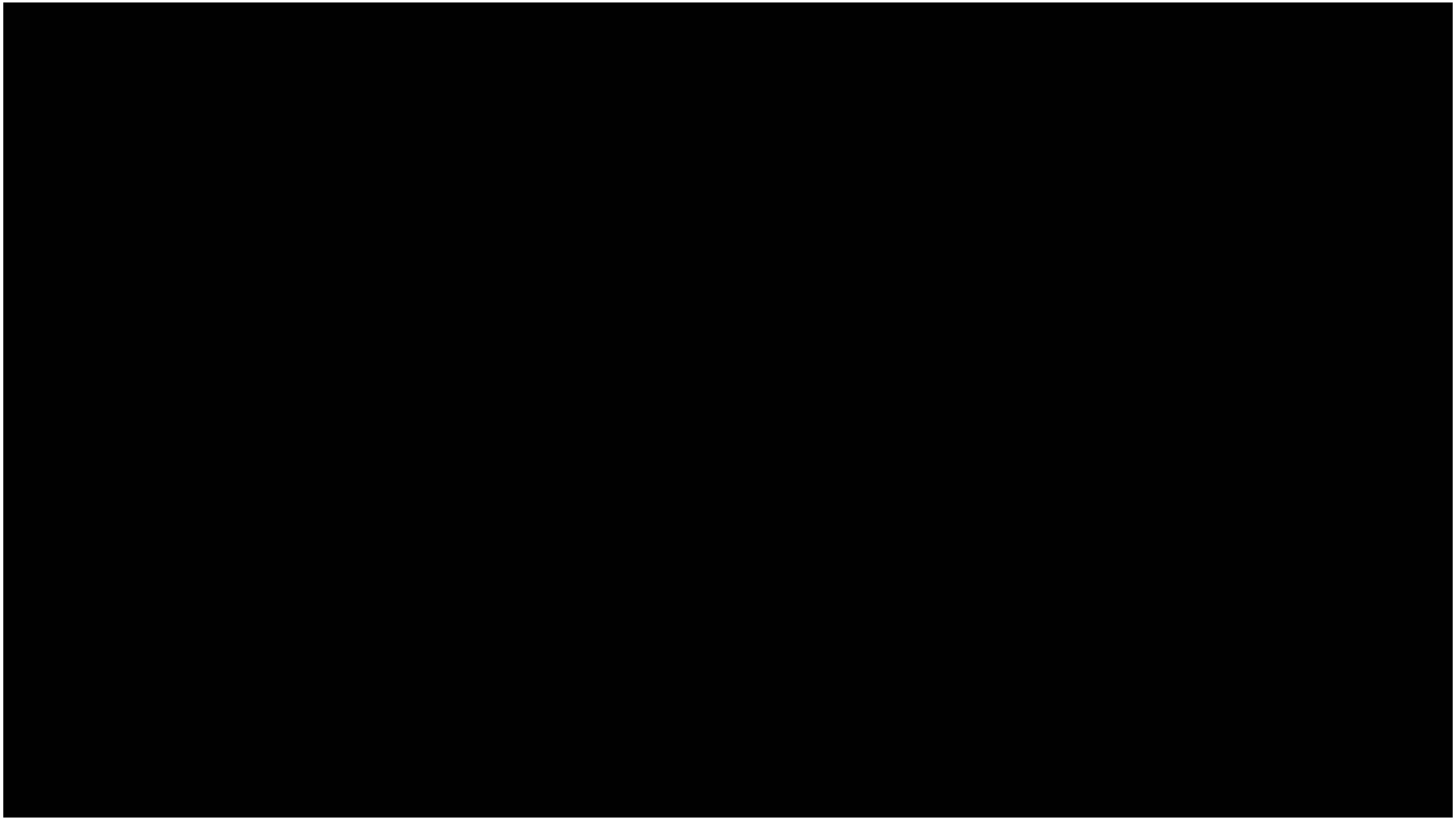
View 3

Example 3: 3D Reconstruction: Detect Correspondences and triangulate





Structure from Motion

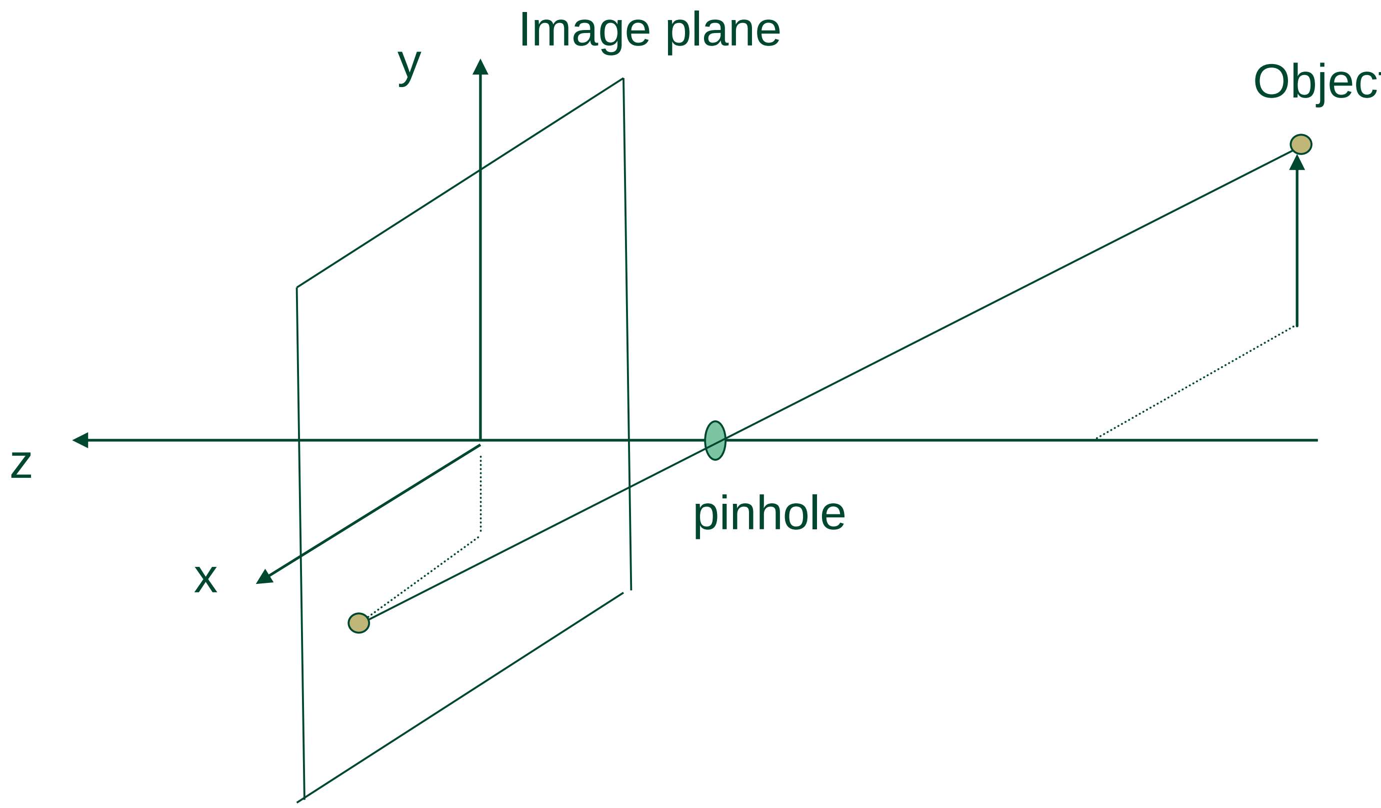


SLAM – Simultanous Localization and Mapping

Augmented Reality

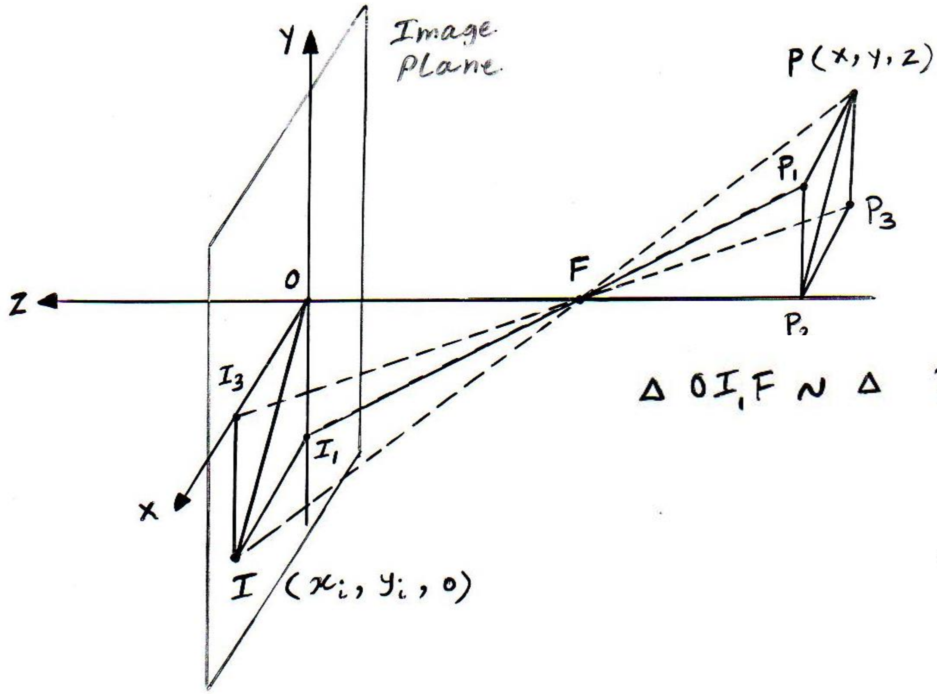
Stereo and motion capture

Pinhole camera model



f - focal length of the camera

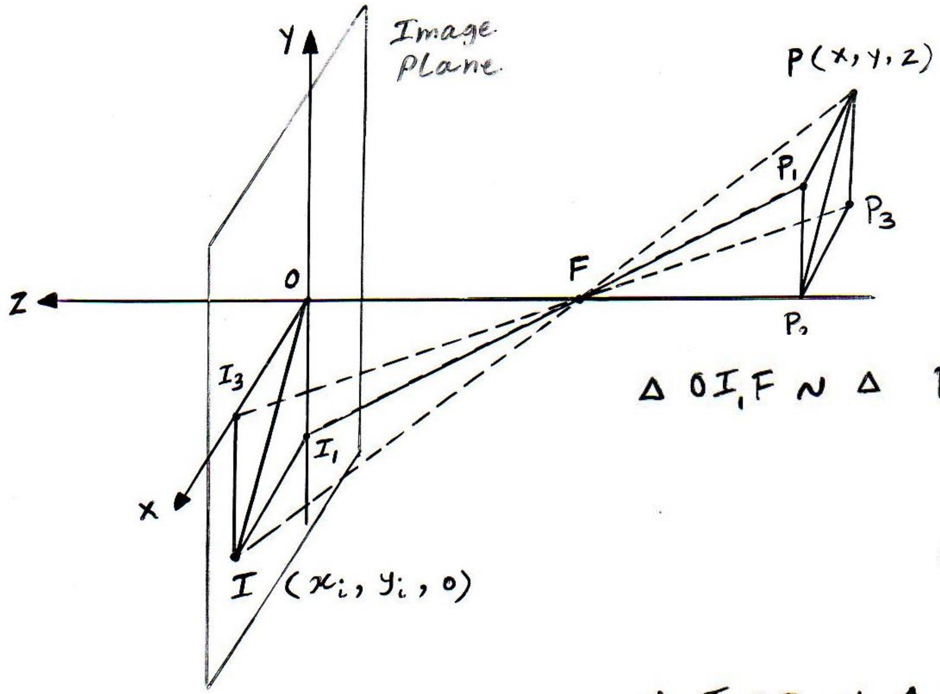
Pinhole camera model



$$\Delta\, OI_1F \sim \Delta\, P_2P_1F \Rightarrow \frac{OI_1}{OF} = \frac{P_2P_1}{P_2F} \Rightarrow \frac{-y_i}{f} = \frac{y}{-z-f}$$

$$\Rightarrow y_i = \frac{yf}{z+f} \qquad (1)$$

Pinhole camera model



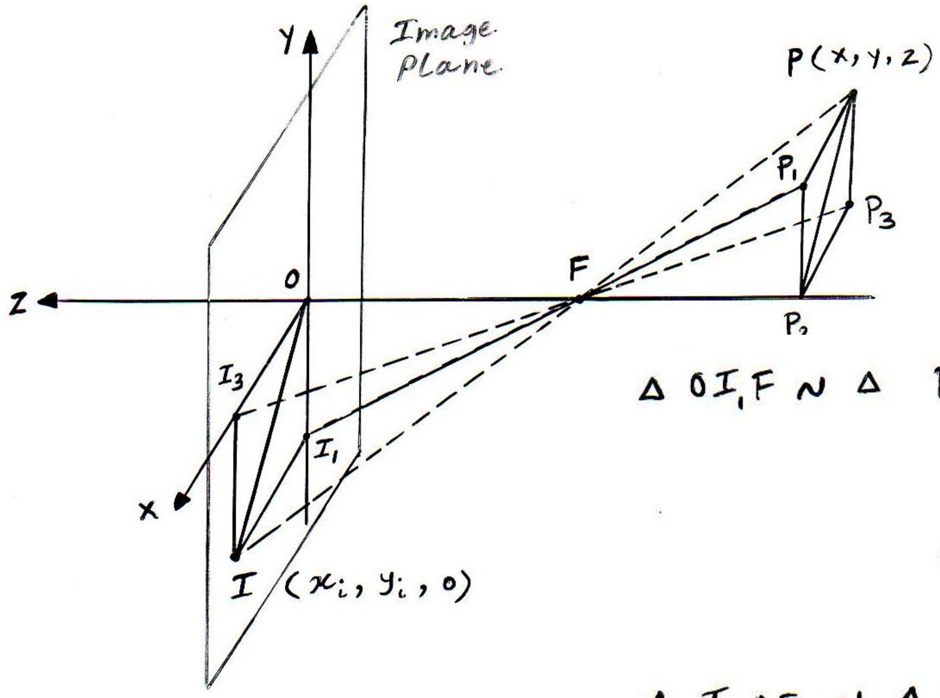
$$\Delta\, OI_1F \sim \Delta\, P_2P_1F \Rightarrow \frac{OI_1}{OF} = \frac{P_2P_1}{P_2F} \Rightarrow \frac{-y_i}{f} = \frac{y}{-z-f}$$

$$\Rightarrow y_i = \frac{yf}{z+f} \qquad (1)$$

$$\Delta\, I_3OF \sim \Delta\, P_3P_2F \Rightarrow \frac{I_3O}{OF} = \frac{P_3P_2}{P_2F} \Rightarrow \frac{x_i}{f} = \frac{-x}{-z-f}$$

$$\Rightarrow x_i = \frac{xf}{z+f} \qquad (2)$$

Pinhole camera model



$$\Delta\, OI_1F \sim \Delta\, P_2P_1F \Rightarrow \frac{OI_1}{OF} = \frac{P_2P_1}{P_2F} \Rightarrow \frac{-y_i}{f} = \frac{y}{-z-f}$$

$$\Rightarrow y_i = \frac{yf}{z+f} \qquad (1)$$

$$\Delta\, I_3OF \sim \Delta\, P_3P_2F \Rightarrow \frac{I_3O}{OF} = \frac{P_3P_2}{P_2F} \Rightarrow \frac{x_i}{f} = \frac{-x}{-z-f}$$

$$\Rightarrow x_i = \frac{xf}{z+f} \qquad (2)$$

Equations (1) and (2): Given position of object point, return position of the corresponding image point.

$$x = \frac{x_i(z+f)}{f} \quad (3) \qquad y = \frac{y_i(z+f)}{f} \quad (4)$$

Pinhole camera model

* Position of a 3D point : (x, y, z)
* Homogeneous coordinates of same point :

$$(Kx, Ky, Kz, K)$$

* To recover actual coordinates :

Divide first three coordinates by the fourth coordinate.

* Let : $K = \dfrac{z+f}{f}$

Pinhole camera model

* Let : $K = \dfrac{z+f}{f}$

— Augmented position vector of image point:

$$V_i = (x, y, z, K)^T$$

— Augmented position vector of object point:

$$V = (x, y, z, 1)^T$$

Pinhole camera model

* Let : $K = \dfrac{z+f}{f}$

— Augmented position vector of image point :

$$V_i = (x, y, z, K)^T$$

— Augmented position vector of object point :

$$V = (x, y, z, 1)^T$$

* We now have a Linear relationship between the two :

$$V_i = P V$$

where

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix}.$$

Pinhole camera model

– Verification:

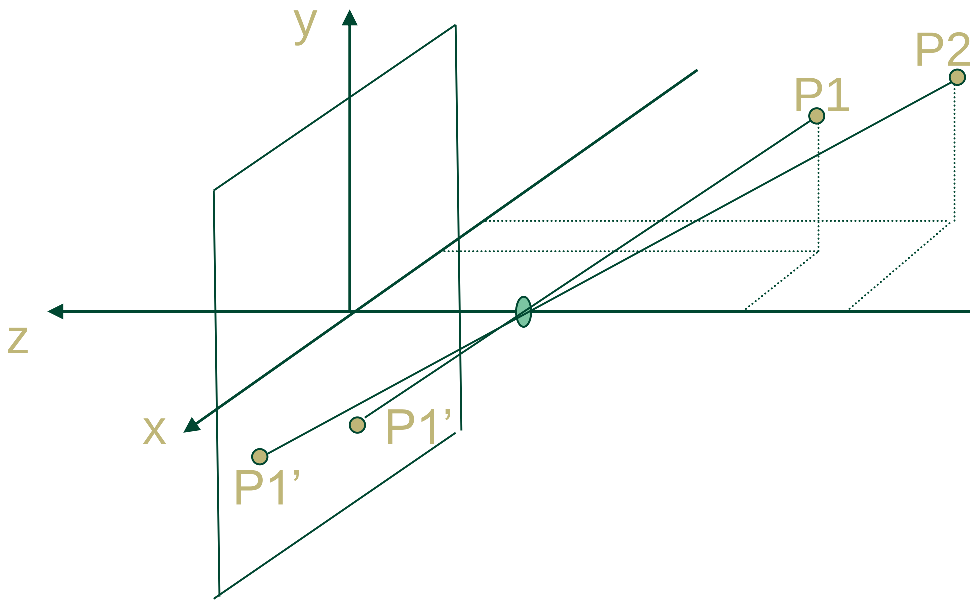
$$\begin{pmatrix} x \\ y \\ z \\ k \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \Rightarrow \begin{aligned} x &= X \\ y &= Y \\ z &= Z \\ k &= \frac{Z}{f} + 1 \end{aligned}$$

– Divide first 3 by 4th:

$$\left(\underbrace{\frac{Xf}{Z+f}}_{x_i}, \underbrace{\frac{Yf}{Z+f}}_{y_i}, \underbrace{\frac{Zf}{Z+f}}_{\text{disregard}} \right)$$

Pinhole camera model

- P1 : x=10 y=5 z=20
- P2 : x=20 y=10 z=10
- f=2



- TASK: Calculate the distance between P1′ and P2′ in the image

Pinhole camera model

– Omit third coordinate altogether;

In general:

$$\begin{pmatrix} K x_i \\ K y_i \\ K \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

<u>Camera Calibration:</u>

The process of determining the values P_{ij}

Textbook: Hartley & Zisserman, Visual Geometry

Homogeneous coordinates

- Allow various image transformations to be easily represented by a matrix
  - Projective space
    - The ordinary plane augmented with points at infinity is known as the *projective plane*.
  - practical aspect of the homogeneous coordinate system is its unification of the translation, scaling and rotation of geometric objects.

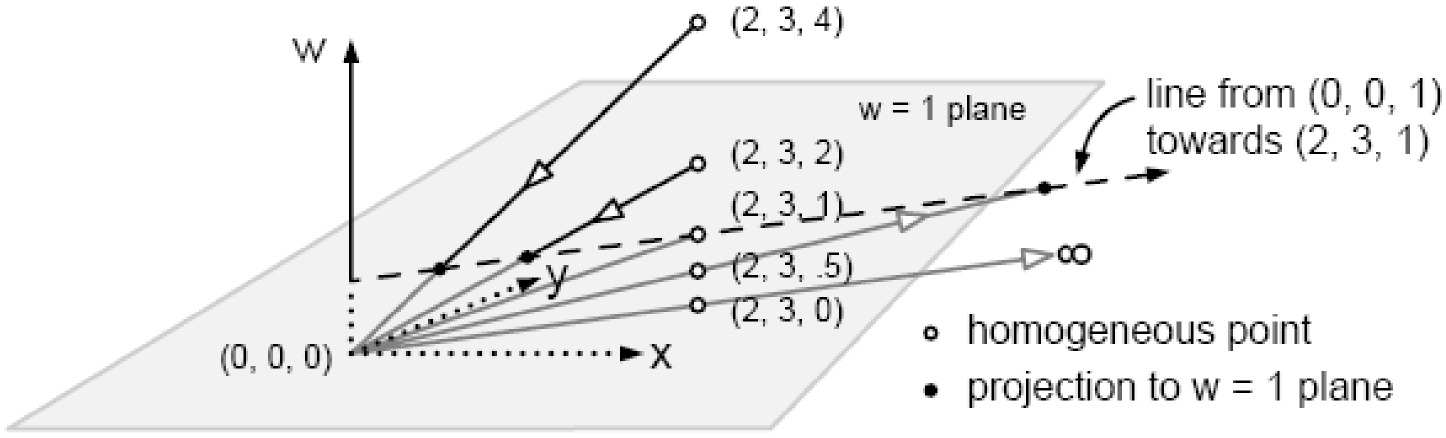
$(x : y : z : \ldots : w)$ is a row vector of length $n + 1$, other than $(0 : 0 : 0 : \ldots : 0)$, where n is the number of dimensions

Two sets of coordinates that are proportional and denote the same point of projective space: for any non-zero scalar c $(cx : cy : cz : \ldots : cw)$ denotes the same point.

The plane at infinity is usually identified with the set of points with $w = 0$.

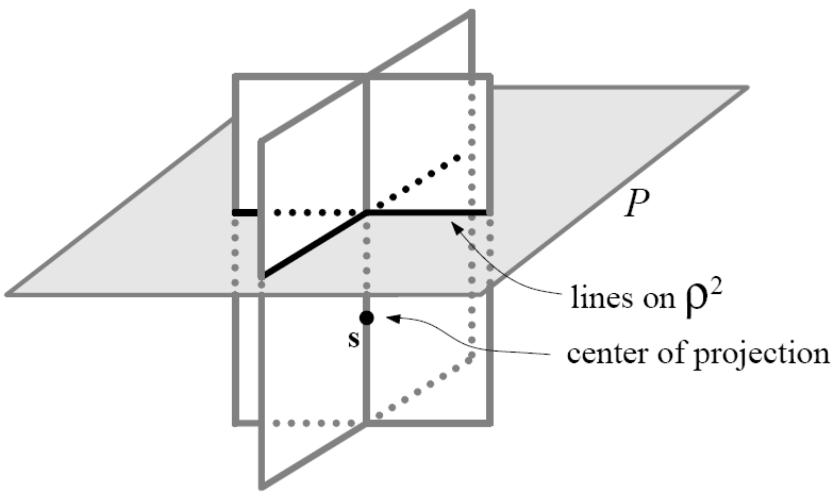
Homogeneous coordinates

- Homogeneous points represent projection, they can also represent points at infinity.
  - Consider a homogeneous point as *w* approaches 0; [2, 3, *w*] is shown for *w* = {4, 2, 1, 1/2, 0}. As *w* approaches 0, the projected Euclidean points move away from the origin in the (2, 3) direction. At *w* = 0, the point is infinitely far and may be treated as a positionless vector.



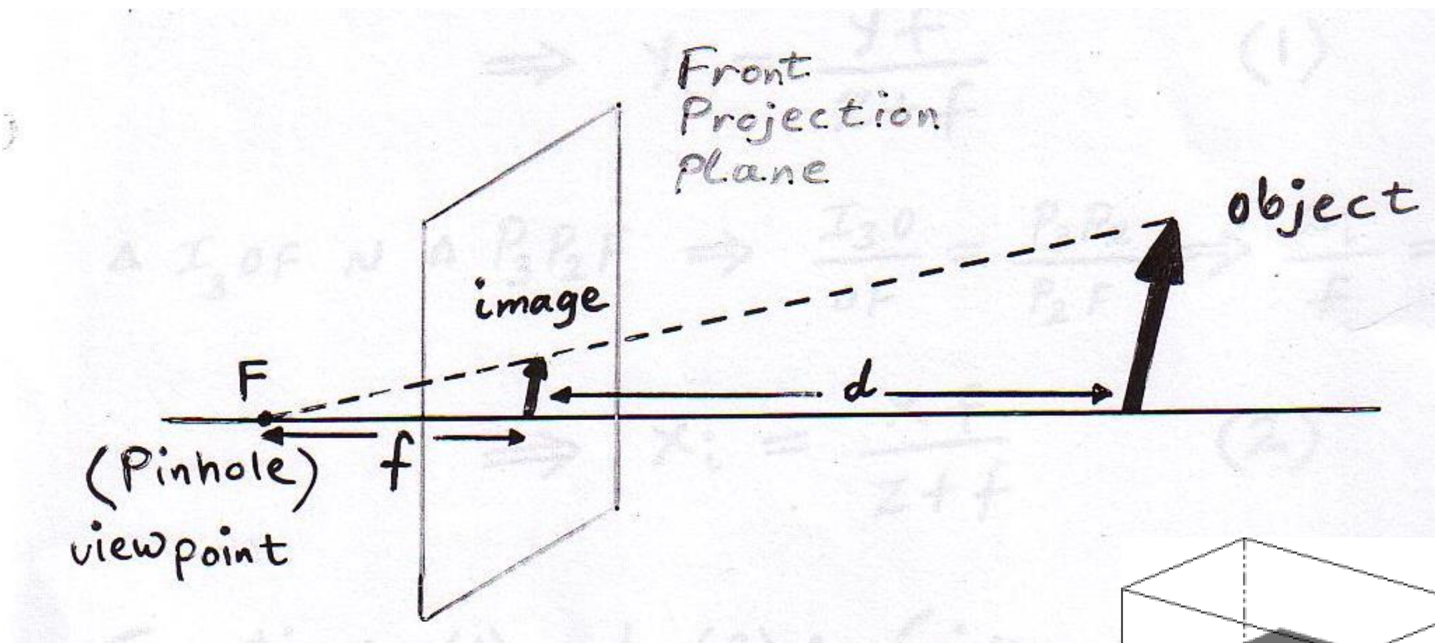
Homogeneous coordinates

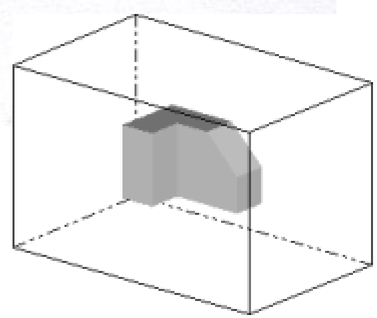
- The projective plane
  - consider all lines and planes passing through a given point s;
  - if they are intersected by a plane *P* that does not pass through s, then each point (or line) on *P* may be associated with a line (or plane) through s
  - The division by *w* means that the conversion of a homogeneous point to its Euclidean equivalent is inherently a projection of the homogenous point onto the *w* = 1 plane.



Orthographic projection

- Representing a three-dimensional object in two dimensions





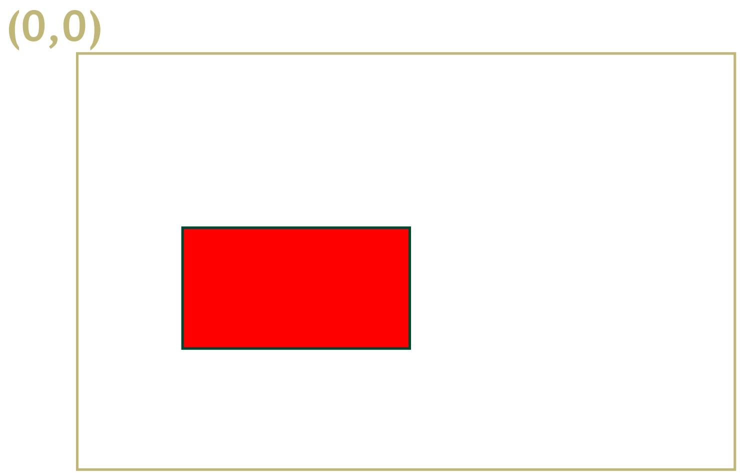
orthographic projection f>>d

Geometric transformations

- Planar transformations:
- Euclidean
- Similarity
- Affine
- Perspective

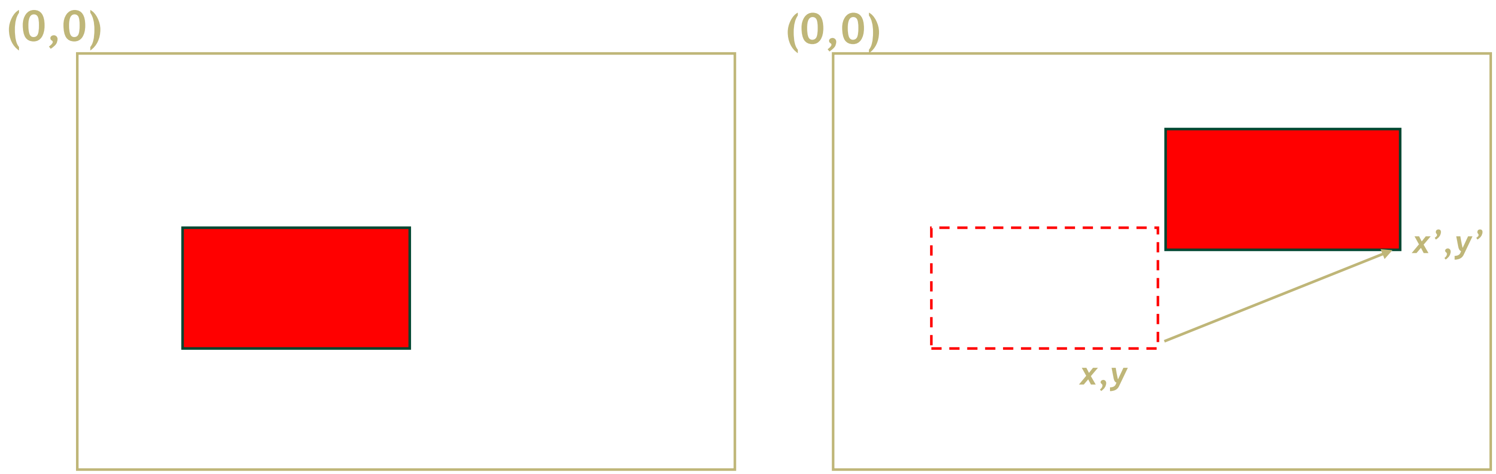
Geometric transformation

- Translation



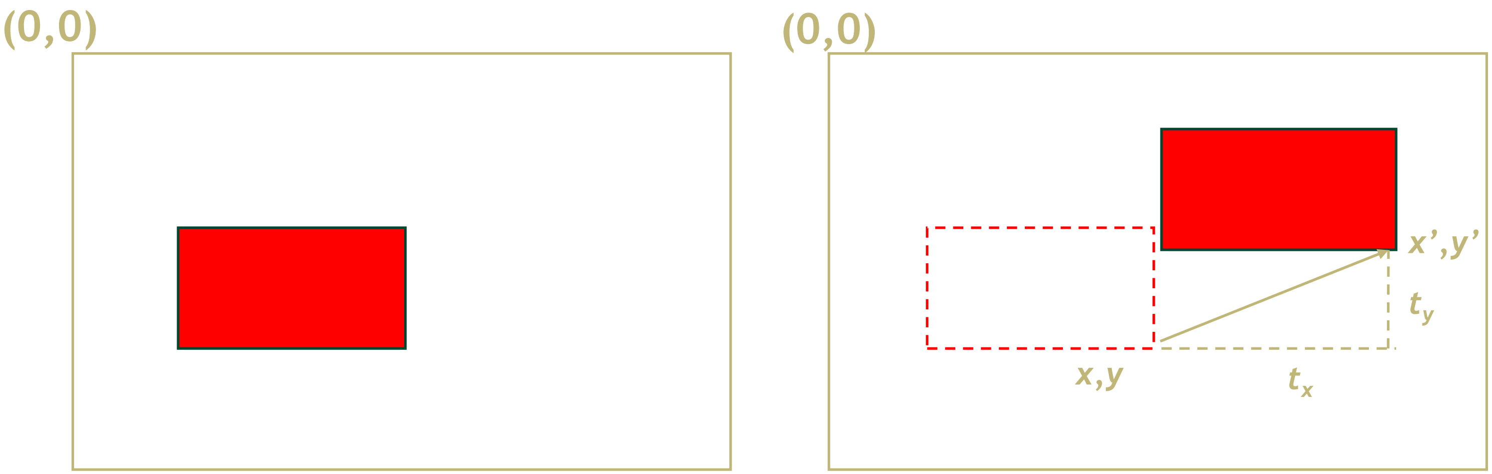
Geometric transformation

- Translation



Geometric transformation

- Translation



$$x' = x + t_x = 1 \cdot x + 0 \cdot y + t_x \cdot 1$$

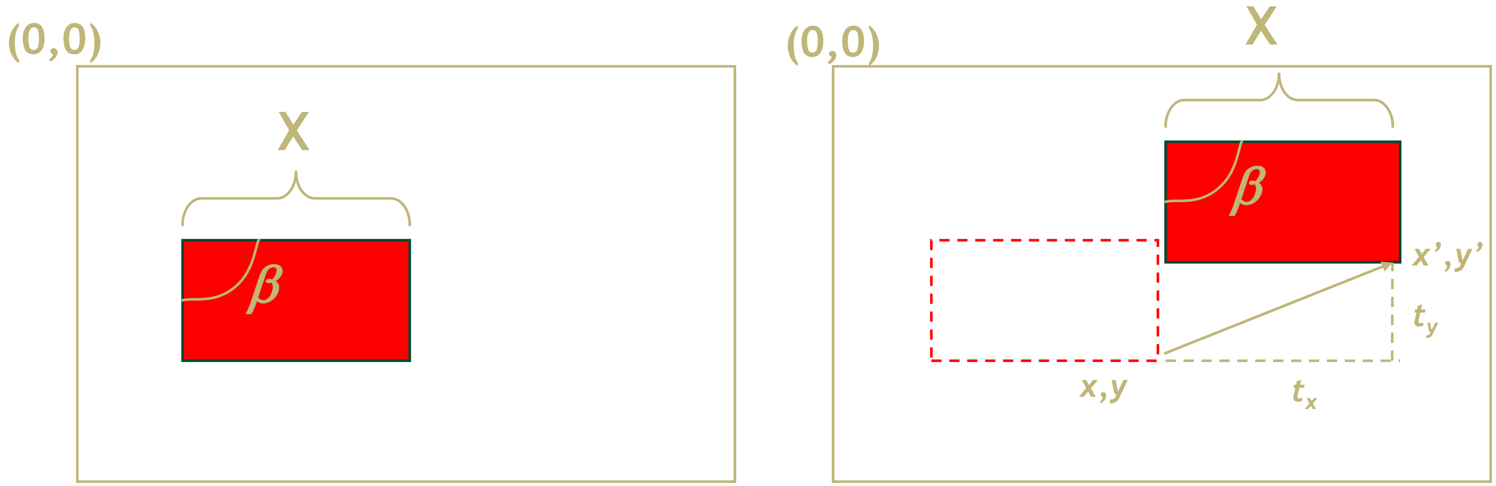
$$y' = y + t_y = 0 \cdot x + 1 \cdot y + t_y \cdot 1$$

In homogenous coordinates ⇨

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Geometric transformation

- Translation



Invariants
- orientation
- Length
- Angle
- Length ratio
- Parallel lines

$$x' = x + t_x = 1\cdot x + 0\cdot y + t_x\cdot 1$$

$$y' = y + t_y = 0\cdot x + 1\cdot y + t_y\cdot 1$$

In homogenous coordinates

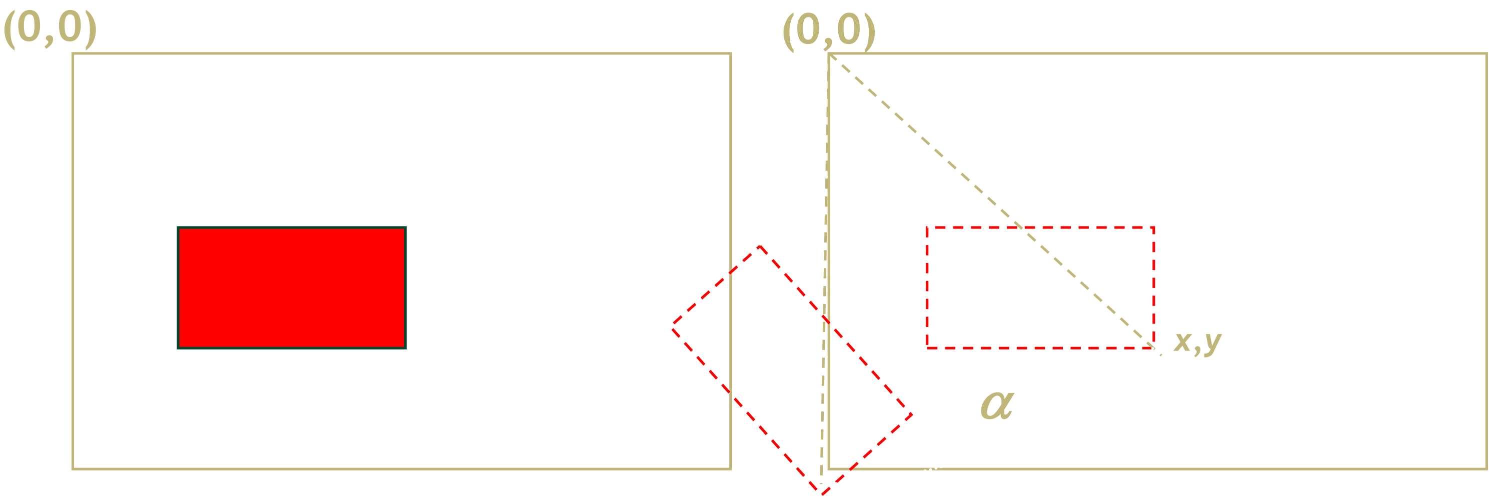
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix notation

$$\mathbf{x}' = \mathbf{x} + \mathbf{t}$$

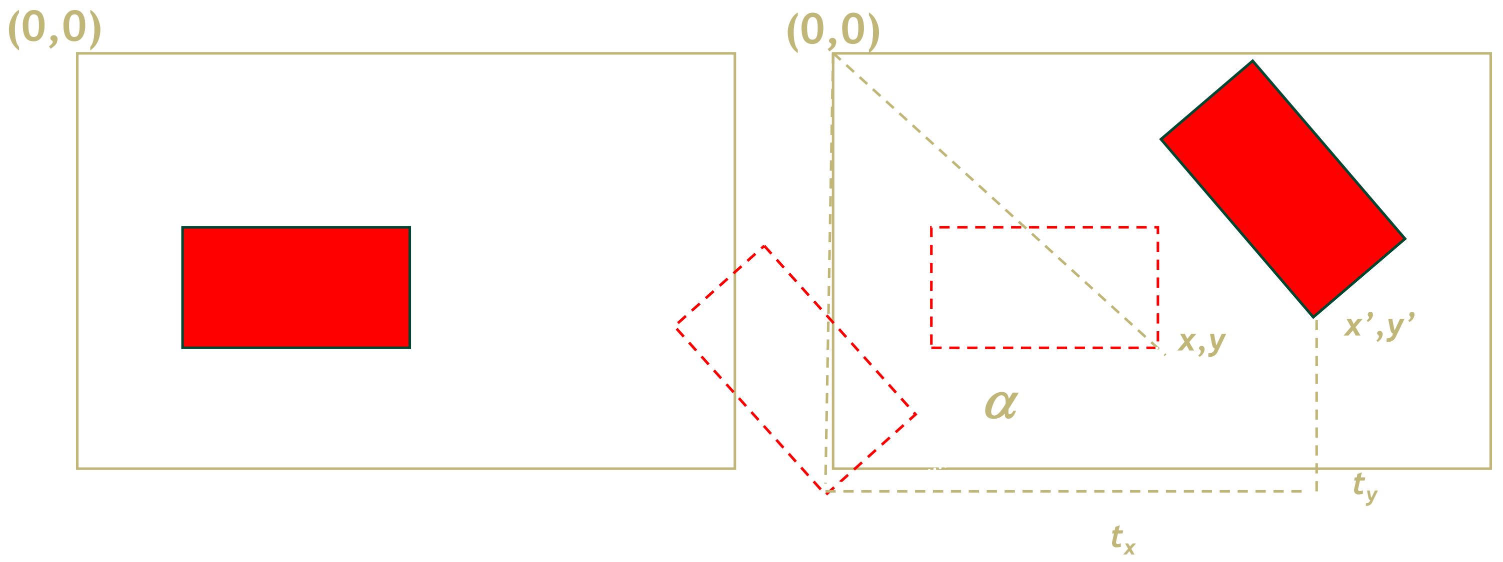
Geometric transformation

- Rotation



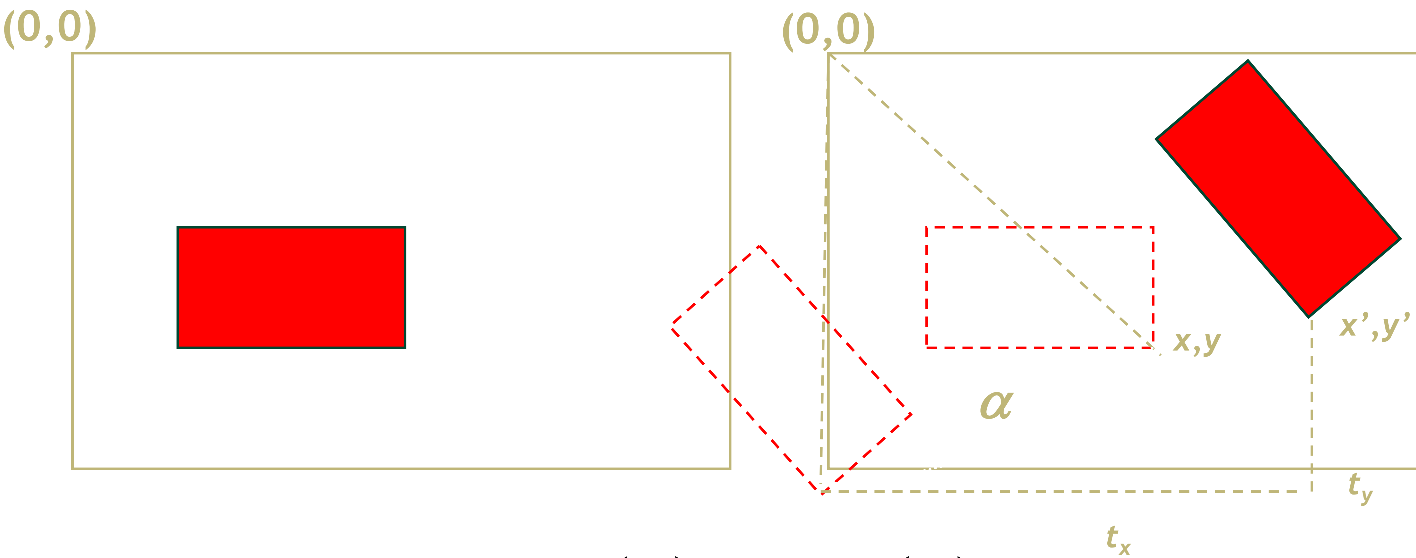
Geometric transformation

- Rotation



Geometric transformation

- Rotation

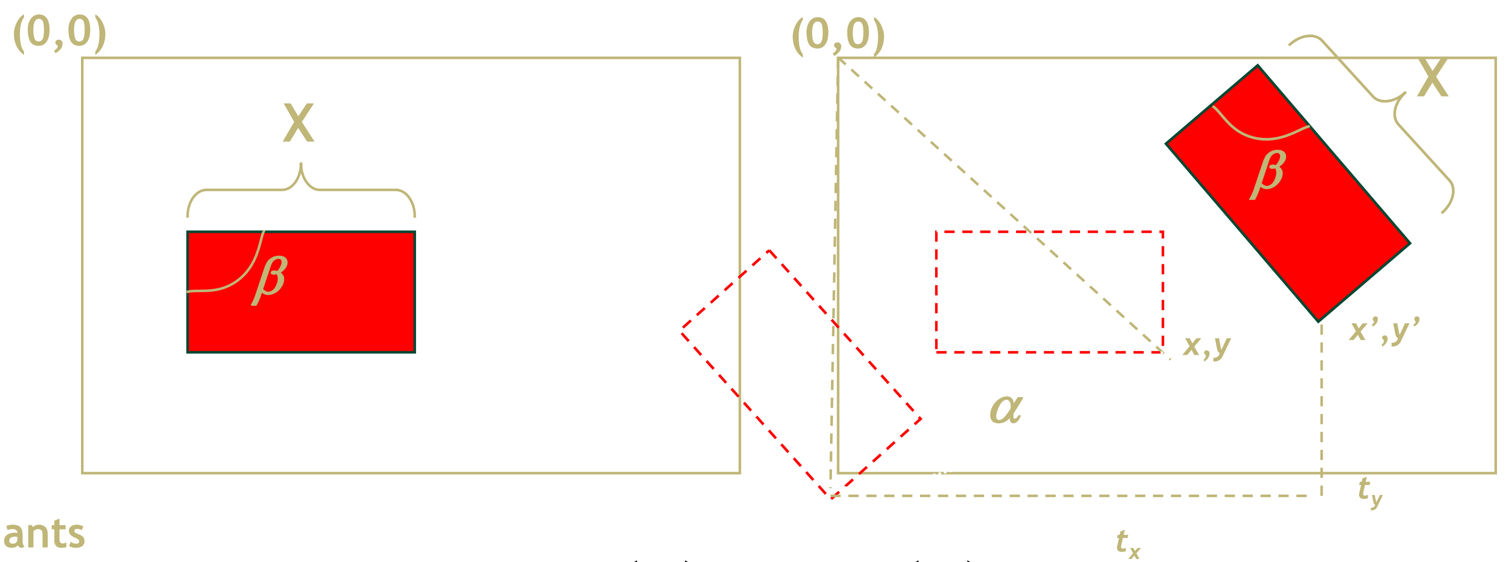


$$x' = \cos(\alpha)\cdot x + \sin(\alpha)\cdot y + t_x$$
$$y' = -\sin(\alpha)\cdot x + \cos(\alpha)\cdot y + t_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & t_x \\ -\sin(\alpha) & \cos(\alpha) & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Geometric transformation

- Rotation



Invariants
- Orientation - no
- Length
- Angle
- Length ratio
- Parallelism

$$x' = \cos(\alpha)\cdot x + \sin(\alpha)\cdot y + t_x$$
$$y' = -\sin(\alpha)\cdot x + \cos(\alpha)\cdot y + t_y$$

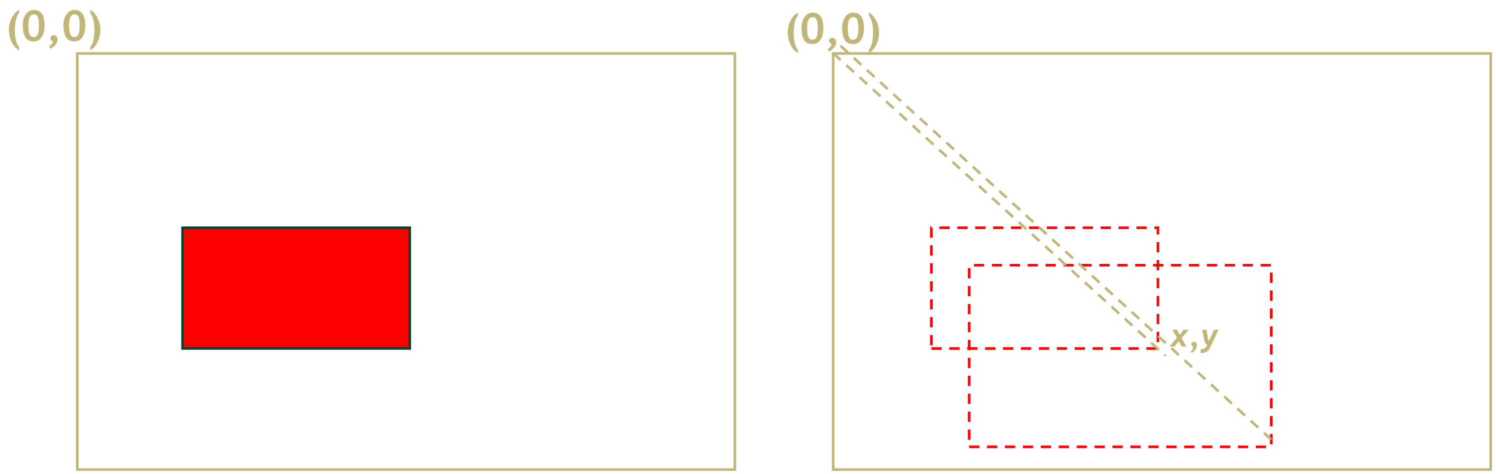
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & t_x \\ -\sin(\alpha) & \cos(\alpha) & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix notation

$$\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$$

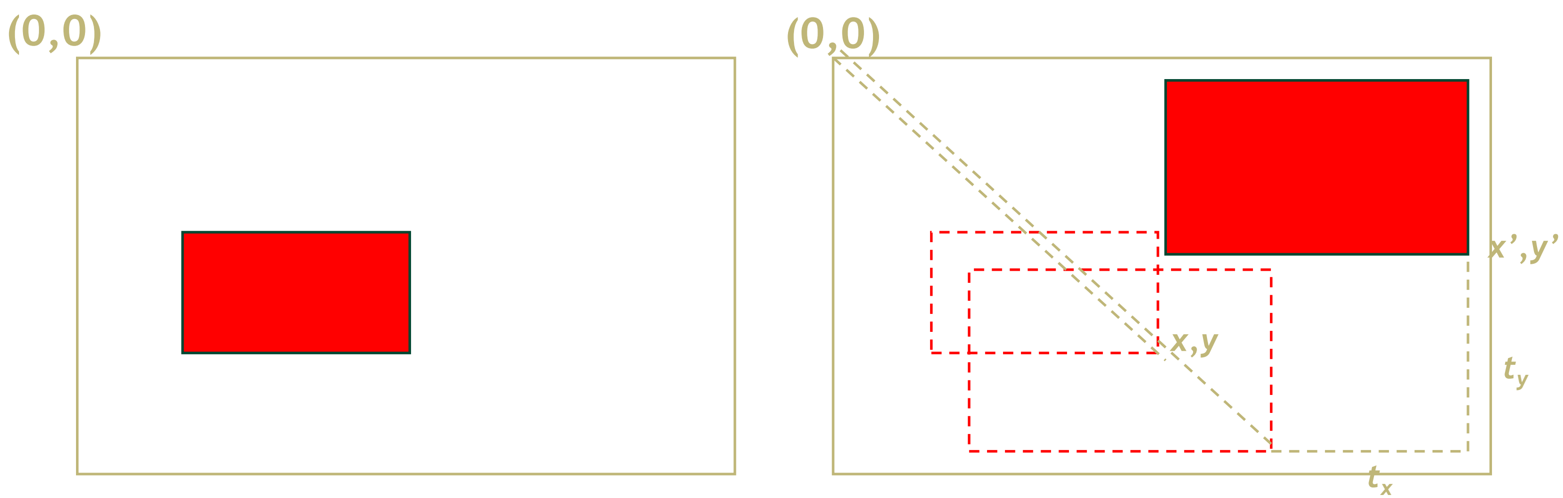
Geometric transformation

- Similarity



Geometric transformation

- Similarity



$$x' = s \cdot x + 0 \cdot y + t_x$$
$$y' = 0 \cdot x + s \cdot y + t_y$$

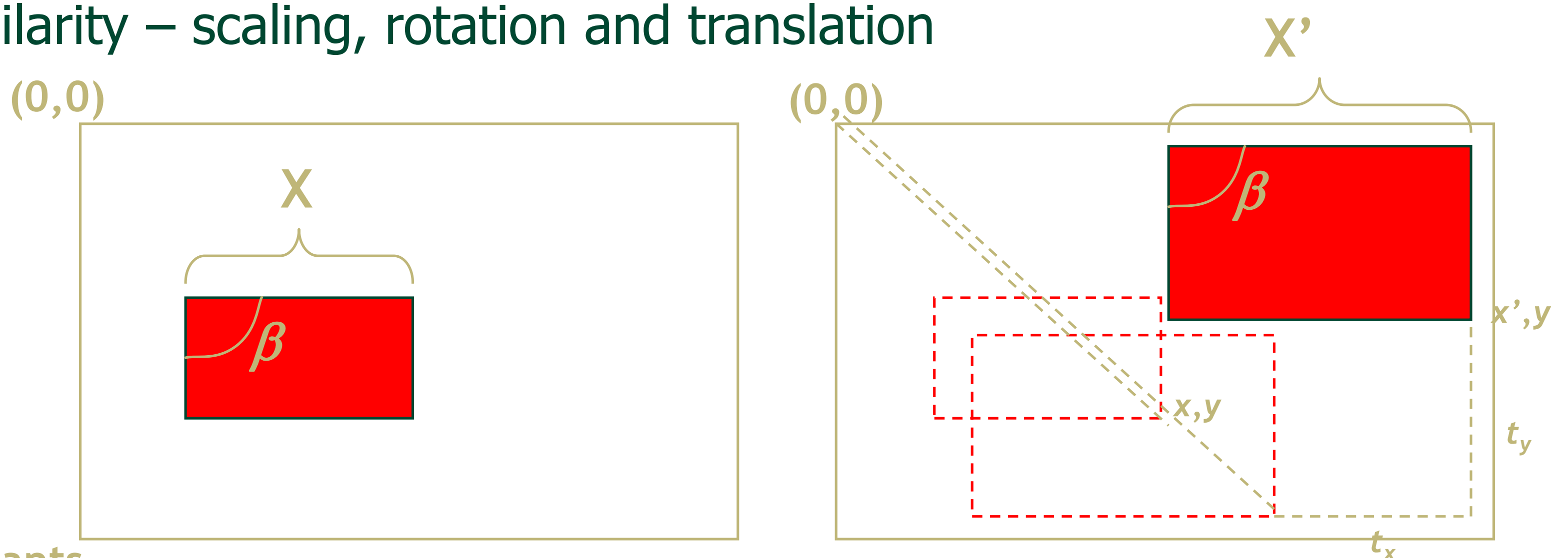
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix notation

$$\mathbf{x}' = \mathbf{S}\mathbf{x} + \mathbf{t}$$

Geometric transformation

- Similarity – scaling, rotation and translation



Invariants
- Orientation - no
- Length - no
- Angle
- Length ratio
- Parallelism

$$x' = \cos(\alpha)\cdot s\cdot x + \sin(\alpha)\cdot s\cdot y + t_x$$
$$y' = -\sin(\alpha)\cdot s\cdot x + \cos(\alpha)\cdot s\cdot y + t_y$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \left\{ \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix} \right\} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s\cdot\cos(\alpha) & s\cdot\sin(\alpha) & t_x \\ -s\cdot\sin(\alpha) & s\cdot\cos(\alpha) & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix notation

$$\mathbf{x'} = \mathbf{SRx} + \mathbf{t}$$

Geometric transformation

- Real example

Geometric transformation

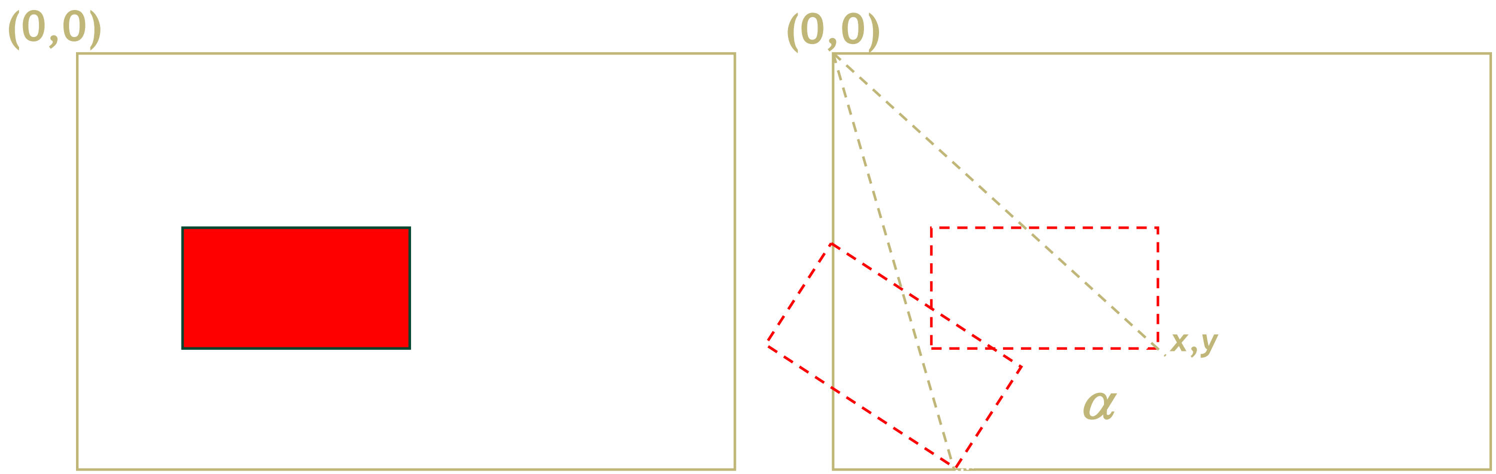
- *(x,y)*

Derive a matrix which will rotate this point by Pi/6, scale by factor 2 and translate by vector [2,1]

$$\begin{bmatrix} ? \\ ? \\ ? \end{bmatrix} = \begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

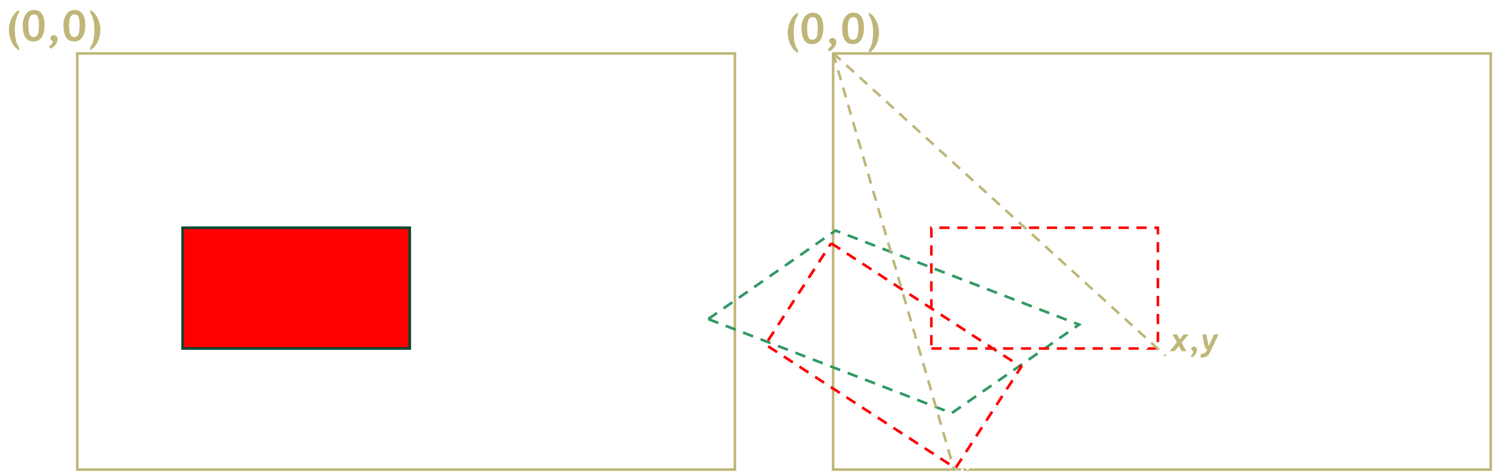
Geometric transformation

- Affine



Geometric transformation

- Affine



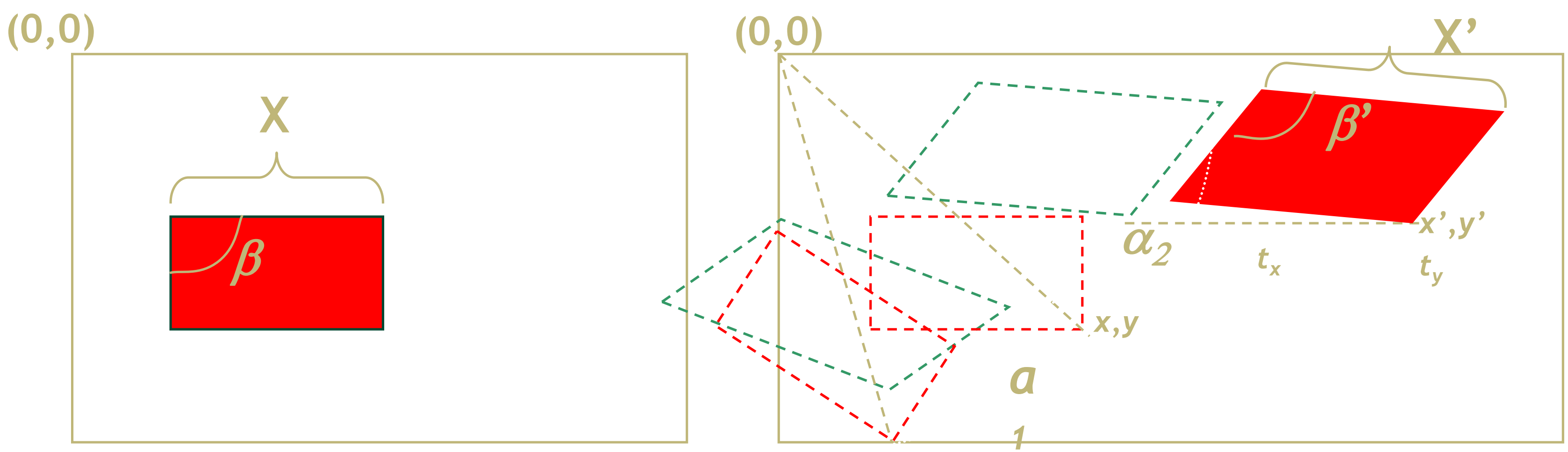
$$\mathbf{x}' = SR(\alpha_1)\mathbf{x}$$

$$\begin{aligned} x' &= \cos(\alpha)\cdot s_x \cdot x + \sin(\alpha)\cdot s_x \cdot y \\ y' &= -\sin(\alpha)\cdot s_y \cdot x + \cos(\alpha)\cdot s_y \cdot y \end{aligned}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \begin{matrix} R(\alpha_2)SR(\alpha_1) \end{matrix} & \begin{matrix} t_x \\ t_y \end{matrix} \\ \begin{matrix} 0 & 0 \end{matrix} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Geometric transformation

- Affine



Invariants
- Orientation - no
- Length - no
- Angle -no
- Length ratio - no
- Parallelism

$$R(\alpha_2)SR(\alpha_1)=\begin{bmatrix}\cos(\alpha_2) & \sin(\alpha_2)\\ -\sin(\alpha_2) & \cos(\alpha_2)\end{bmatrix}\begin{bmatrix}s_x & 0\\ 0 & s_y\end{bmatrix}\begin{bmatrix}\cos(\alpha_1) & \sin(\alpha_1)\\ -\sin(\alpha_1) & \cos(\alpha_1)\end{bmatrix}$$

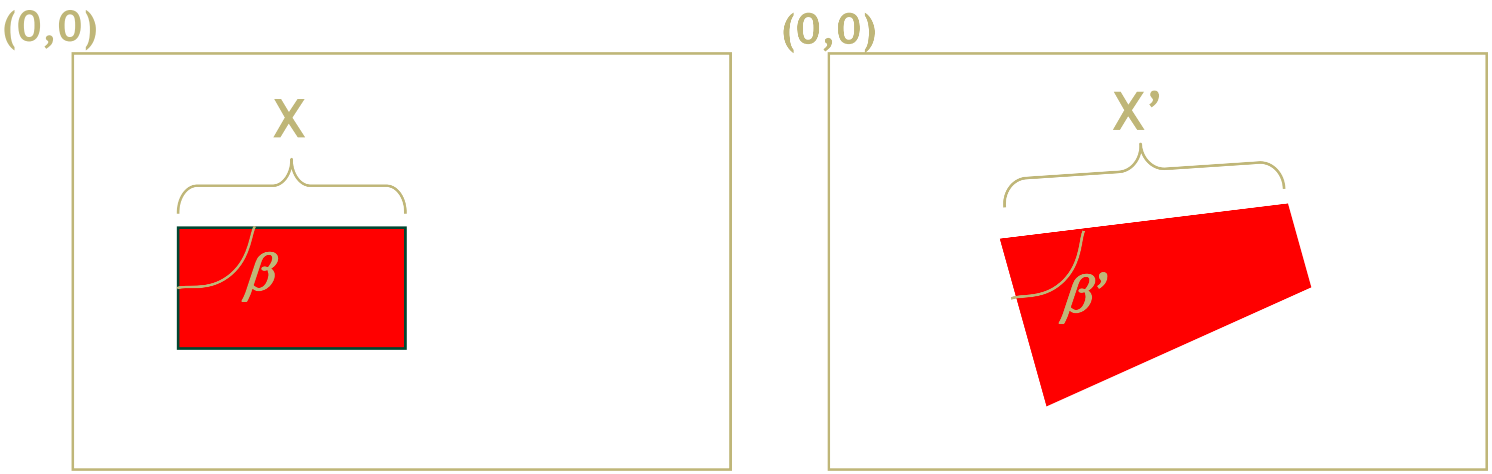
$$\begin{bmatrix}x'\\ y'\\ 1\end{bmatrix}=\begin{bmatrix}R(\alpha_2)SR(\alpha_1) & & \begin{matrix}t_x\\ t_y\end{matrix}\\ 0 \quad 0 & & 1\end{bmatrix}\begin{bmatrix}x\\ y\\ 1\end{bmatrix}$$

Matrix notation

$$\mathbf{x}'=\mathbf{R}_2\mathbf{S}\mathbf{R}_1\mathbf{x}+\mathbf{t}$$

Geometric transformation

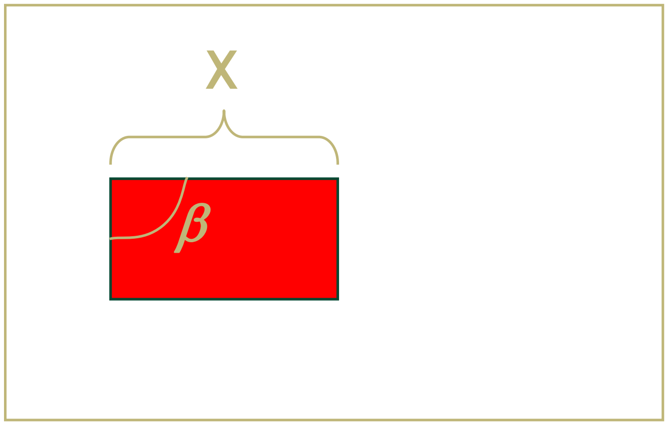
- Projective

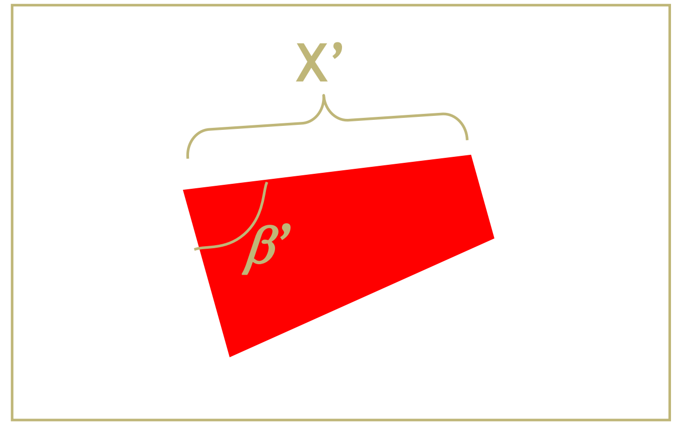


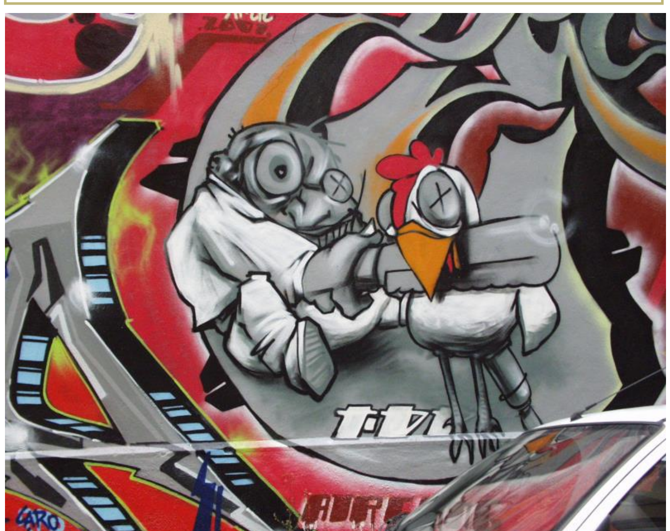
Invariants: ?

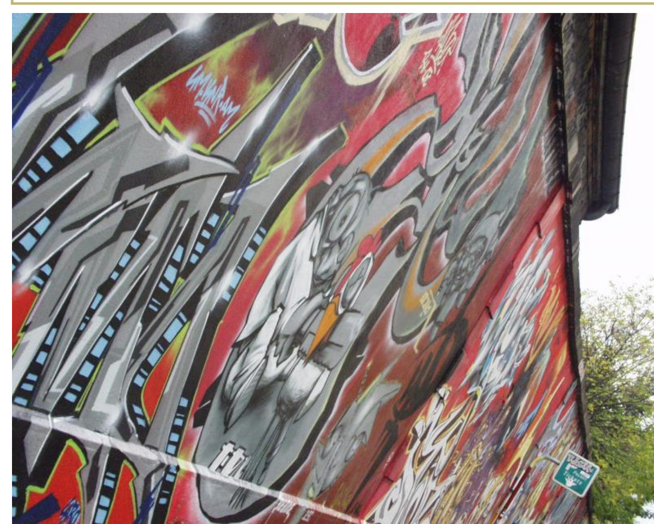
Geometric transformation

- Real example





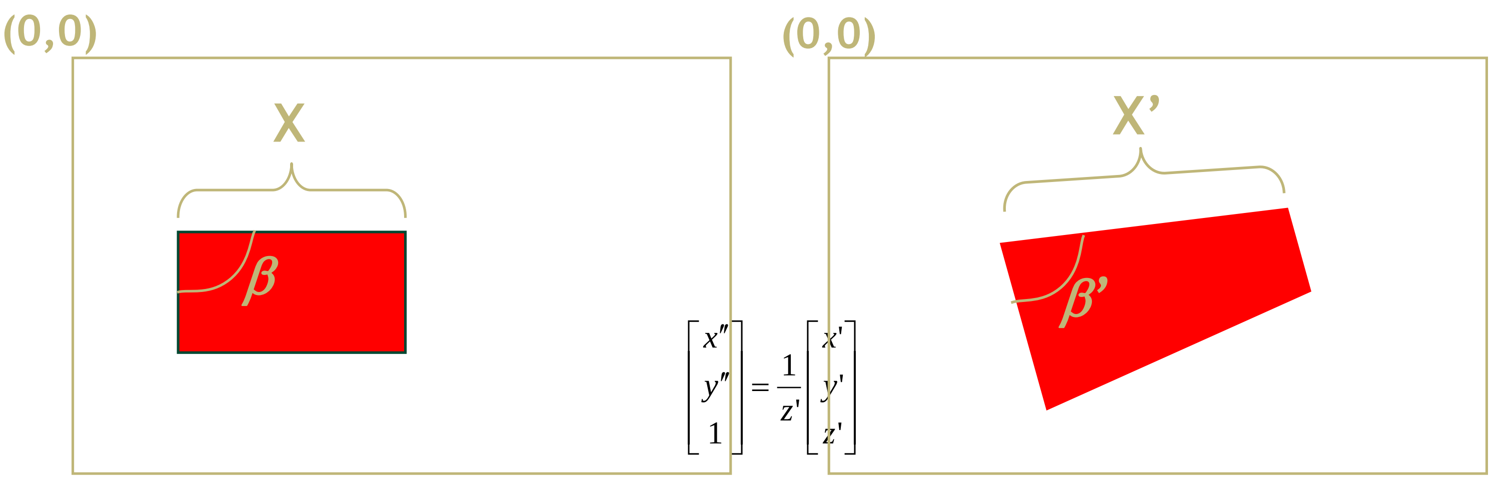




Questions: goo.gl/K61te5

Geometric transformation

- Projective



$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Invariants: Length - no, Angle -no, Length ratio - no, Parallelism -no, orientation - no

Homogenous coordinates

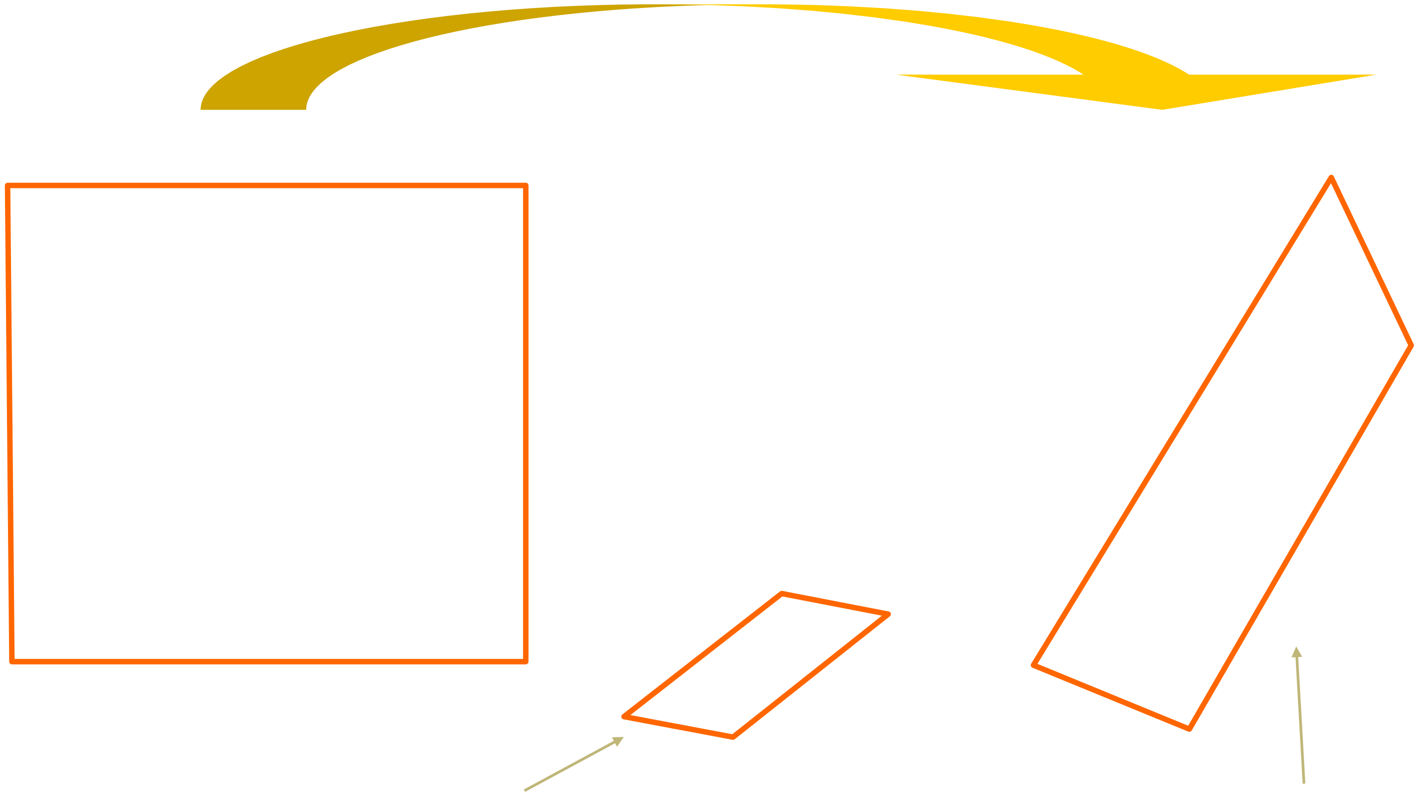
Image coordinates

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Matrix notation

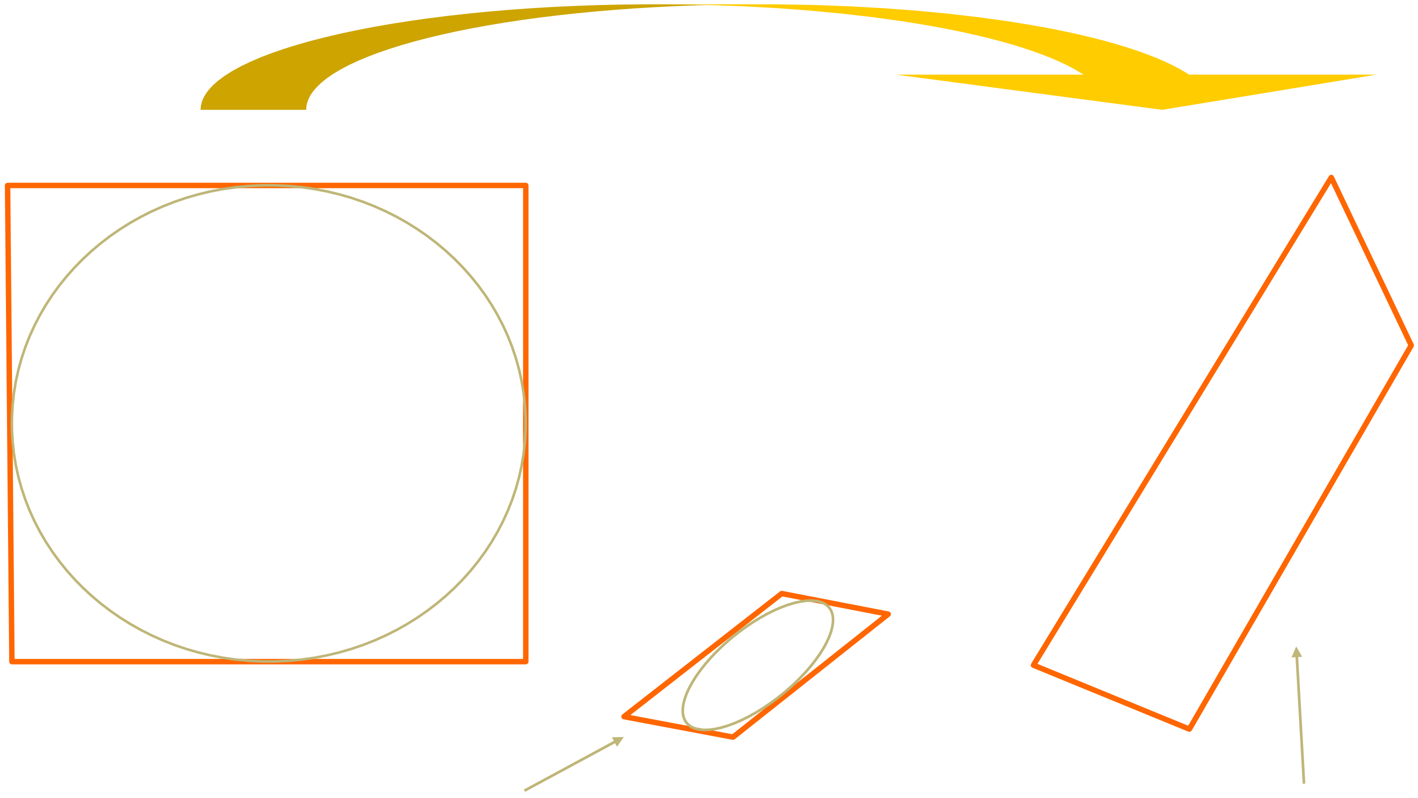
$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

Projective transformation



- Affine - 6 Degrees of Freedom
  - translation (2) + rotation(1) + scale(1)
- Projective – 8 Degrees of Freedom

Projective transformation



- Affine - 6 Degrees of Freedom
  - translation (2) + rotation(1) + scale(1)
- Projective – 8 Degrees of Freedom

From projective to affine

- Approximation of projective transformation by affine transformation
  - Can be done only locally

Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Image coordinates

$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Matrix notation

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

From projective to affine

- Approximation of projective transformation by affine transformation
  - Can be done only locally

Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Image coordinates

$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Matrix notation

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

Affine

$$x'' = \frac{a_{11}\ x + a_{12} y + a_{13}}{0 \cdot x + 0 \cdot y + 1}$$

$$y'' = \frac{a_{21}\ x + a_{22} y + a_{23}}{0 \cdot x + 0 \cdot y + 1}$$

Projective

$$x'' = \frac{h_{11}\ x + h_{12} y + h_{13}}{h_{31}\ x + h_{32} y + 1}$$

$$y'' = \frac{h_{21}\ x + h_{22} y + h_{23}}{h_{31}\ x + h_{32} y + 1}$$

From projective to affine

- Approximation of projective transformation by affine transformation
  - Can be done only locally

Affine

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homography

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

h_{31}, h_{32}

Image coordinates

$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Matrix notation

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

Affine

$$x'' = \frac{a_{11}\ x + a_{12} y + a_{13}}{0 \cdot x + 0 \cdot y + 1}$$

$$y'' = \frac{a_{21}\ x + a_{22} y + a_{23}}{0 \cdot x + 0 \cdot y + 1}$$

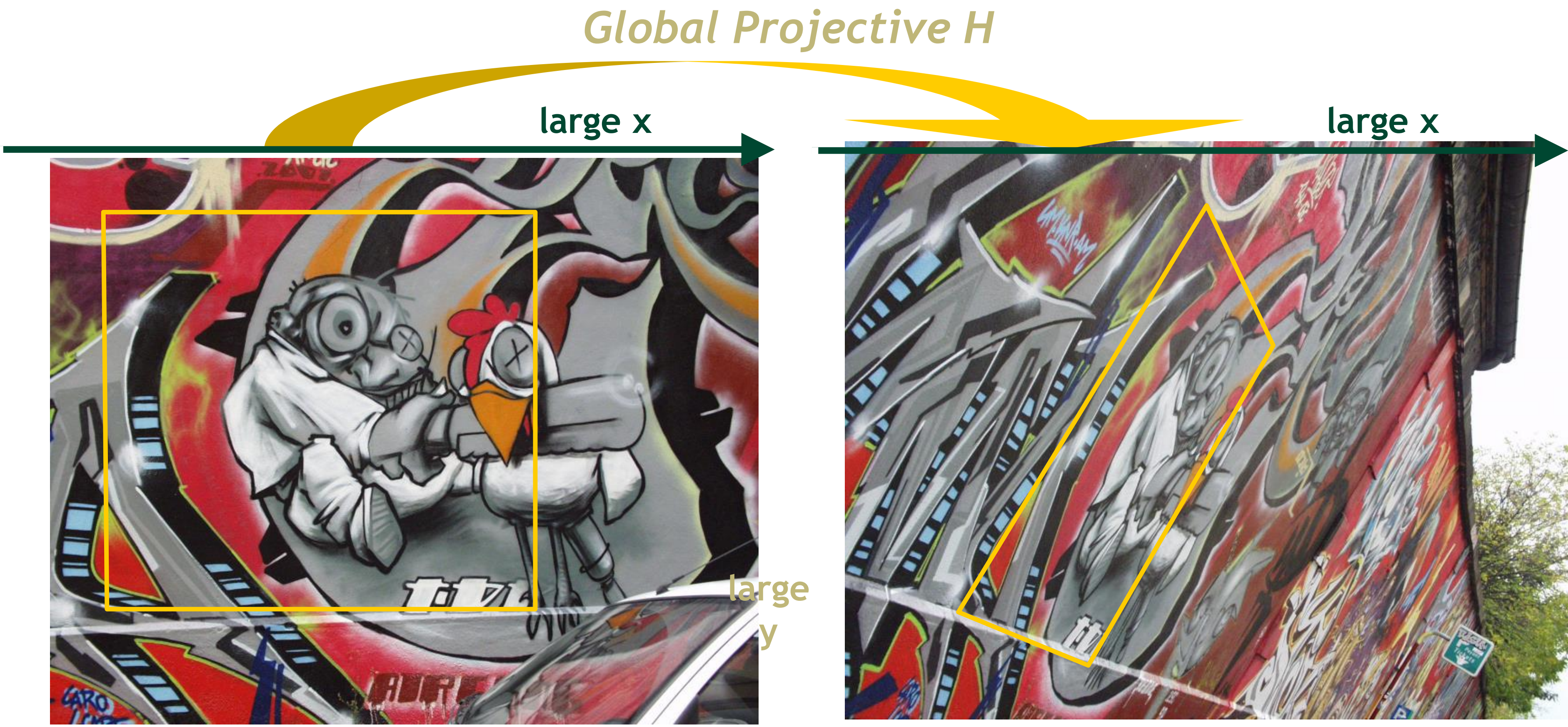
Projective

$$x'' = \frac{h_{11}\ x + h_{12} y + h_{13}}{h_{31}\ x + h_{32} y + 1}$$

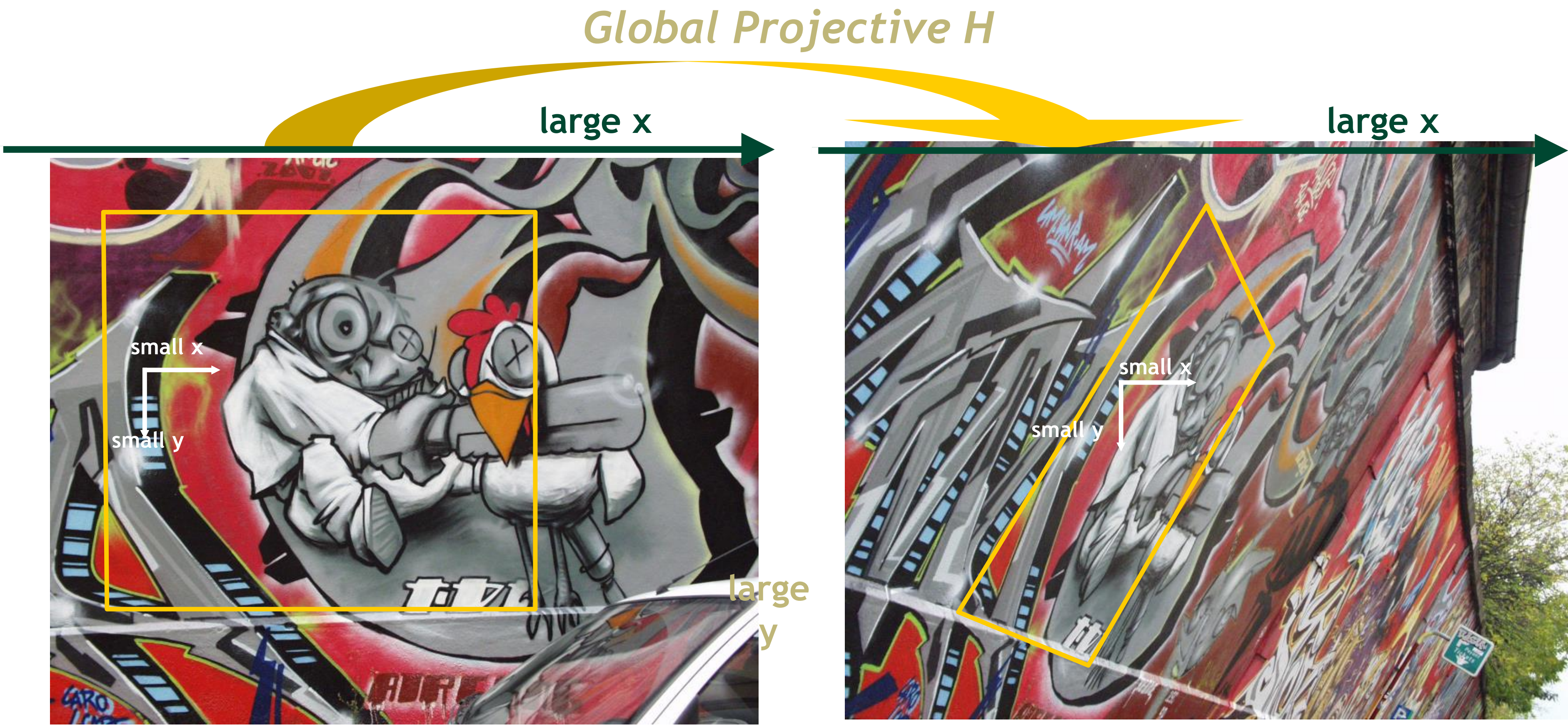
$$y'' = \frac{h_{21}\ x + h_{22} y + h_{23}}{h_{31}\ x + h_{32} y + 1}$$

$h_{31}\ x + h_{32} y$

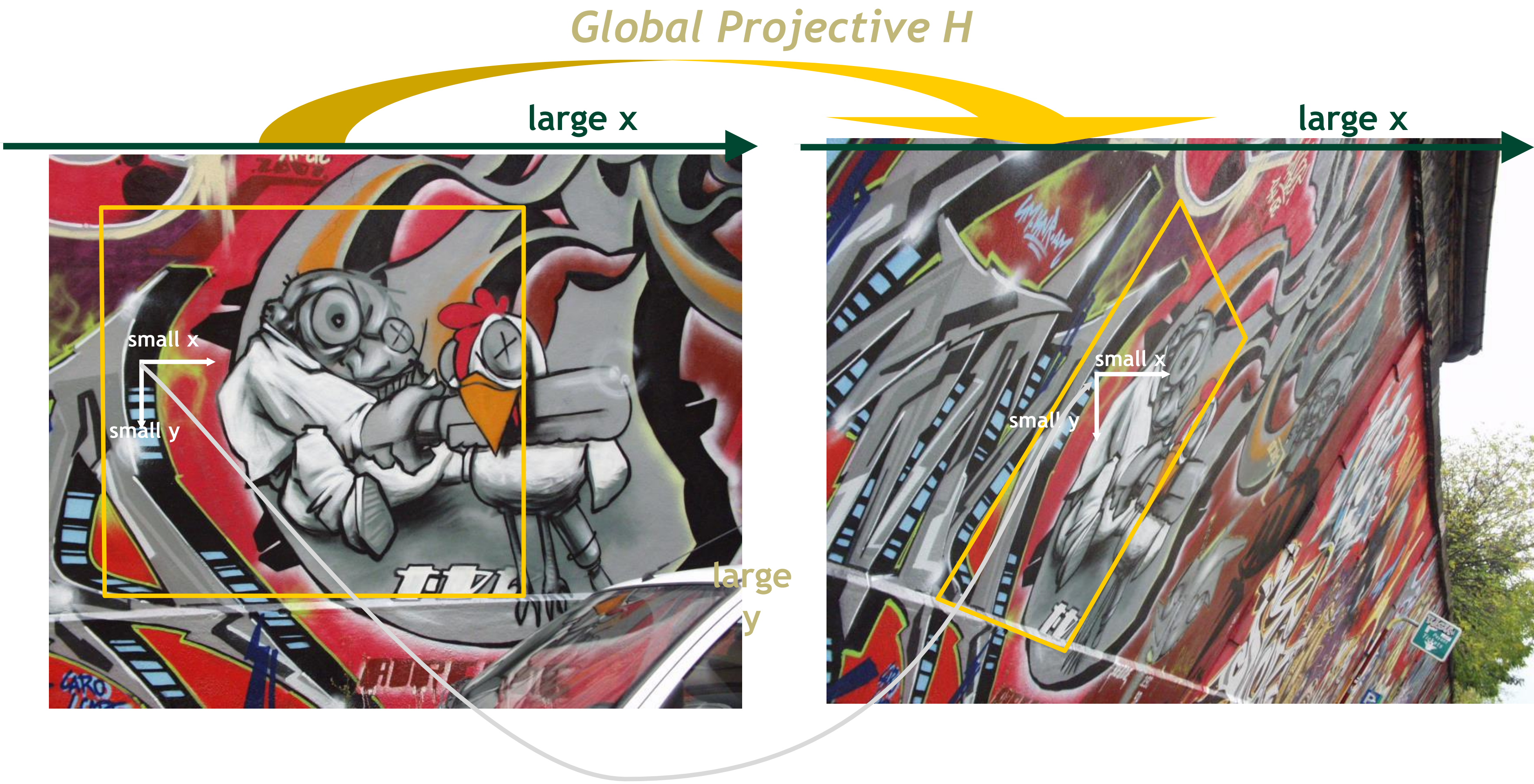
From projective to affine



From projective to affine

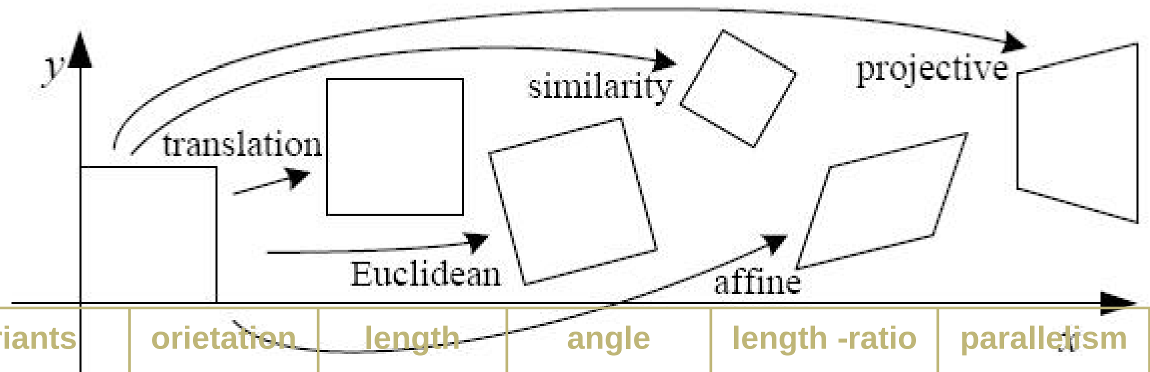


From projective to affine



Planar image transformations

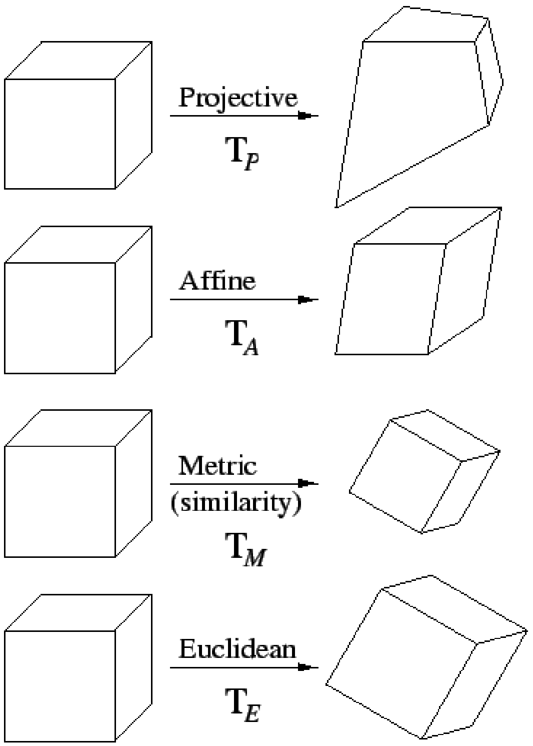
- Transformation of planar scenes
  - Fully defined by a 3x3 matrix



Invariants / Transformation	orietation	length	angle	length -ratio	parallelism	straight lines
Translation & rotation = Euclidean 3DoF	no	yes	yes	yes	yes	yes
Similarity 4DoF	no	no	yes	yes	yes	yes
Affine 6DoF	no	no	no	no	yes	yes
Projective, 8DoF	no	no	no	no	no	yes

Taxonomy of planar projective transforms II

group	matrix	properties
perspective 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{12} & h_{22} & h_{23} \\ h_{13} & h_{23} & h_{33} \end{bmatrix}$	concurrency, collinearity, in-cidence, tangency, inflection
affine 6 DOF	$\begin{bmatrix} a_{11} & a_{12} & t_{13} \\ a_{12} & a_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	parallelism, ratio of areas, ratio of lengths on collinear or parallel sgements
similarity 4 DOF	$\begin{bmatrix} sr_{11} & sr_{12} & t_{13} \\ sr_{12} & sr_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	angles, ratio of lengths
Euclidean 3 DOF	$\begin{bmatrix} r_{11} & r_{12} & t_{13} \\ r_{12} & r_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix}$	length, area



Notes:

•Properties of the more general transforms are inherited by transformations lower in the table

•R = [r_{ij}] is a rotation matrix, i.e. $\mathbf{R}\,\mathbf{R}^{>}=1$, also

$$\mathrm{R} = \begin{matrix} \sin(\theta) & \cos\theta \\ -\cos\theta & \sin\theta \end{matrix}$$

Revision

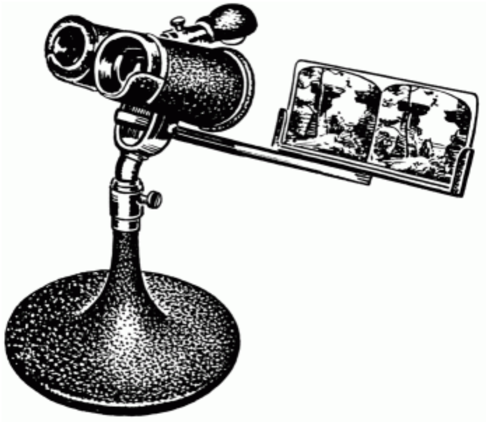
- Pinhole camera model
- Derive equations to compute point position in the image given its position in 3D
  - Convert it to homogeneous matrix formula
- Planar transformations, what are their parameters and invariants?
- Transform a point using homogeneous coordinates in matrix notation.
- Covert transformation parameters (rotation angle, translation vector, scale factor) to matrix notation in homogeneous system.

Stereo and 3D vision

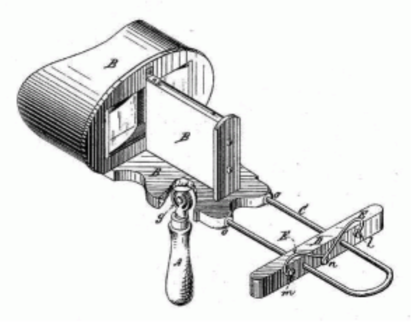
- wet-plate process invented in 1851 with portable dark room

two pictures mounted for parallel viewing
pictures taken with a two-lensed camera - two points of view separated by 2.5”,

Stereo and 3D vision



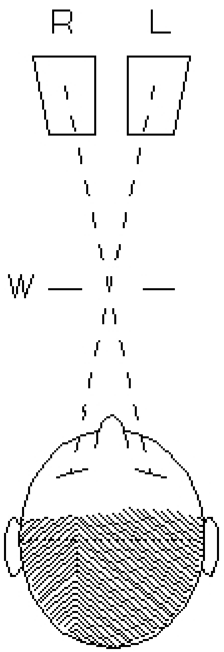
Early form of stereoscope.



The Holmes-Bates Stereoscope
US patent US00232649

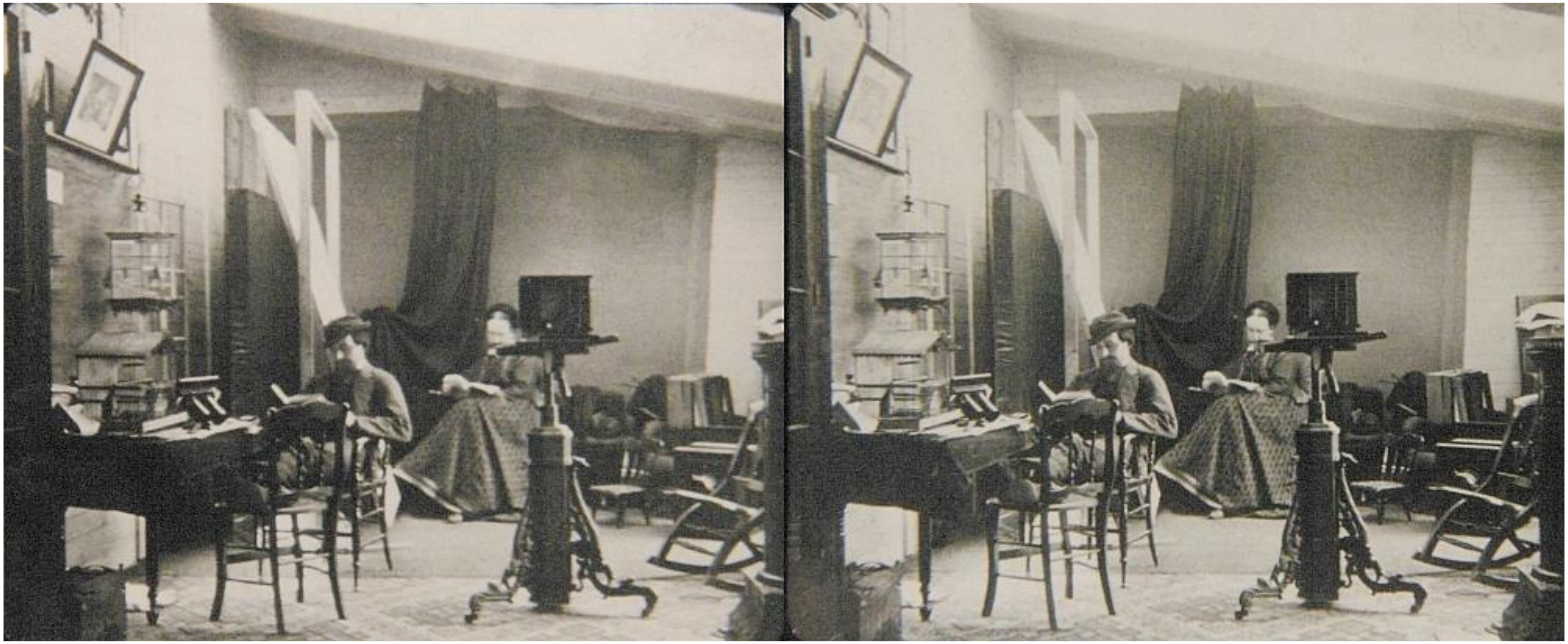
Stereo and 3D vision

- two pictures mounted for parallel viewing
- pictures were taken with a two-lensed camera, recording two points of view separated by 2.5”,
- left and right pictures interchanged, for cross eyed viewing.
  - allows larger images on the computer screen



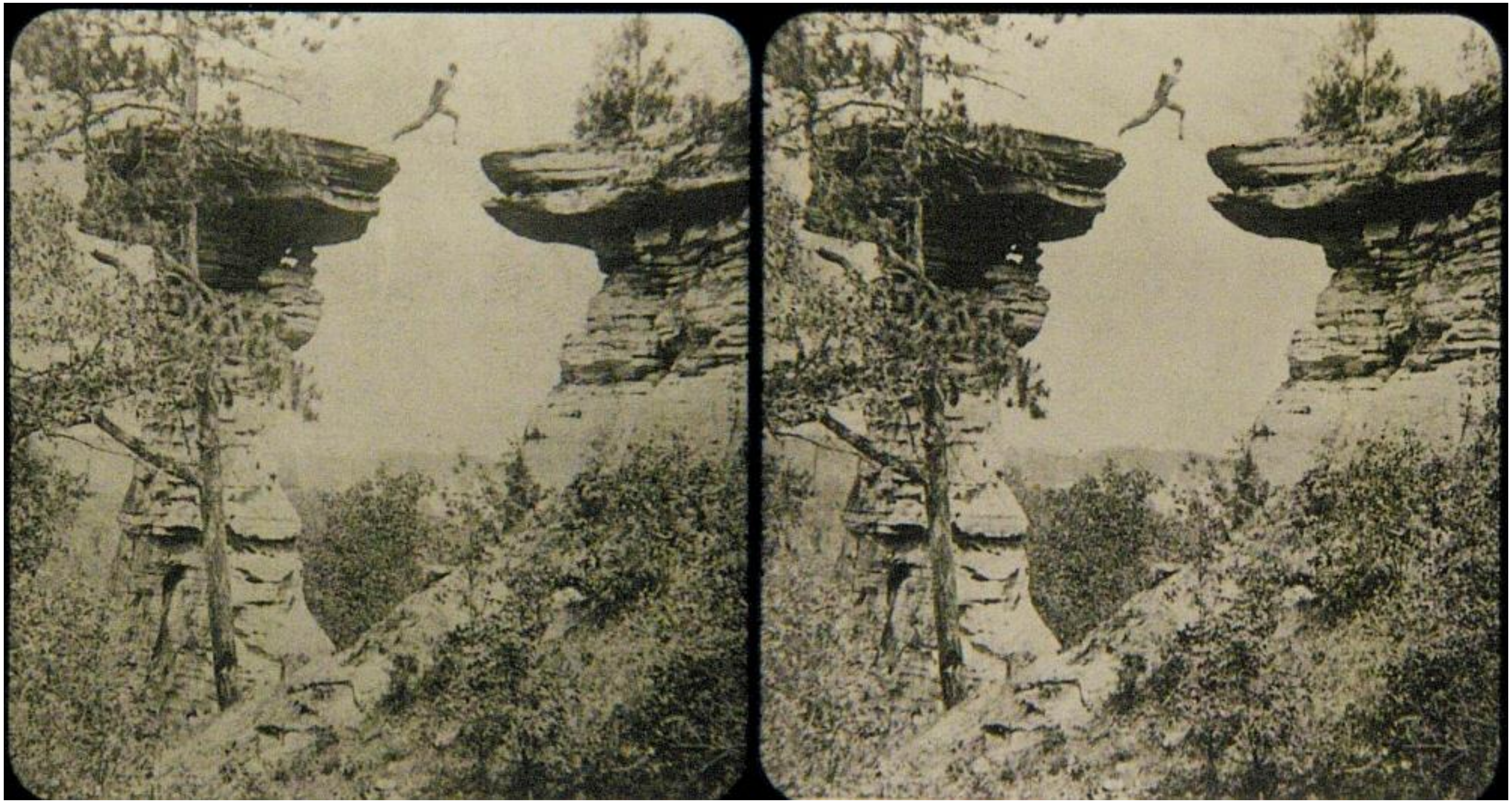
Stereo and 3D vision

- Stereo studio 1870?



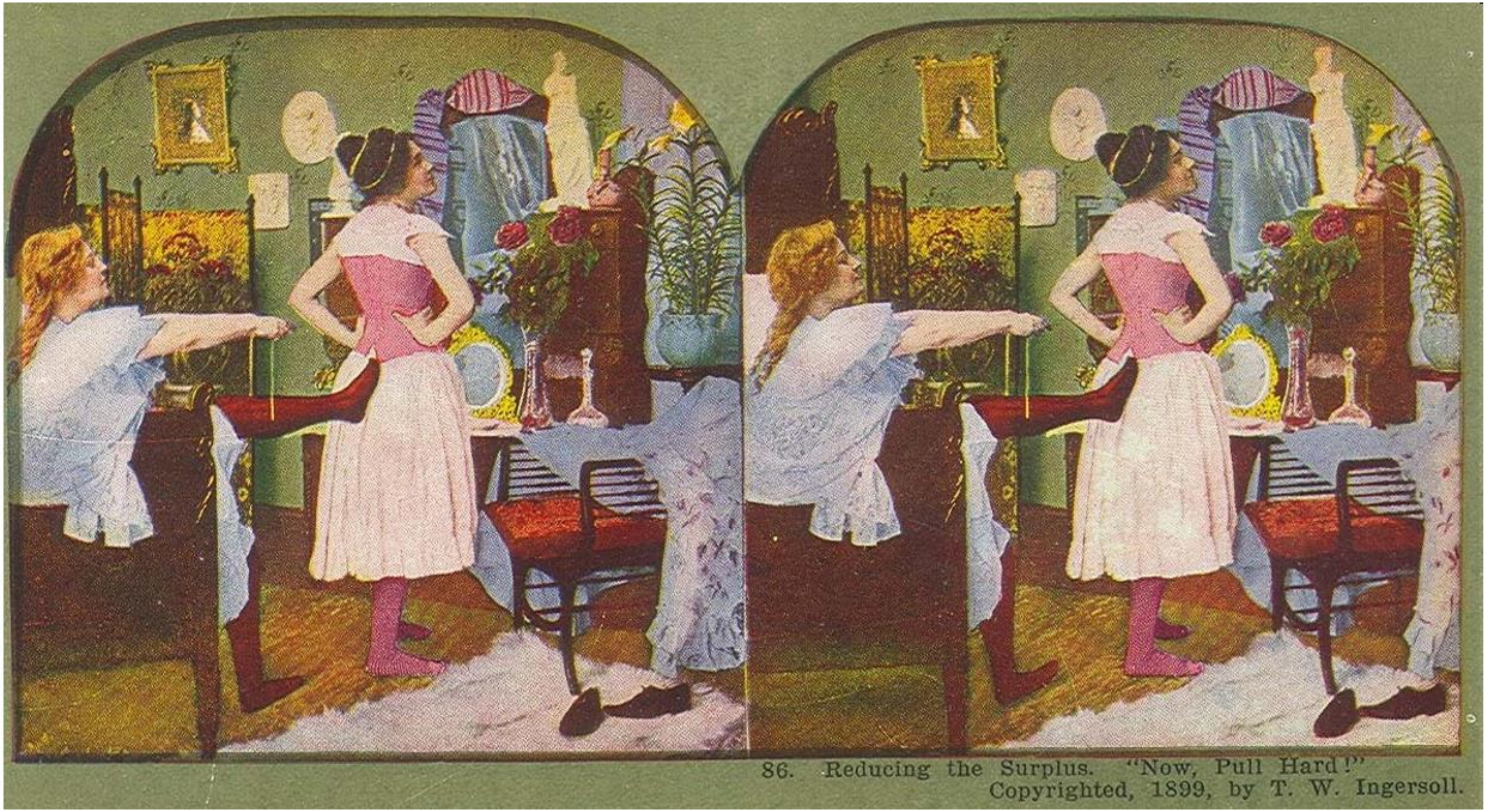
Stereo and 3D vision

- 1880?



Stereo and 3D vision

- Colour pictures

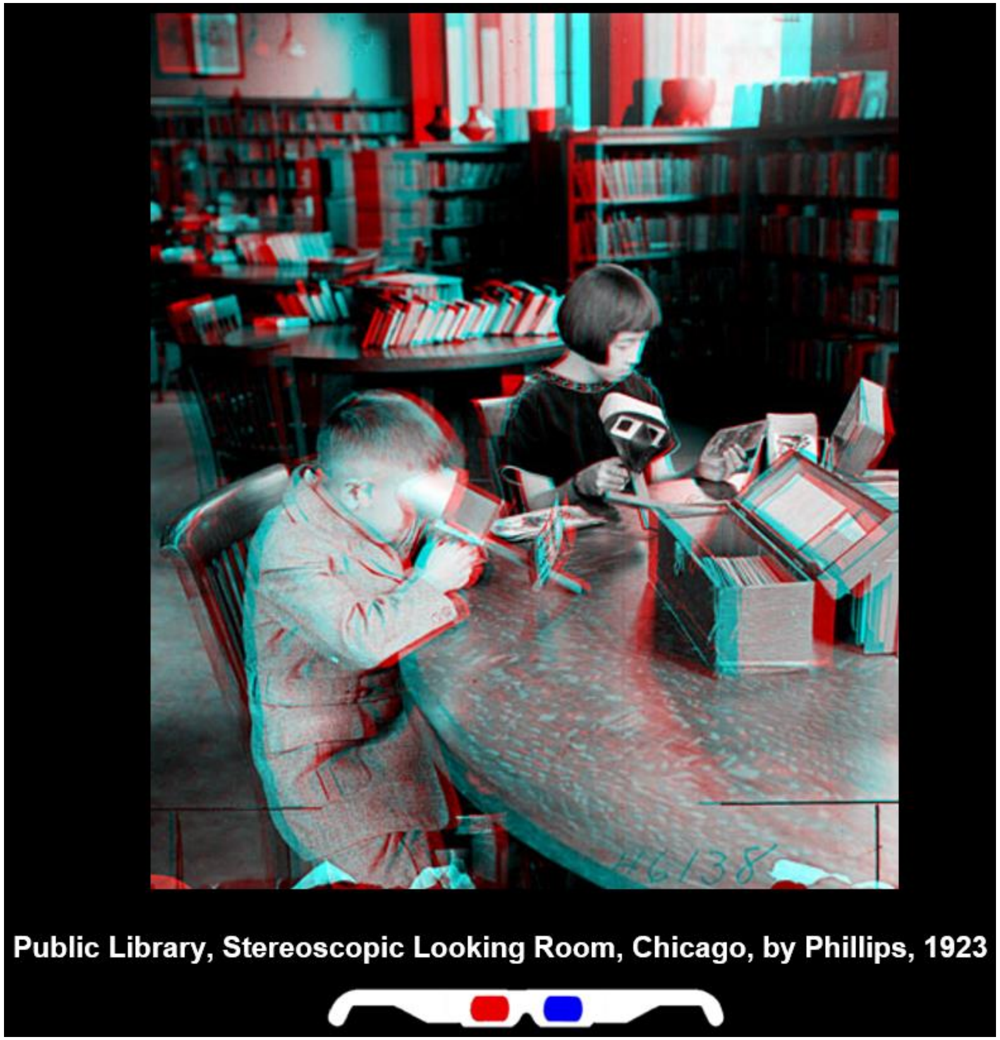


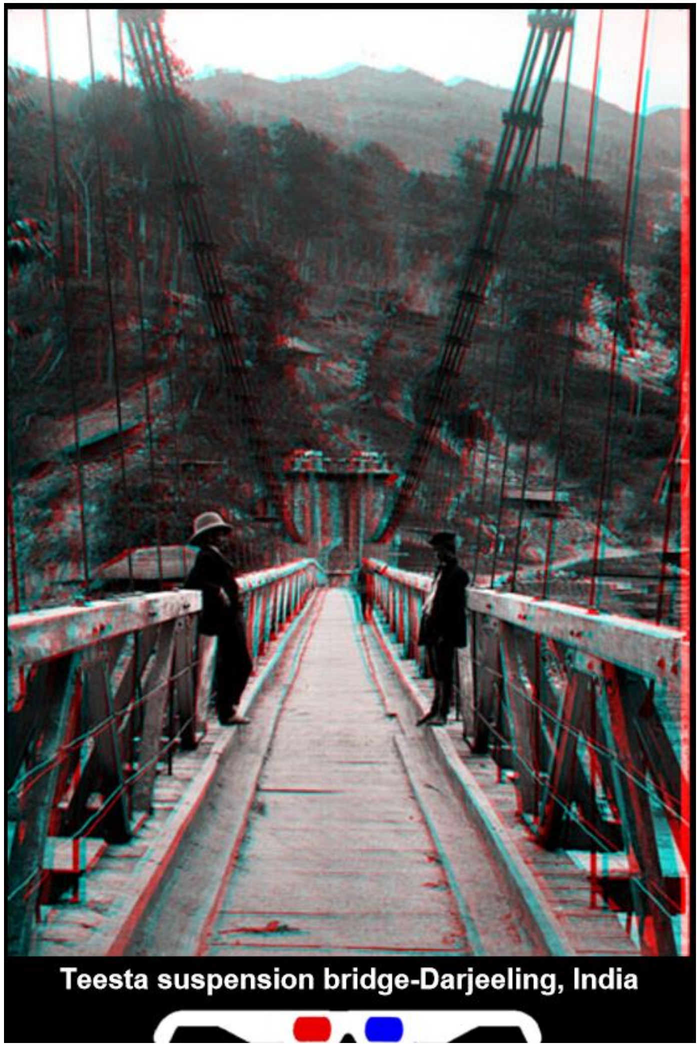
Anaglyphs

Anaglyphs provide a stereoscopic 3D effect when viewed with 2-color glasses (each lens a chromatically opposite color, usually red and cyan).

contain two differently filtered colored images

http://en.wikipedia.org/wiki/Anaglyph_image

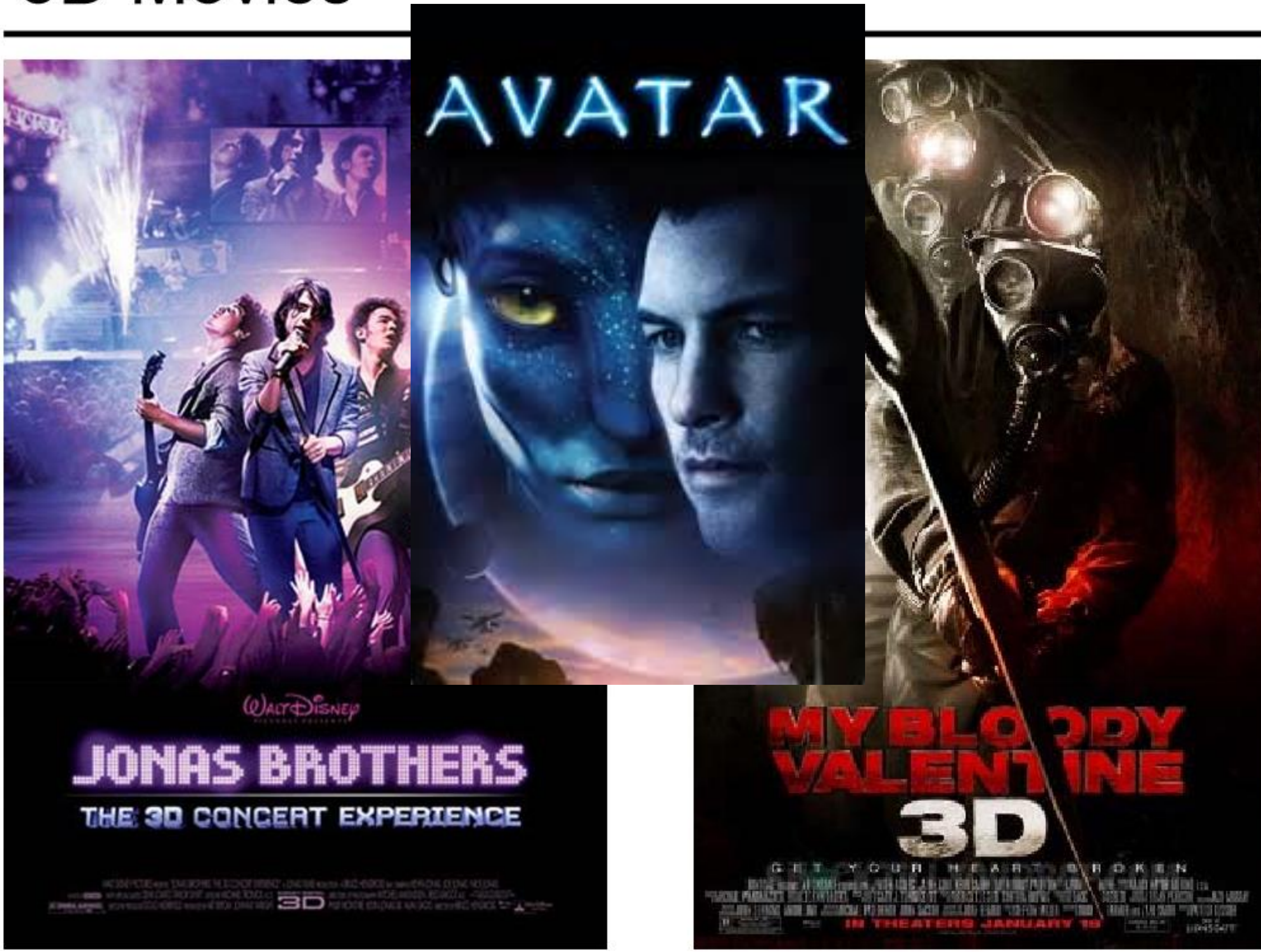

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923


Teesta suspension bridge-Darjeeling, India

Woman getting eye exam during immigration procedure at Ellis Island, c. 1905 - 1920 , UCR Museum of Phography

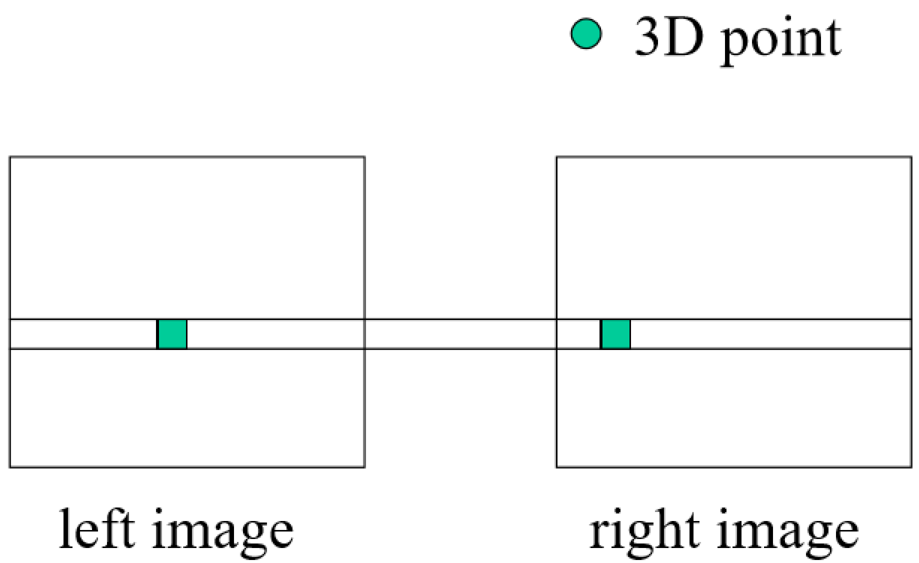
Mark Twain at Pool Table", no date, UCR Museum of Photography

3D Movies


AVATAR
Walt Disney
JONAS BROTHERS
THE 3D CONCERT EXPERIENCE
3D
MY BLOODY VALENTINE
3D
GET YOUR HEART BROKEN
IN THEATERS JANUARY 16

How do we get 3D from Stereo Images?

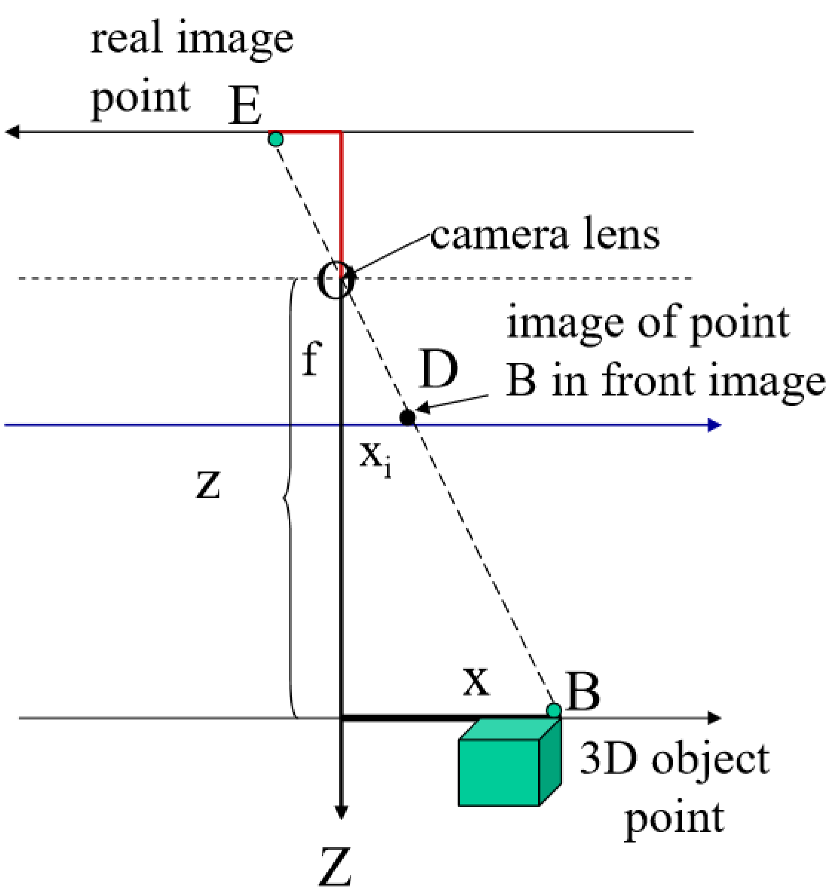
Perception of depth arises from "disparity" of a given 3D point in your right and left retinal images



disparity: the difference in image location of the *same 3D point* when projected under perspective to two different cameras

$$\boxed{d = xleft - xright}$$

Perspective projection



This is the axis of the real image plane.

O is the center of projection.

This is the axis of the front image plane, which we use.

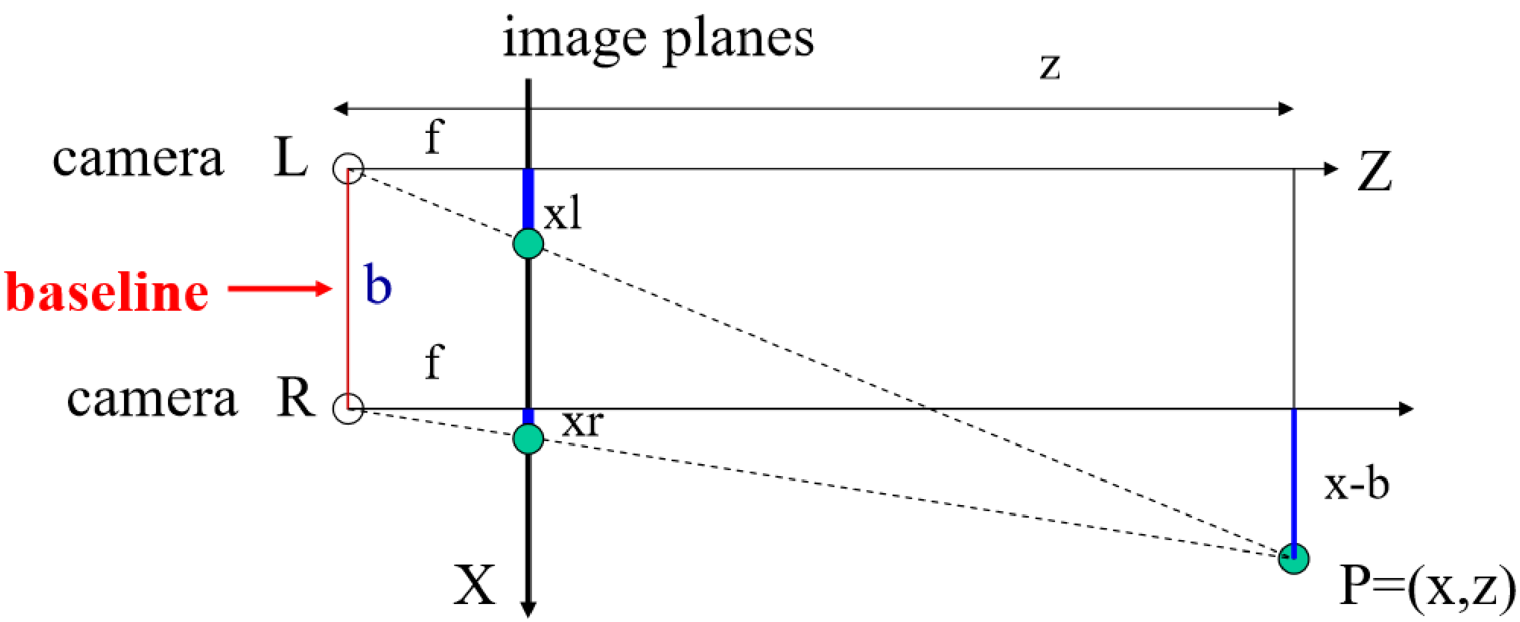
$$\boxed{\frac{x_i}{f} = \frac{x}{z}}$$

(from similar triangles)

(Note: For convenience, we orient Z axis as above and use f instead of –f as in lecture 5)

Projection for Stereo Images

Simple Model: Optic axes of 2 cameras are parallel



$$\frac{z}{f} = \frac{x}{xl}$$

$$\frac{z}{f} = \frac{x-b}{xr}$$

$$\frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

Y-axis is perpendicular to the page.

(from similar triangles)

3D from Stereo Images: Triangulation

For stereo cameras with parallel optical axes, focal length f, baseline b, corresponding image points (xl,yl) and (xr,yr), the location of the 3D point can be derived from previous slide's equations:

Depth $z = f*b / (xl - xr) = f*b/d$

$x = xl*z/f$ or $b + xr*z/f$

$y = yl*z/f$ or $yr*z/f$

This method of determining depth from disparity d is called **triangulation.**

Note that **depth is inversely proportional to disparity**

3D from Stereo Images

Depth $z = f*b / (xl - xr) = f*b/d$

$x = xl*z/f$ or $b + xr*z/f$

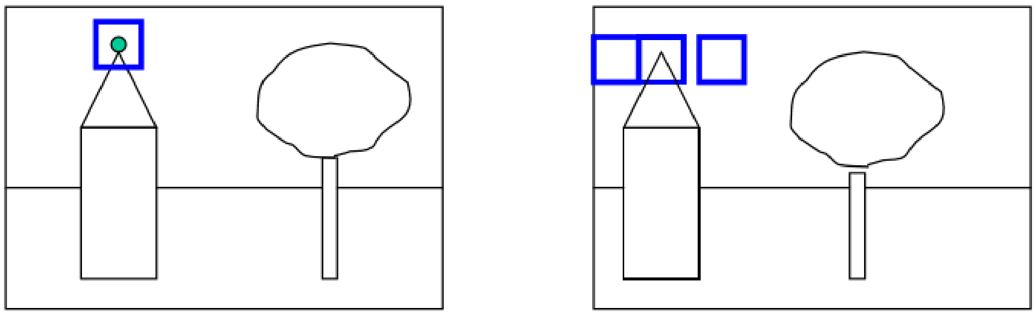
$y = yl*z/f$ or $yr*z/f$

Two main problems:

1. Need to know focal length f, baseline b
   - use prior knowledge or camera calibration
2. Need to find corresponding point (xr,yr) for each (xl,yl) $\Rightarrow$ Correspondence problem

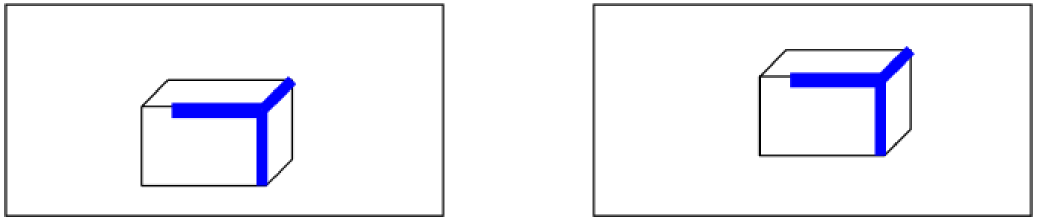
Solving the stereo correspondence problem

1. Cross correlation or SSD using small windows.



dense

2. Symbolic feature matching, usually using segments/corners.



sparse

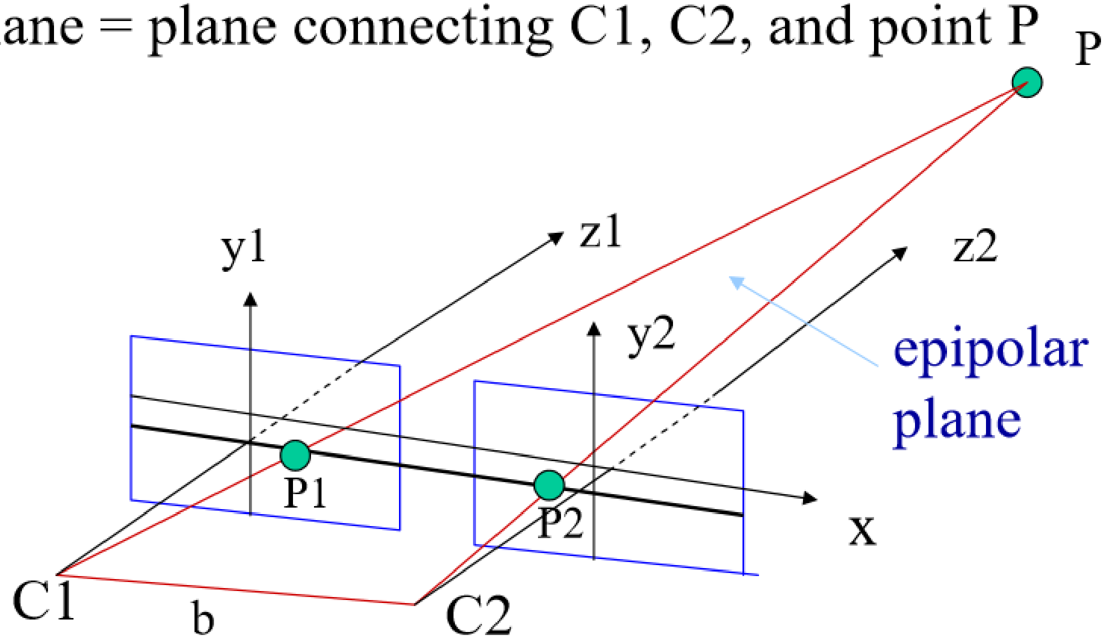
3. Use the newer interest operators, e.g., SIFT.

sparse

Given a point in the left image, do you need to search the entire right image for the corresponding point?

Epipolar Constraint for Correspondence

Epipolar plane = plane connecting C1, C2, and point P



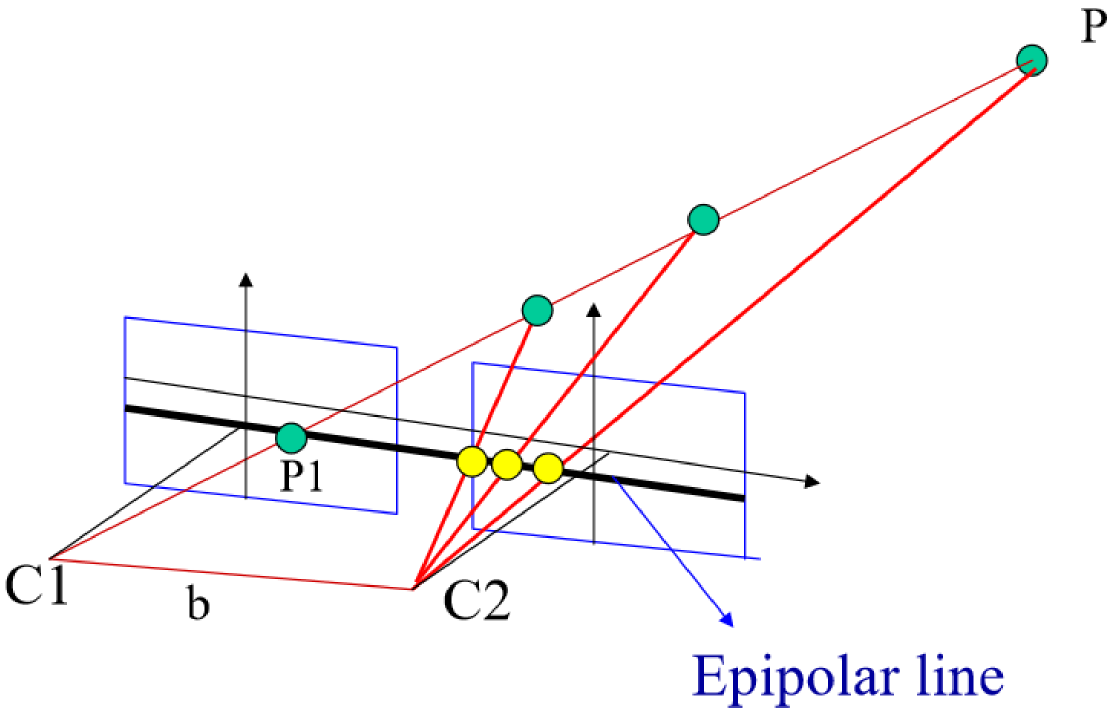
.

Epipolar plane cuts through image planes forming an epipolar line in each plane

Match for P1 (or P2) in the other image must lie on epipolar line

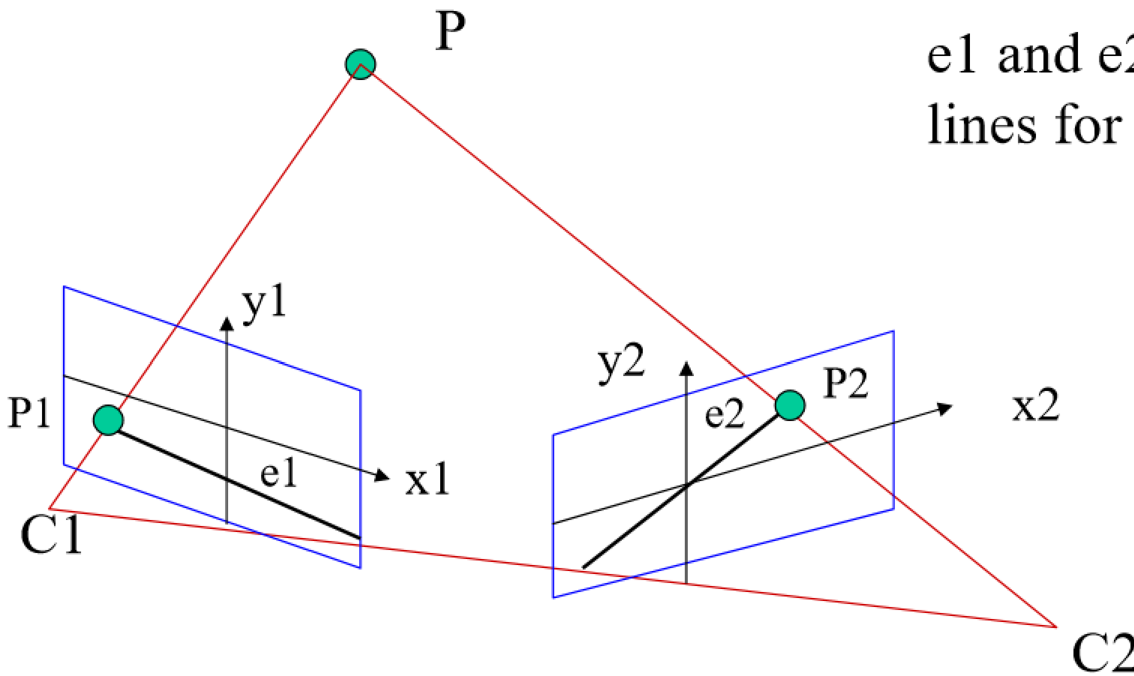
Epipolar Constraint for Correspondence

Match for P1 in the other image must lie on epipolar line
So need search only along this line



What if the optical axes of the 2 cameras are not parallel to each other?

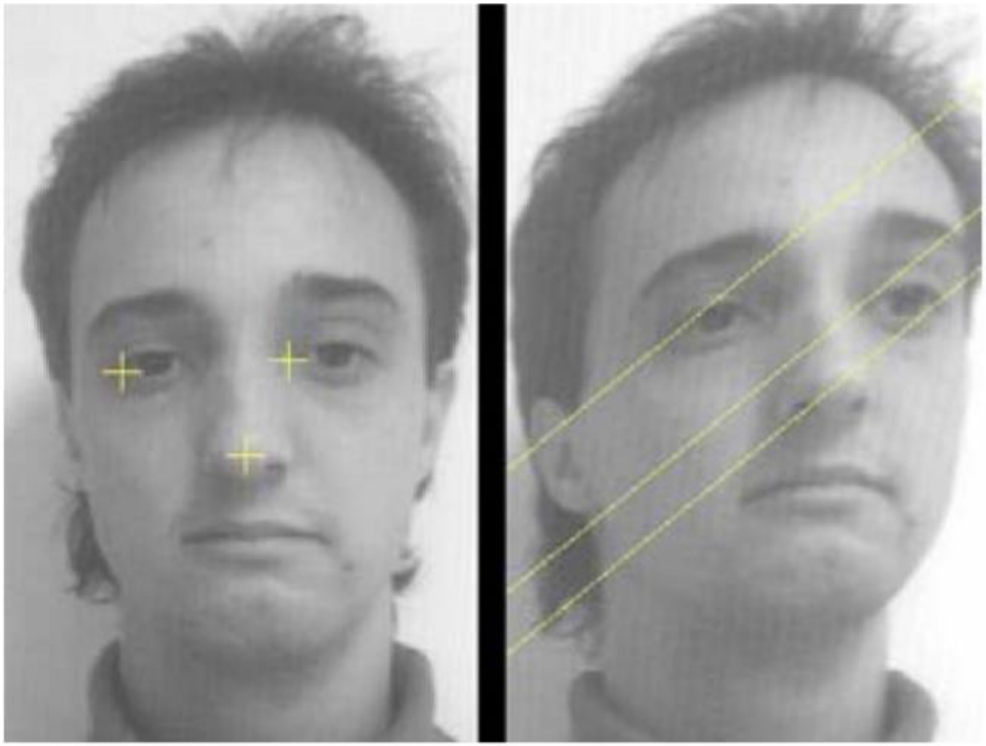
Epipolar constraint still holds…



e1 and e2 are the epipolar lines for point P

But the epipolar lines may no longer be horizontal

Example

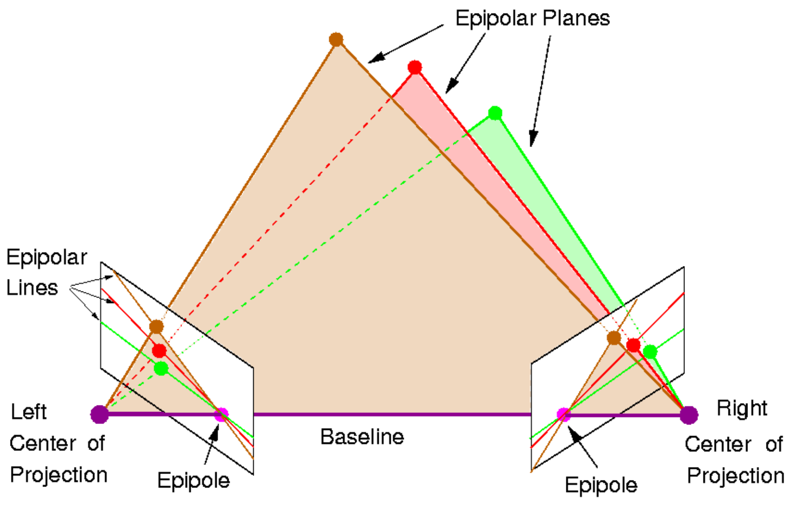


(from a slide by Pascal Fua)

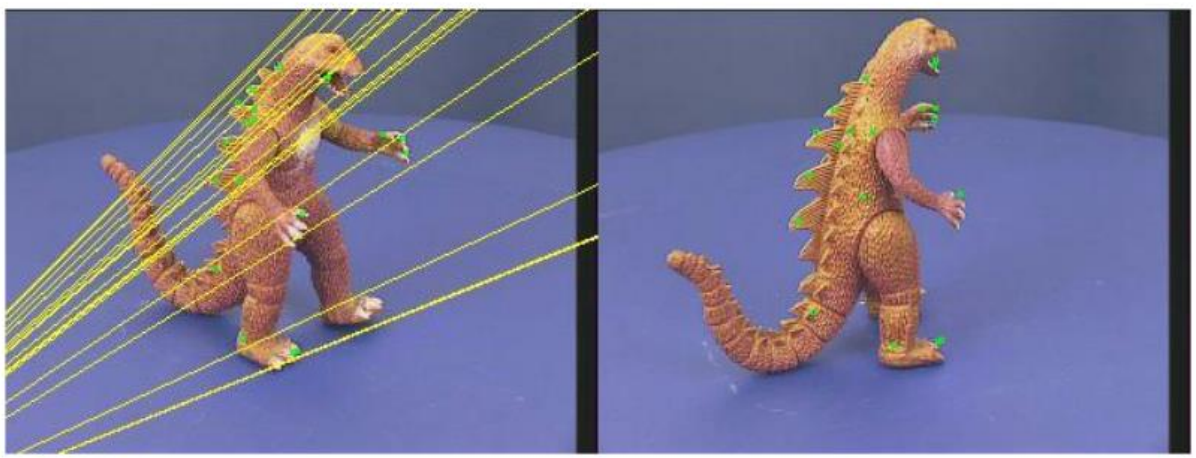
Yellow epipolar lines for the three points shown on the left image

Given a point P1 in left image on epipolar line e1, can find epipolar line e2 provided we know relative orientations of cameras

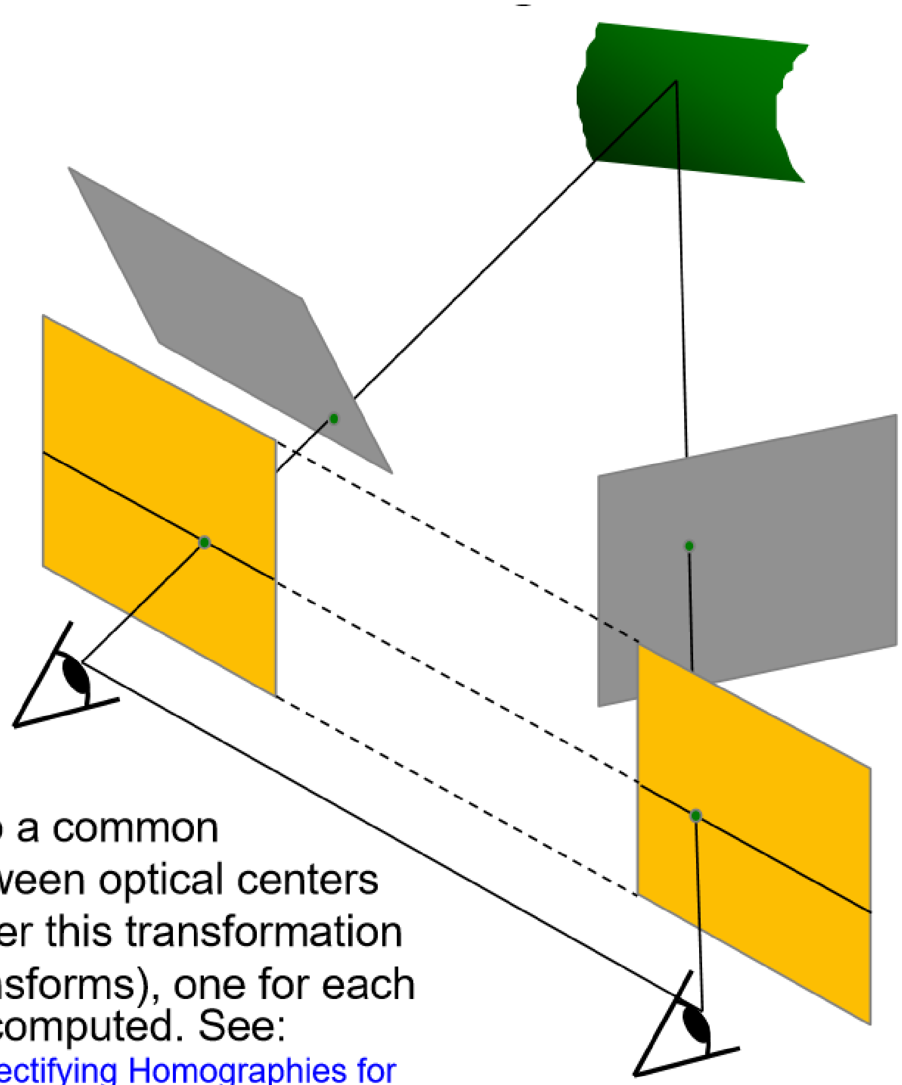
Epipolar constraints



Epipolar lines all intersect at epipoles.

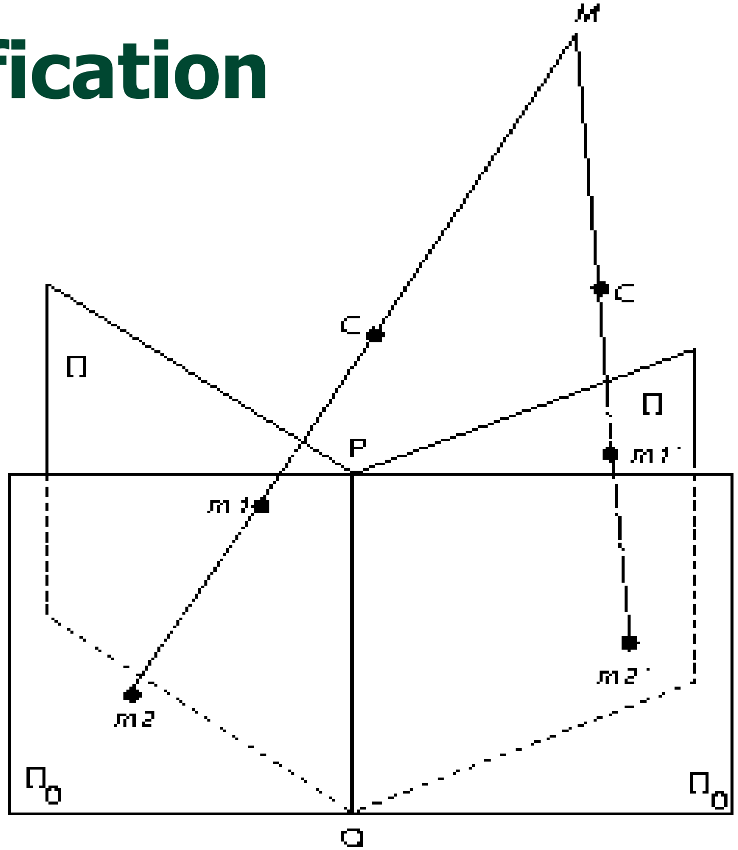


Stereo image rectification



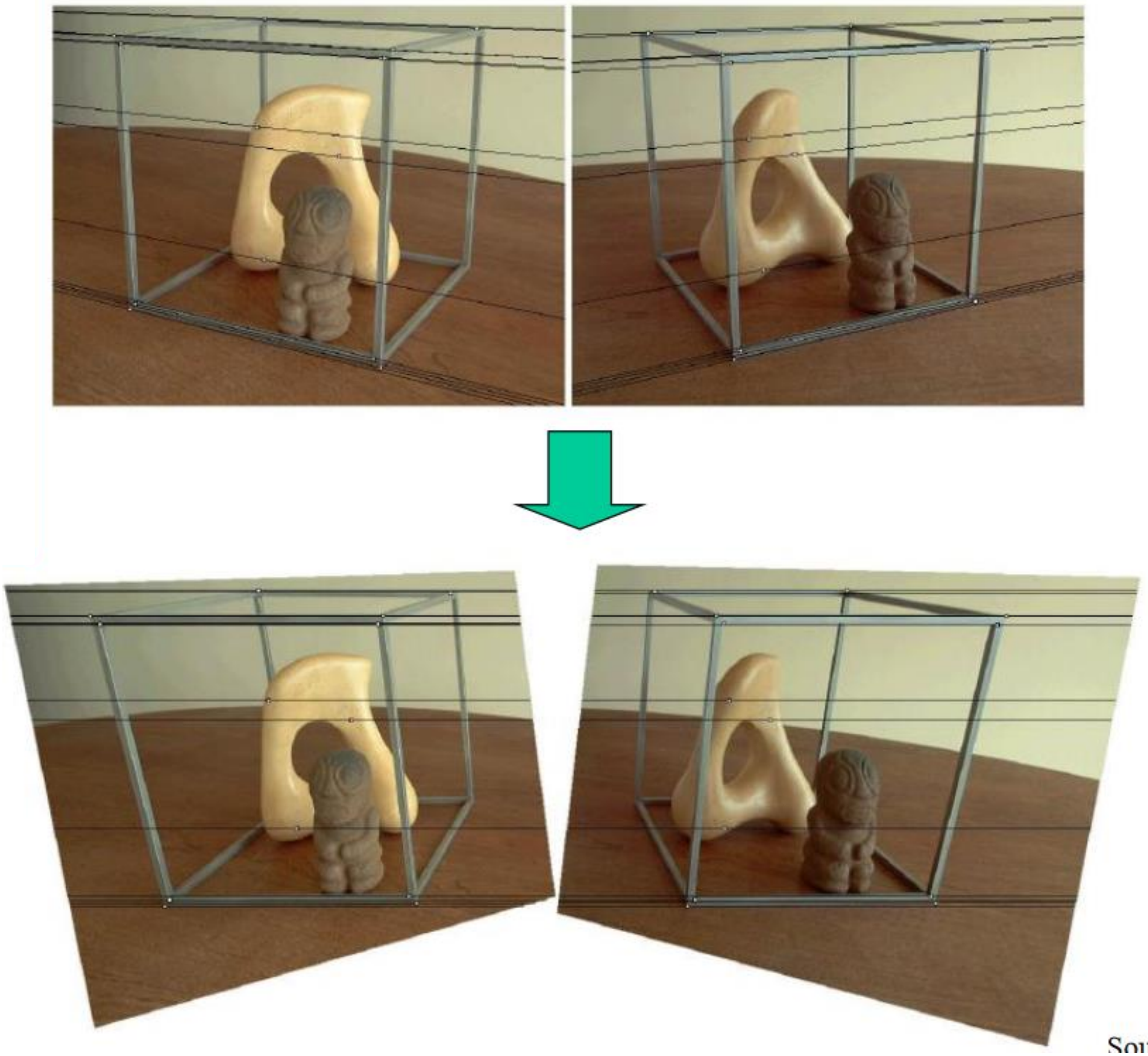
- Reproject image planes onto a common plane parallel to the line between optical centers
- Epipolar line is horizontal after this transformation
- Two homographies (3x3 transforms), one for each input image reprojection, is computed. See:
  - ➢ C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](). IEEE Conf. Computer Vision and Pattern Recognition, 1999.

Stereo image rectification

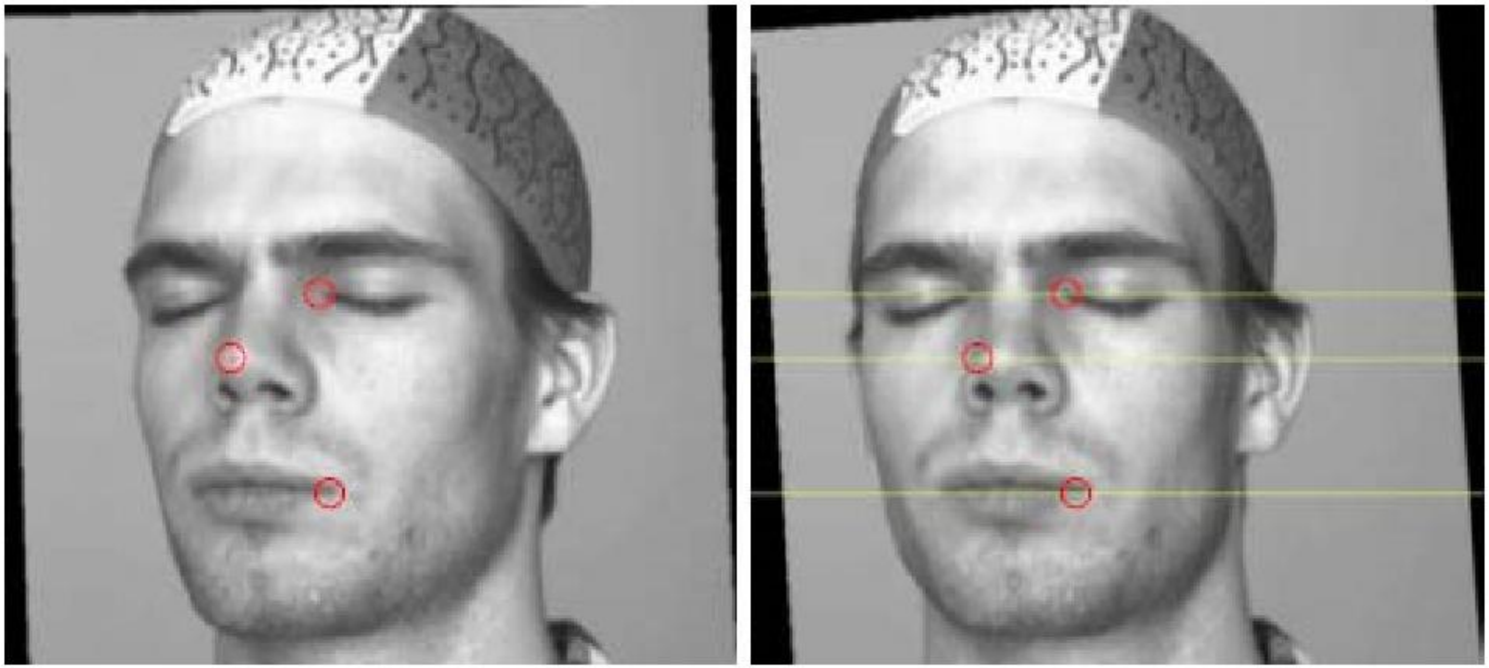


1. Compute line PQ - intersection between the original image planes.
2. compute the baseline vector *CC*' - between camera centres.
3. compute the equation of the plane that contains PQ and parallel to *CC*'.

Example

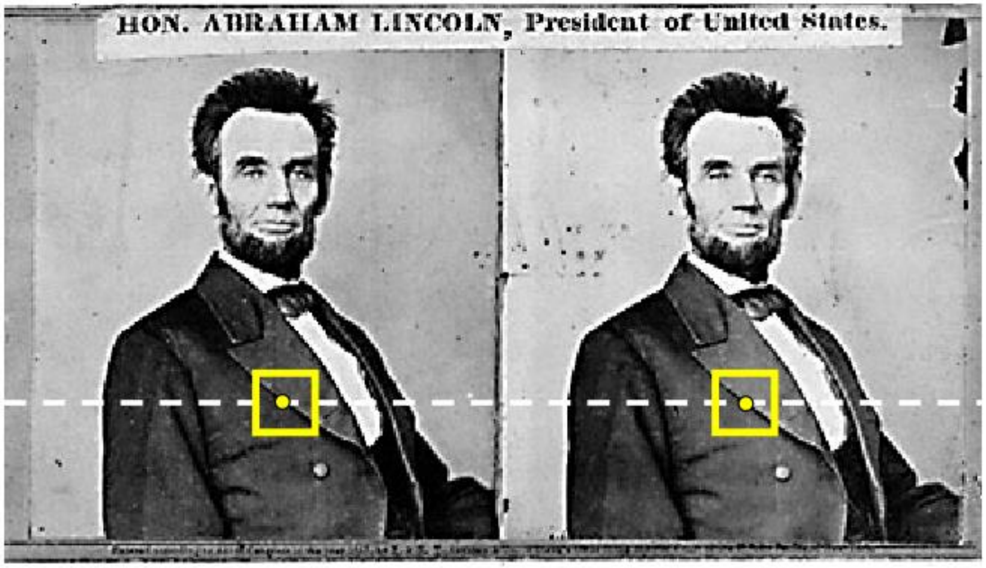


Example



After rectification, need only search for matches along horizontal scan line

Your basic stereo algorithm



For each epipolar line

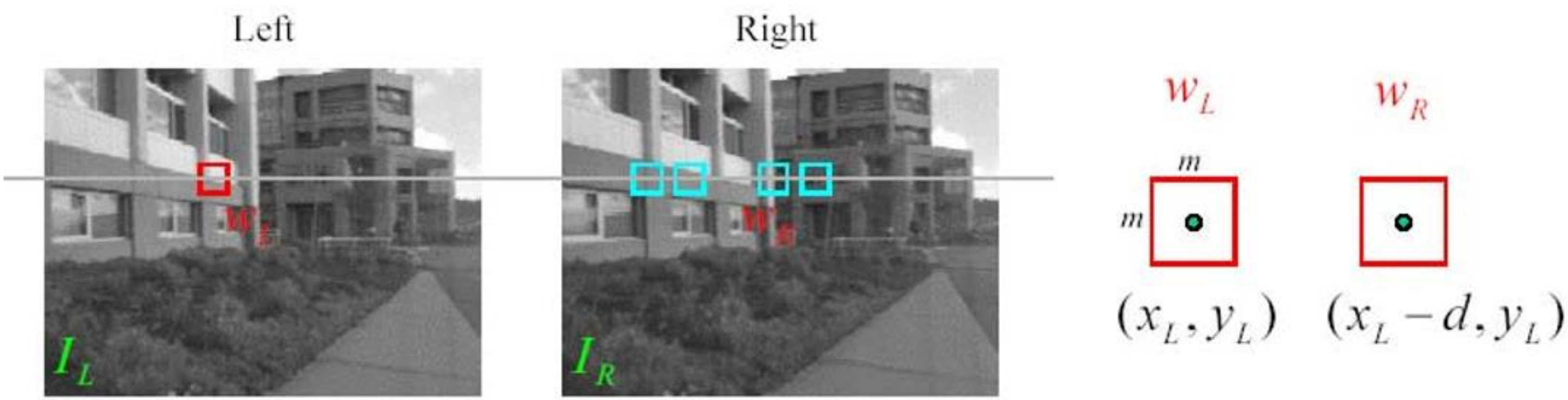
For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

A good survey and evaluation: http://vision.middlebury.edu/stereo/

Matching using Sum of Squared Differences (SSD)



w_L and w_R are corresponding m by m windows of pixels.

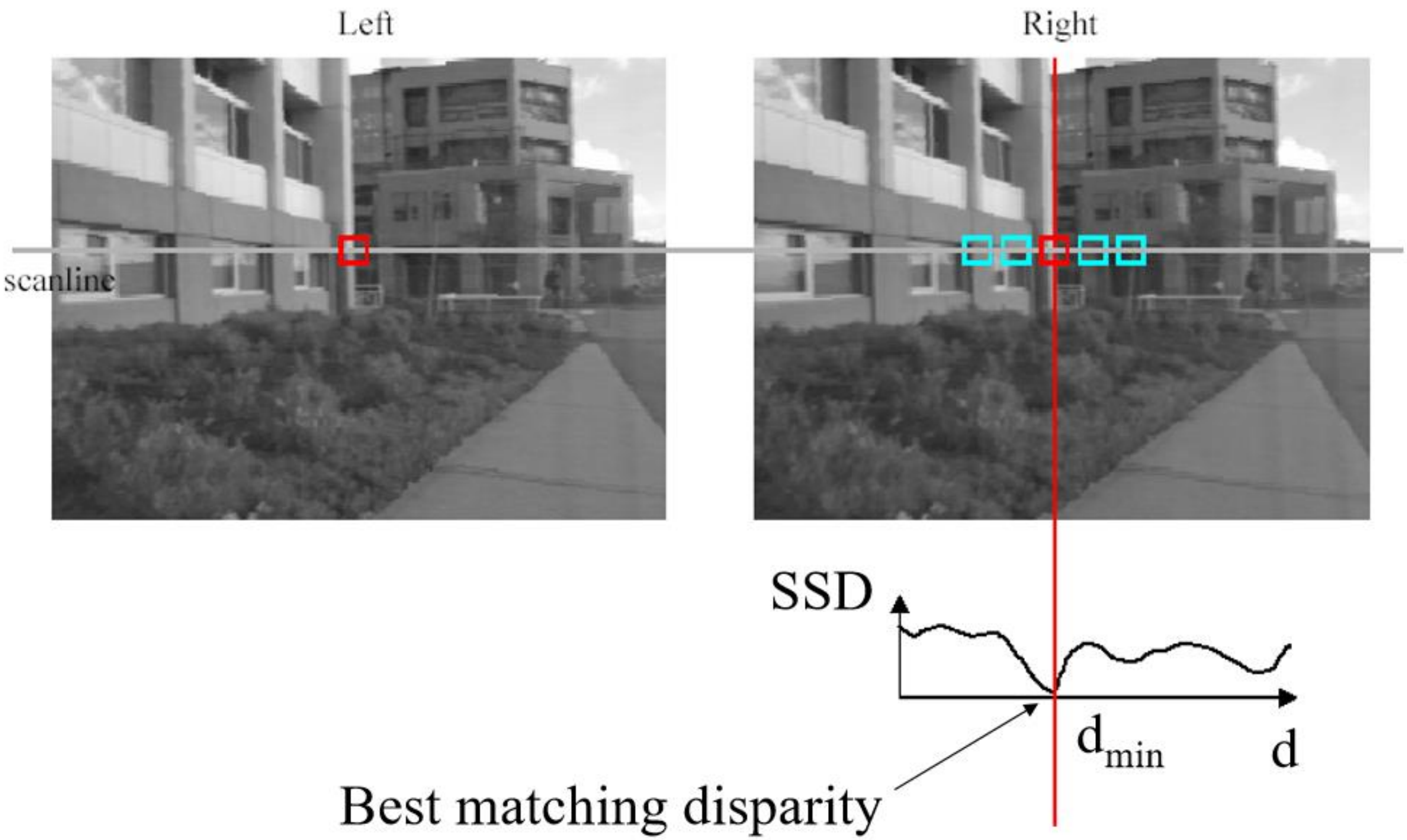
We define the window function :

$$W_m(x, y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2}, y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x, y, d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u, v) - I_R(u - d, v)]^2$$

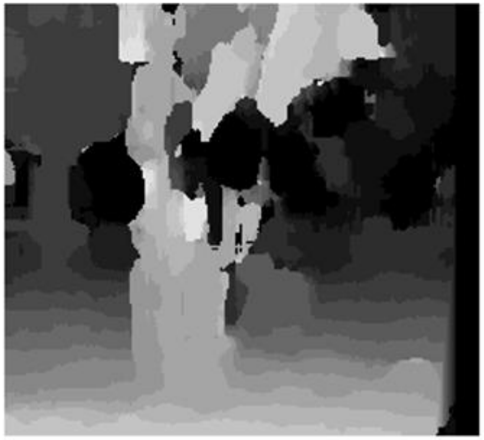
Stereo matching based on SSD



Problems with window size

Input stereo pair

W = 3



W = 20

Effect of window size W

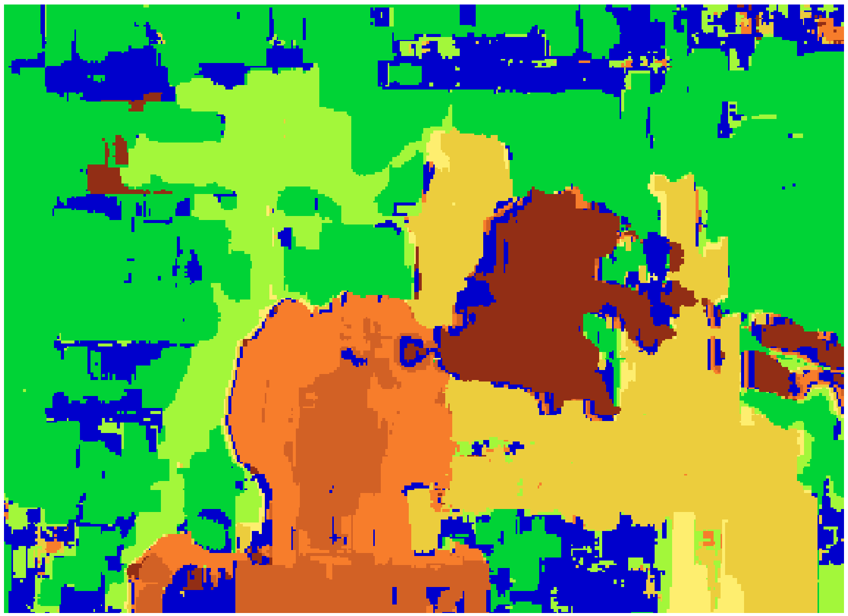
- Smaller window
  - \+ Good precision, more detail
  - – Sensitive to noise
- Larger window
  - \+ Robust to noise
  - – Reduced precision, less detail

3D Scene and disparity map

Scene

Ground truth

Results with window-based stereo matching



Window-based matching
(best window size)

Ground truth

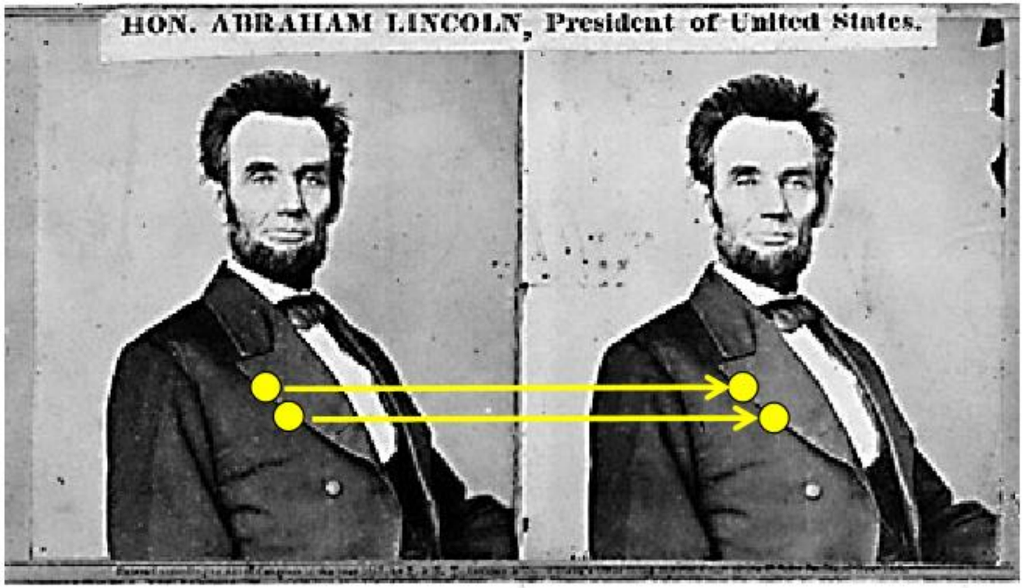
Better methods exist...

State of the art method:

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, 1999

Ground truth

State of the art: Stereo as energy minimization



What defines a good stereo correspondence?

1. Match quality
   – Want each pixel to find a good match in the other image
2. Smoothness
   – If two pixels are adjacent, they should (usually) be displaced about the same amount i.e., have similar disparities

Stereo as energy minimization

Expressing this mathematically

1. Match quality
   - Want each pixel to find a good match in the other image

$$matchCost = \sum_{x,y} \|I(x,y) - J(x + d_{xy}, y)\|$$

2. Smoothness
   - If two pixels are adjacent, they should has similar disparities

$$smoothnessCost = \sum_{neighbor\ pixels\ p,q} |d_p - d_q|$$

We want to minimize $Energy = matchCost + smoothnessCost$

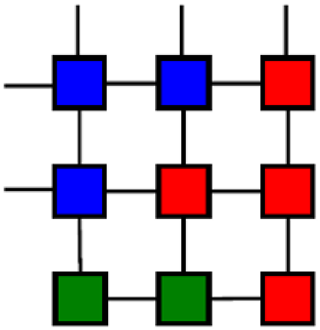
Stereo as energy minimization

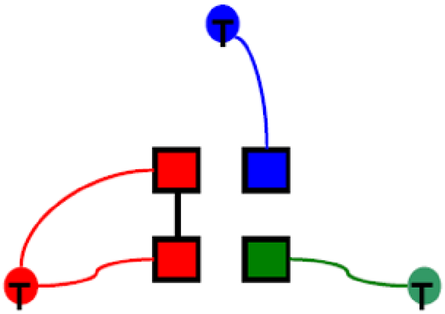
We want to minimize:

$$Energy = matchCost + smoothnessCost$$

- This is a special type of energy function known as an MRF (Markov Random Field)
  - Effective and fast algorithms have been recently developed:
    - » Graph cuts, belief propagation….
    - » for more details (and code): http://vision.middlebury.edu/MRF/

Image as a graph with disparity labels





Min-cost graph cut yields a labeling of each pixel with best disparity

Stereo reconstruction pipeline

Steps

- Calibrate cameras
- Rectify images
- Compute disparity
- Estimate depth

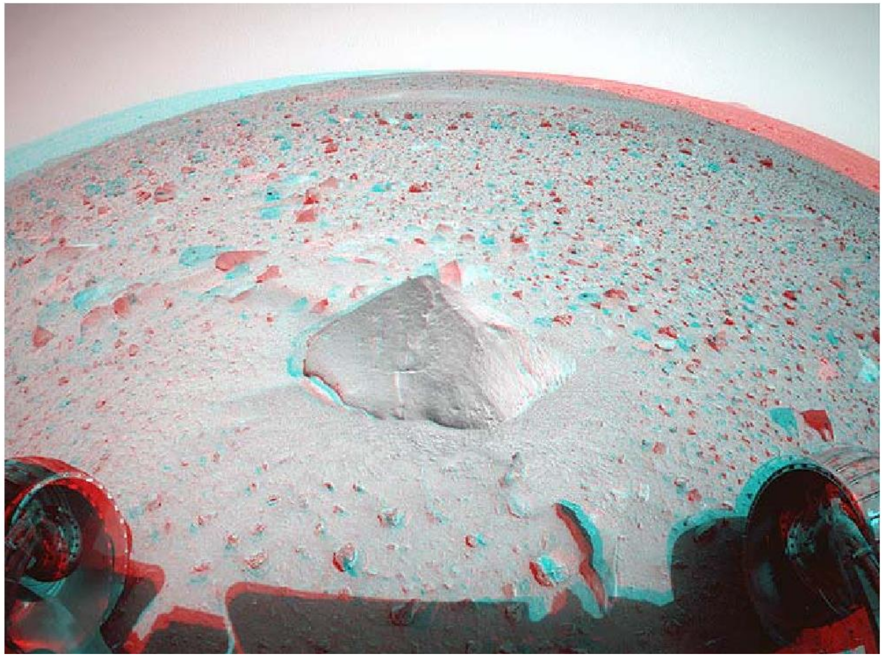
What will cause errors?

- Camera calibration errors
- Poor image resolution
- Occlusions
- Violations of brightness constancy (specular reflections)
- Large motions
- Low-contrast image regions

Robot navigation and 3D scene capture

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

Anaglyph from Mars Rover

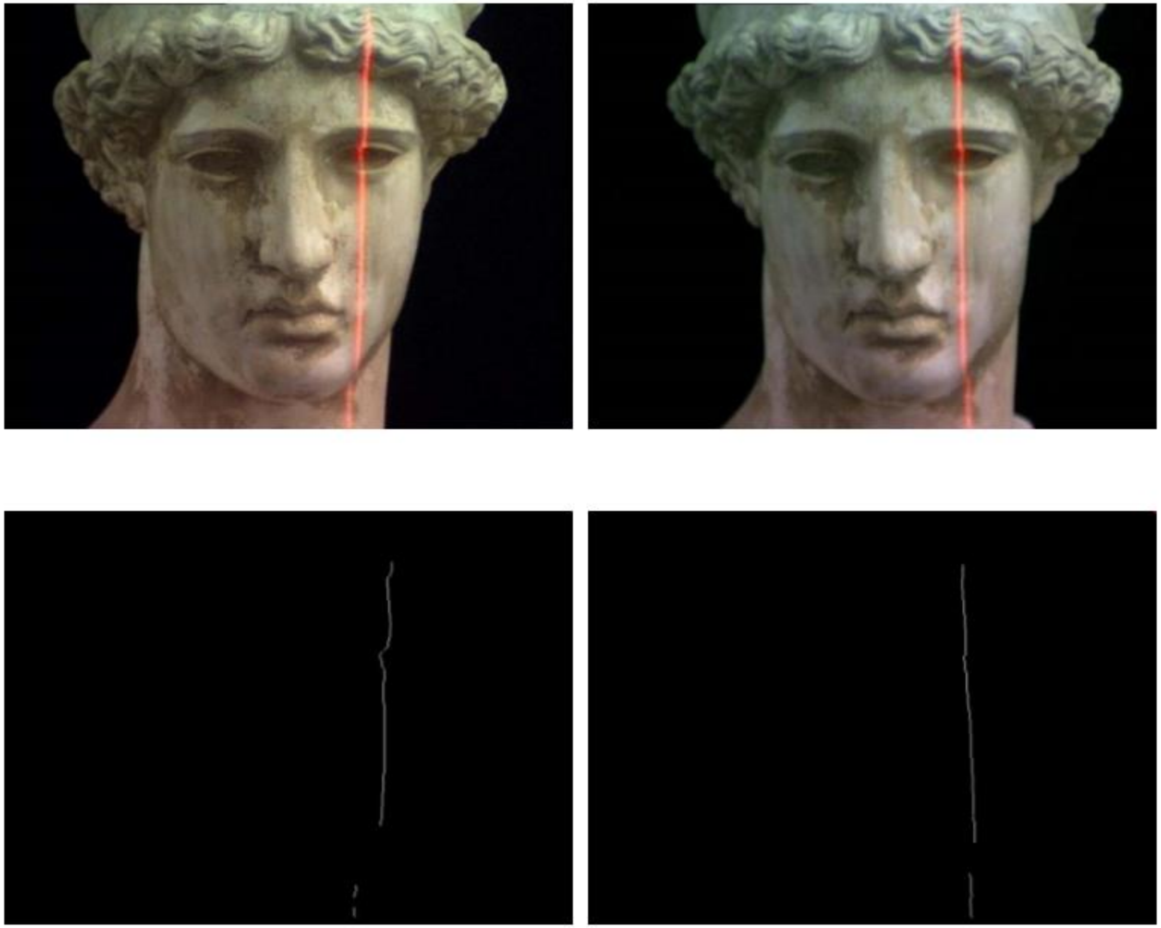


What if 3D object has little or no texture?

Matching points might be difficult or impossible

Can we still recover depth information?

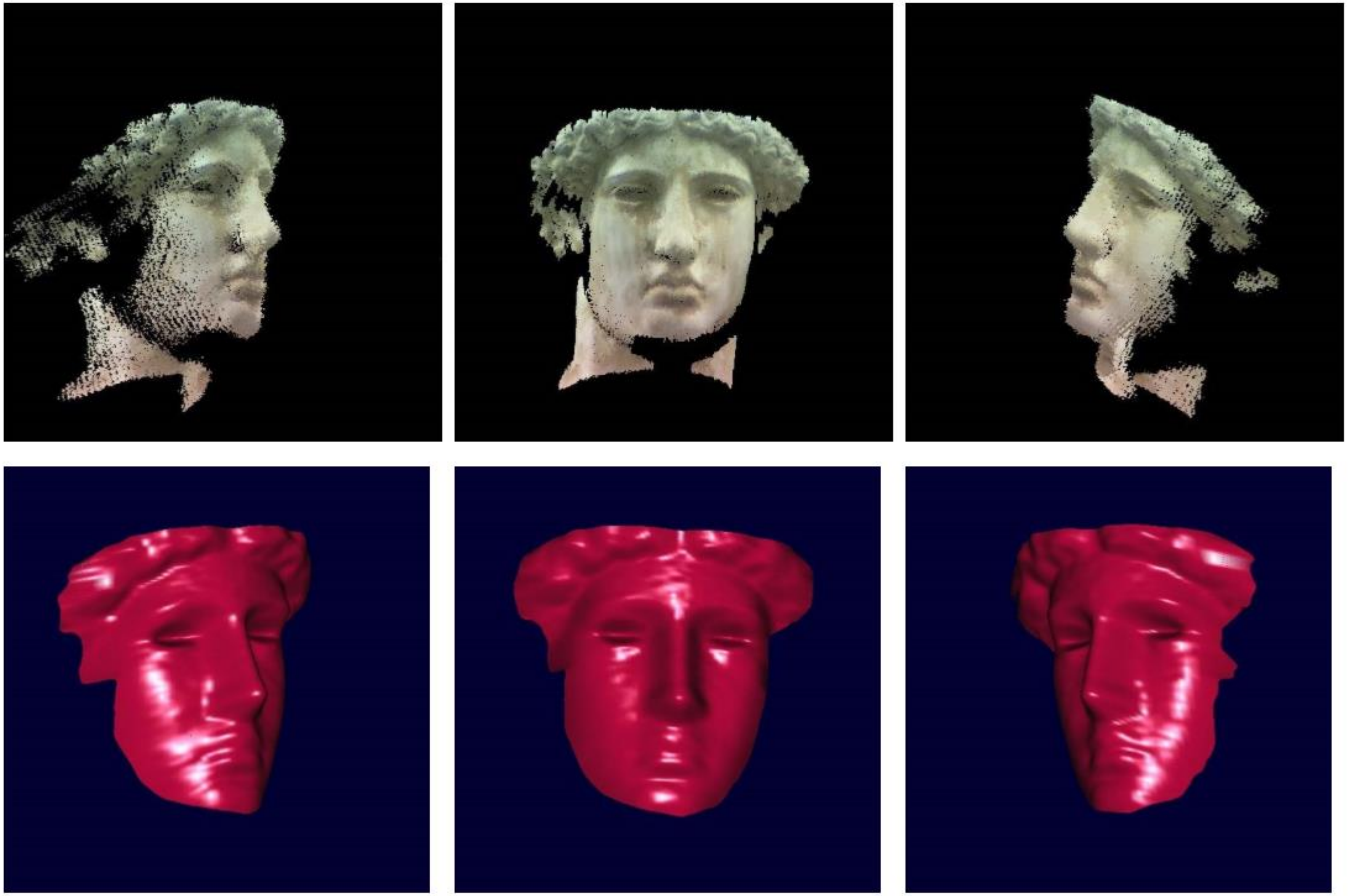
Idea: Use structured light!



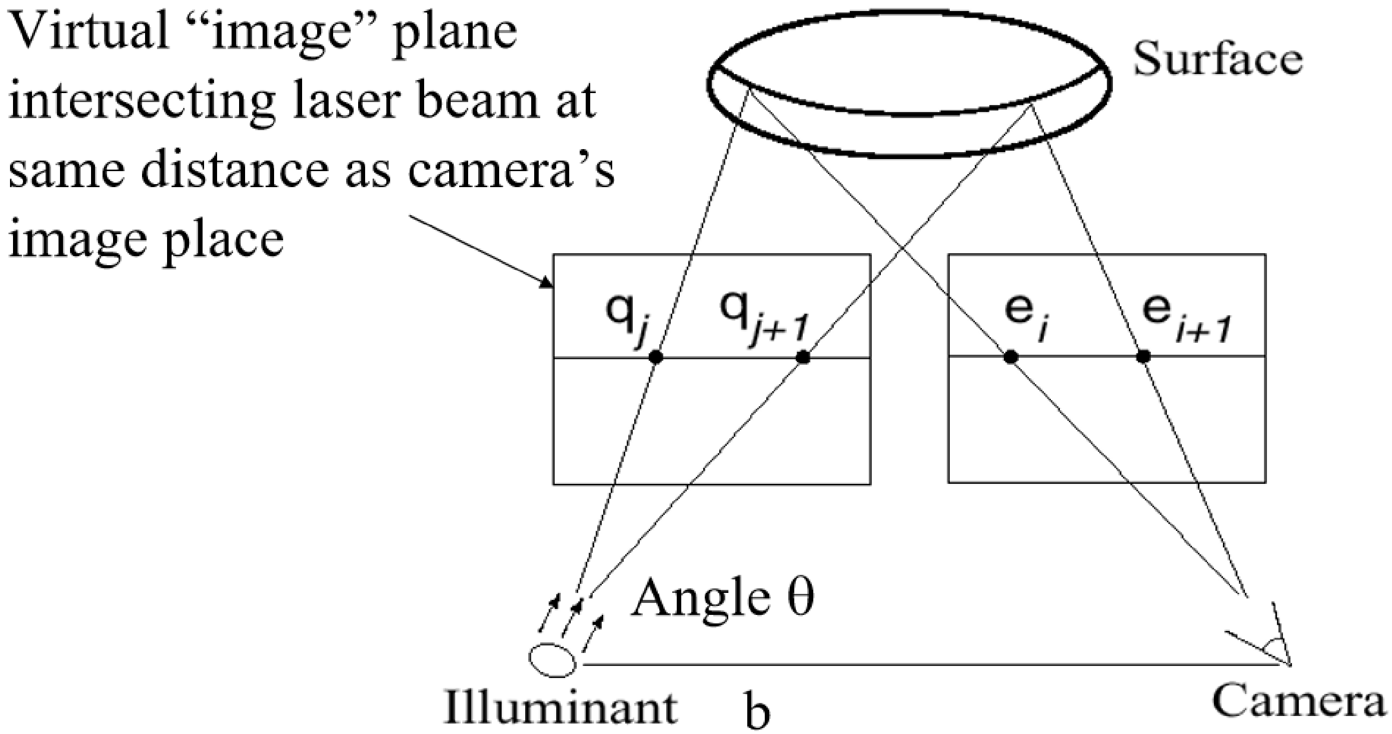
Disparity between laser points on the same scanline in the images determines the 3-D coordinates of the laser point on object

http://www.cs.wright.edu/~agoshtas/stereorange.html

Recovered 3D Model

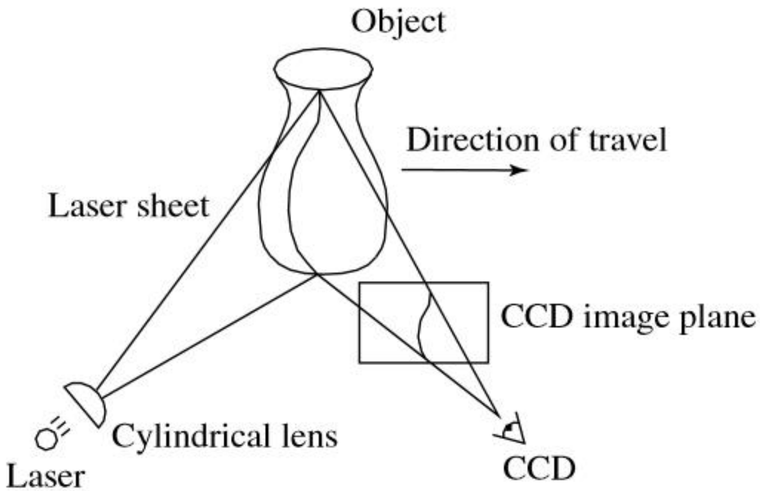


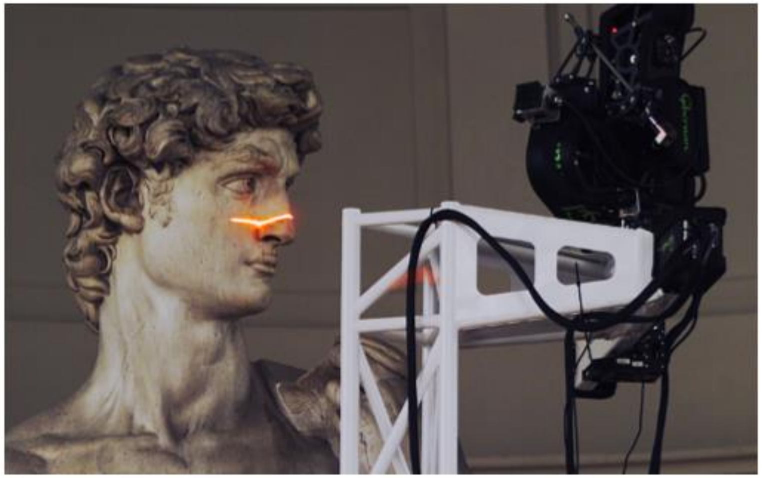
Actually, we can make do with just 1 camera



From calibration of both camera and light projector, we can compute 3D coordinates laser points on the surface

The Digital Michelangelo Project





http://graphics.stanford.edu/projects/mich/

Optical triangulation

- Project a single stripe of laser light
- Scan it across the surface of the object

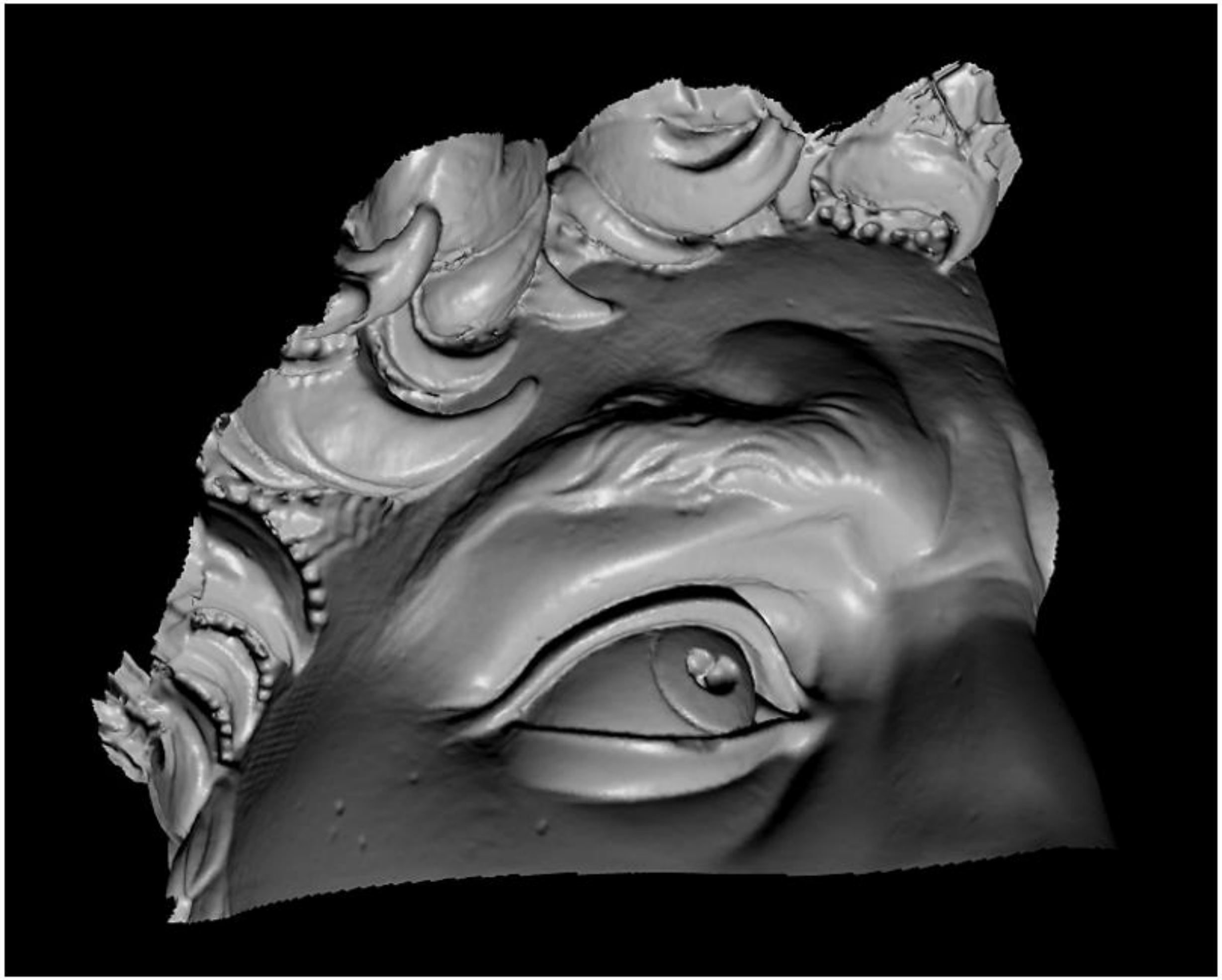
Laser scanned models

The Digital Michelangelo Project, Levoy et al.

Laser scanned models

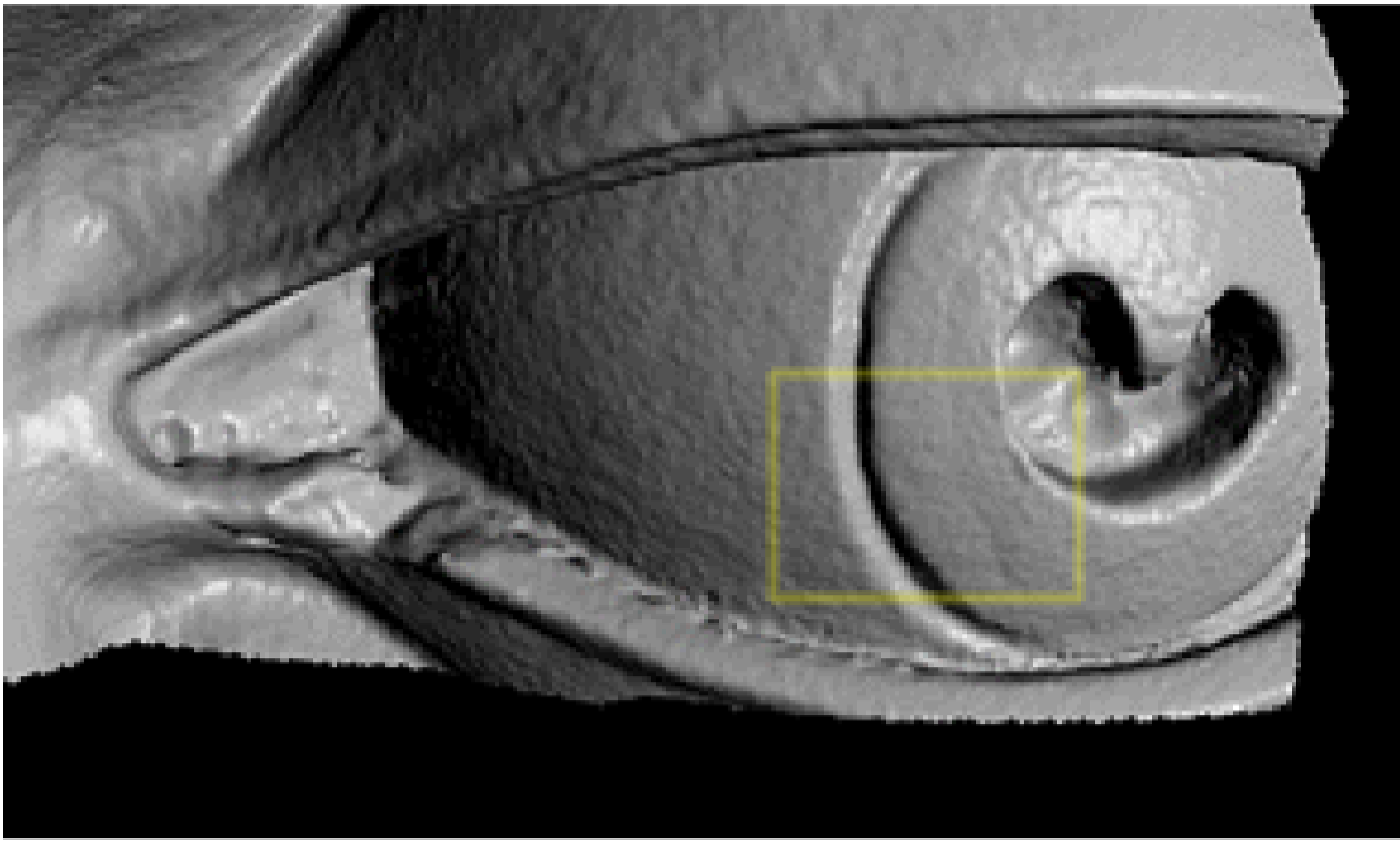
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



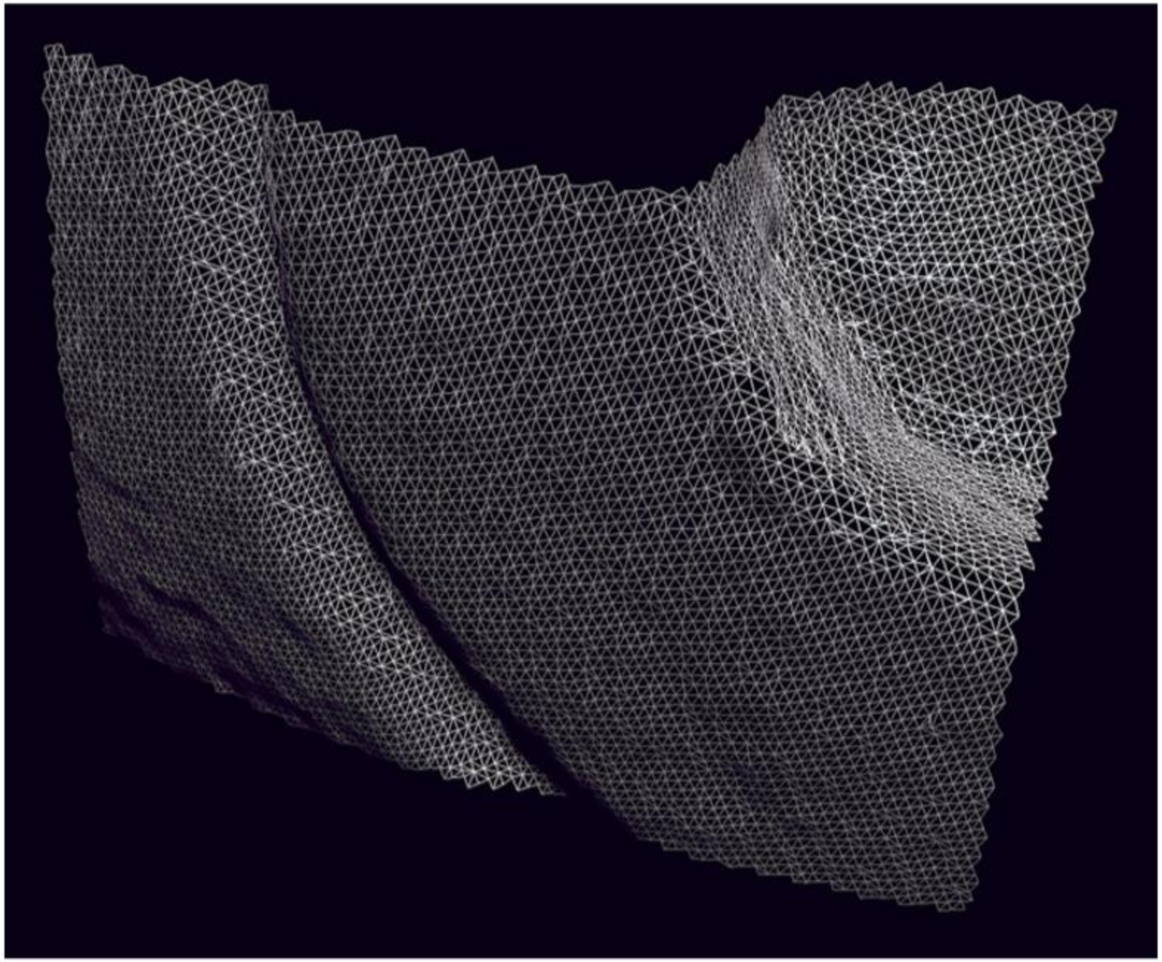
The Digital Michelangelo Project, Levoy et al.

Laser scanned models



The Digital Michelangelo Project

Laser scanned models



The Digital Michelangelo Project, Levoy et al.

Revision

- Stereo camera model
  - Projection
  - Triangulation
  - Epipolar constraints
  - Rectification
  - Stereo as energy minimization