SSY097 - Image Analysis

Lecture 9 - Camera Geometry

Torsten Sattler
(slides adapted from Olof Enqvist)

Last Lecture

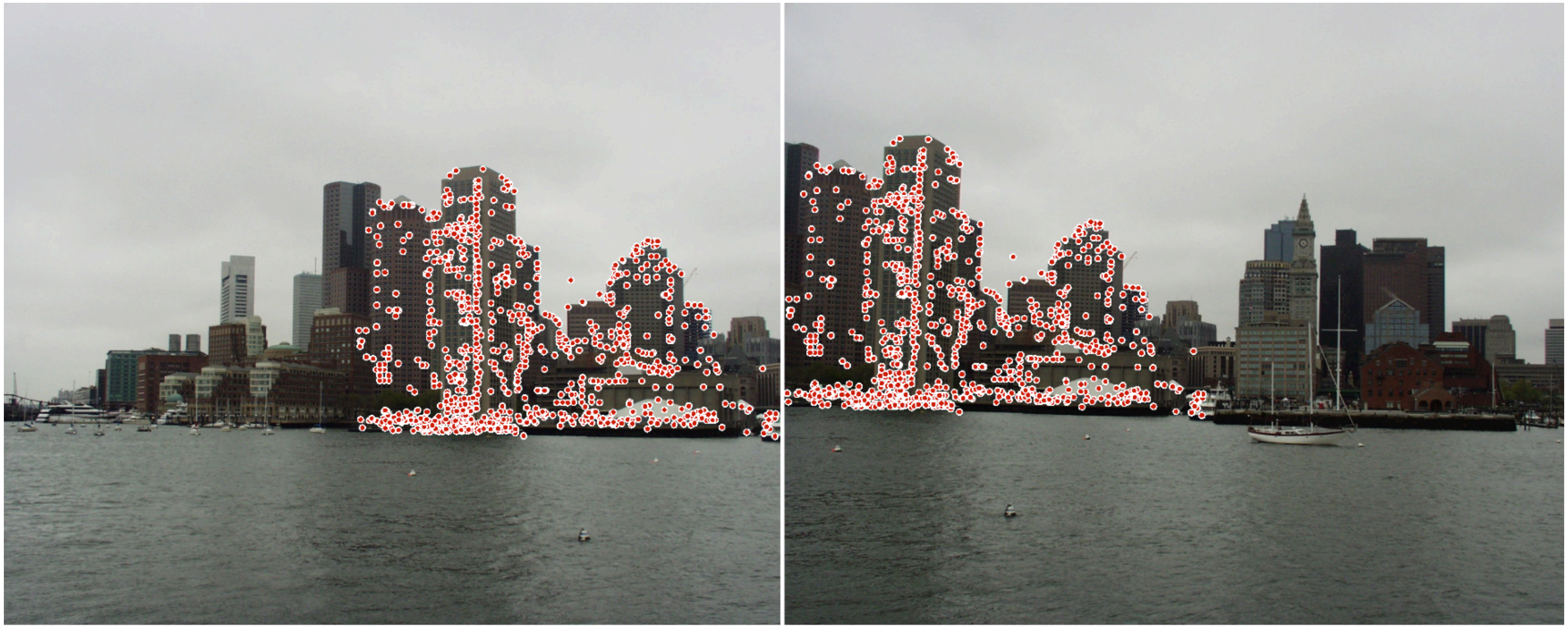


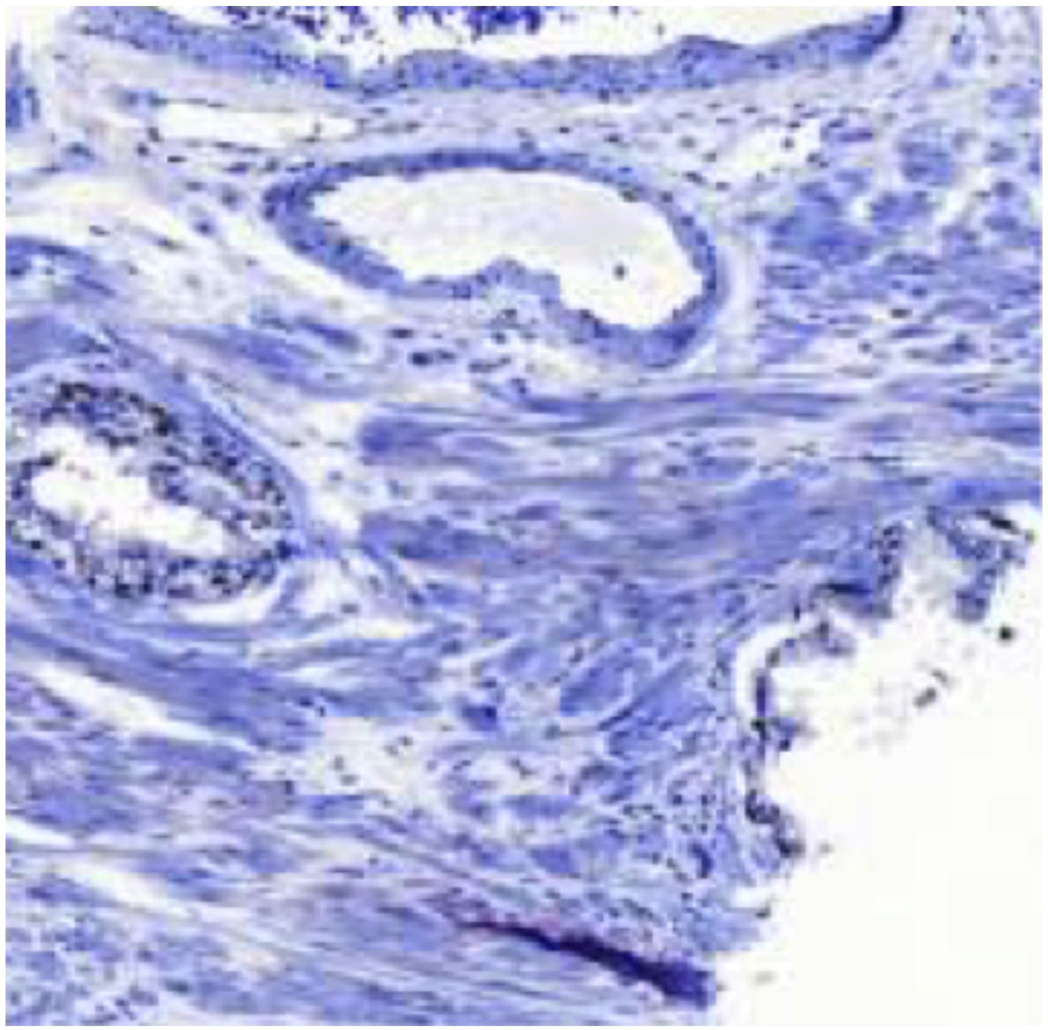
Image alignment

Last Lecture

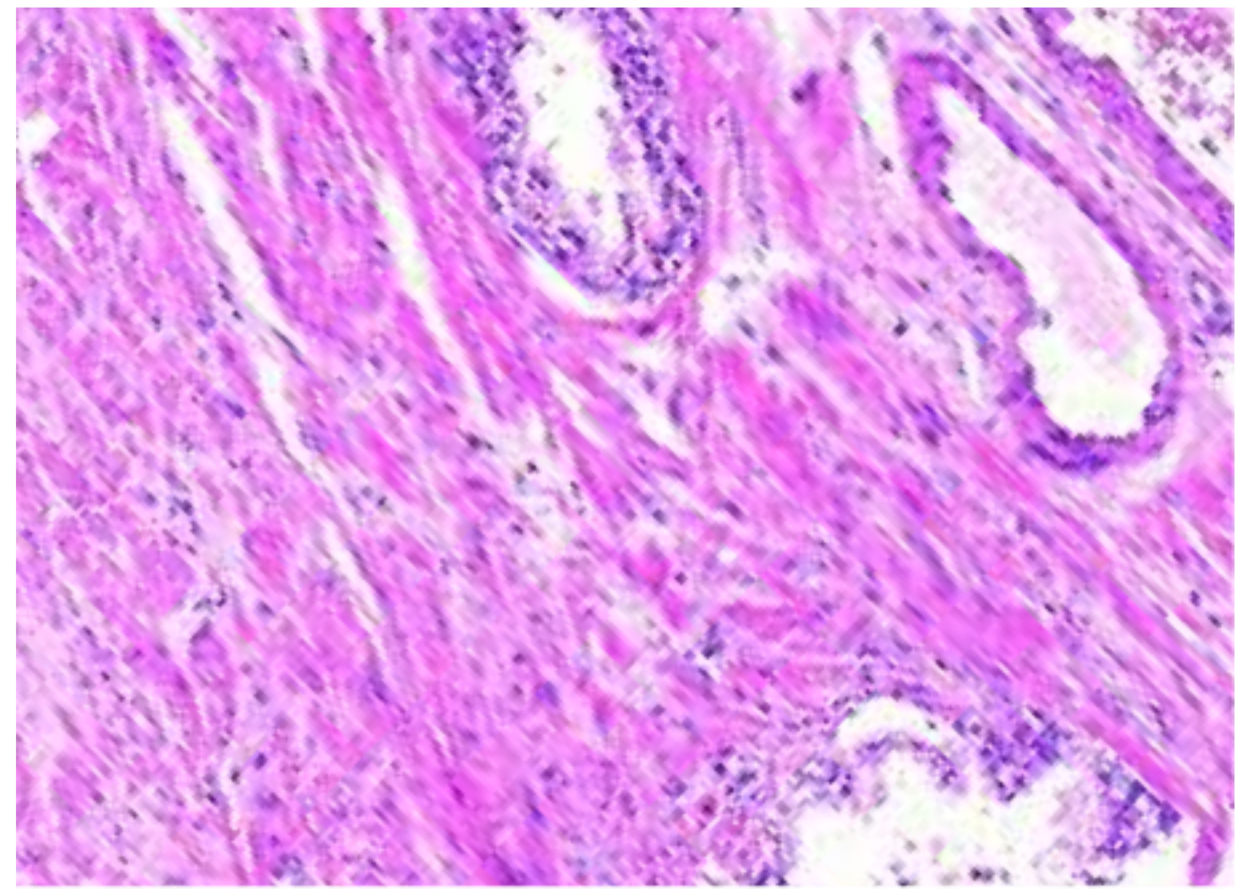
Image alignment

Last Lecture

transformation



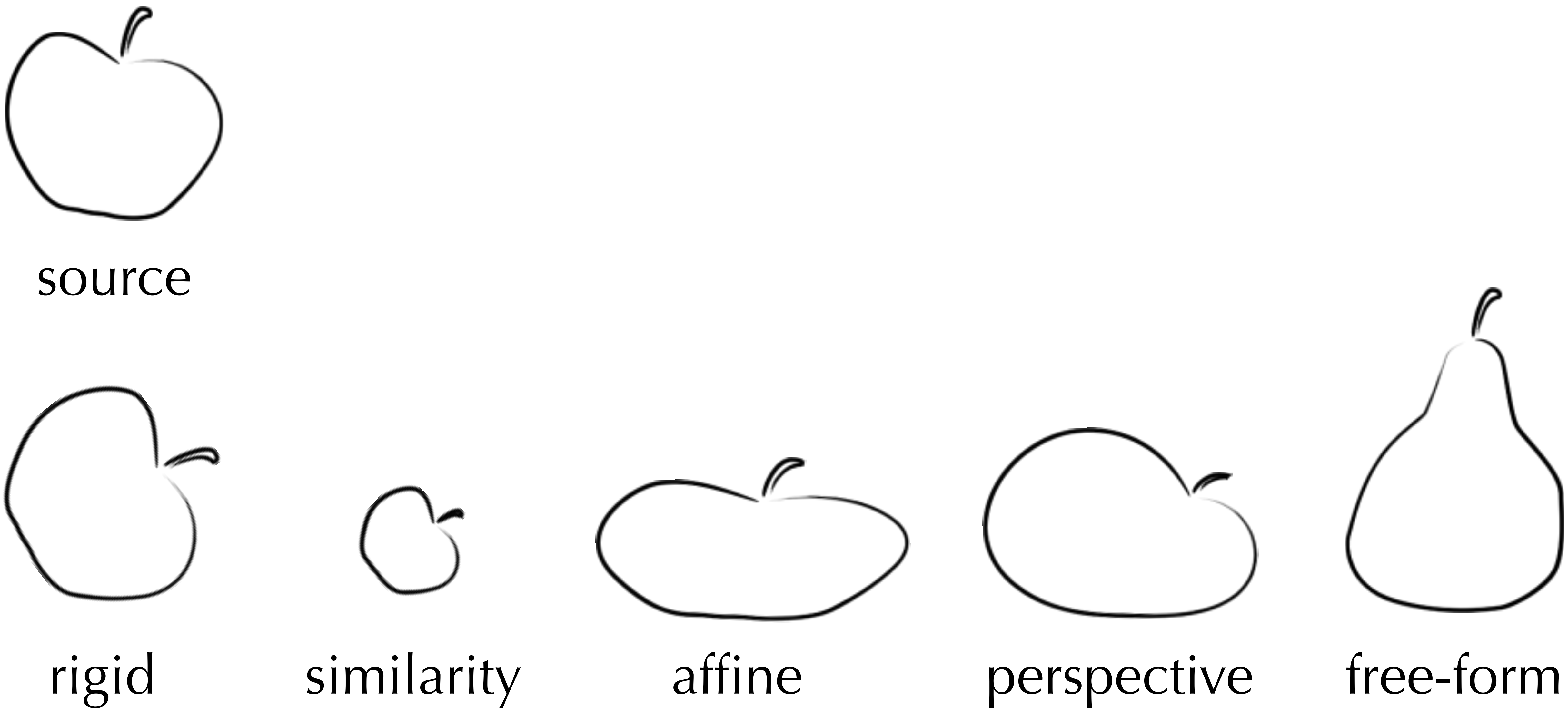
source image



target image

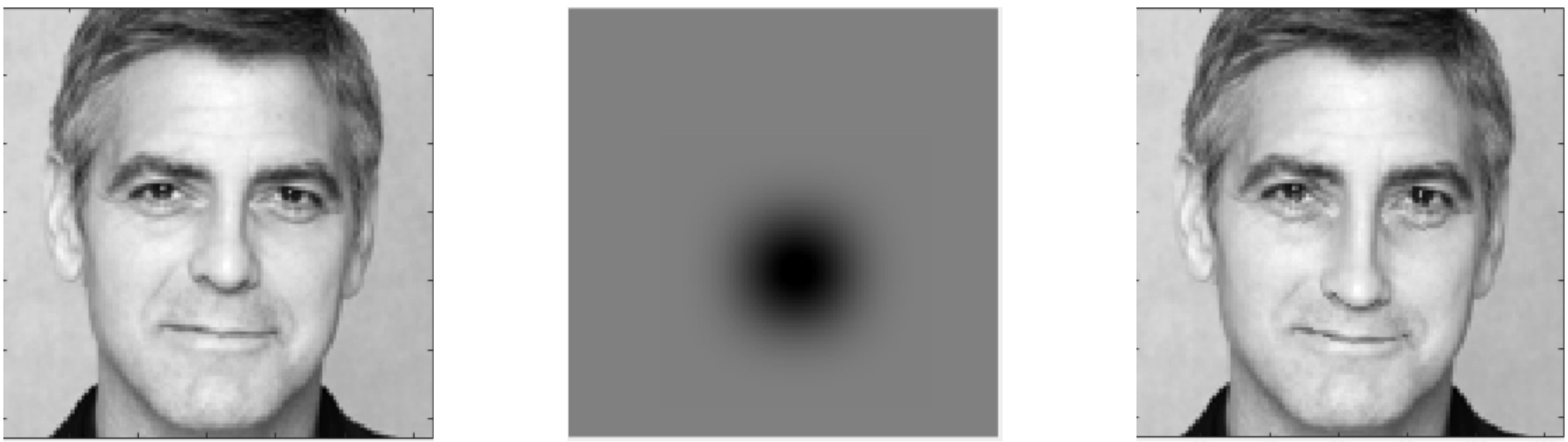
Pixel transfer

Last Lecture



Transformation hierarchy

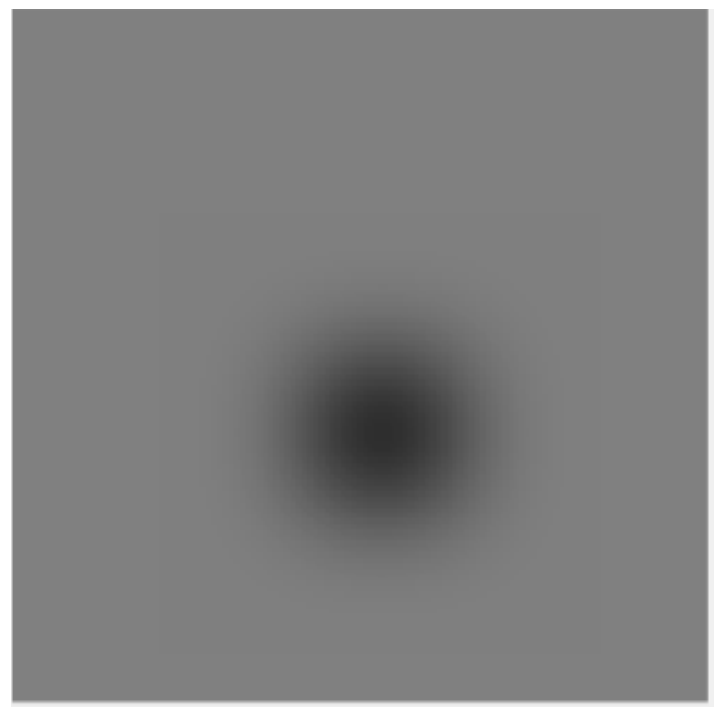
Last Lecture



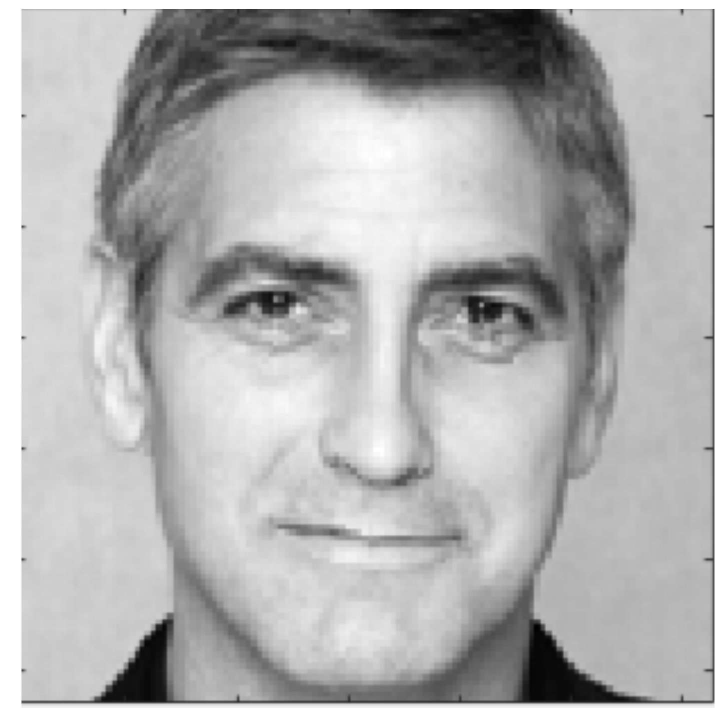
Δ_y

Free-form registration

Last Lecture

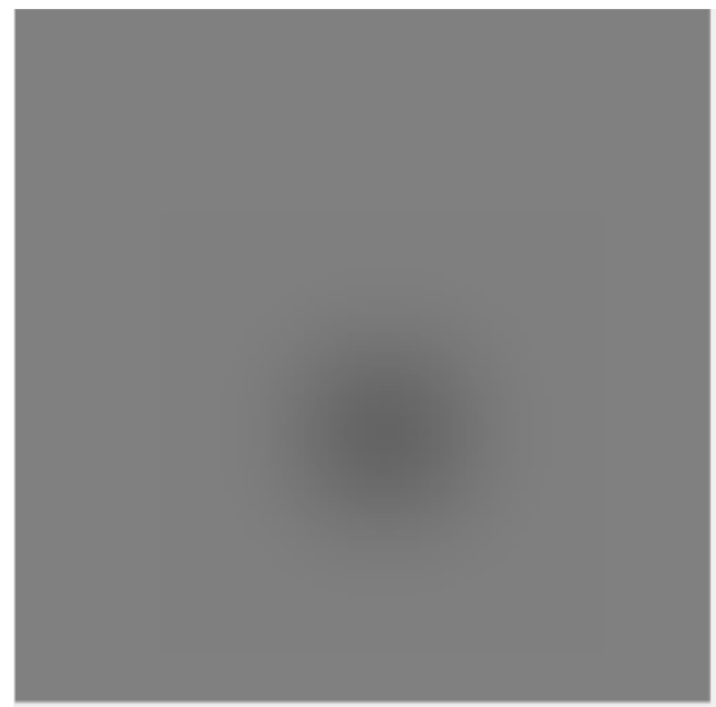


$$\Delta_y$$

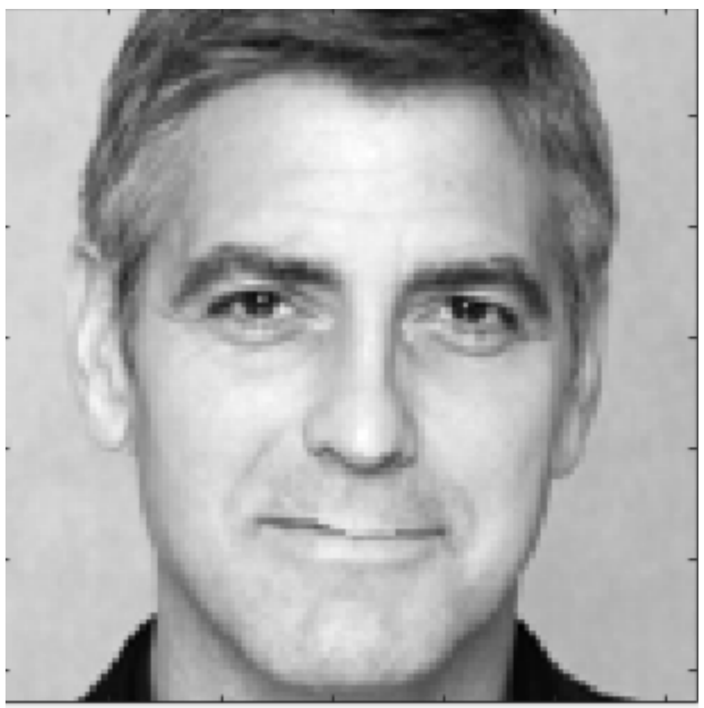


Free-form registration

Last Lecture

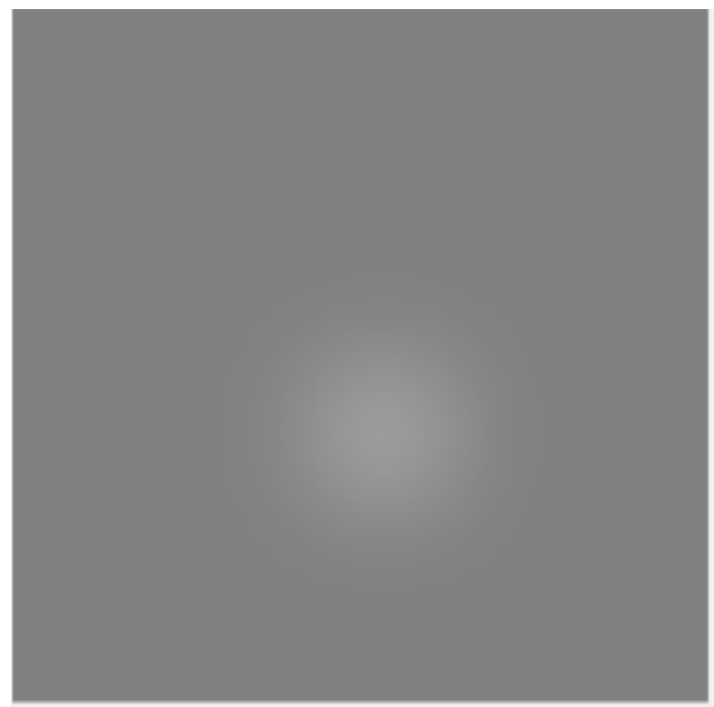


Δ_y

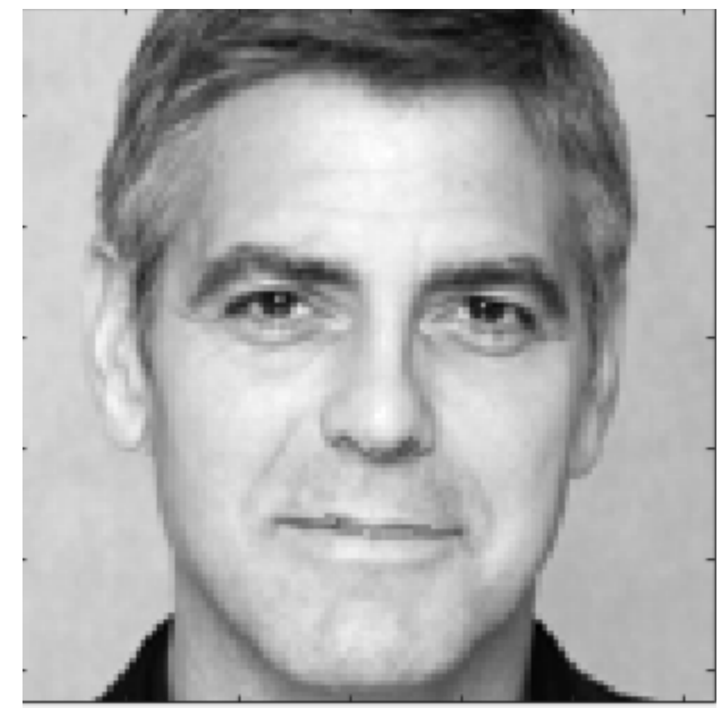


Free-form registration

Last Lecture

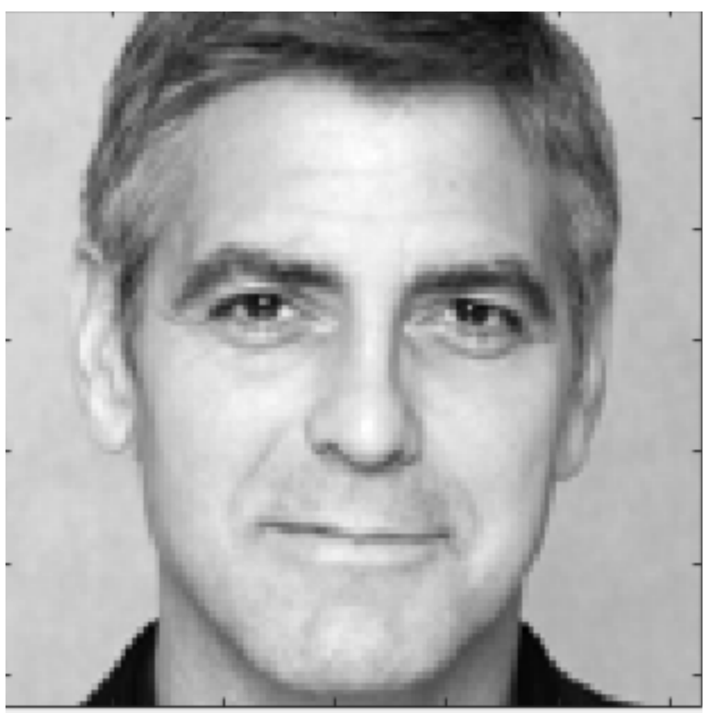


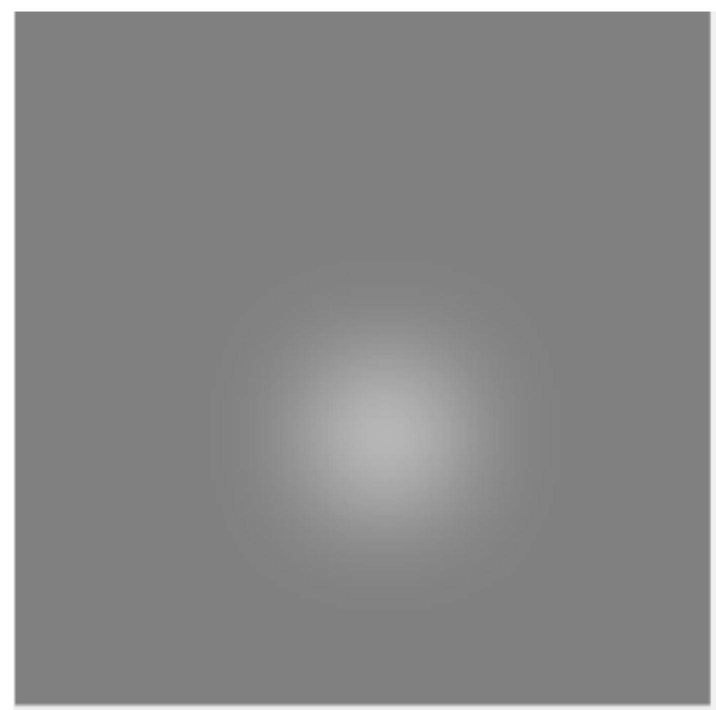
Δ_y



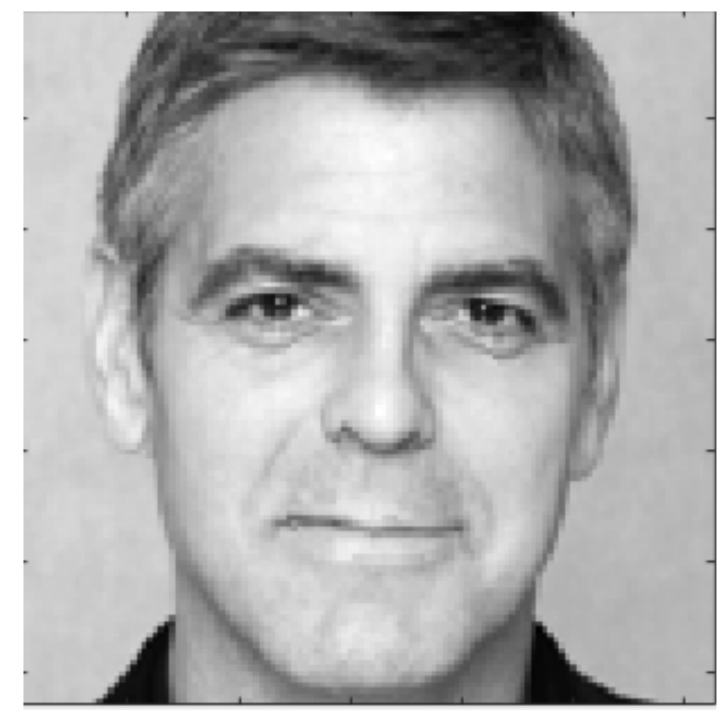
Free-form registration

Last Lecture



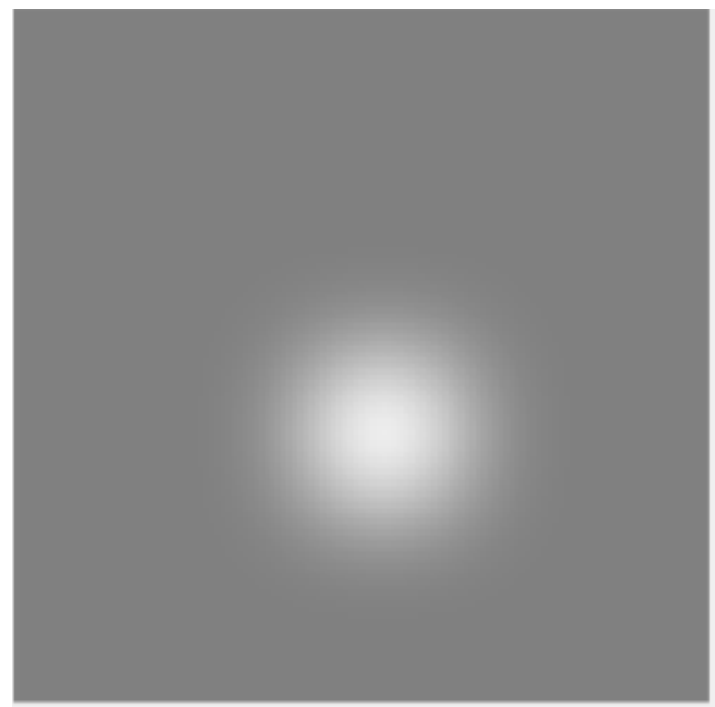
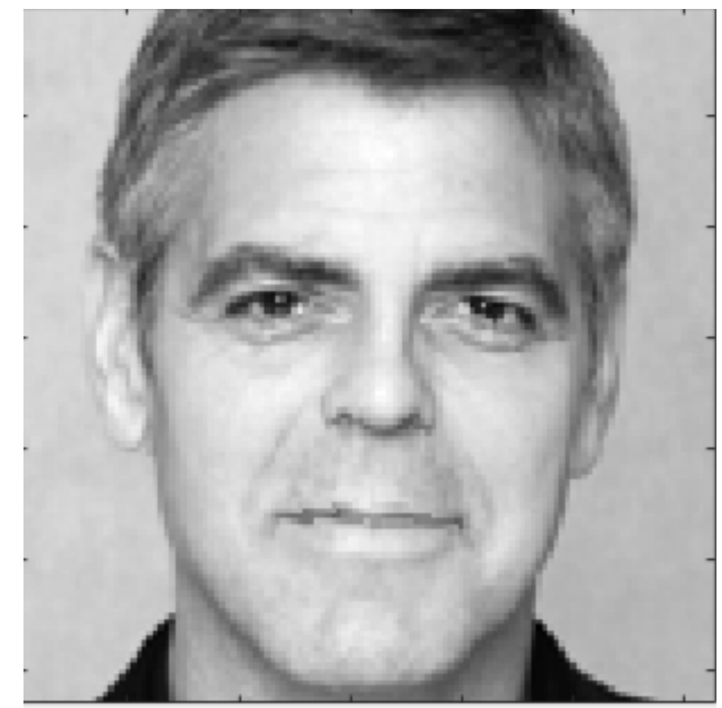


Δ_y



Free-form registration

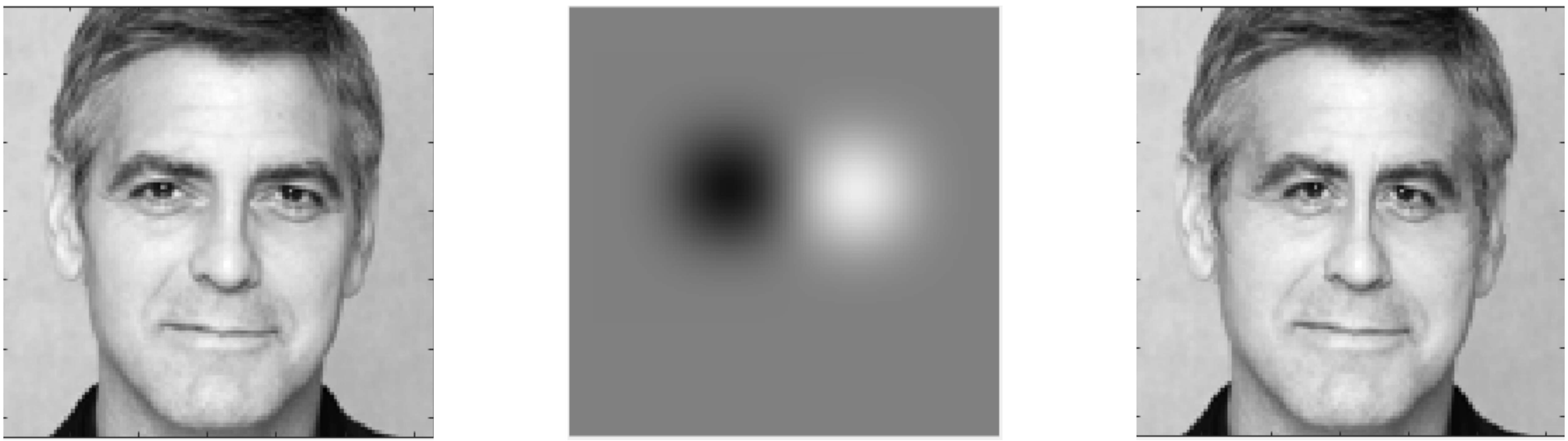
Last Lecture

Δ_y

Free-form registration

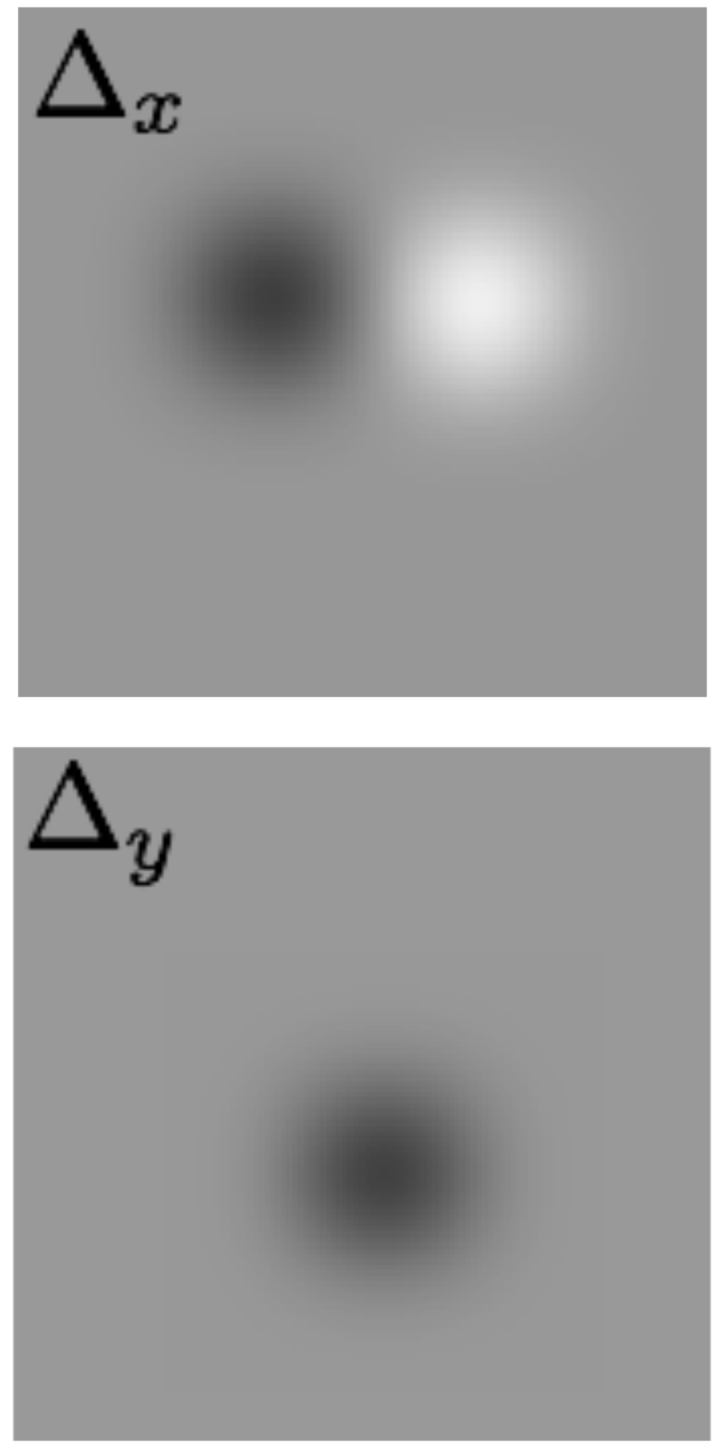
Last Lecture



Δ_x

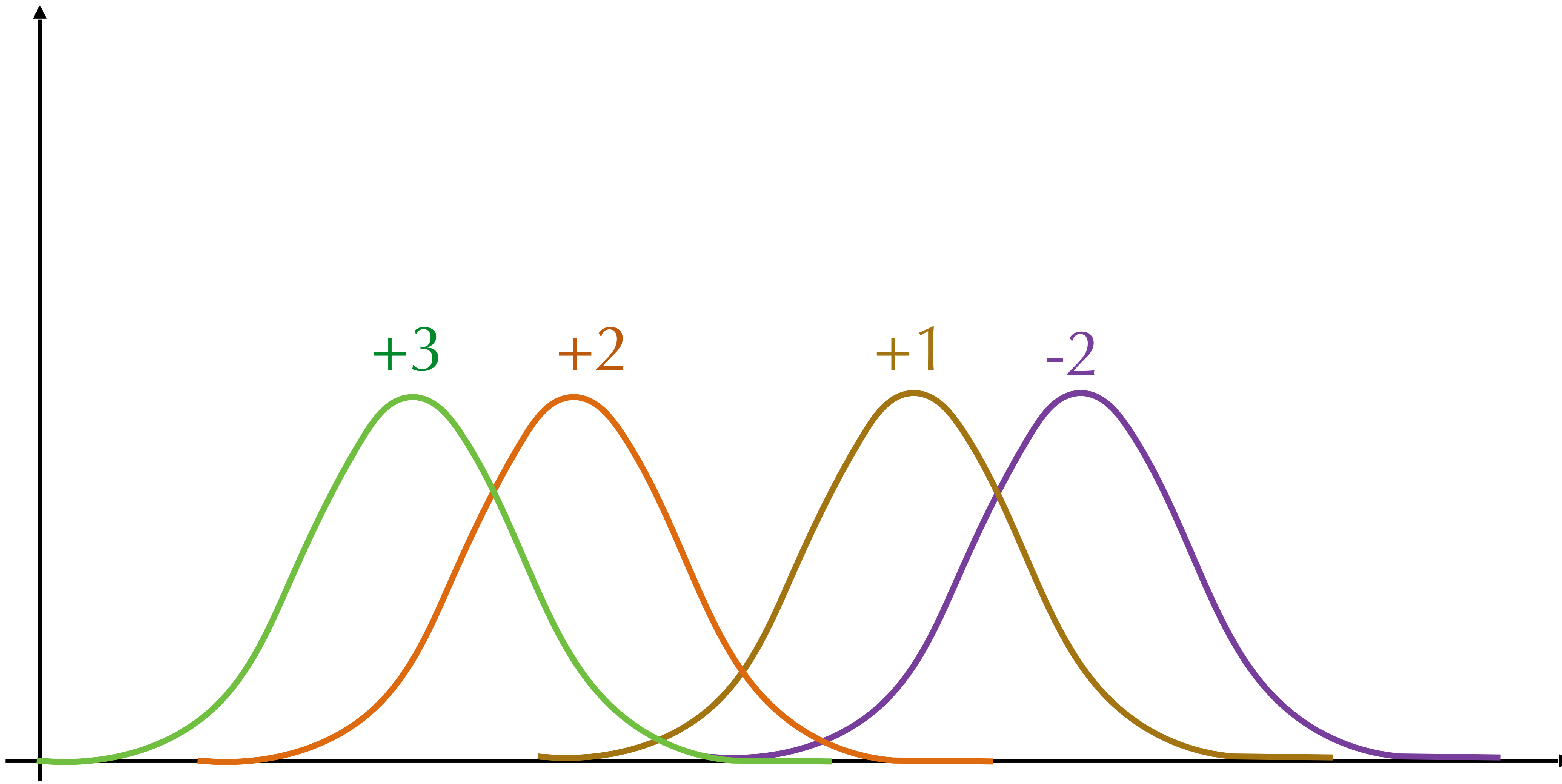
Free-form registration

Last Lecture



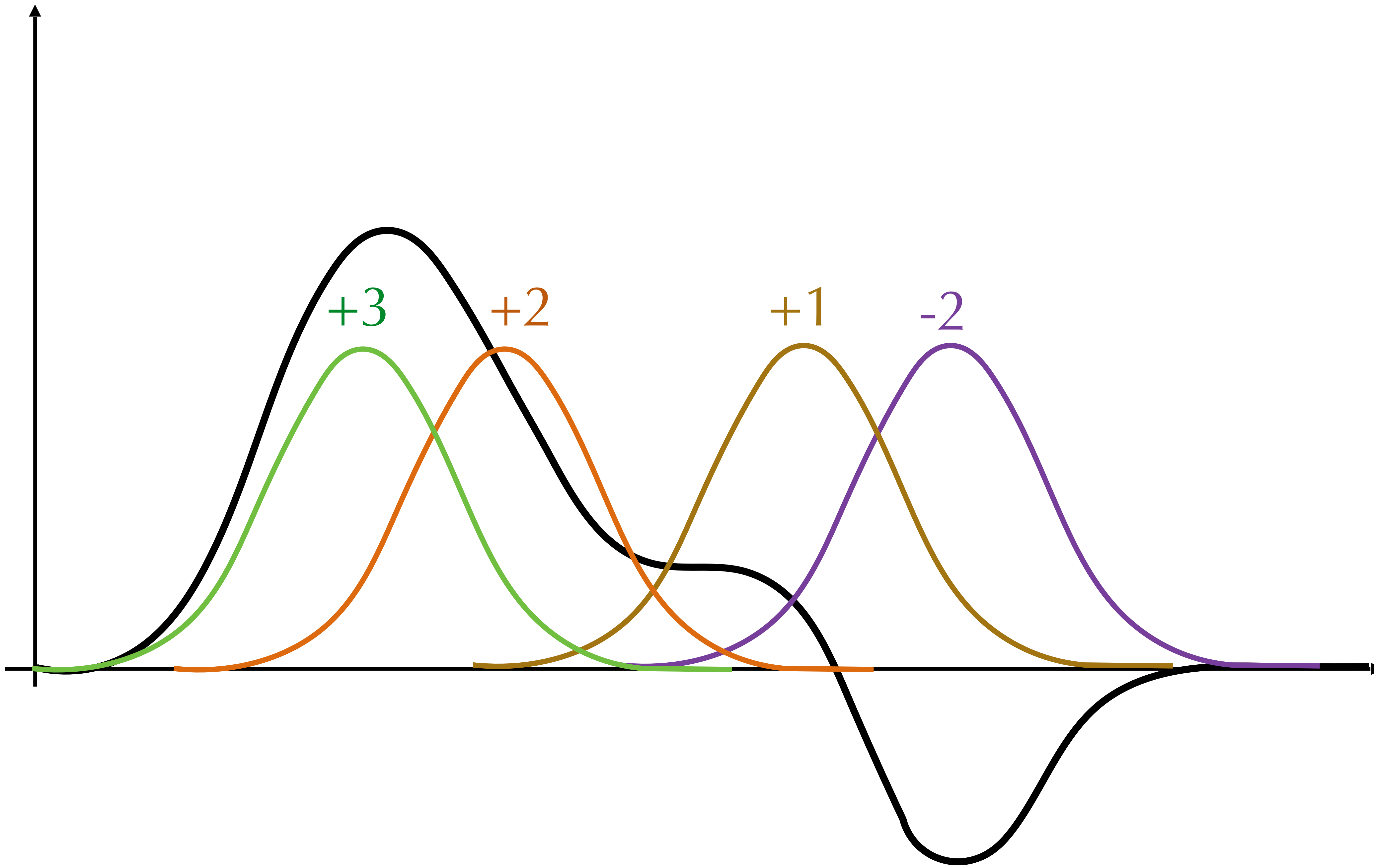
Free-form registration

Last Lecture



$$\Delta_x(x, y) = \sum_{k=1}^{n} \alpha_k B_k(x, y)$$

Last Lecture

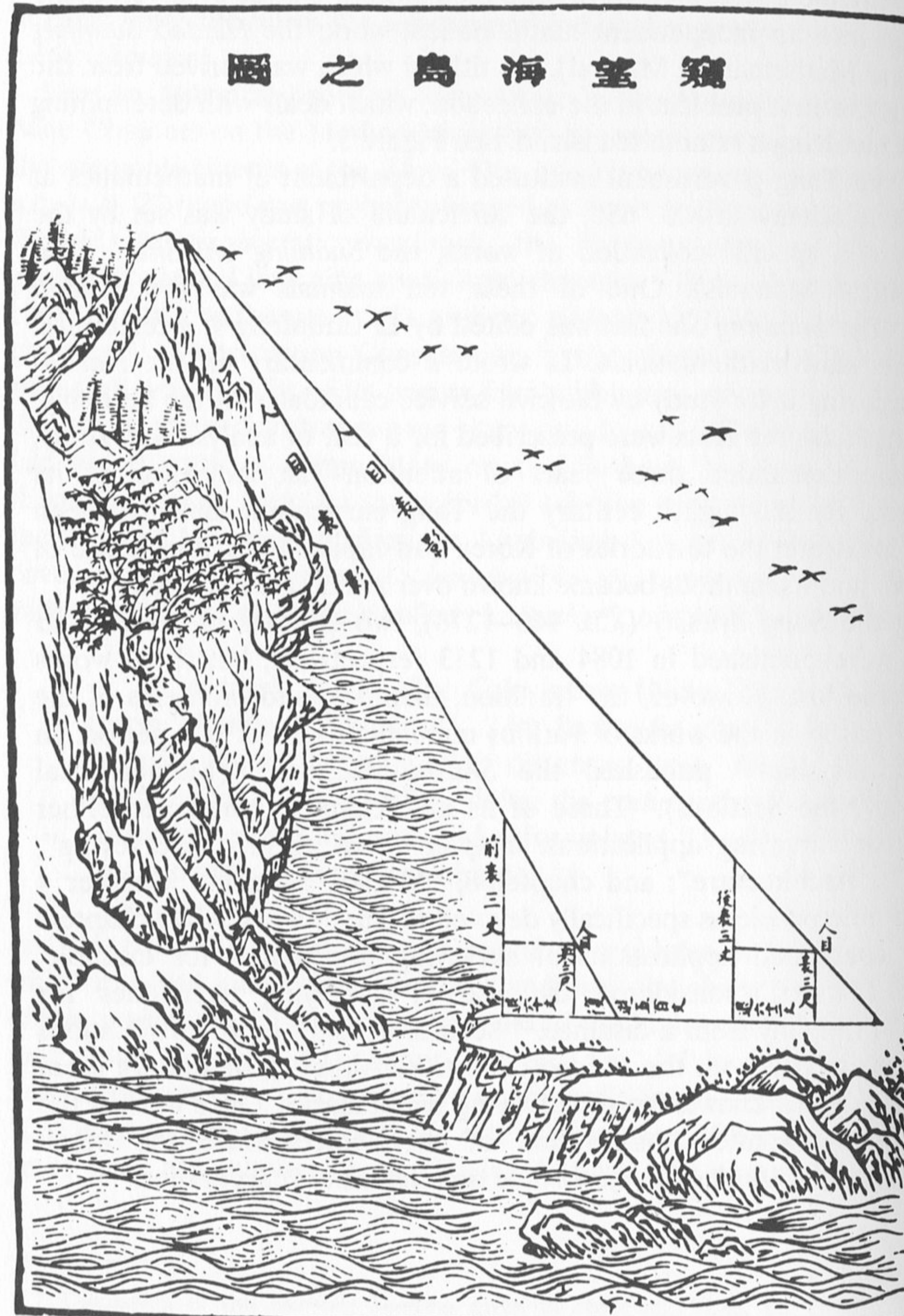


$$\Delta_x(x,y) = \sum_{k=1}^{n} \alpha_k B_k(x,y)$$

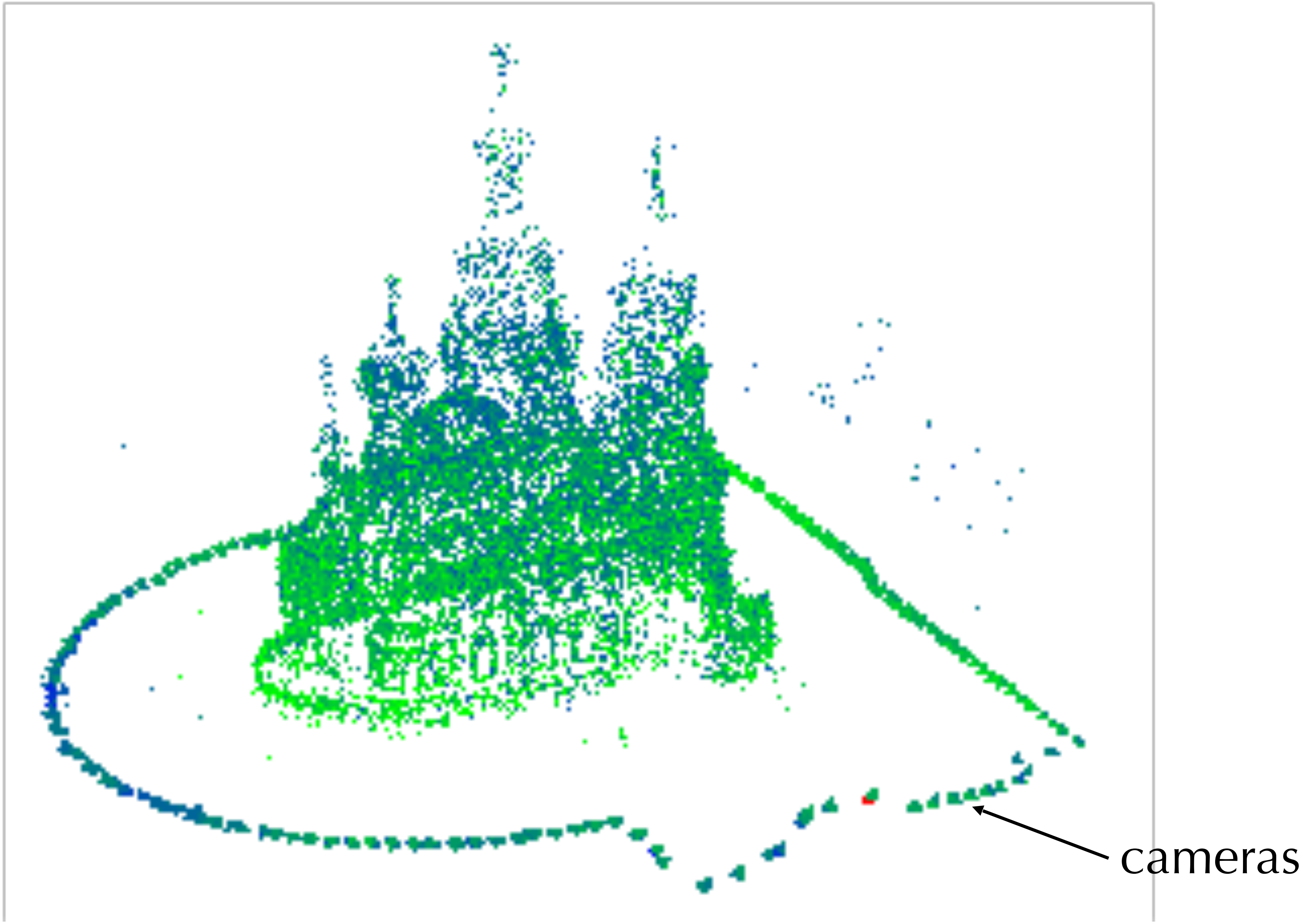
Today

Camera Geometry

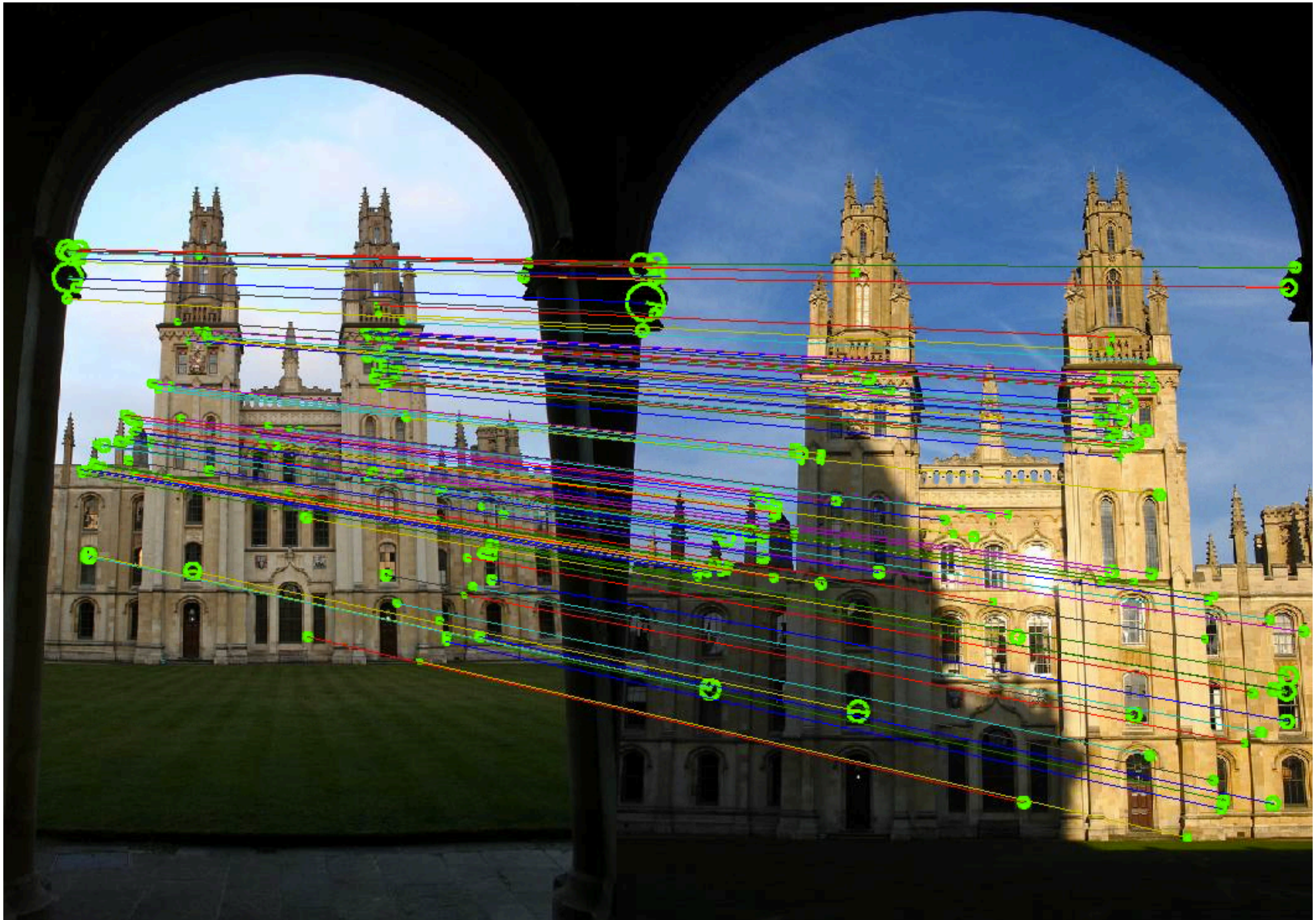
Triangulation



Structure-from-Motion



The Measurements



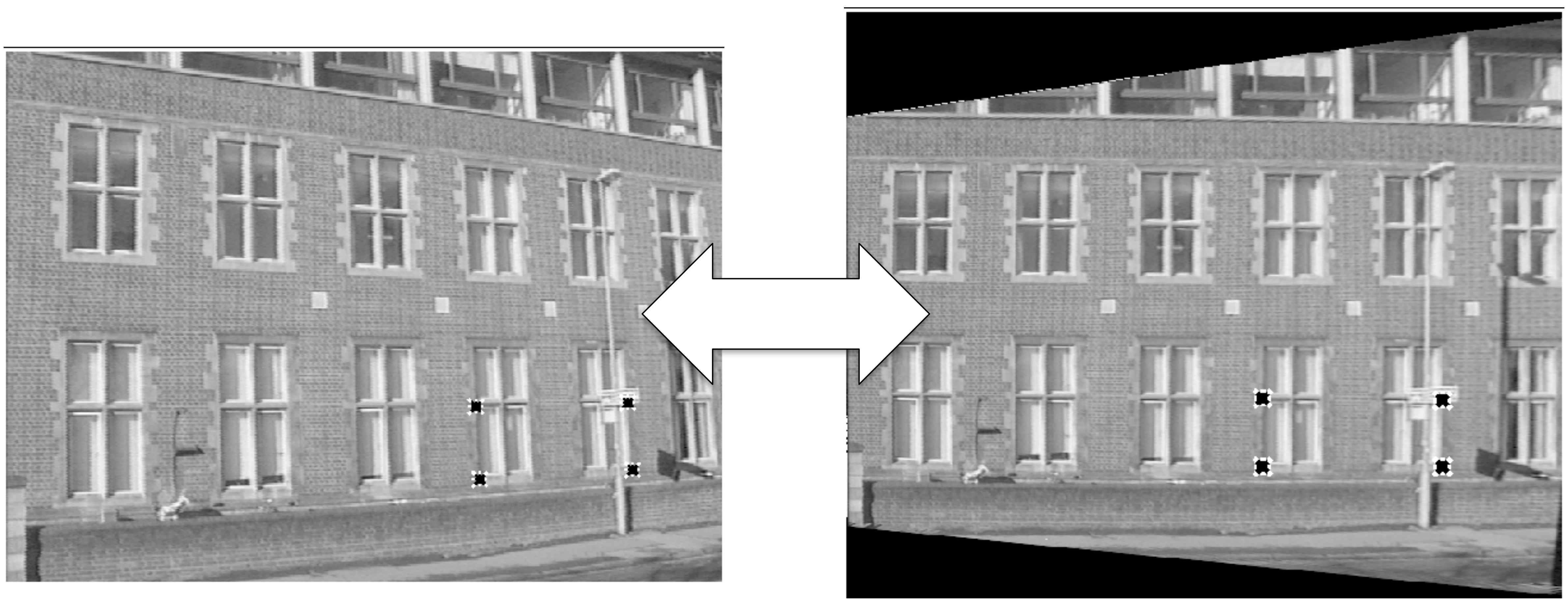
Camera Obscura

Today

- 2D Projective Geometry
- Camera Geometry

2D Projective Geometry?

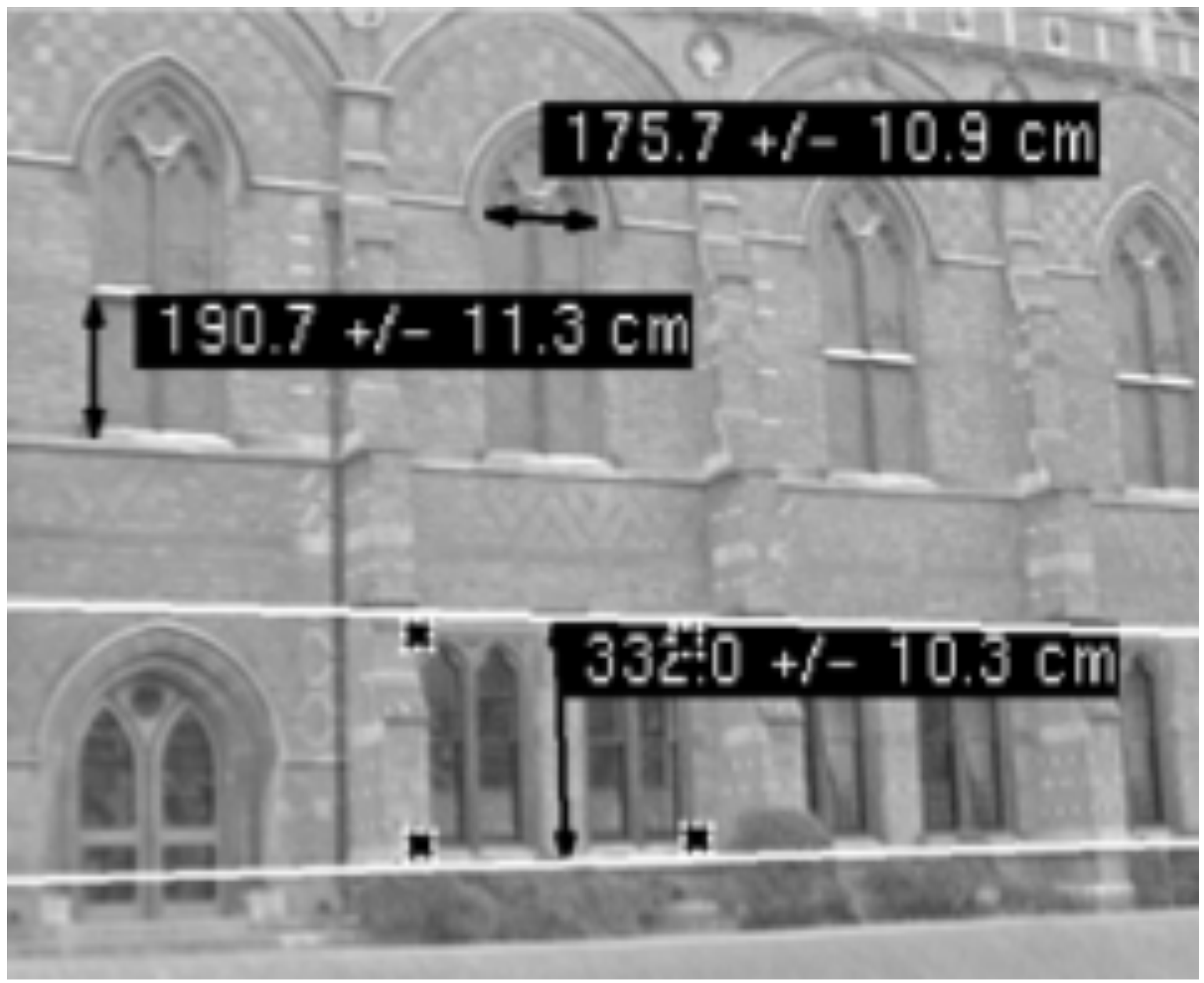
Projections of planar surfaces



A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD Thesis 1999.

2D Projective Geometry?

Measure distances



A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD Thesis 1999.

2D Projective Geometry?

Discover details

Piero della Francesca, La Flagellazione di Cristo (1460)

A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD Thesis 1999.

2D Projective Geometry?

Discover details

Piero della Francesca, La Flagellazione di Cristo (1460)

A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD Thesis 1999.

2D Projective Geometry?

Discover details

Piero della Francesca, La Flagellazione di Cristo (1460)

A. Criminisi. *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. PhD Thesis 1999.

2D Projective Geometry?

Image Stitching

2D Projective Geometry?

Image Stitching

Hierarchy of 2D Transformations

Hierarchy of 2D Transformations

rigid $$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

Hierarchy of 2D Transformations

rigid $$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

similarity $$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Hierarchy of 2D Transformations

rigid $$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

similarity $$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = s \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

affine $$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Hierarchy of 2D Transformations

rigid $$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

similarity $$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = s\begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

affine $$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

perspective $$\hat{w}\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}a & b & c\\ d & e & f\\ g & h & i\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

Extended Coordinates

``Extend" 2D coordinate to 3D coordinate

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \underbrace{\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}}_{\text{point}}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix}}_{\text{vector}}$$

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix} = \begin{pmatrix} x''' \\ y''' \\ 0 \end{pmatrix}$$

Extended Coordinates

``Extend" 2D coordinate to 3D coordinate

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

point

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix}$$

vector

Write affine transformations as matrix multiplication

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix} = \begin{pmatrix} x''' \\ y''' \\ 0 \end{pmatrix}$$

Extended Coordinates

``Extend" 2D coordinate to 3D coordinate

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

point

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix}$$

vector

Write affine transformations as matrix multiplication

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix} = \begin{pmatrix} x''' \\ y''' \\ 0 \end{pmatrix}$$

Extended Coordinates

``Extend" 2D coordinate to 3D coordinate

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

point

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} - \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix}$$

vector

Write affine transformations as matrix multiplication

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

$$\Leftrightarrow \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ 0 \end{pmatrix} = \begin{pmatrix} x''' \\ y''' \\ 0 \end{pmatrix}$$

Hierarchy of 2D Transformations

rigid
$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

similarity
$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = s\begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

affine
$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}a & b\\ c & d\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}t_x\\ t_y\end{pmatrix}$$

perspective
$$\hat{w}\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}a & b & c\\ d & e & f\\ g & h & i\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

Hierarchy of 2D Transformations

rigid

$$\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}\cos\alpha & -\sin\alpha & t_x\\ \sin\alpha & \cos\alpha & t_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

similarity

$$\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}s\cos\alpha & -s\sin\alpha & t_x\\ s\sin\alpha & s\cos\alpha & t_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

affine

$$\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}a & b & t_x\\ c & d & t_y\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

perspective

$$\hat{w}\begin{pmatrix}\hat{x}\\ \hat{y}\\ 1\end{pmatrix} = \begin{pmatrix}a & b & c\\ d & e & f\\ g & h & i\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

Homogenous Coordinates

Homogenous coordinates

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \;,\quad w \neq 0$$

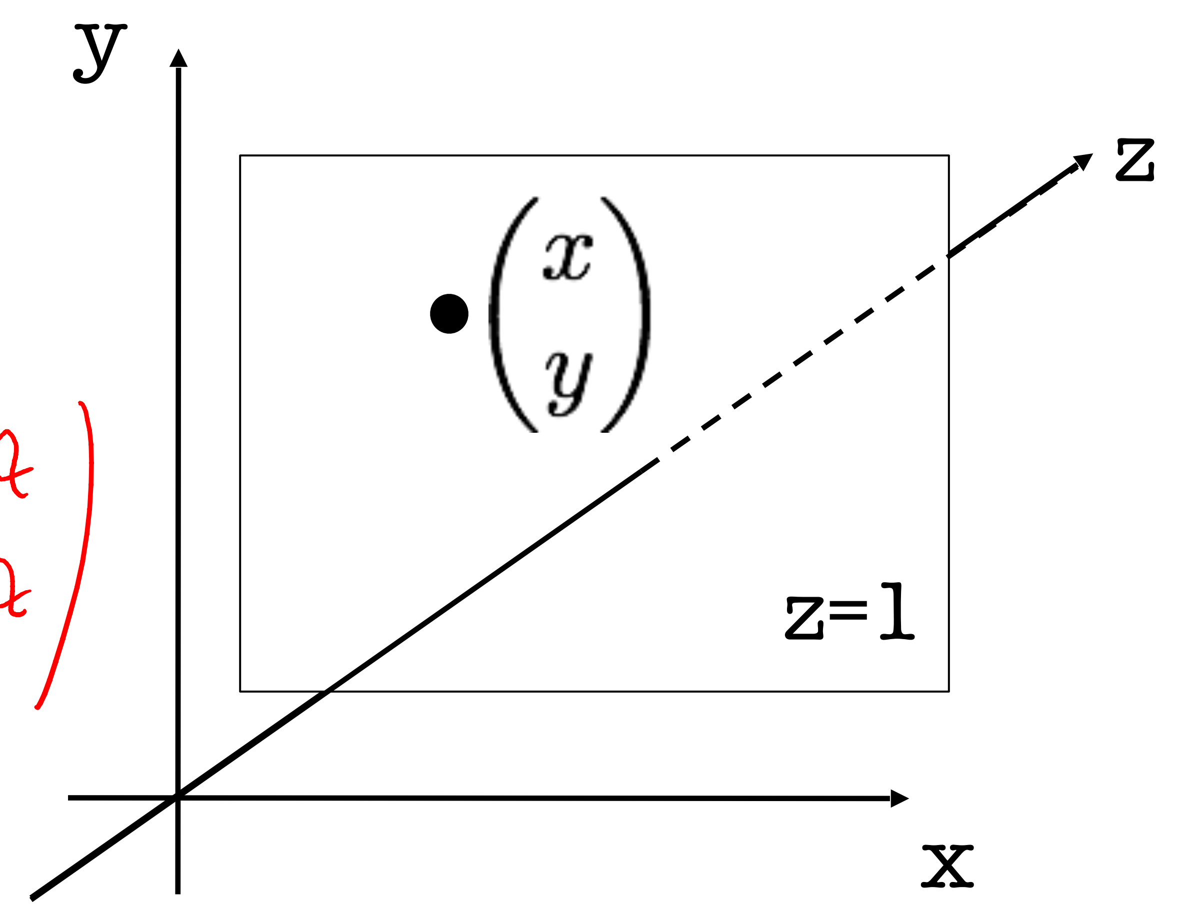
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

Homogenous Coordinates

Homogenous coordinates

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad , \quad w \neq 0$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

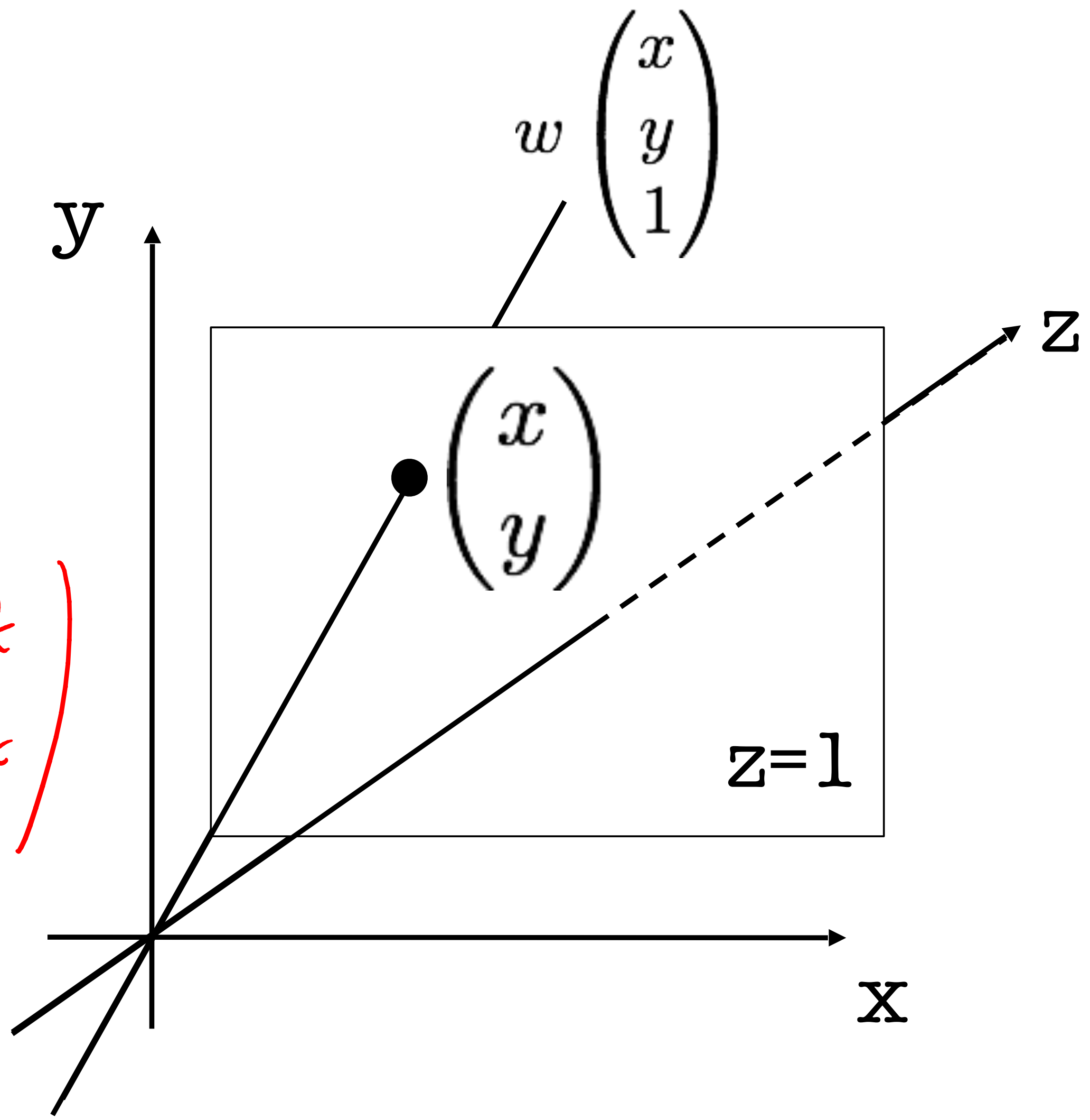


Homogenous Coordinates

Homogenous coordinates

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad , \quad w \neq 0$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$



Homogenous Coordinates

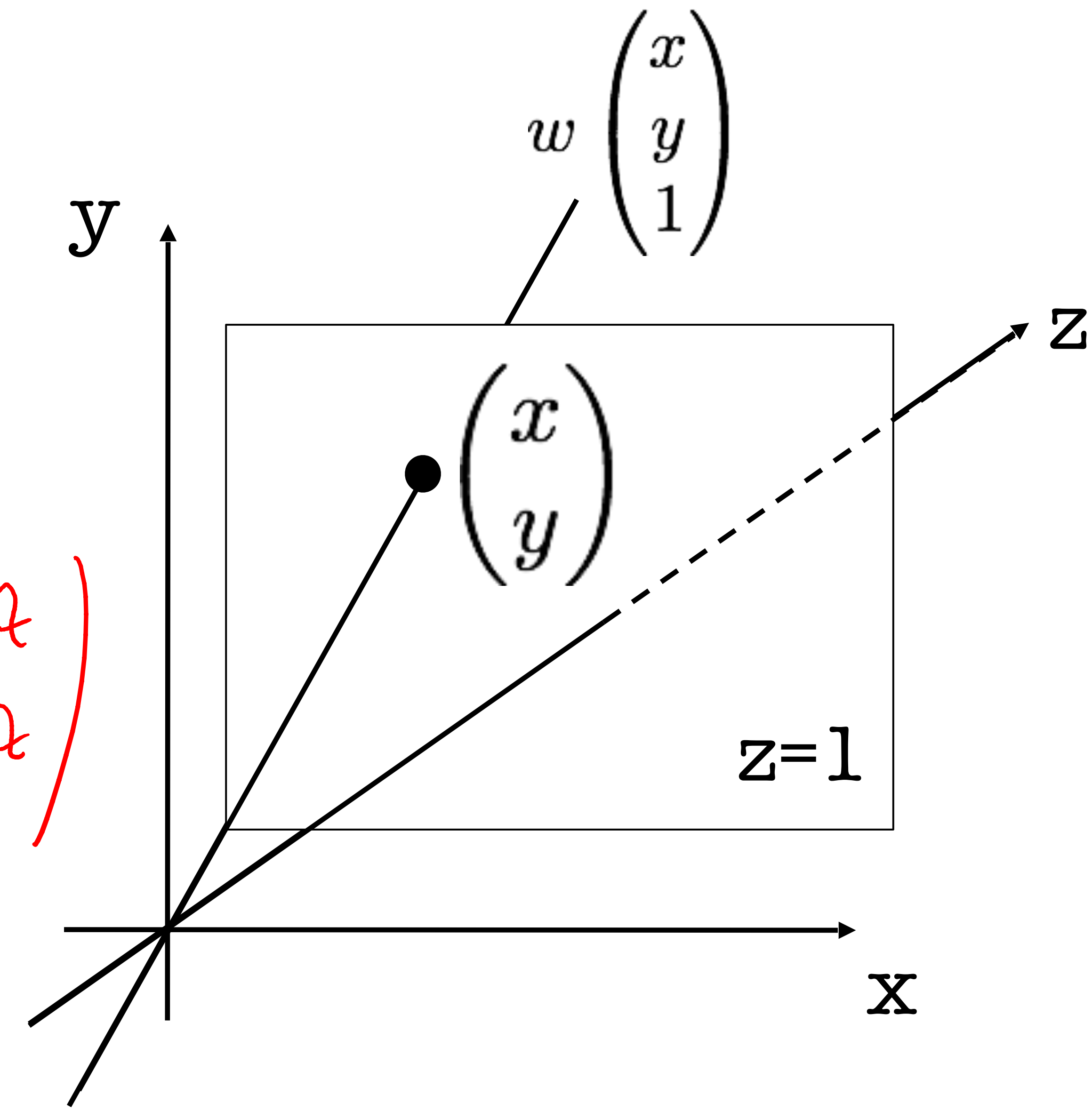
Homogenous coordinates

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad , \quad w \neq 0$$

De-homogenization:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

$$w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} x/w \\ y/w \end{pmatrix}$$



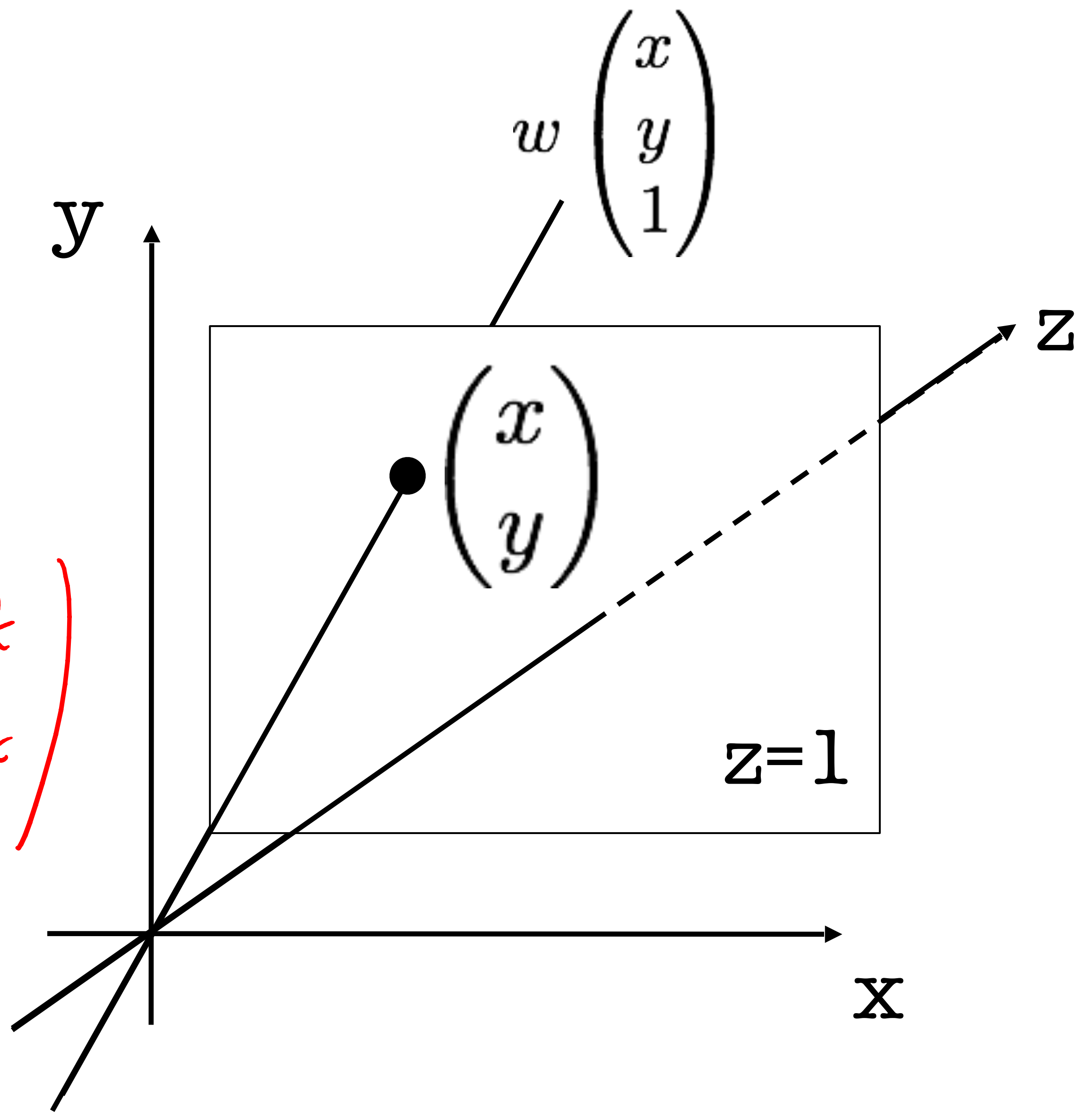
Homogenous Coordinates

Homogenous coordinates

$$\begin{pmatrix} x \\ y \end{pmatrix} \rightarrow w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad , \quad w \neq 0$$

De-homogenization: $\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix}$

$$w \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} x/w \\ y/w \end{pmatrix}$$



2D projective space: $\mathcal{P}^2 = \mathcal{R}^3 \setminus \{(0,0,0)^T\}$

Point ⇔ Line Duality

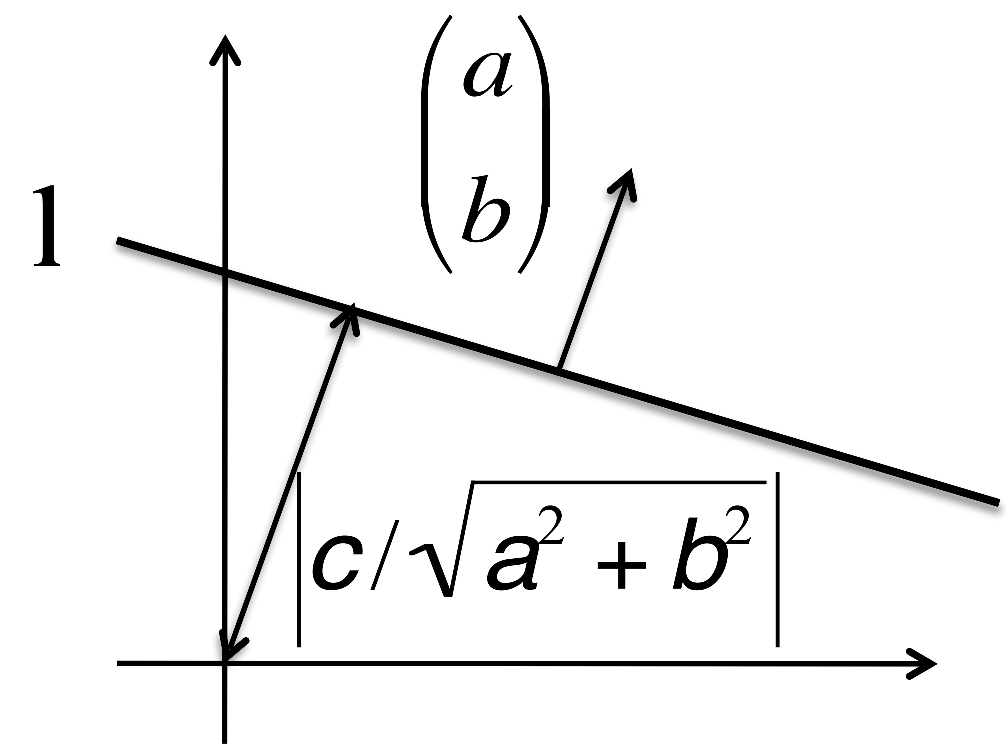
(Homogenous) 2D line representation:

$$ax + by + c = 0$$

Point ⇔ Line Duality

(Homogenous) 2D line representation:

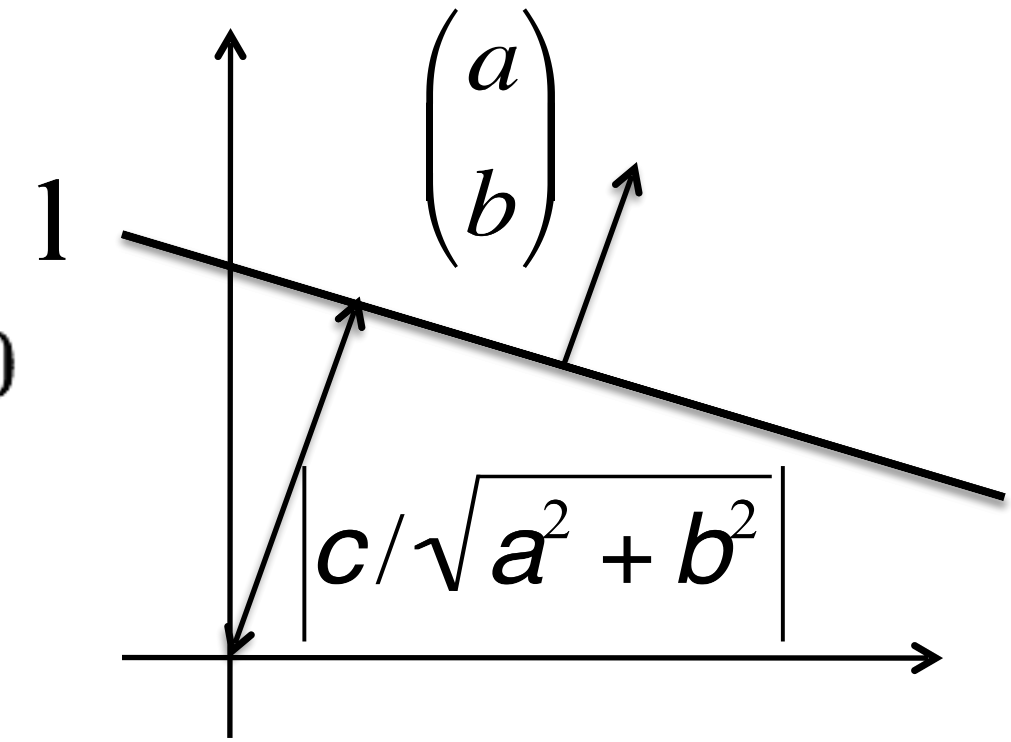
$$ax + by + c = 0$$



Point ⇔ Line Duality

(Homogenous) 2D line representation:

$$ax + by + c = 0 \Leftrightarrow (a, b, c) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

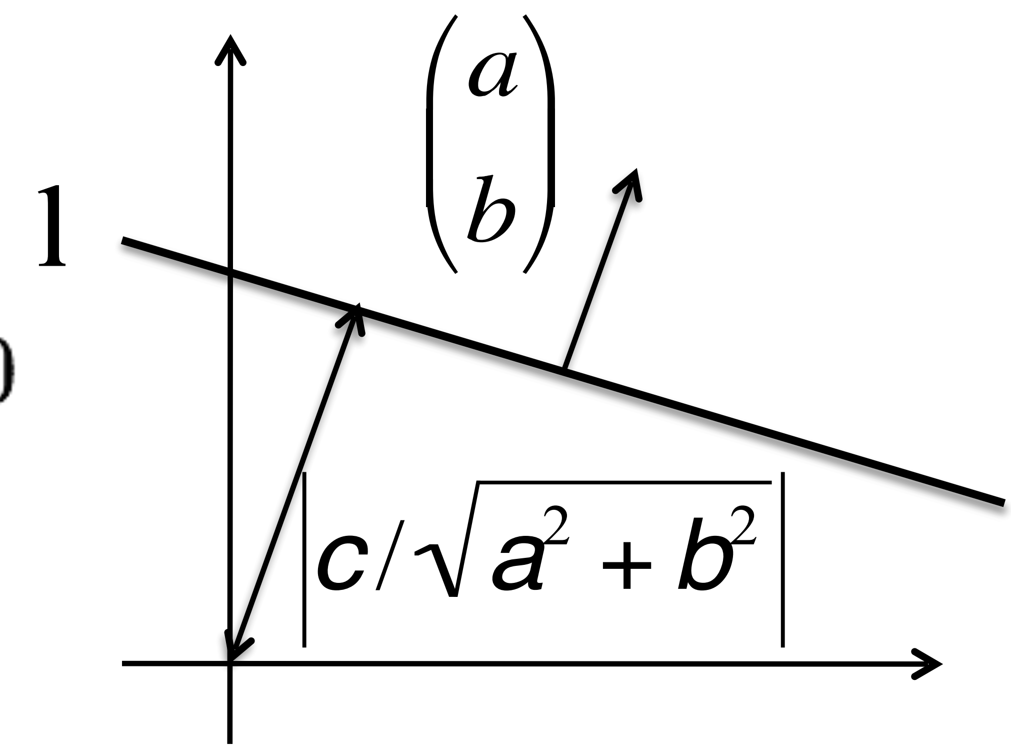


Point ⇔ Line Duality

(Homogenous) 2D line representation:

$$ax + by + c = 0 \Leftrightarrow (a, b, c) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

$$\Leftrightarrow l^T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

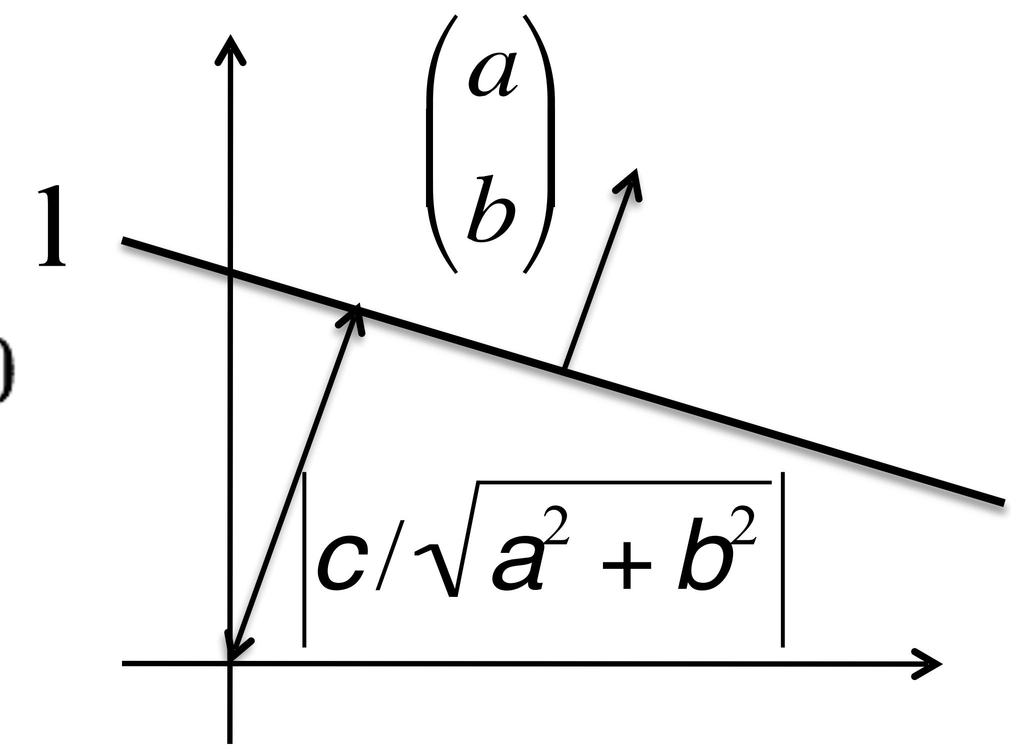


Point ⇔ Line Duality

(Homogenous) 2D line representation:

$$ax + by + c = 0 \Leftrightarrow (a, b, c) \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

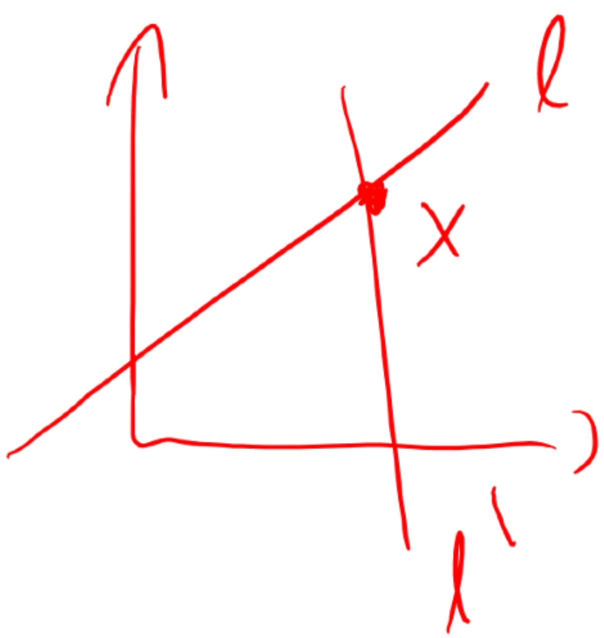
$$\Leftrightarrow l^T \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$



2D lines are points in $\mathcal{R}^3$

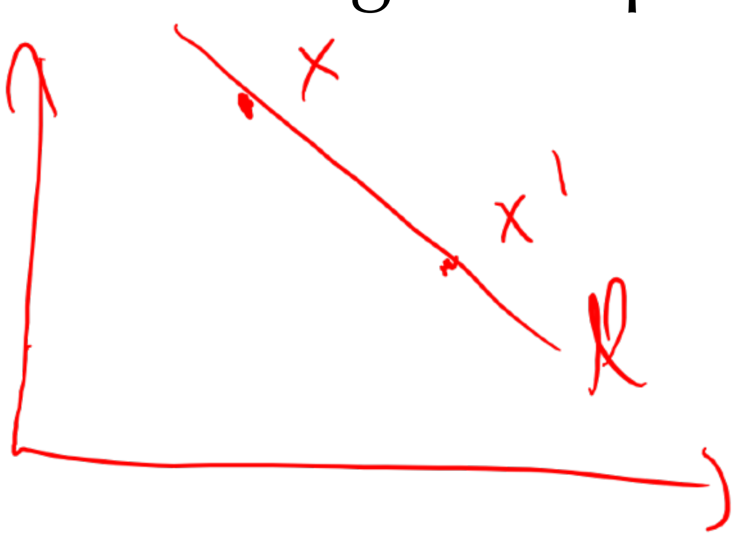
Point ⇔ Line Duality

Intersection of two lines



$$\left.\begin{aligned} l^T x &= 0 \\ l'^T x &= 0 \end{aligned}\right\} \quad x = l \times l'$$

Lines through two points



$$\left.\begin{aligned} l^T x &= 0 \\ l^T x' &= 0 \end{aligned}\right\} \quad l = x \times x'$$

2D Projective Transformations

Perspective mapping (projectivity, **homography**)

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$\hat{x} = (sH)x \quad , \; s \neq 0$

2D Projective Transformations

Perspective mapping (projectivity, **homography**)

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

or $\hat{\mathbf{x}} = \mathbf{H}\mathbf{x}$

$\hat{x} = (sH)x \quad , s \neq 0$

2D Projective Transformations

Perspective mapping (projectivity, **homography**)

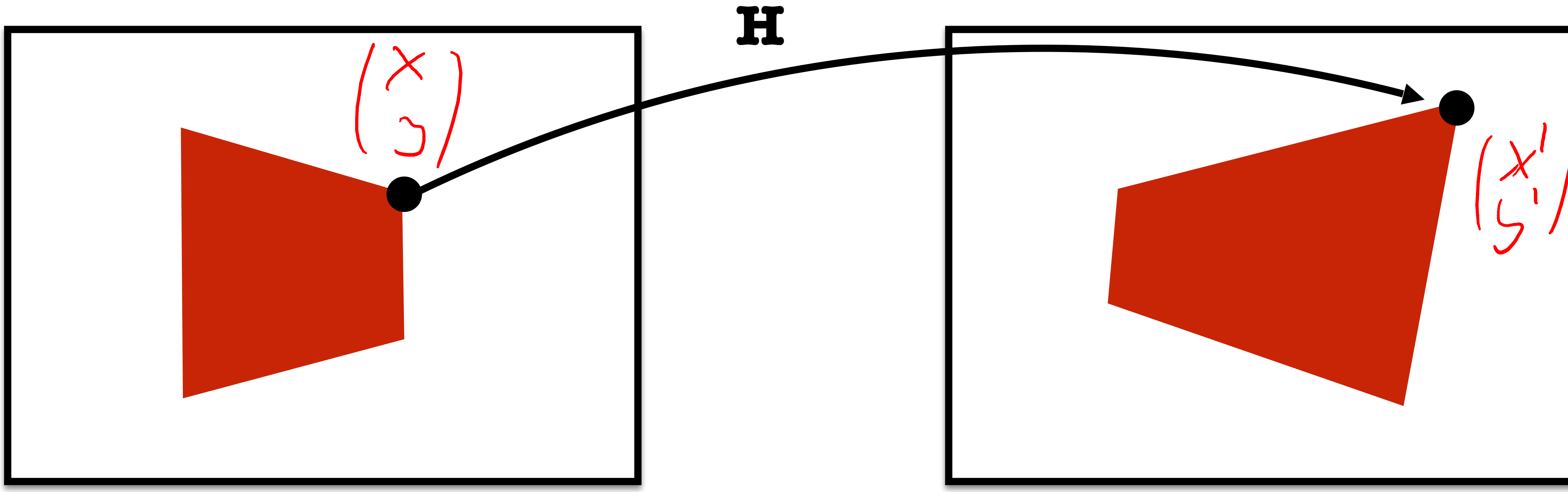
$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

or $\hat{\mathbf{x}} = \mathbf{H}\mathbf{x}$

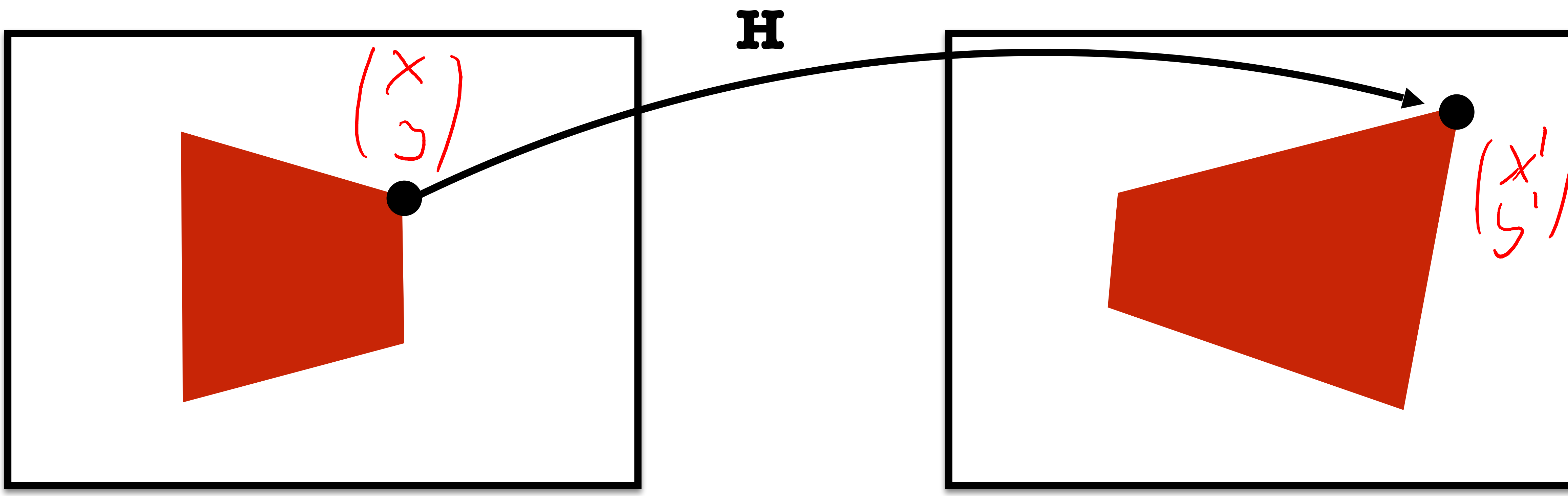
$\hat{x} = (sH)x \quad , s \neq 0$

- $\mathbf{H}$ needs to be invertible
- $\mathbf{H}$ has 8 Degrees-of-Freedom (DoF)

Using Homogenous Coordinates



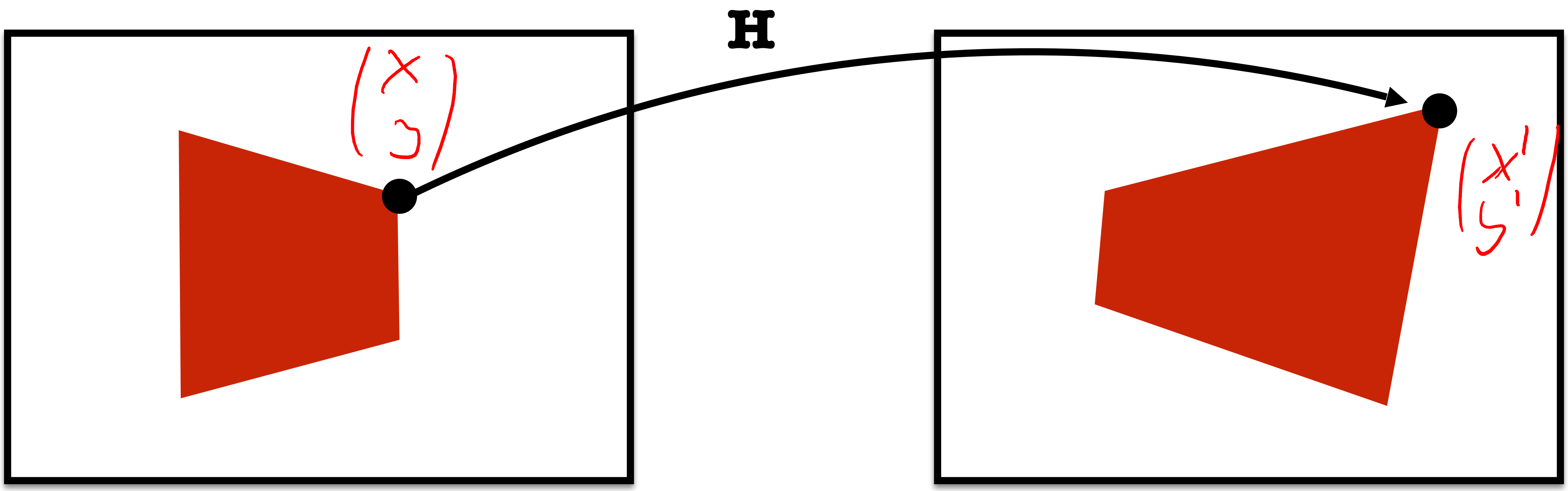
Using Homogenous Coordinates



- "Homogenize": $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

- Apply $\mathbf{H}$:

Using Homogenous Coordinates



- "Homogenize": $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- Apply **H**: $\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \mathtt{H} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- De-homogenize: $\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} \mapsto \begin{pmatrix} x''/z'' \\ y''/z'' \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$

Homography Estimation

$h^i \in \mathbb{R}^{1\times 3}$

$x'_i = x''_i \cdot s, \; s \neq 0$

3×9
system

Homography Estimation

Compute homography $\mathbf{H}$ from 2D-2D matches:

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \qquad \mathbf{x}'_i = \left(x'_i, y'_i, w'_i\right)^\top \qquad \mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}$$

$h^i \in \mathbb{R}^{1\times 3}$

$x'_i = x''_i \cdot s, \; s \neq 0$

3 x 3 system

Homography Estimation

Compute homography $\mathbf{H}$ from 2D-2D matches:

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \qquad \mathbf{x}'_i = \left(x'_i, y'_i, w'_i\right)^\top \qquad \mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}$$

$h^i \in \mathbb{R}^{1\times 3}$

$x'_i = x''_i \cdot s, \; s \neq 0$

Only defined up to scale, use cross-product instead

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = 0$$

3×3 system

Homography Estimation

Compute homography $\mathbf{H}$ from 2D-2D matches:

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \qquad \mathbf{x}'_i = \left(x'_i, y'_i, w'_i\right)^\top \qquad \mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}$$

$h^i \in \mathbb{R}^{1\times 3}$

$x'_i = x''_i \cdot s, \; s \neq 0$

Only defined up to scale, use cross-product instead

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = 0 \quad \Leftrightarrow \quad \mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\top}\mathbf{x}_i - w'_i\mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\top}\mathbf{x}_i - y'_i\mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix}$$

3 x 3
system

Homography Estimation

Compute homography **H** from 2D-2D matches:

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i \qquad \mathbf{x}'_i = \left(x'_i, y'_i, w'_i\right)^\top \qquad \mathbf{H}\mathbf{x}_i = \begin{pmatrix} \mathbf{h}^{1\top}\mathbf{x}_i \\ \mathbf{h}^{2\top}\mathbf{x}_i \\ \mathbf{h}^{3\top}\mathbf{x}_i \end{pmatrix}$$

$h^i \in \mathbb{R}^{1\times 3}$

$x'_i = x''_i \cdot s, \; s \neq 0$

Only defined up to scale, use cross-product instead

$$\mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = 0 \;\Leftrightarrow\; \mathbf{x}'_i \times \mathbf{H}\mathbf{x}_i = \begin{pmatrix} y'_i\mathbf{h}^{3\top}\mathbf{x}_i - w'_i\mathbf{h}^{2\top}\mathbf{x}_i \\ w'_i\mathbf{h}^{1\top}\mathbf{x}_i - x'_i\mathbf{h}^{3\top}\mathbf{x}_i \\ x'_i\mathbf{h}^{2\top}\mathbf{x}_i - y'_i\mathbf{h}^{1\top}\mathbf{x}_i \end{pmatrix}$$

$$\Rightarrow \begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix}\begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

3 x 9 system

Homography Estimation

Single match gives 3 equations (2 linearly independent)

$$\begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \\ -y_i'\mathbf{x}_i^\top & x_i'\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Homography Estimation

Single match gives 3 equations (2 linearly independent)

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Stack equations from 4 matches (8x9 or 12x9 matrix of rank 2)

$$\mathbf{A}\mathbf{h} = \mathbf{0}$$

Homography Estimation

Single match gives 3 equations (2 linearly independent)

$$\begin{bmatrix} \mathbf{0}^\top & -w'_i\mathbf{x}_i^\top & y'_i\mathbf{x}_i^\top \\ w'_i\mathbf{x}_i^\top & \mathbf{0}^\top & -x'_i\mathbf{x}_i^\top \\ -y'_i\mathbf{x}_i^\top & x'_i\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Stack equations from 4 matches (8x9 or 12x9 matrix of rank 2)

$$\mathbf{A}\mathbf{h} = \mathbf{0}$$

No exact solution due to noise. Use least squares solution instead:

Homography Estimation

Single match gives 3 equations (2 linearly independent)

$$\begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \\ -y_i'\mathbf{x}_i^\top & x_i'\mathbf{x}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Stack equations from 4 matches (8x9 or 12x9 matrix of rank 2)

$$\mathtt{A}\mathbf{h} = \mathbf{0}$$

No exact solution due to noise. Use least squares solution instead:

- Constraint needed to avoid $\mathbf{h} = \mathbf{0}$, use $\|\mathbf{h}\| = 1$
- Minimize $\| \mathtt{A}\mathbf{h} \|$

Direct Linear Transform (DLT)

Objective

Given n≥4 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$, determine the 2D homography matrix $\mathtt{H}$ such that $\mathbf{x}_i' = \mathtt{H}\mathbf{x}_i$

Algorithm

- For each correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ compute $\mathtt{A}_i$. Usually only two first rows needed.
- Assemble *n* 2x9 matrices $\mathtt{A}_i$ into a single $\mathtt{2nx9}$ matrix $\mathtt{A}$ $svd(A) = U \Sigma V^T$
- Obtain SVD of $\mathtt{A}$. Solution for $\mathtt{h}$ is last column of $\mathtt{V}$
- Determine $\mathtt{H}$ from $\mathtt{h}$

Hartley and Zisserman. *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, 2004.

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x'_i & -y'_i & -1 & y'_i x_i & y'_i y_i & y'_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$ $\sim 10^2$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x'_i & -y'_i & -1 & y'_i x_i & y'_i y_i & y'_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$ $\sim 10^2$ 1

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i' x_i & y_i' y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i' x_i & -x_i' y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$ $\sim 10^2$ 1 $\sim 10^2$

Coordinate Normalization

$$\begin{bmatrix}0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i'\end{bmatrix}\begin{pmatrix}\mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3\end{pmatrix} = \mathbf{0}$$

$\sim 10^2 \quad \sim 10^2 \quad 1 \quad \sim 10^2 \quad \sim 10^2$

Coordinate Normalization

$$\begin{array}{c} \begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \\ \begin{matrix} \sim 10^2 & \sim 10^2 & 1 & \sim 10^2 & \sim 10^2 & 1 & & & \end{matrix} \end{array} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$$\begin{array}{ccccccccc} \sim 10^2 & \sim 10^2 & 1 & \sim 10^2 & \sim 10^2 & 1 & \sim 10^4 & & \end{array}$$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x'_i & -y'_i & -1 & y'_i x_i & y'_i y_i & y'_i \\ x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$ $\sim 10^2$ 1 $\sim 10^2$ $\sim 10^2$ 1 $\sim 10^4$ $\sim 10^4$

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix}\begin{pmatrix}\mathbf{h}^1\\ \mathbf{h}^2\\ \mathbf{h}^3\end{pmatrix} = \mathbf{0}$$

$$\begin{array}{ccccccccc} \sim 10^2 & \sim 10^2 & 1 & \sim 10^2 & \sim 10^2 & 1 & \sim 10^4 & \sim 10^4 & \sim 10^2 \end{array}$$

Coordinate Normalization

$$\begin{array}{c}\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i' x_i & y_i' y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i' x_i & -x_i' y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0} \\ \begin{array}{ccccccccc} \sim 10^2 & \sim 10^2 & 1 & \sim 10^2 & \sim 10^2 & 1 & \sim 10^4 & \sim 10^4 & \sim 10^2 \end{array}\end{array}$$

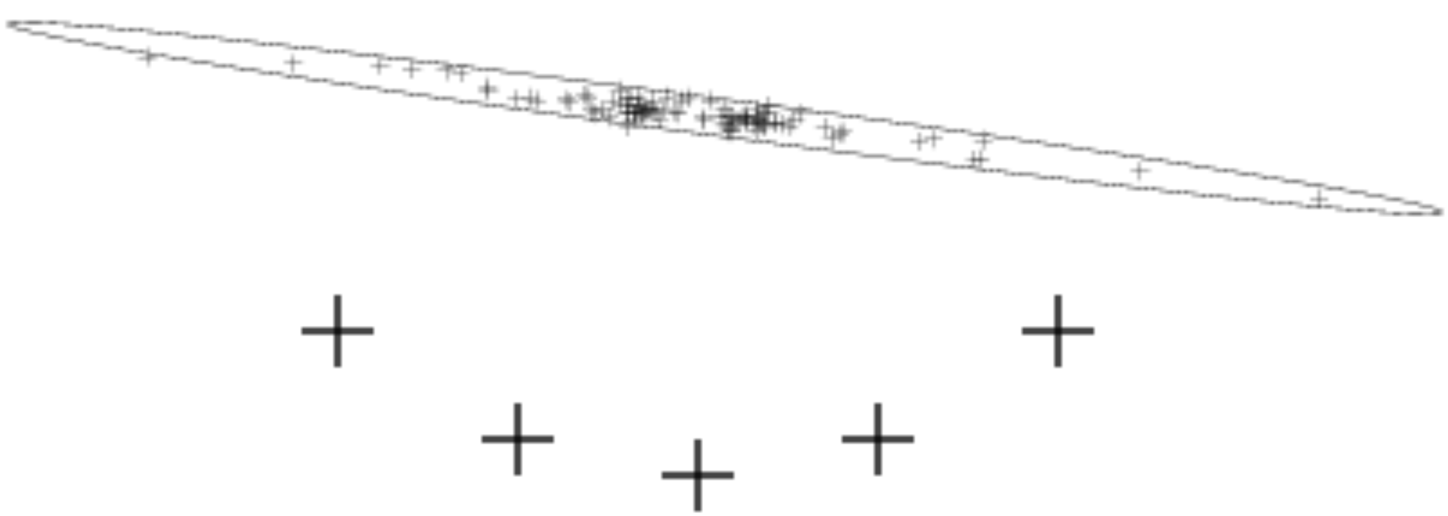
orders of magnitude difference!

Coordinate Normalization

$$\begin{bmatrix} 0 & 0 & 0 & -x_i' & -y_i' & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}$$

$\sim 10^2$ $\sim 10^2$ 1 $\sim 10^2$ $\sim 10^2$ 1 $\sim 10^4$ $\sim 10^4$ $\sim 10^2$

orders of magnitude difference!



Monte Carlo simulation
for identity computation
based on 5 points

Normalized Direct Linear Transform (DLT)

<u>Objective</u>

Given $n \geq 4$ 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$, determine the 2D homography matrix $\mathtt{H}$ such that $\mathbf{x}_i' = \mathtt{H}\mathbf{x}_i$

<u>Algorithm</u>

- Normalize points: $\tilde{\mathbf{x}}_i = \mathbf{T}_{\text{norm}}\mathbf{x}_i, \tilde{\mathbf{x}}_i' = \mathbf{T}_{\text{norm}}'\mathbf{x}_i'$
- Apply DLT algorithm to $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_i'$,
- Denormalize solution: $\mathbf{H} = \mathbf{T}_{\text{norm}}'^{-1}\tilde{\mathbf{H}}\mathbf{T}_{\text{norm}}$

Hartley and Zisserman. *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, 2004.

Normalized Direct Linear Transform (DLT)

<u>Objective</u>

Given n≥4 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}_i'\}$, determine the 2D homography matrix H such that $\mathbf{x}_i' = H\mathbf{x}_i$

<u>Algorithm</u>

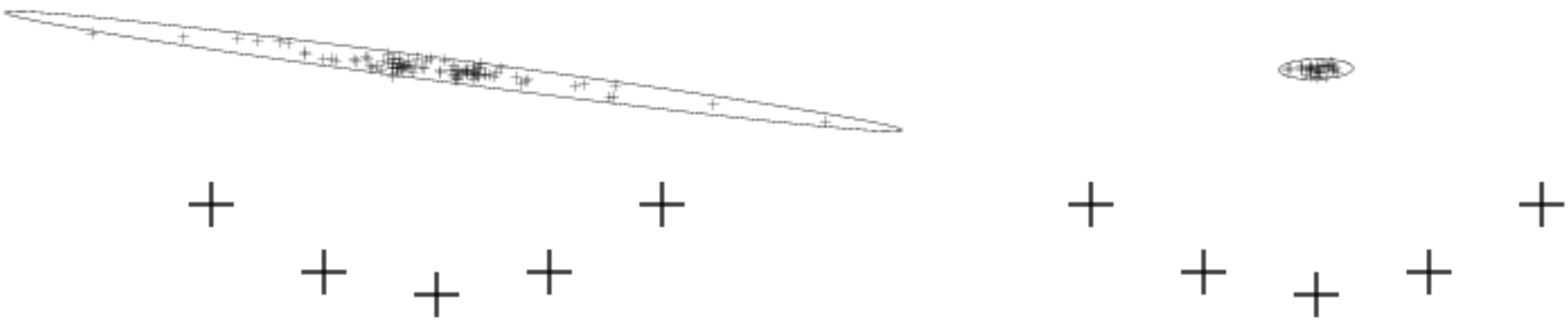
- Normalize points: $\tilde{\mathbf{x}}_i = \mathbf{T}_{norm}\mathbf{x}_i, \tilde{\mathbf{x}}_i' = \mathbf{T}_{norm}'\mathbf{x}_i'$
- Apply DLT algorithm to $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}_i'$,
- Denormalize solution: $\mathbf{H} = \mathbf{T}_{norm}'^{-1}\tilde{\mathbf{H}}\mathbf{T}_{norm}$

Normalization (independently per image):

- Translate points such that centroid is at origin
- Isotropic scaling such that mean distance to origin is $\sqrt{2}$

Hartley and Zisserman. *Multiple View Geometry in Computer Vision*, 2nd edition, Cambridge University Press, 2004.

Impact of Coordinate Normalization



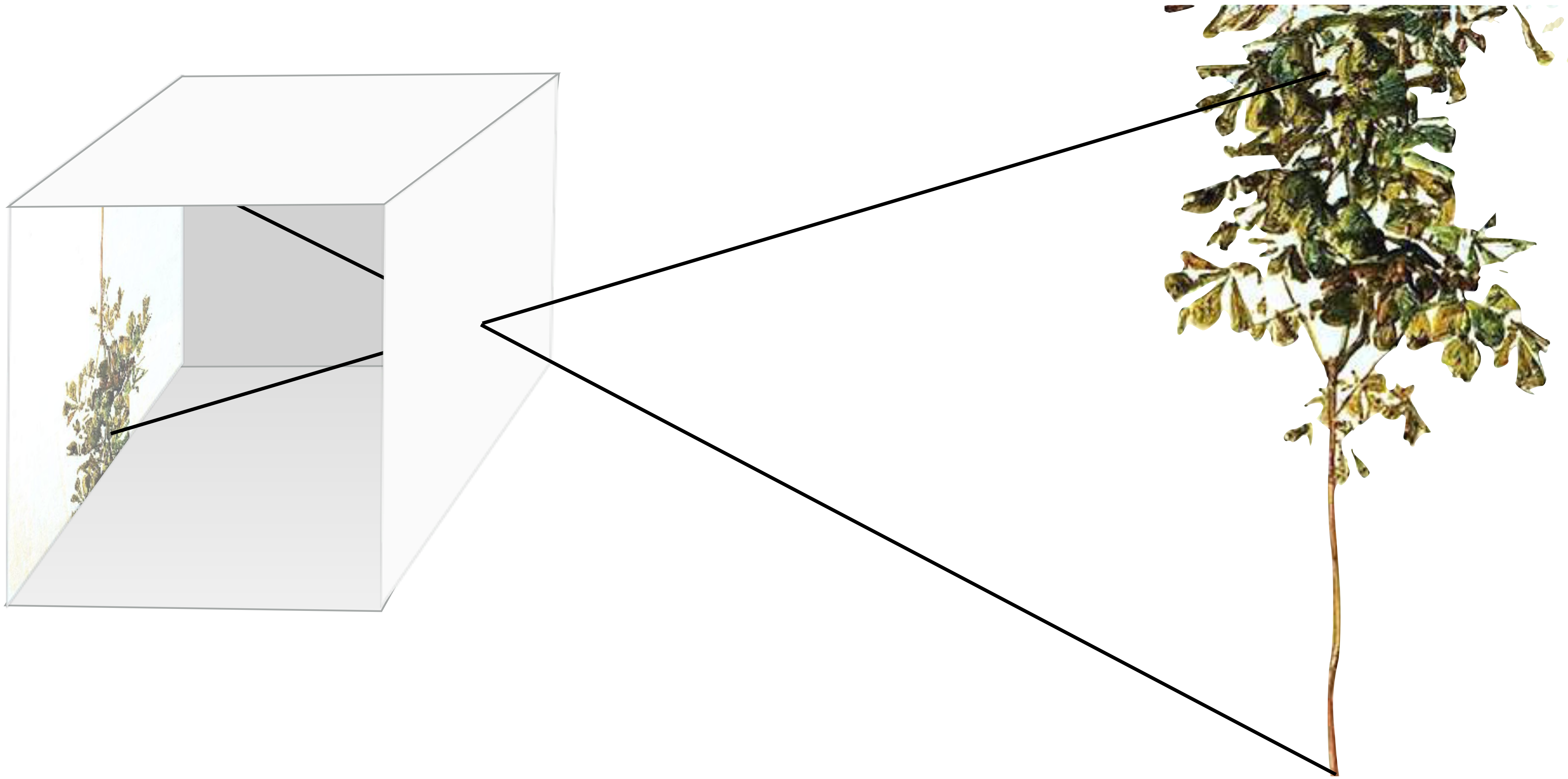
Monte Carlo simulation
for identity computation based on 5 points
(not normalized ↔ normalized)

Today

- 2D Geometry
- **Camera Geometry**

Camera Obscura

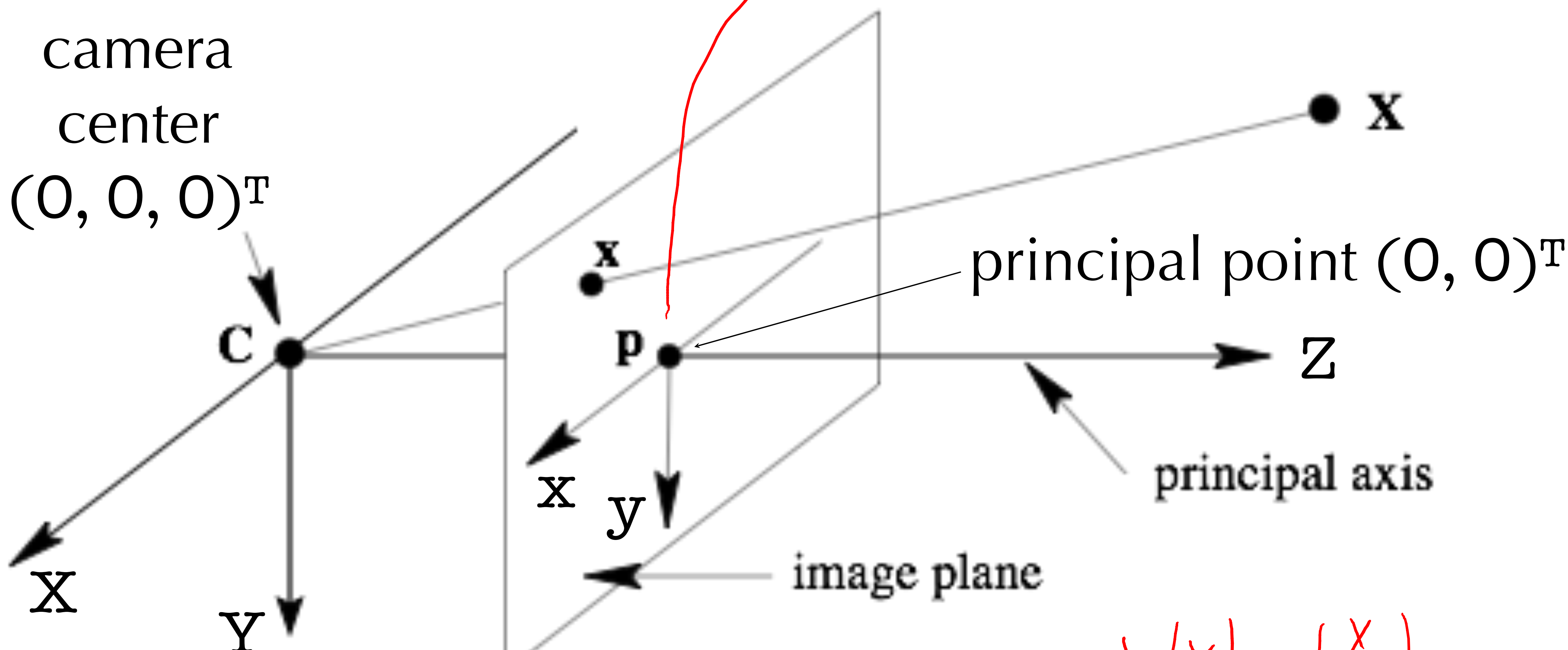
Pinhole Camera Principle



Pinhole Camera

Pinhole Camera Model

$= (6, 0)^T$

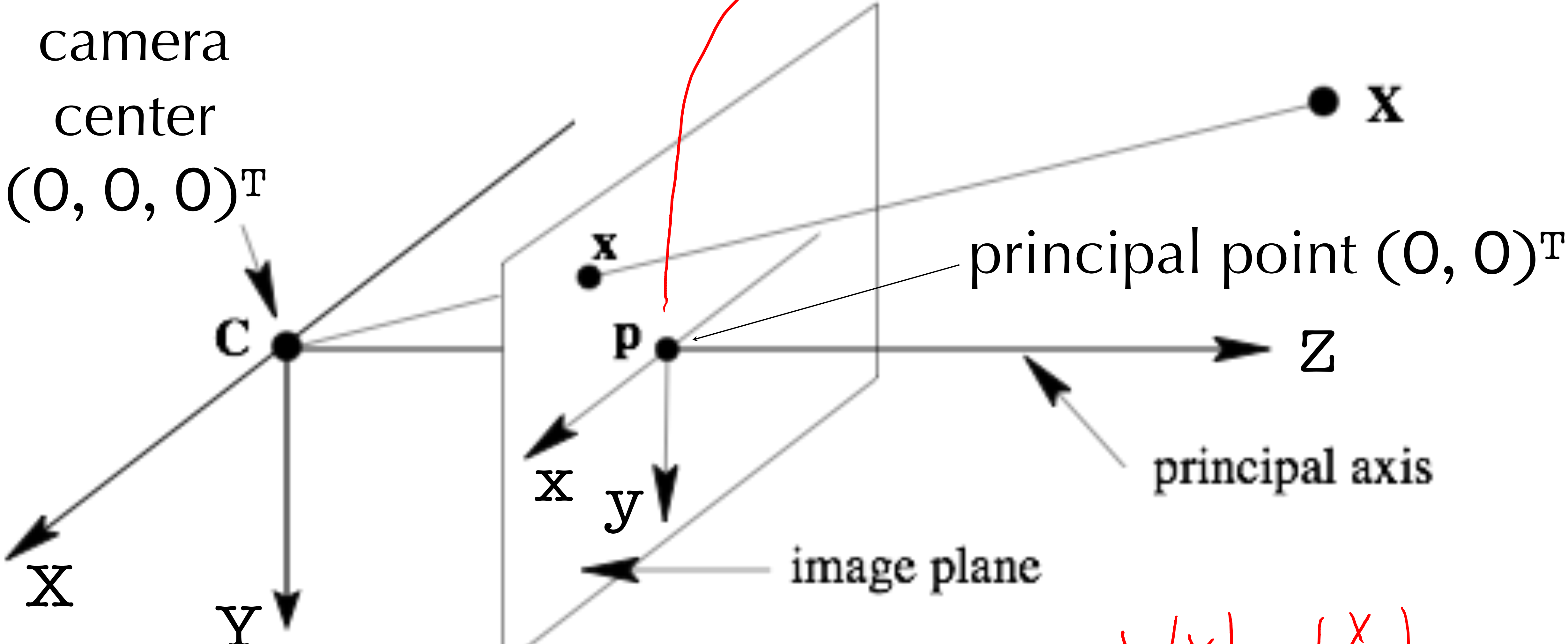


$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

figure adapted from Hartley and Zisserman, 2004

Pinhole Camera Model

$= (0, 0)^T$

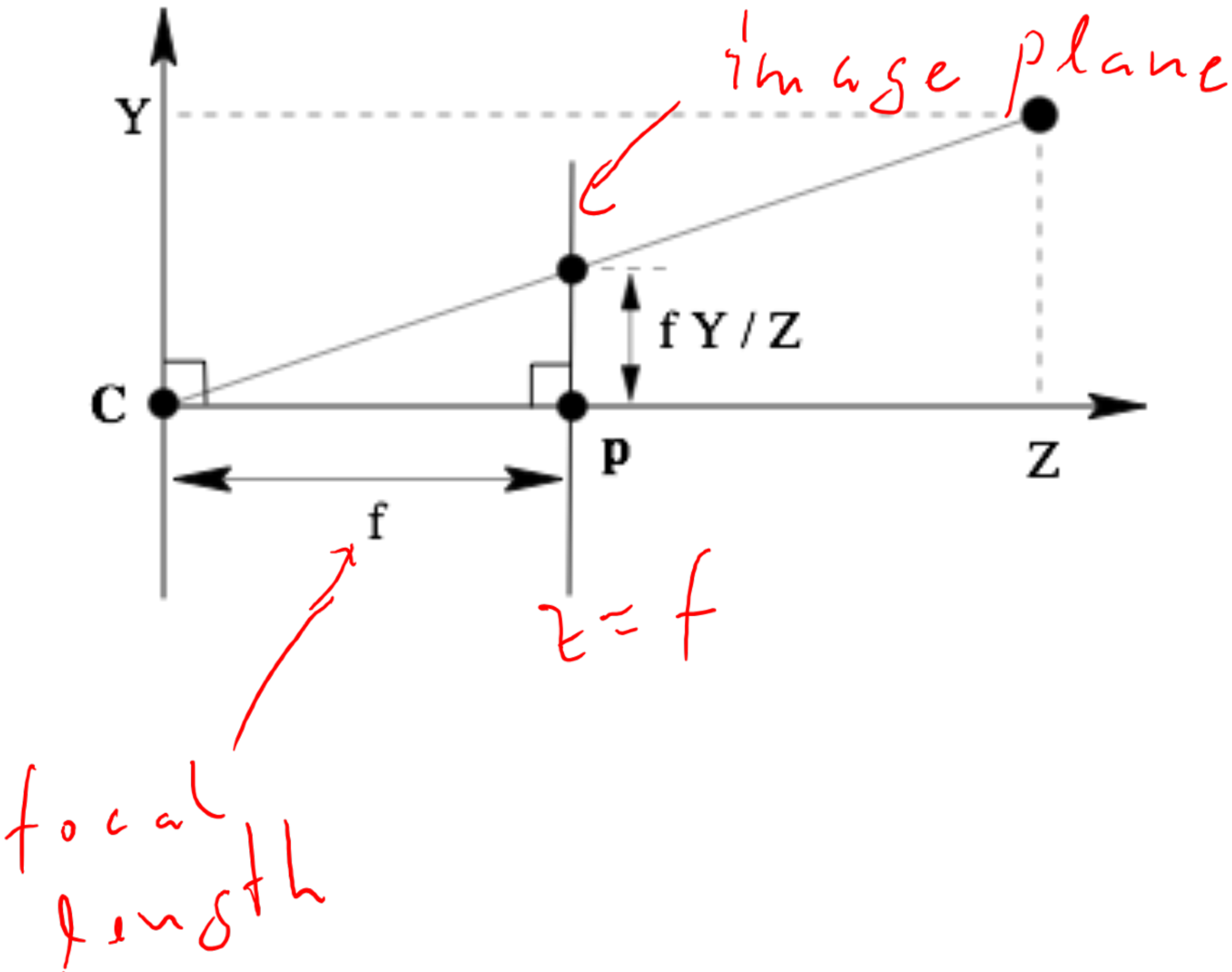


$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Projection in homogenous coordinates: $\lambda \mathbf{x} = \mathbf{X}$

figure adapted from Hartley and Zisserman, 2004

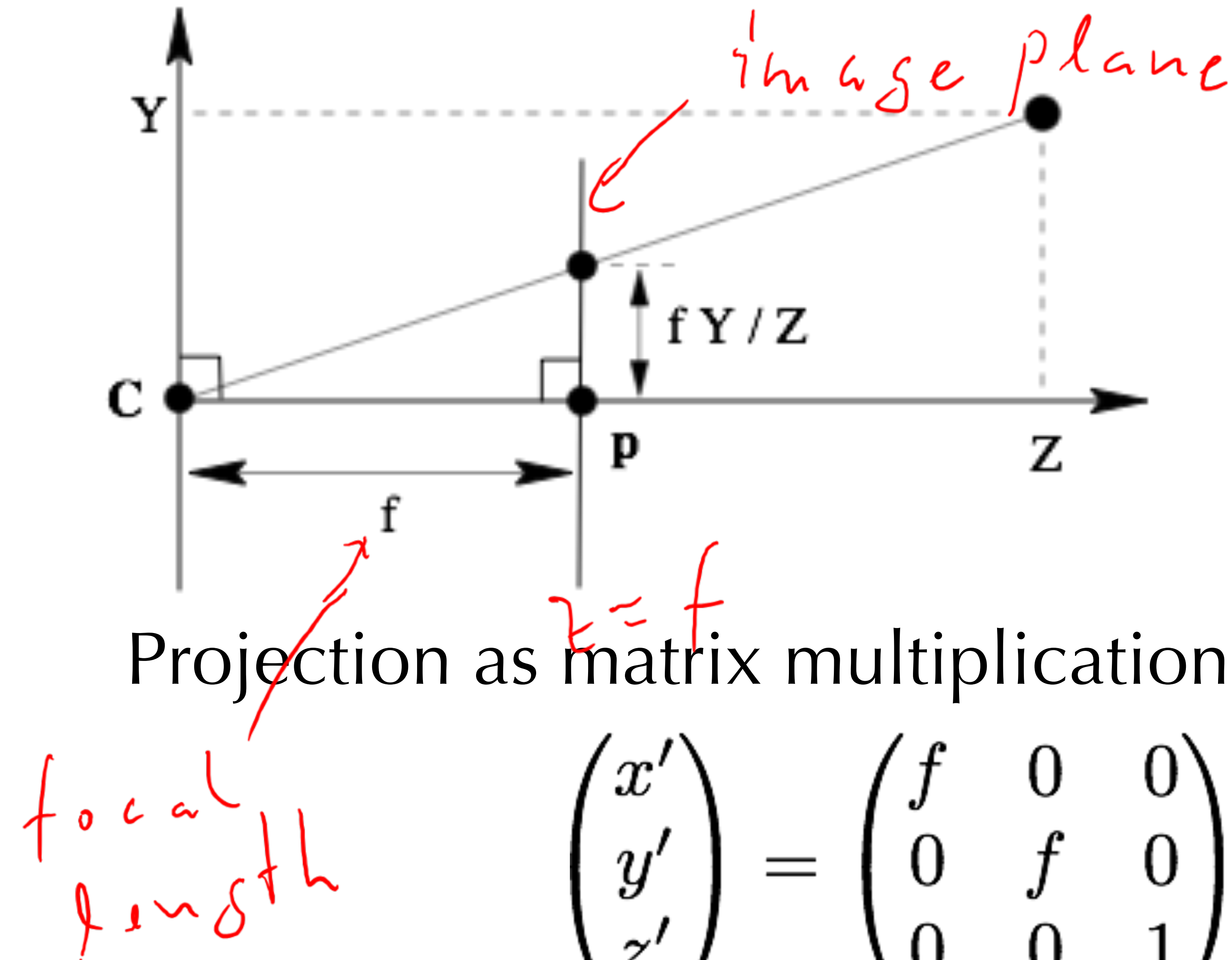
Pinhole Camera Model



$$= \begin{pmatrix} fX \\ fy \\ z \end{pmatrix} = \begin{pmatrix} fX/z \\ fy/z \end{pmatrix}$$

figure from Hartley and Zisserman, 2004

Pinhole Camera Model



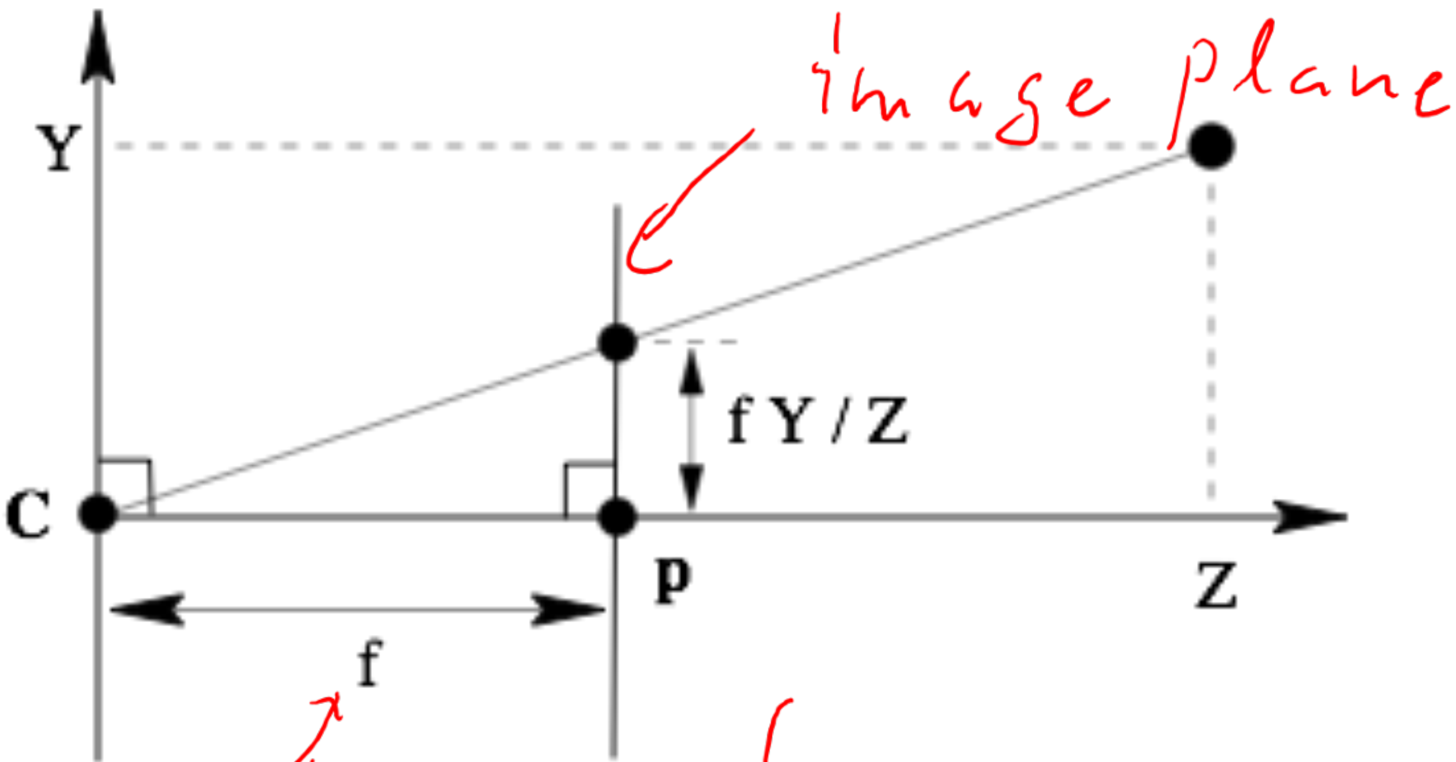
Projection as matrix multiplication:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix}$$

$$= \begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix}$$

figure from Hartley and Zisserman, 2004

Pinhole Camera Model



z = f

focal length

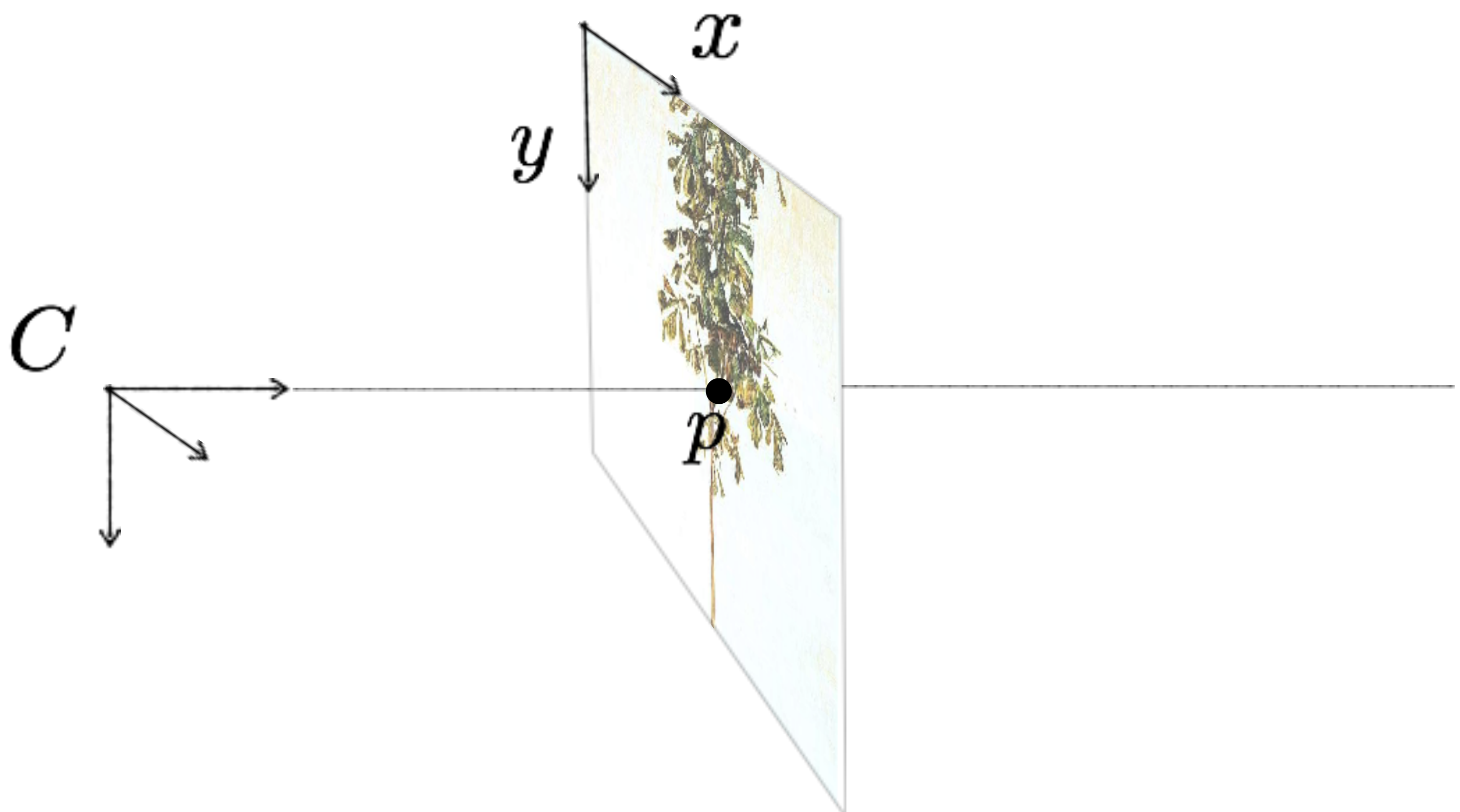
Projection as matrix multiplication:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ z \end{pmatrix}$$

De-homogenization: $$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x'/z' \\ y'/z' \end{pmatrix} = \begin{pmatrix} fX/z \\ fY/z \end{pmatrix}$$

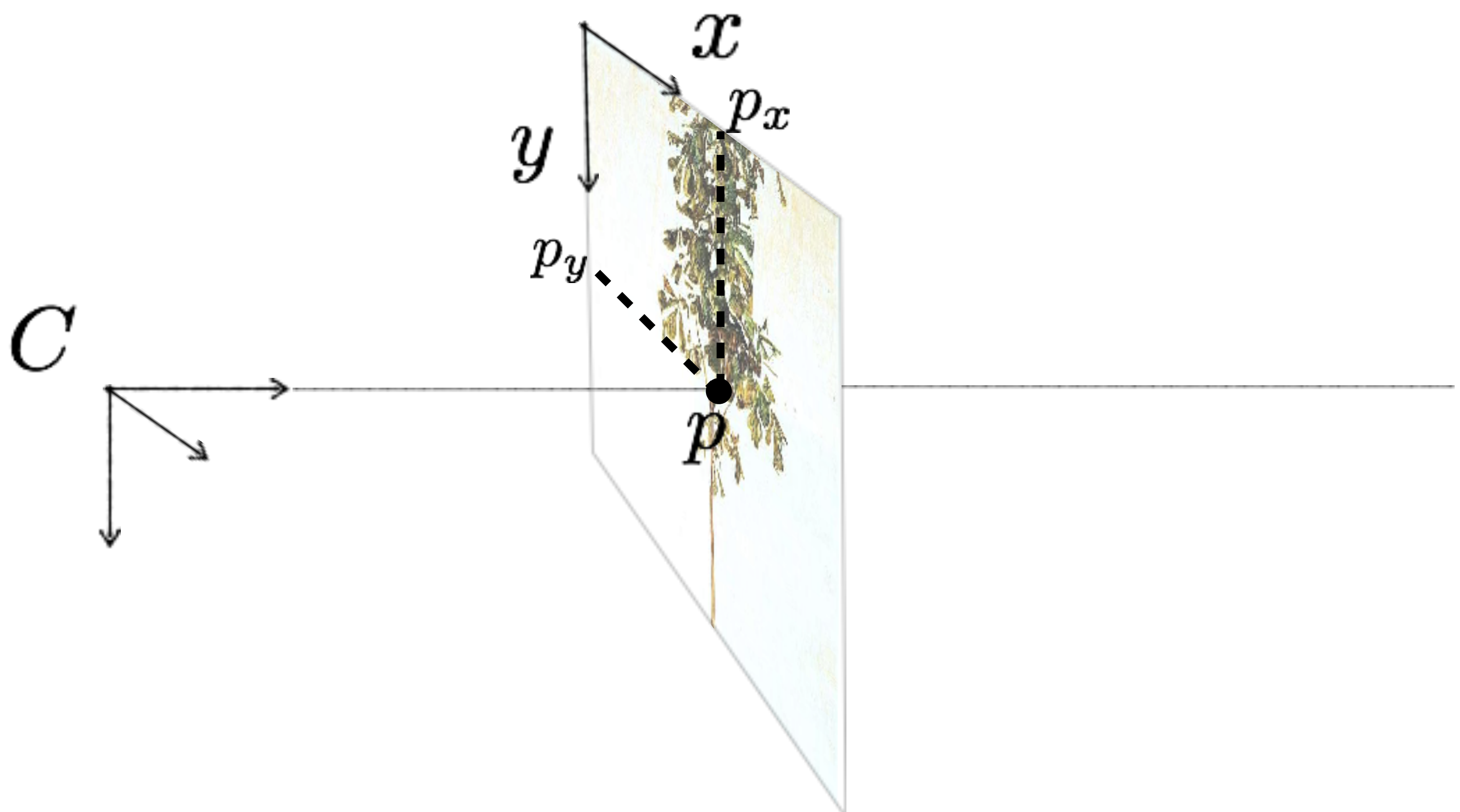
figure from Hartley and Zisserman, 2004

To Pixel Coordinates



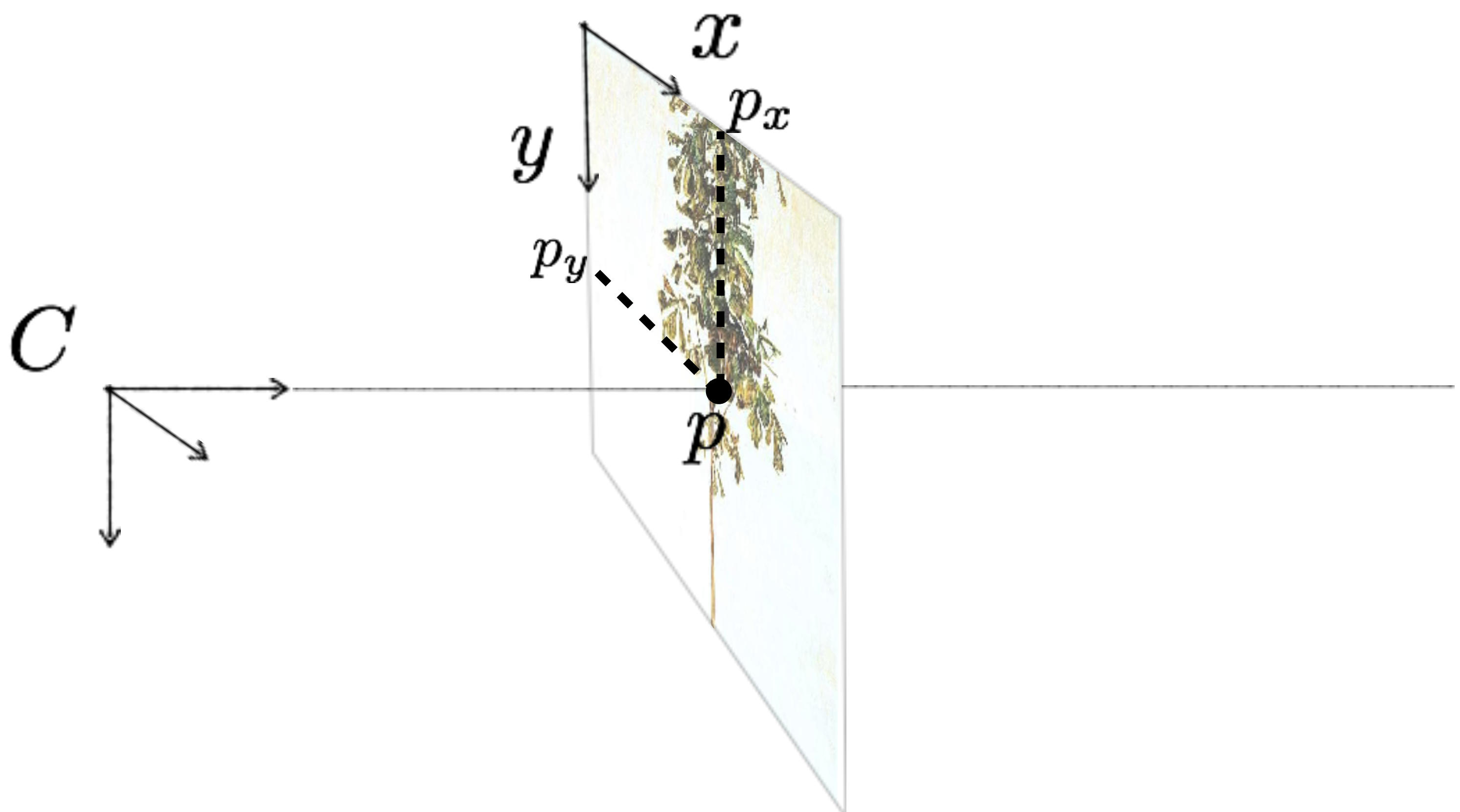
$$= \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} fX/Z + p_x \\ fY/Z + p_y \\ 1 \end{pmatrix}$$

To Pixel Coordinates



$$= \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} fX/Z + p_x \\ fY/Z + p_y \\ 1 \end{pmatrix}$$

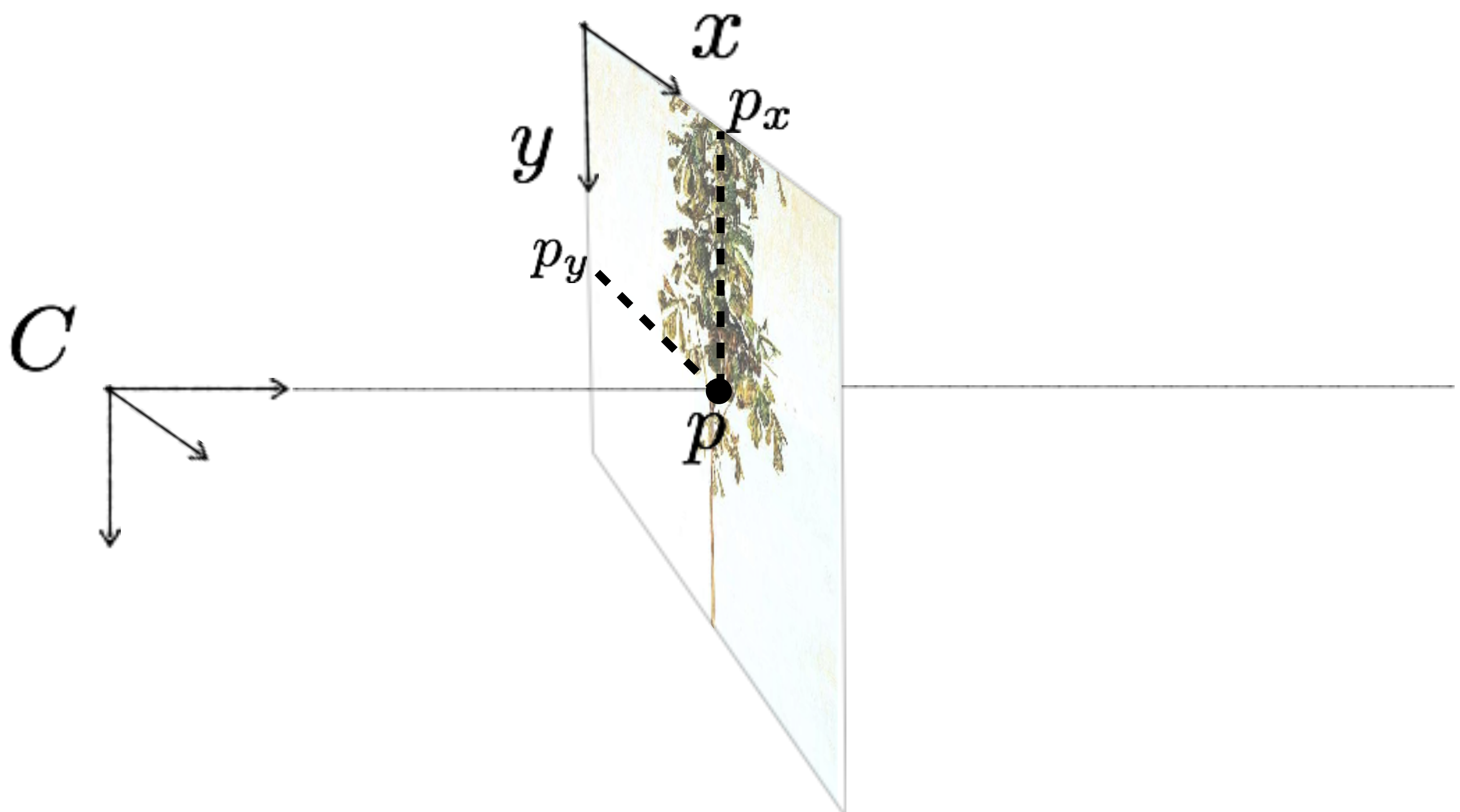
To Pixel Coordinates



Mapping to pixel coordinates: $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + p_y \\ y + p_y \end{pmatrix}$

$$= \begin{pmatrix} fX + zp_x \\ fY + zp_y \\ z \end{pmatrix} = \begin{pmatrix} fX/z + p_x \\ fY/z + p_y \\ 1 \end{pmatrix}$$

To Pixel Coordinates



Mapping to pixel coordinates: $\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x + p_y \\ y + p_y \end{pmatrix}$

Projection as matrix multiplication:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} fX + z p_x \\ fY + z p_y \\ z \end{pmatrix} = \begin{pmatrix} fX/z + p_x \\ fY/z + p_y \\ 1 \end{pmatrix}$$

Intrinsic Camera Calibration

General intrinsic camera calibration matrix:

skew:

non-rectangular pixels for $s \neq 0$

$$\mathrm{K} = \begin{pmatrix} f & s & p_x \\ 0 & \alpha f & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

aspect ratio

Intrinsic Camera Calibration

General intrinsic camera calibration matrix:

skew: non-rectangular pixels for $s \neq 0$

$$K = \begin{pmatrix} f & s & p_x \\ 0 & \alpha f & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

aspect ratio

α

In practice:

$$K = \begin{pmatrix} f & 0 & w/2 \\ 0 & f & h/2 \\ 0 & 0 & 1 \end{pmatrix}$$

MATLAB R2014b

HOME PLOTS APPS

New Script | New | Open | Find Files | Compare | Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace | Analyze Code | Run and Time | Clear Commands | Simulink Library | Layout | Preferences | Set Path | Parallel | Help | Community | Request Support | Add-Ons

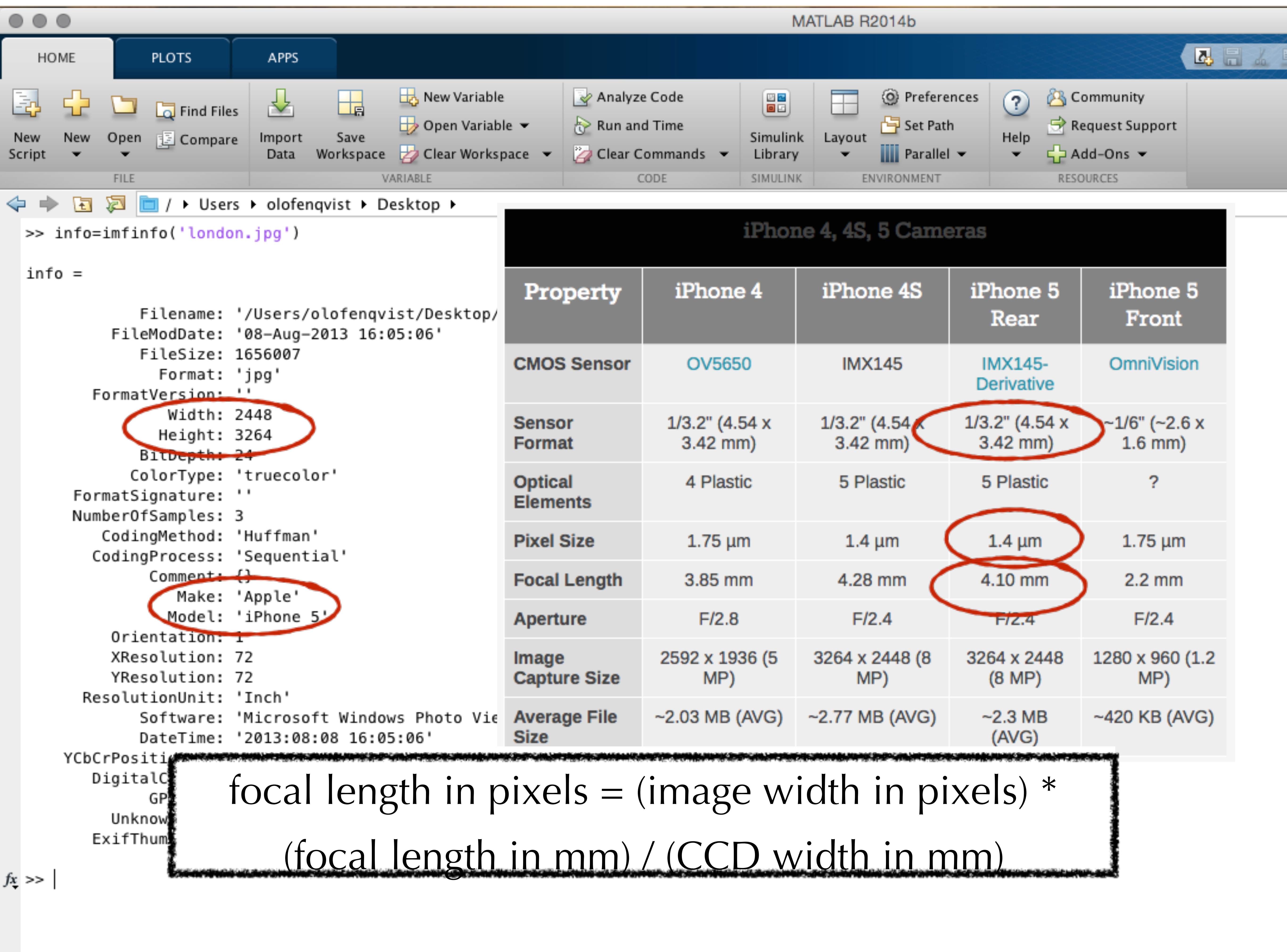
FILE | VARIABLE | CODE | SIMULINK | ENVIRONMENT | RESOURCES

/ ▸ Users ▸ olofenqvist ▸ Desktop ▸

```
>> info=imfinfo('london.jpg')

info =

           Filename: '/Users/olofenqvist/Desktop/
        FileModDate: '08-Aug-2013 16:05:06'
           FileSize: 1656007
             Format: 'jpg'
      FormatVersion: ''
              Width: 2448
             Height: 3264
           BitDepth: 24
          ColorType: 'truecolor'
    FormatSignature: ''
    NumberOfSamples: 3
       CodingMethod: 'Huffman'
      CodingProcess: 'Sequential'
            Comment: {}
               Make: 'Apple'
              Model: 'iPhone 5'
        Orientation: 1
        XResolution: 72
        YResolution: 72
     ResolutionUnit: 'Inch'
           Software: 'Microsoft Windows Photo Vie
           DateTime: '2013:08:08 16:05:06'
   YCbCrPositioning: 'Centered'
      DigitalCamera: [1x1 struct]
            GPSInfo: [1x1 struct]
        UnknownTags: [1x1 struct]
      ExifThumbnail: [1x1 struct]

fx >>
```

iPhone 4, 4S, 5 Cameras

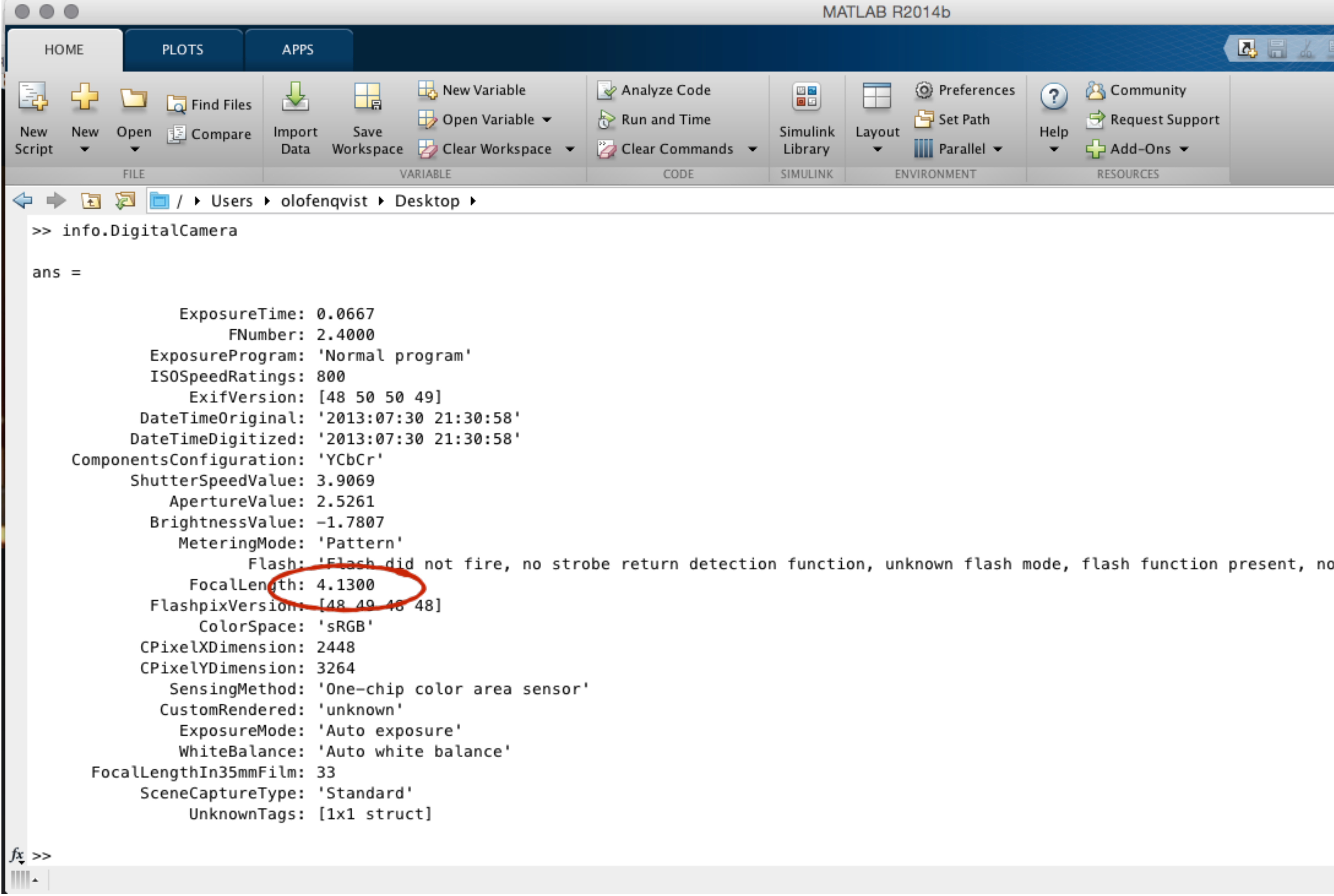
Property	iPhone 4	iPhone 4S	iPhone 5 Rear	iPhone 5 Front
CMOS Sensor	OV5650	IMX145	IMX145-Derivative	OmniVision
Sensor Format	1/3.2" (4.54 x 3.42 mm)	1/3.2" (4.54 x 3.42 mm)	1/3.2" (4.54 x 3.42 mm)	~1/6" (~2.6 x 1.6 mm)
Optical Elements	4 Plastic	5 Plastic	5 Plastic	?
Pixel Size	1.75 µm	1.4 µm	1.4 µm	1.75 µm
Focal Length	3.85 mm	4.28 mm	4.10 mm	2.2 mm
Aperture	F/2.8	F/2.4	F/2.4	F/2.4
Image Capture Size	2592 x 1936 (5 MP)	3264 x 2448 (8 MP)	3264 x 2448 (8 MP)	1280 x 960 (1.2 MP)
Average File Size	~2.03 MB (AVG)	~2.77 MB (AVG)	~2.3 MB (AVG)	~420 KB (AVG)



iPhone 4, 4S, 5 Cameras

Property	iPhone 4	iPhone 4S	iPhone 5 Rear	iPhone 5 Front
CMOS Sensor	OV5650	IMX145	IMX145-Derivative	OmniVision
Sensor Format	1/3.2" (4.54 x 3.42 mm)	1/3.2" (4.54 x 3.42 mm)	1/3.2" (4.54 x 3.42 mm)	~1/6" (~2.6 x 1.6 mm)
Optical Elements	4 Plastic	5 Plastic	5 Plastic	?
Pixel Size	1.75 µm	1.4 µm	1.4 µm	1.75 µm
Focal Length	3.85 mm	4.28 mm	4.10 mm	2.2 mm
Aperture	F/2.8	F/2.4	F/2.4	F/2.4
Image Capture Size	2592 x 1936 (5 MP)	3264 x 2448 (8 MP)	3264 x 2448 (8 MP)	1280 x 960 (1.2 MP)
Average File Size	~2.03 MB (AVG)	~2.77 MB (AVG)	~2.3 MB (AVG)	~420 KB (AVG)

focal length in pixels = (image width in pixels) * (focal length in mm) / (CCD width in mm)

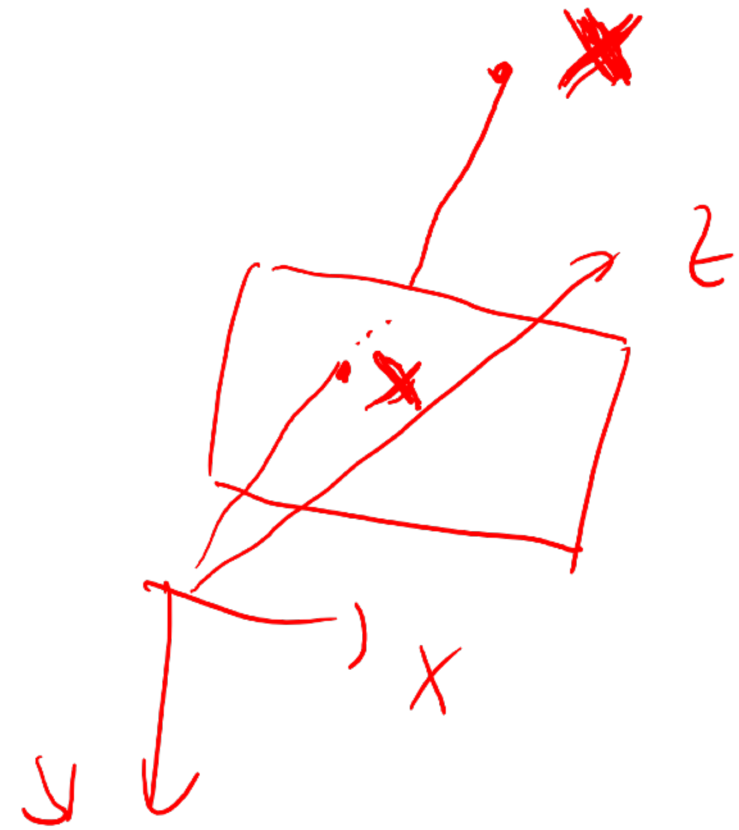


For details, see: http://www.cs.cornell.edu/~snavely/bundler/focal.html

Forward and Backward Projections

A 3D point maps to a 2D point:

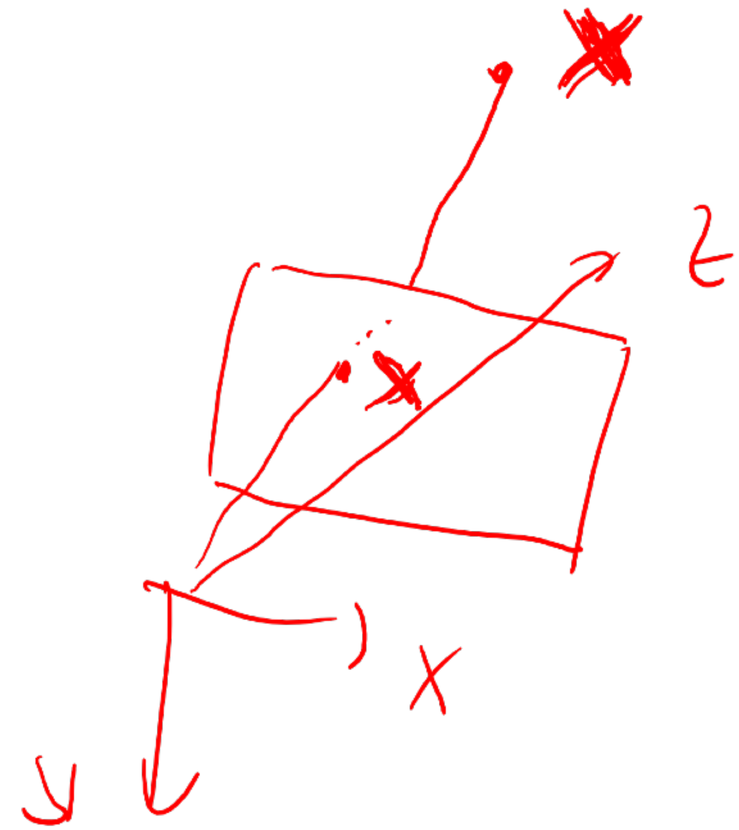
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathtt{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$



Forward and Backward Projections

A 3D point maps to a 2D point:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathtt{K} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$



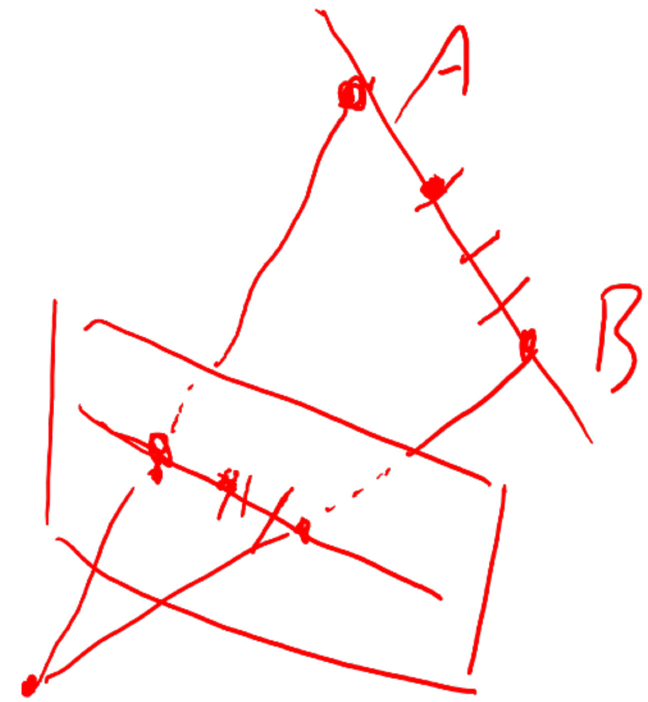
A 2D point maps to a ray:

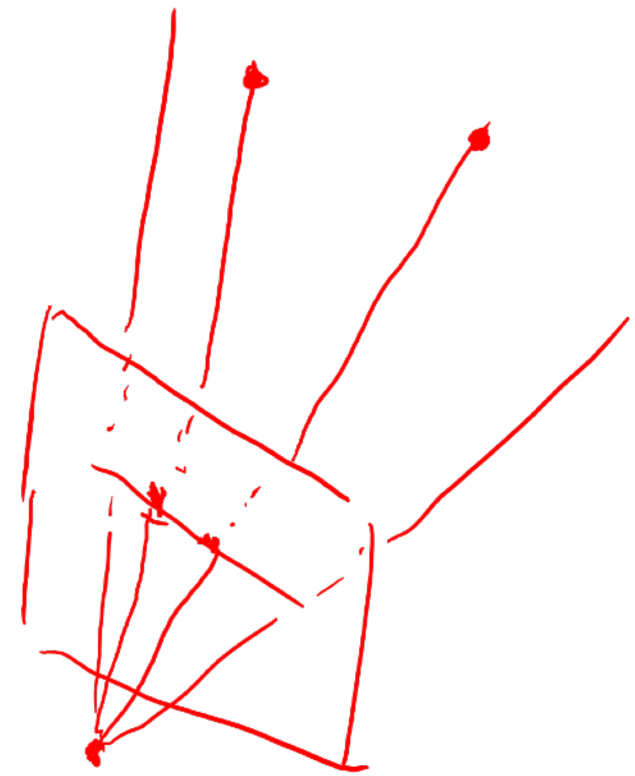
$$\delta \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathtt{K}^{-1} \begin{pmatrix} x \\ y \\ \cancel{z} \end{pmatrix} 1$$

Forward and Backward Projections

A 3D line maps to a 2D line:

$$\mathrm{K}\left(\mathbf{A}+\lambda\mathbf{B}\right)=\mathrm{K}\mathbf{A}+\lambda\mathrm{K}\mathbf{B}$$

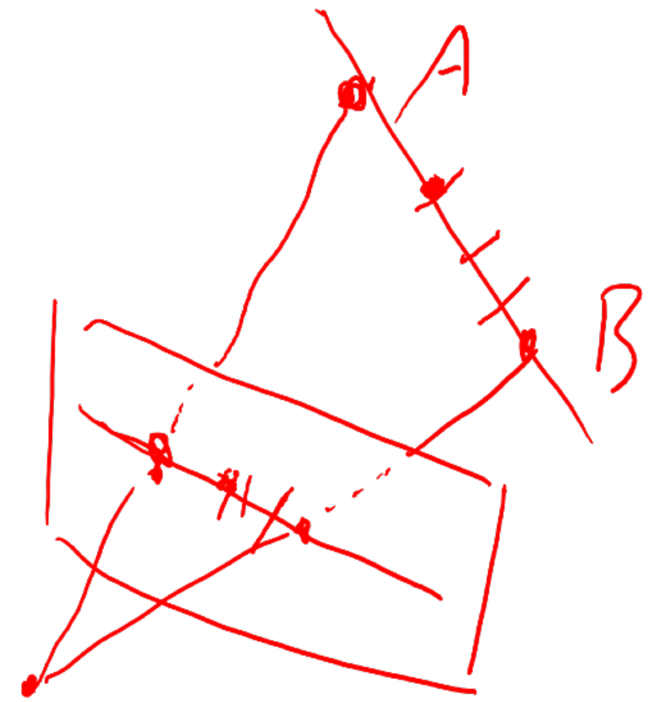




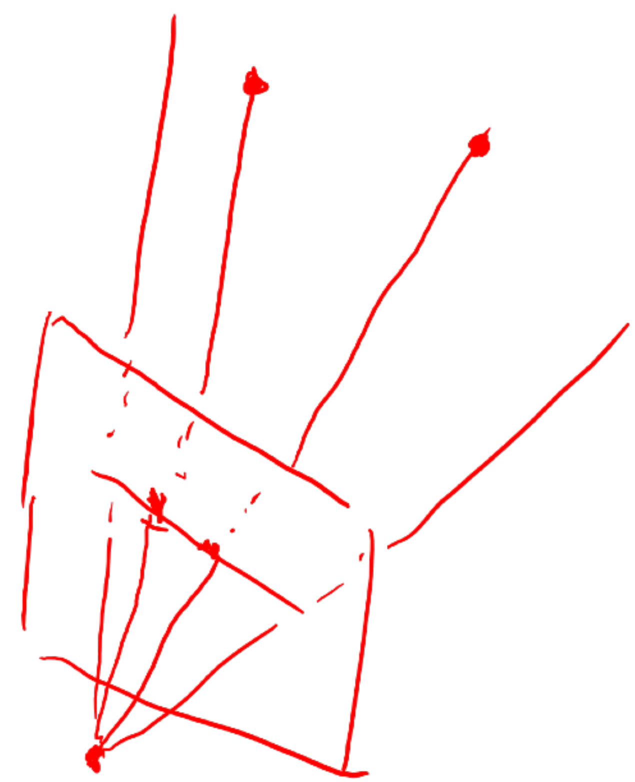
Forward and Backward Projections

A 3D line maps to a 2D line:

$$\mathrm{K}\left(\mathbf{A}+\lambda\mathbf{B}\right)=\mathrm{K}\mathbf{A}+\lambda\mathrm{K}\mathbf{B}$$



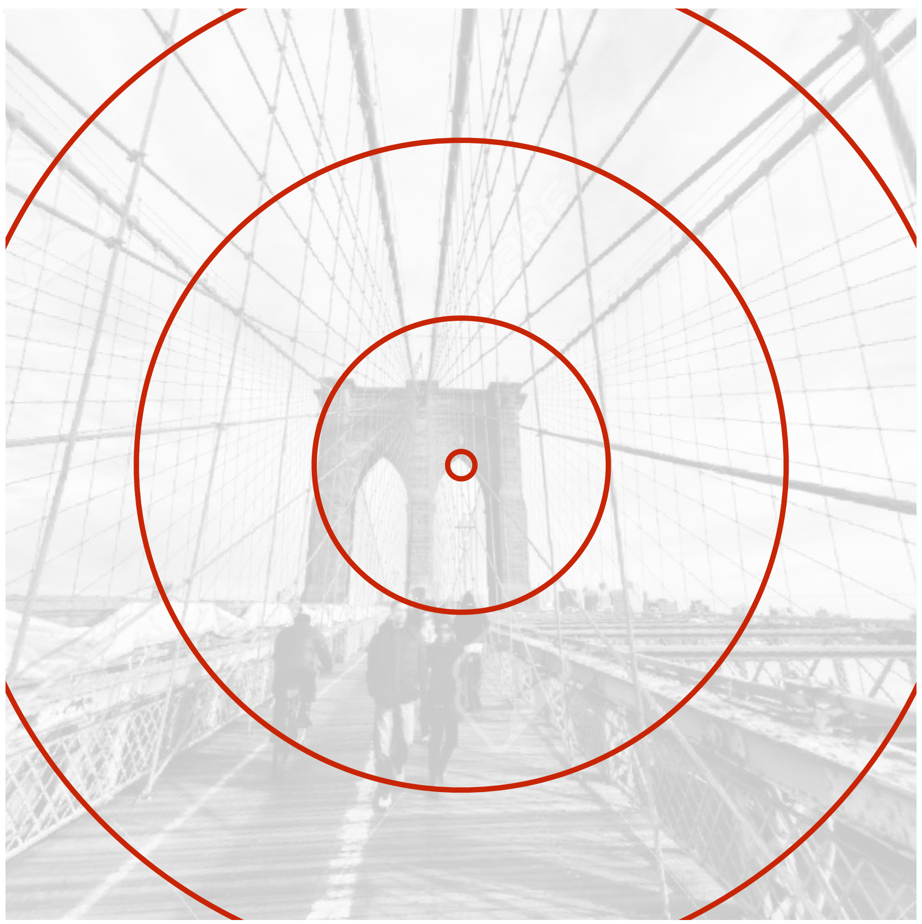
A 2D line maps to a 3D plane:



Lens Distortion

Accurate Physical Model?

Radial Distortion



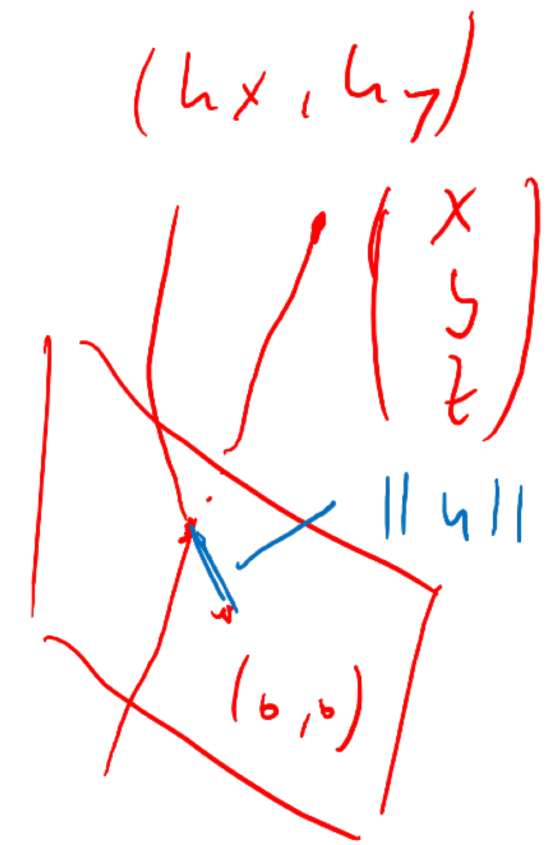
Pinhole camera

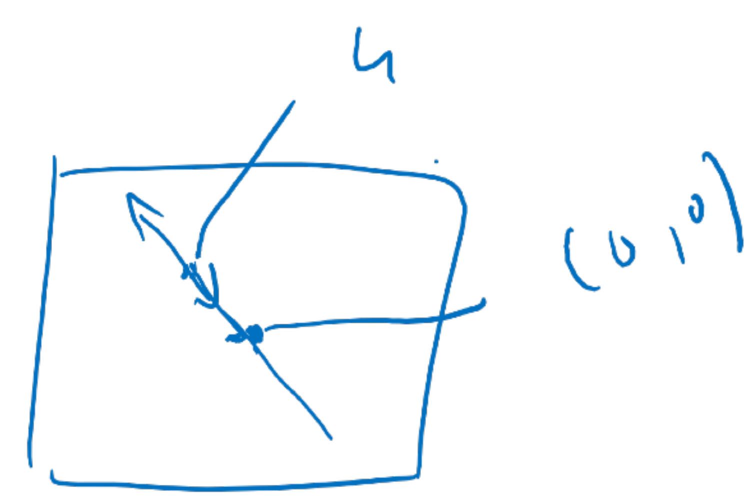
Fisheye lens camera

Polynomial Radial Distortion Model

Project 3D point into camera coordinates

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}$$





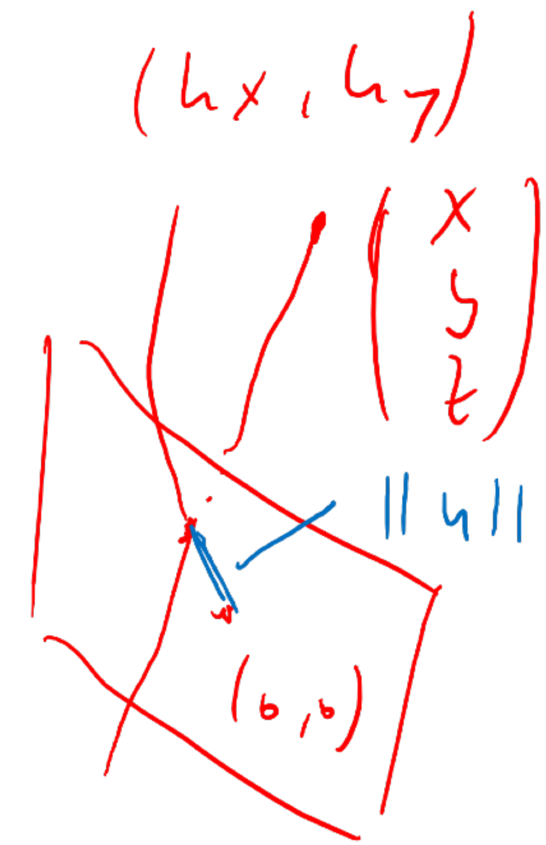
Polynomial Radial Distortion Model

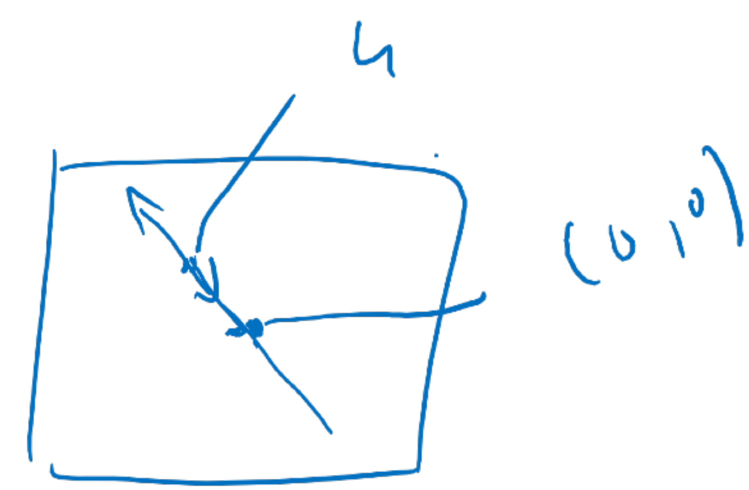
Project 3D point into camera coordinates

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}$$

Compute radial distortion factor

$$r(\mathbf{u}) = 1 + \kappa_1 ||\mathbf{u}||^2 + \kappa_2 ||\mathbf{u}||^4$$

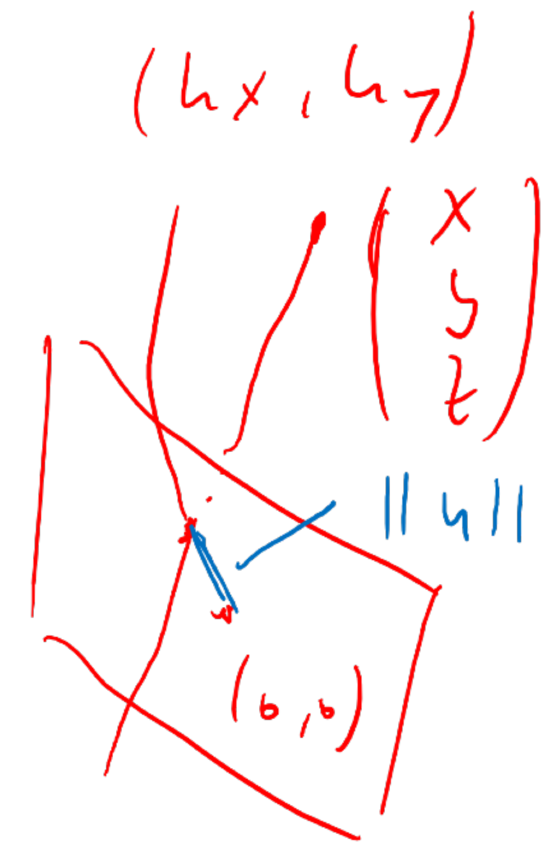




Polynomial Radial Distortion Model

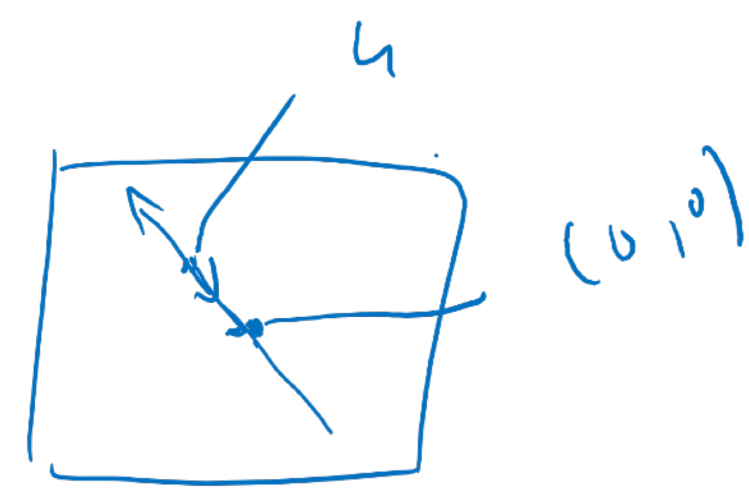
Project 3D point into camera coordinates

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} X/Z \\ Y/Z \end{pmatrix}$$



Compute radial distortion factor

$$r(\mathbf{u}) = 1 + \kappa_1 ||\mathbf{u}||^2 + \kappa_2 ||\mathbf{u}||^4$$

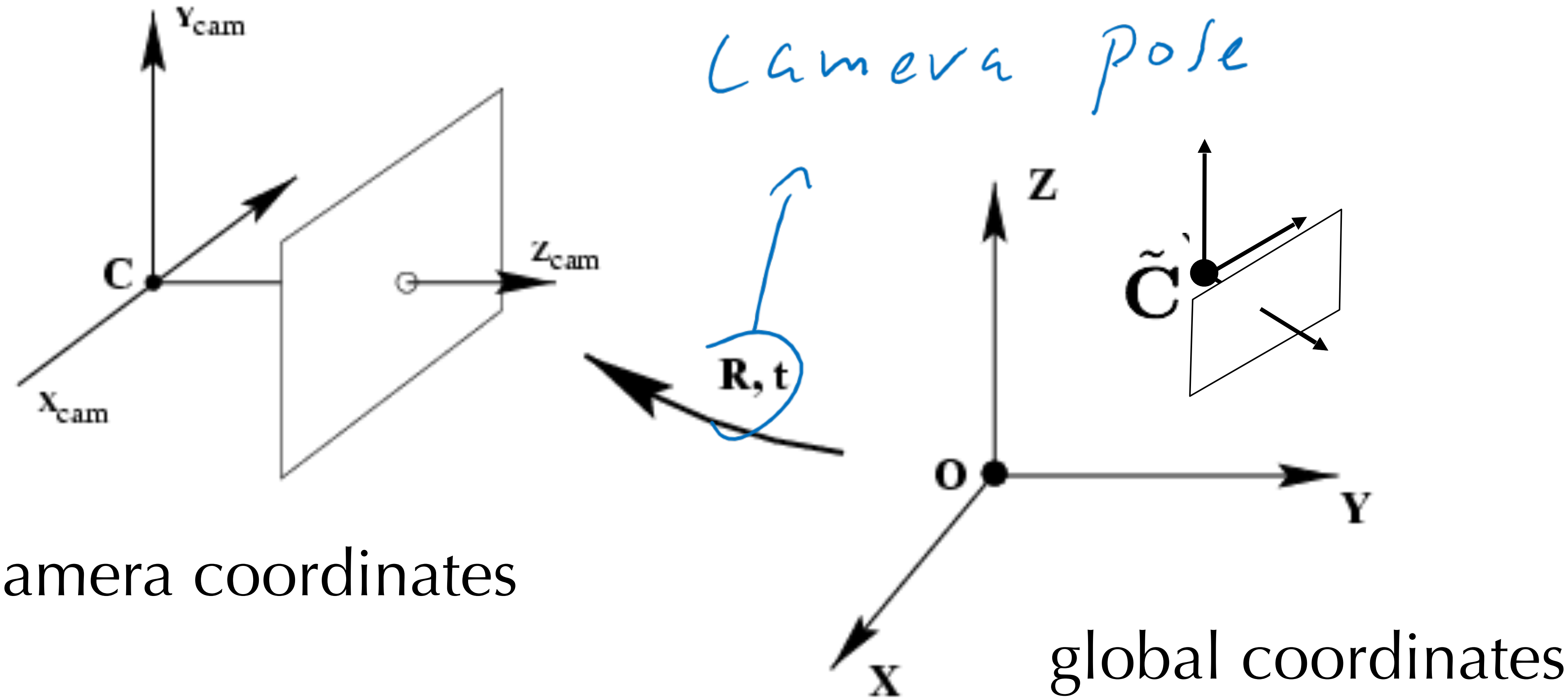


Compute pixel position

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \cdot r(\mathbf{u}) \cdot u_x + p_x \\ f \cdot r(\mathbf{u}) \cdot u_y + p_y \end{pmatrix}$$

Removing Lens Distortion

Camera Rotation and Translation

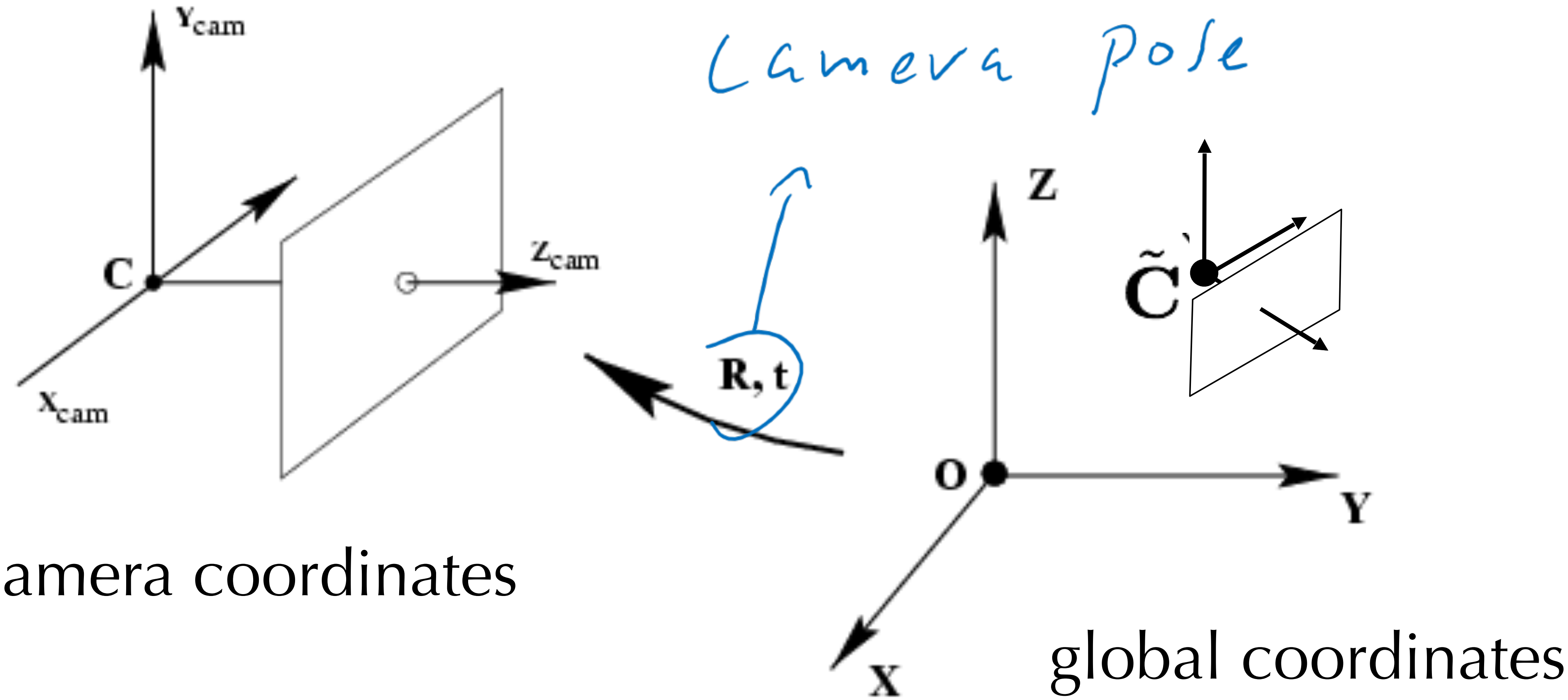


camera position $\tilde{C}$
camera translation t

$$\left.\begin{array}{l}\text{camera position } \tilde{C}\\ \text{camera translation } t\end{array}\right\}\; t = -R\tilde{C}, \quad \tilde{C} = -R^T t$$

figure adapted from Hartley and Zisserman, 2004

Camera Rotation and Translation



Transformation from global to camera coordinates:

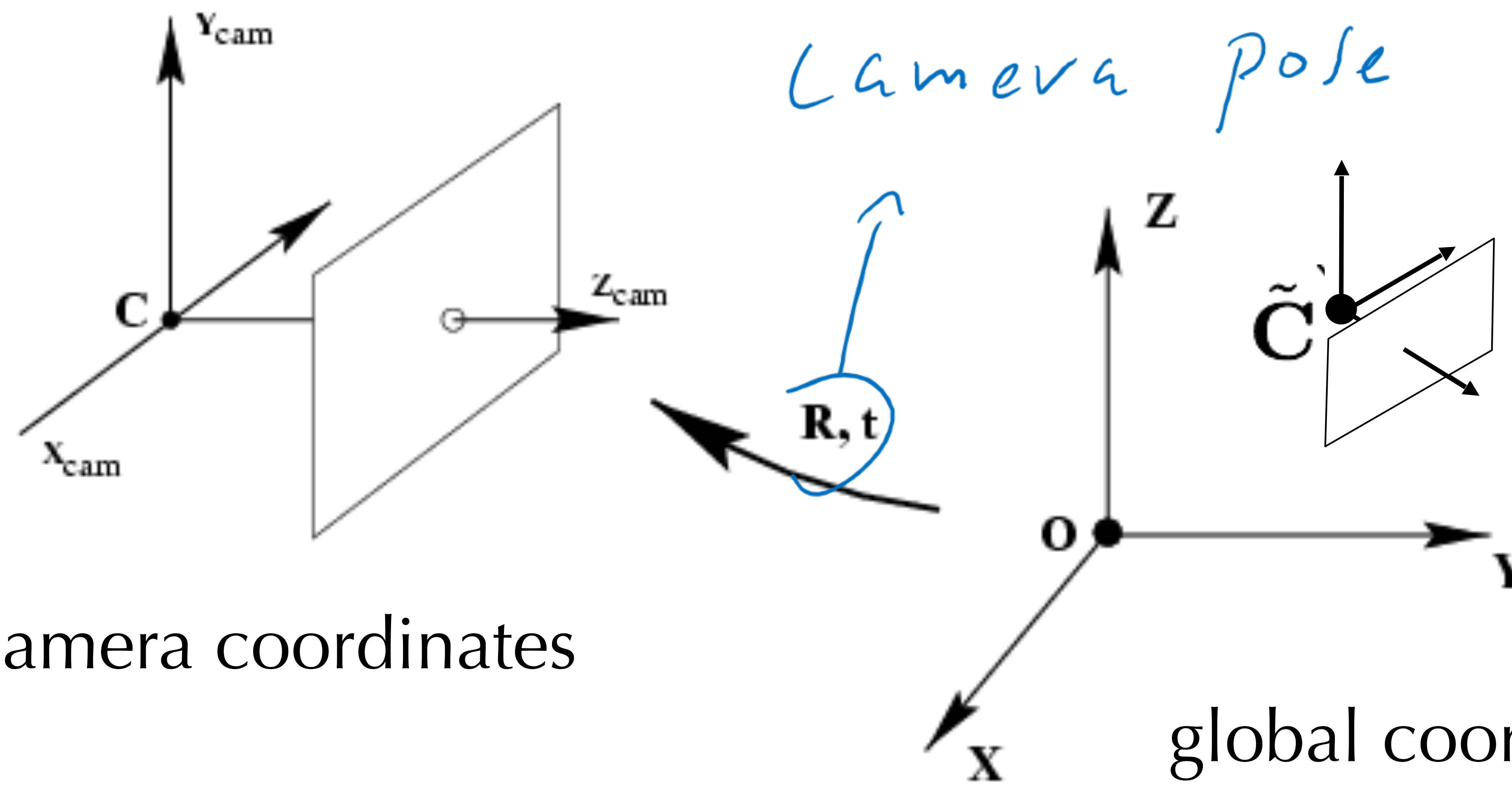
$$\mathbf{X}_{\text{cam}} = \mathrm{R}\left(\mathbf{X}_{\text{global}} - \tilde{\mathbf{C}}\right)$$

camera position $\tilde{C}$

camera translation t $\Big\}$ $t = -R\tilde{C}$, $\tilde{C} = -R^T t$

figure adapted from Hartley and Zisserman, 2004

Camera Rotation and Translation



camera coordinates

global coordinates

Transformation from global to camera coordinates:

$$\begin{aligned} \mathbf{X}_{\text{cam}} &= \mathrm{R}\left(\mathbf{X}_{\text{global}} - \tilde{\mathbf{C}}\right) \\ &= \mathrm{R}\mathbf{X} + \mathbf{t} \end{aligned}$$

camera position $\tilde{c}$

camera translation t $\Big\}$ $t = -R\tilde{c}$, $\tilde{c} = -R^T t$

figure adapted from Hartley and Zisserman, 2004

The Projection Matrix

Projection from 3D global coordinates to pixels:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathrm{K}\left(\mathrm{R}\mathbf{X}_{\mathrm{global}} + \mathbf{t}\right)$$

3x3 3x4 3x1

3x4

figure adapted from Hartley and Zisserman, 2004

The Projection Matrix

Projection from 3D global coordinates to pixels:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathtt{K}\left(\mathtt{R}\mathbf{X}_{\text{global}} + \mathbf{t}\right)$$

$$= \overset{3\times 3}{\mathtt{K}}\,\overset{3\times 4}{[\mathtt{R}|\mathbf{t}]}\overset{3\times 1}{\begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}}$$

3×4

figure adapted from Hartley and Zisserman, 2004

The Projection Matrix

Projection from 3D global coordinates to pixels:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathrm{K}\left(\mathrm{R}\mathbf{X}_{\text{global}} + \mathbf{t}\right)$$

3x3 3x4 3x1

$$= \overset{3\times 3}{\mathrm{K}} \overbrace{[\mathrm{R}|\mathbf{t}]}^{3\times 4} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

$$= \underbrace{\mathrm{K}\left[\mathrm{R}| - \mathrm{R}\tilde{\mathbf{C}}\right]}_{3\times 4} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

figure adapted from Hartley and Zisserman, 2004

The Projection Matrix

Projection from 3D global coordinates to pixels:

$$
\begin{aligned}
\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \mathtt{K}\left(\mathtt{R}\mathbf{X}_{\text{global}} + \mathbf{t}\right) \\
&= \overset{3\times 3}{\mathtt{K}} \overset{3\times 4}{\left[\mathtt{R}|\mathbf{t}\right]} \overset{3\times 1}{\begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}} \\
&= \mathtt{K}\left[\mathtt{R}| -\mathtt{R}\tilde{\mathbf{C}}\right] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \\
&= \overset{3\times 4}{\mathtt{P}} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}
\end{aligned}
$$

figure adapted from Hartley and Zisserman, 2004

The Projection Matrix

Projection from 3D global coordinates to pixels:

$$
\begin{aligned}
\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} &= \mathrm{K}\left(\mathrm{R}\mathbf{X}_{\text{global}} + \mathbf{t}\right) \\
&= \overset{3\times 3}{\mathrm{K}}\,\overset{3\times 4}{[\mathrm{R}|\mathbf{t}]} \overset{3\times 1}{\begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}} \\
&= \mathrm{K}\left[\mathrm{R}| - \mathrm{R}\tilde{\mathbf{C}}\right] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix} \\
&= \overset{3\times 4}{\mathrm{P}} \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}
\end{aligned}
$$

projection matrix

figure adapted from Hartley and Zisserman, 2004

Rolling Shutter Effect

[video]

Rolling Shutter Effect

[video]

Rolling Shutter Effect

Global shutter

Rolling shutter

youtu.be/7TGKFdrY9aw

Slide credit: Cenek Albl

Rolling Shutter Effect

Global shutter | Rolling shutter

youtu.be/7TGKFdrY9aw

Slide credit: Cenek Albl

Rolling Shutter Effect

Global shutter Rolling shutter

youtu.be/7TGKFdrY9aw

Slide credit: Cenek Albl

Rolling Shutter Effect

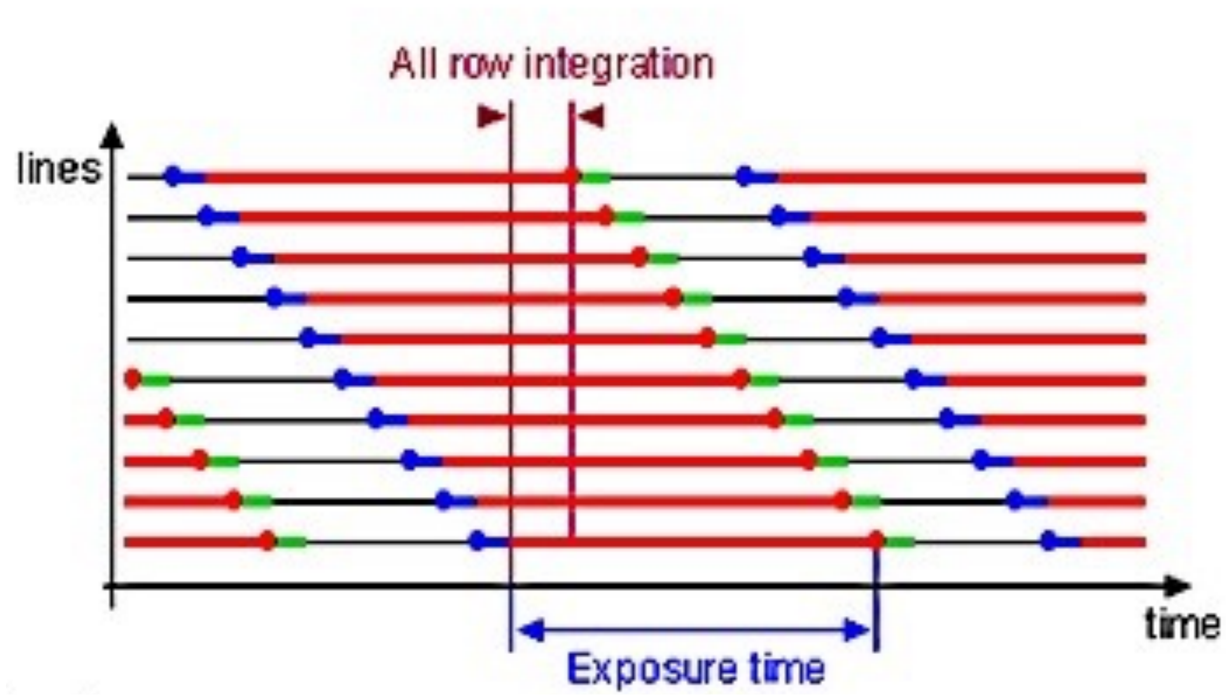


Image recorded line by line

Slide credit: Cenek Albl

Rolling Shutter Effect

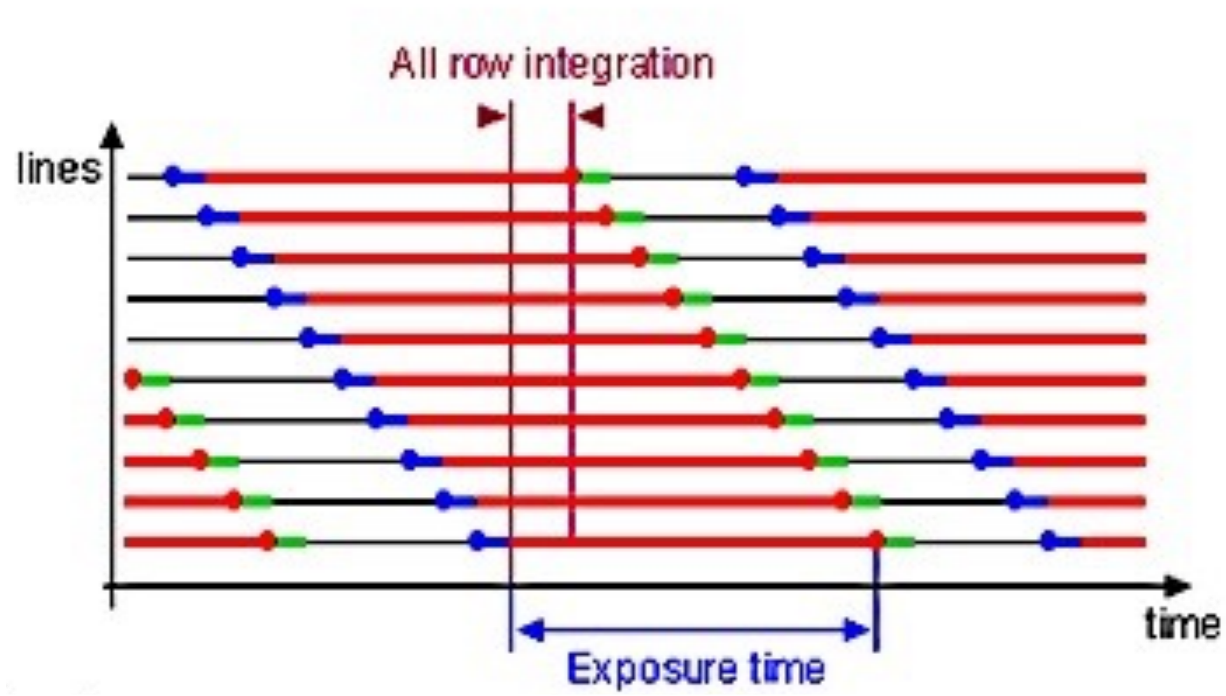


Image recorded line by line

Rolling shutter effect

Slide credit: Cenek Albl

Rolling Shutter Effect

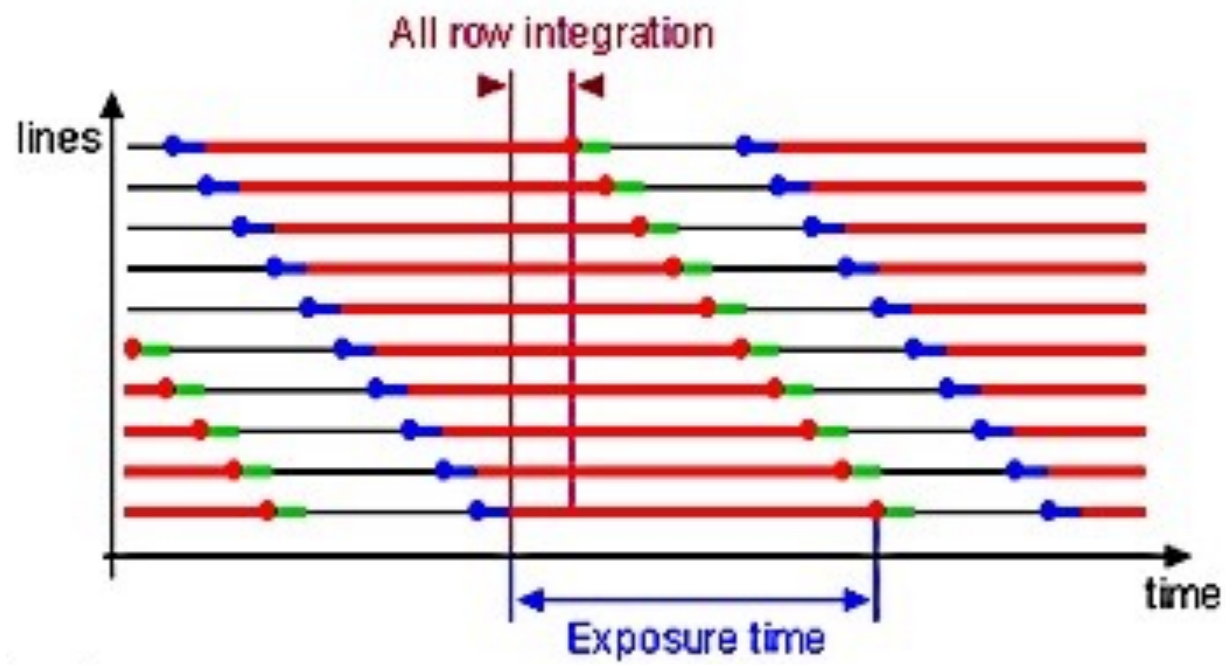


Image recorded line by line

Rolling shutter effect

- Rolling shutter cameras cheaper
- Faster frame rates

Slide credit: Cenek Albl

Rolling Shutter Effect

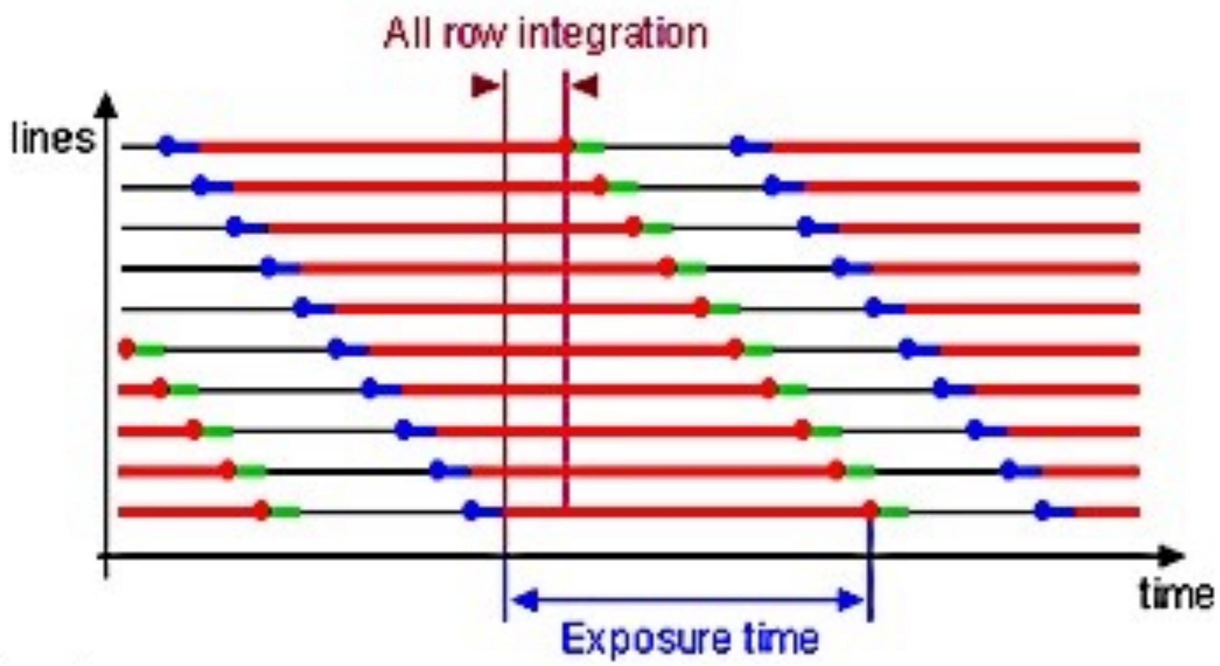


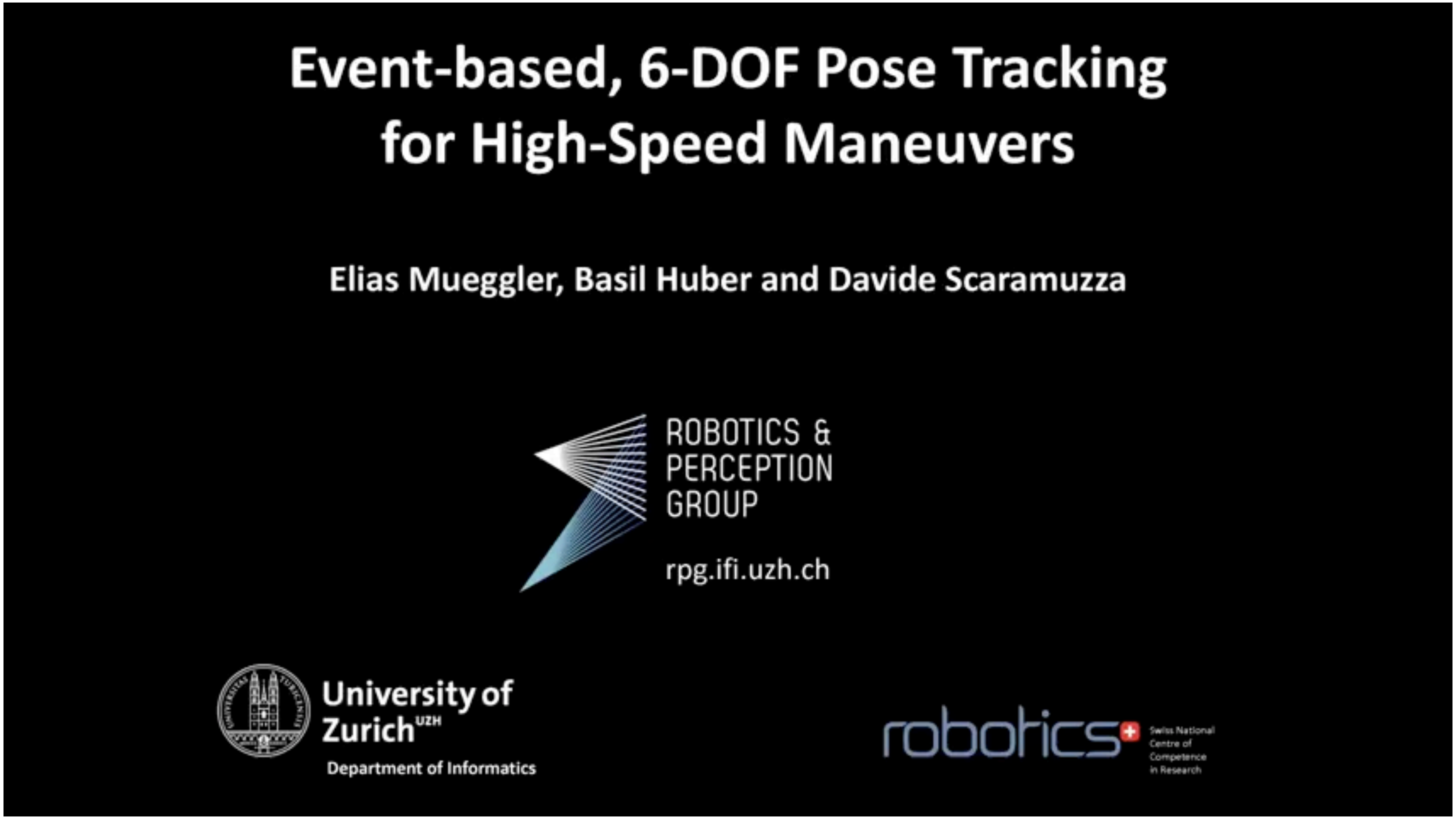
Image recorded line by line

Rolling shutter effect

- Rolling shutter cameras cheaper
- Faster frame rates
- Better adaption to illumination changes

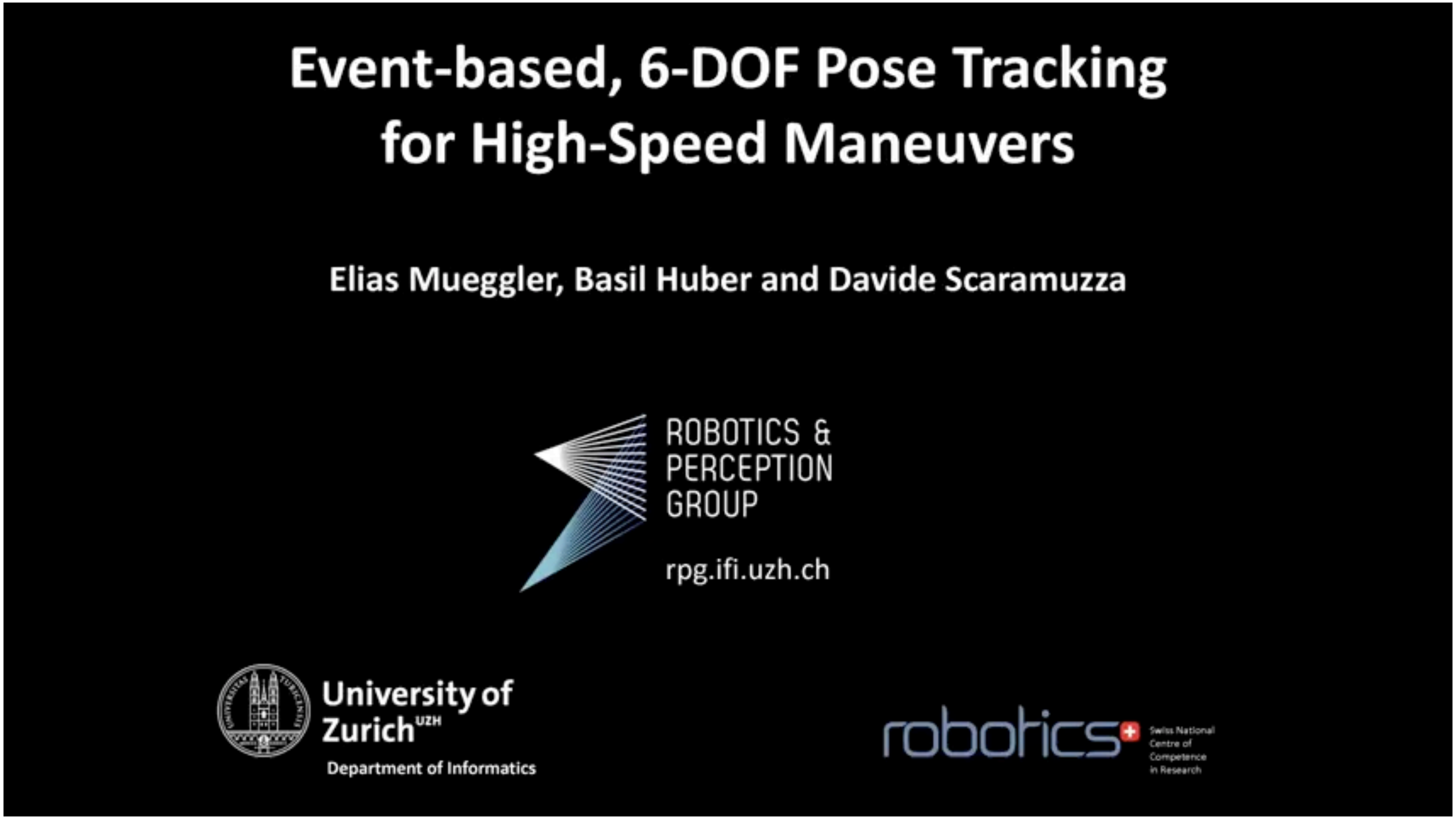
Slide credit: Cenek Albl

Event Cameras



video

Event Cameras



video

Lessons Learned

- Main lessons from this lecture
  - 2D projective transformations via homogeneous coordinates
  - 3D-to-2D projection via homogenous coordinates
  - Transformation from global to camera coordinates
  - What are $\underbrace{\text{intrinsic}}_{K}$ and $\underbrace{\text{extrinsic camera}}_{R,\ t}$ parameters?
  - What is a projection matrix?

Lessons Learned

- Main lessons from this lecture
  - 2D projective transformations via homogeneous coordinates
  - 3D-to-2D projection via homogenous coordinates
  - Transformation from global to camera coordinates
  - What are intrinsic and extrinsic camera parameters?
  - What is a projection matrix?

K

R, t

- Next lecture: More 3D Geometry