CSE455: Computer Vision

Geometric Primitives & Transformations

Fatemeh Ghezloo

Reference: Szeliski 2.1

What is the most popular topic at CVPR?

	Publication	h5-index	h5-median
1.	Nature	467	707
2.	The New England Journal of Medicine	439	876
3.	Science	424	665
4.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	422	681
5.	The Lancet	368	688
6.	Nature Communications	349	456
7.	Advanced Materials	326	415
8.	Cell	316	503
9.	Neural Information Processing Systems	309	503
10.	International Conference on Learning Representations	303	563

h5-index: largest number h such that h articles published in the last 5 years have at least h citations each.

https://scholar.google.com/citations?view_op=top_venues&hl=en

CVPR 2023 by the Numbers

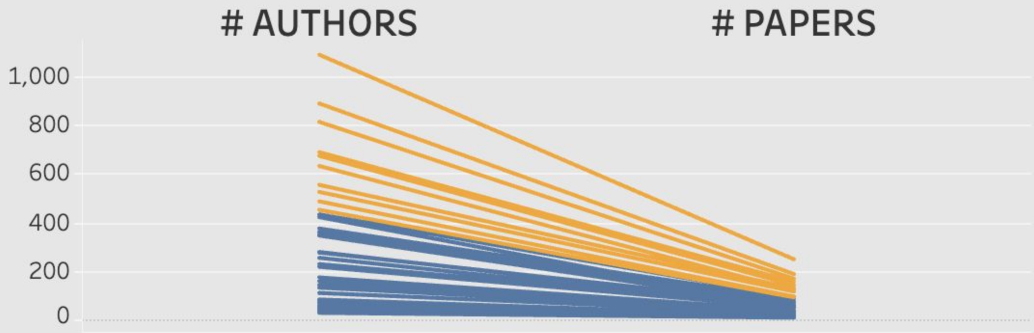
Selecting a category below changes the paper list on the right.

Select
- (•) All
- ( ) Award Candidate
- ( ) Highlight
- ( ) Paper

PICK INSTITUTIONS
(All)

SELECT ↓ Top 10 overall by number of authors

		AUTHORS	PAPERS
1	3D from multi-view and sensors	1,090	246
2	Image and video synthesis and generation	889	185
3	Humans: Face, body, pose, gesture, movement	813	166
4	Transfer, meta, low-shot, continual, or long-tail learning	688	153
5	Recognition: Categorization, detection, retrieval	673	139
6	Vision, language, and reasoning	631	118
7	Low-level vision	553	126
8	Segmentation, grouping and shape analysis	524	113
9	Deep learning architectures and techniques	485	92
10	Multi-modal learning	450	89
11	3D from single images	431	91
12	Medical and biological vision, cell microscopy	420	53
13	Video: Action and event understanding	373	83
14	Autonomous driving	359	69
15	Self-supervised or unsupervised representation learning	349	71
16	Datasets and evaluation	344	54
17	Scene analysis and understanding	276	54
18	Adversarial attack and defense	274	61
19	Efficient and scalable vision	252	48
20	Computational imaging	226	53
21	Video: Low-level analysis, motion, and tracking	215	46
22	Vision applications and systems	171	35
23	Vision + graphics	155	32
24	Robotics	141	23
25	Transparency, fairness, accountability, privacy, ethics in vision	129	30
26	Explainable computer vision	107	24
27	Embodied vision: Active agents, simulation	80	14
28	Document analysis and understanding	72	12
29	Machine learning (other than deep learning)	65	14
30	Physics-based vision and shape-from-X	55	12



3D from multi-view and sensors

Award Candidate | Highlight | Paper

33	NeuMap: Neural Coordinate Mapping by Auto-Transdecoder for Camera Localization
76	Object Pose Estimation with Statistical Guarantees: Conformal Keypoint Detection and Geometric Uncertainty Propagation
120	NeuralUDF: Learning Unsigned Distance Fields for Multi-view Reconstruction of Surfaces with Arbitrary Topologies
143	NEF: Neural Edge Fields for 3D Parametric Curve Reconstruction from Multi-view Images
330	Looking Through the Glass: Neural Surface Reconstruction Against High Specular Reflections
357	Multi-View Azimuth Stereo via Tangent Space Consistency

https://cvpr2023.thecvf.com/Conferences/2023/AcceptedPapers

Why do we care about Geometry?

- Self-driving cars: navigation, collision avoidance
- Robots: navigation, manipulation
- Graphics & AR/VR: augment or generate images
- Photogrammetry (architecture, surveys)
- Pattern Recognition (web, medical imaging, etc)

What will we learn today?

- Why Geometric Vision Matters
- Geometric Primitives in 2D & 3D
- 2D & 3D Transformations

General Advice / Observations

- Fundamentals: need to (eventually) feel easy
- Try to do the math in parallel live in class!
- If not grokking this: practice later, ask on Ed, OH
- Lots of good (hard?) exercises in Szeliski's book

What will we learn today?

Why Geometric Vision Matters

Geometric Primitives in 2D & 3D

2D & 3D Transformations

Images are
2D projections of
the 3D world

Simplified Image Formation

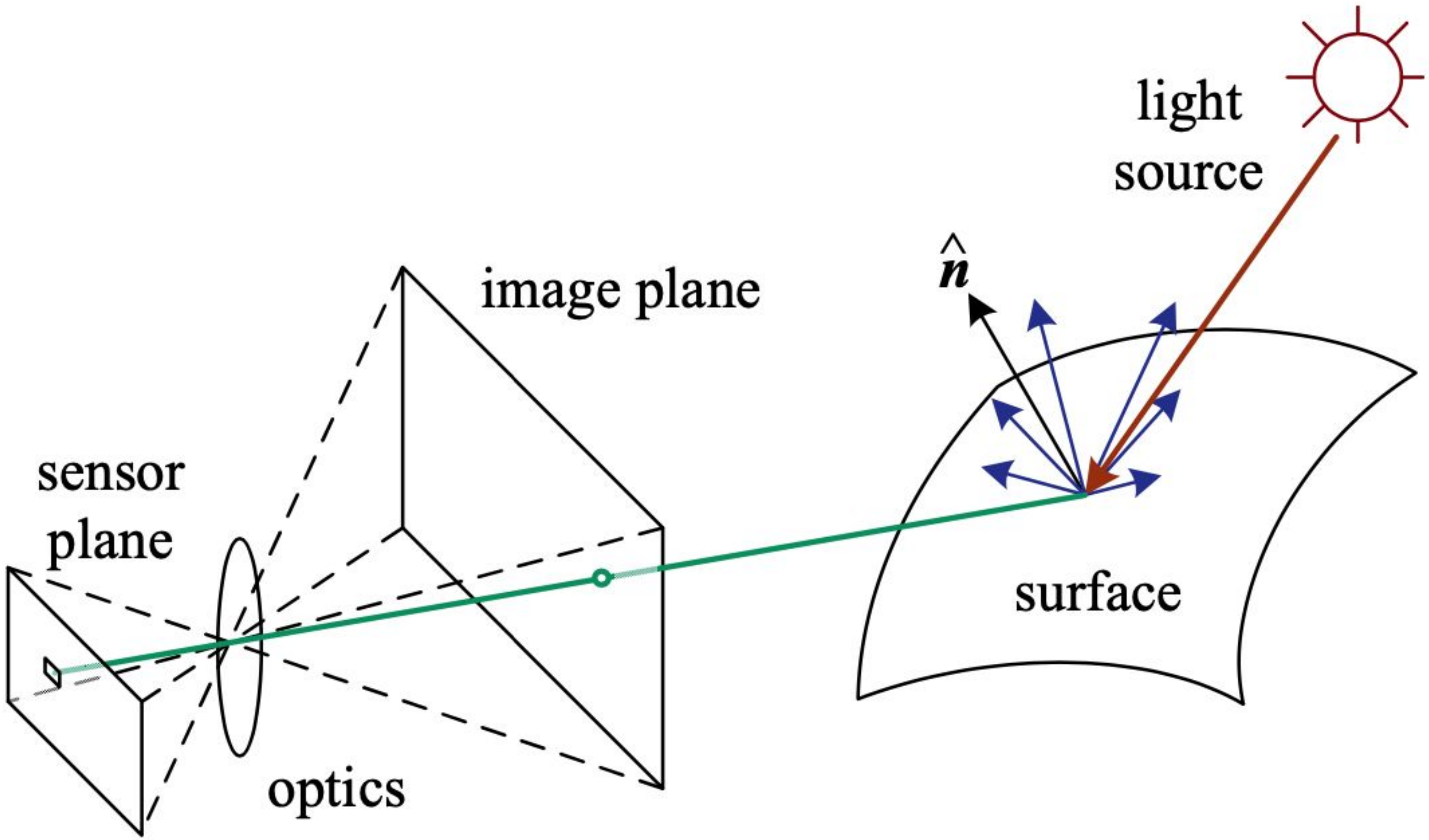
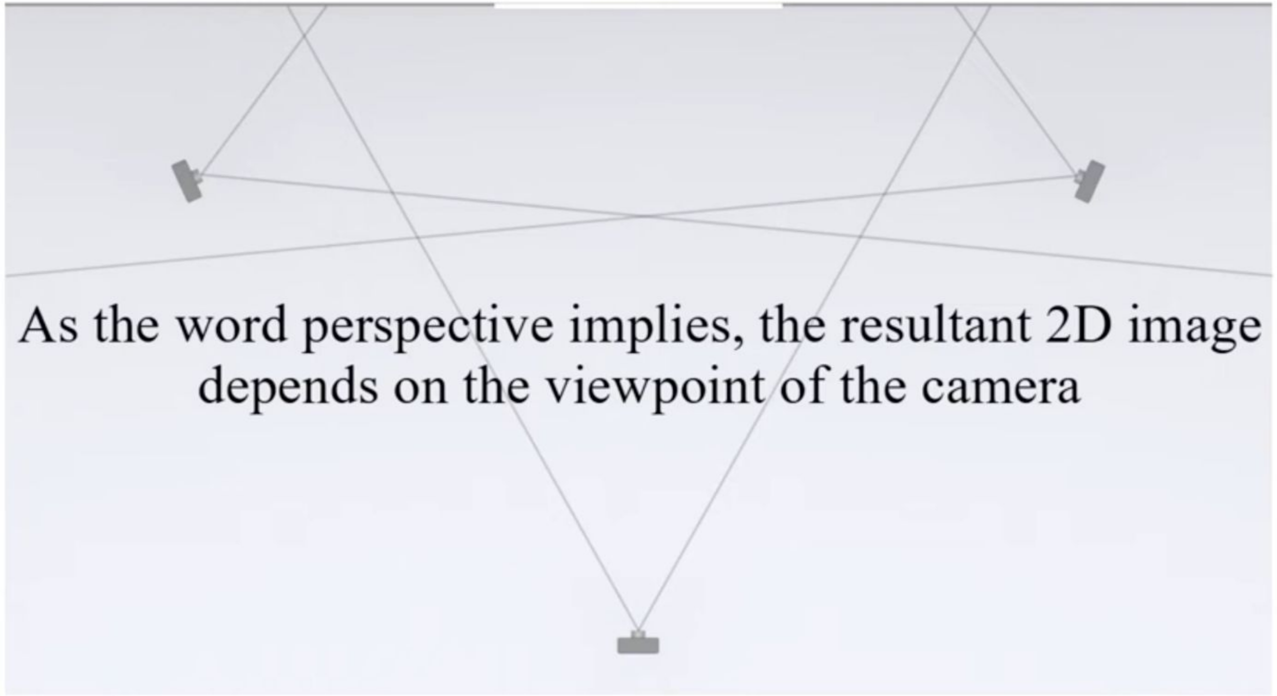


Figure: R. Szeliski

Perspective Projection



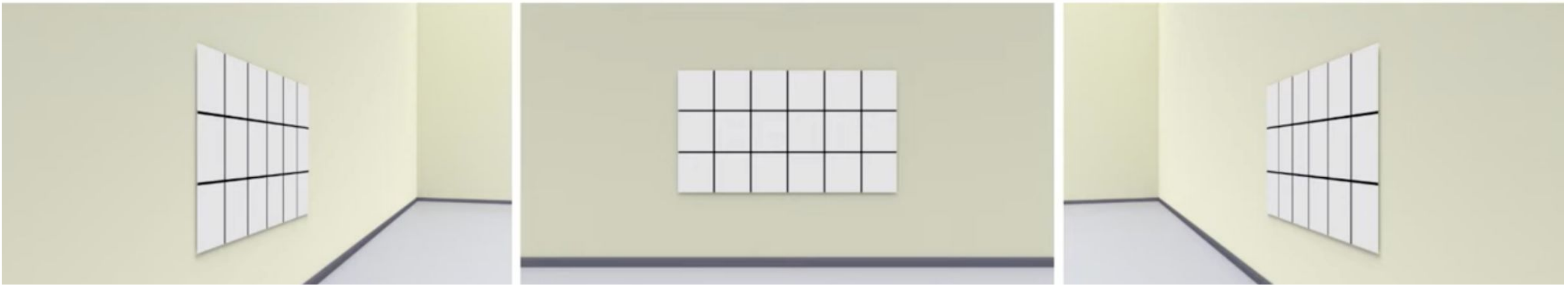
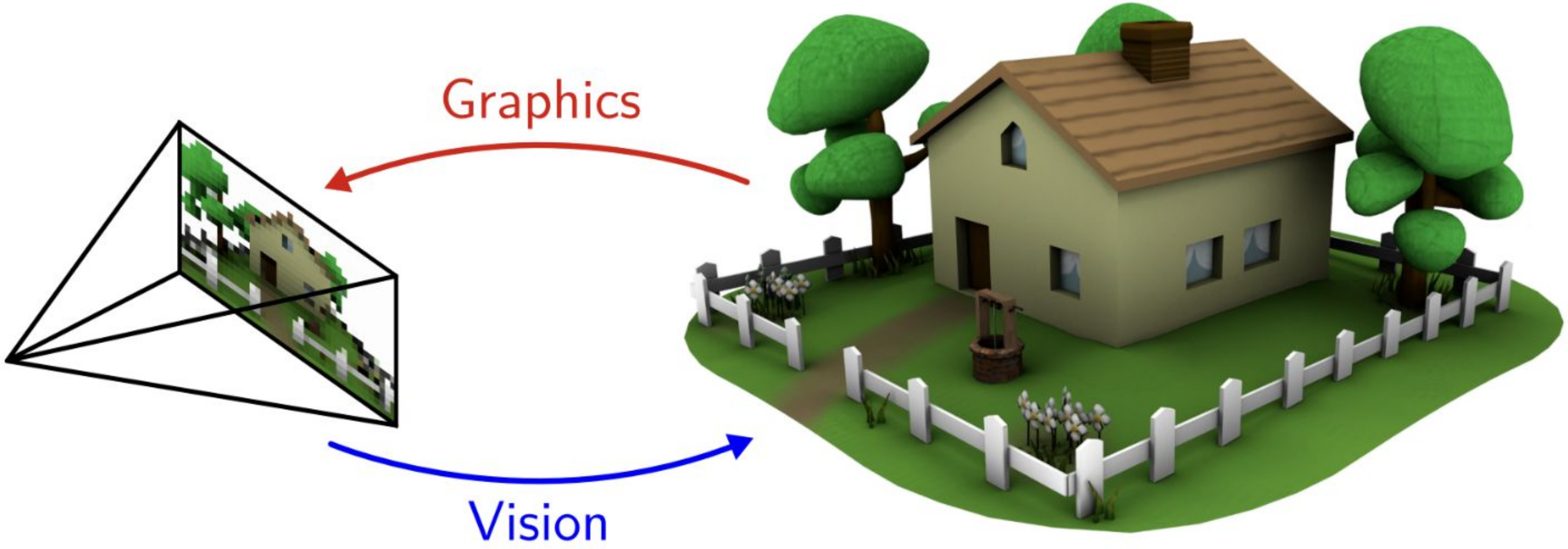


Figure: https://www.youtube.com/@huseyin_ozde...

Can we understand the 3D world from 2D images?

CV is an ill-posed inverse problem

2D Image **3D Scene**



Pixel Matrix

217	191	252	255	239
102	80	200	146	138
159	94	91	121	138
179	106	136	85	41
115	129	83	112	67
94	114	105	111	89

Objects Material

Shape/Geometry Motion

Semantics 3D Pose

Slide credit: Andreas Geiger

What will we learn today?

Why Geometric Vision Matters

Geometric Primitives in 2D & 3D

2D & 3D Transformations

Points in Cartesian and Homogeneous Coordinates

2D points: $\mathbf{x} = (x, y) \in \mathcal{R}^2$ or column vector $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

3D points: $\mathbf{x} = (x, y, z) \in \mathcal{R}^3$(often noted $\mathbf{X}$ or $\mathbf{P}$)

Homogeneous coordinates: append a 1

$$\bar{\mathbf{x}} = (x, y, 1) \qquad \bar{\mathbf{x}} = (x, y, z, 1)$$

Why?

Homogeneous coordinates in 2D

2D Projective Space: $\mathcal{P}^2 = \mathcal{R}^3 - (0,0,0)$ (same story in 3D with $\mathcal{P}^3$)

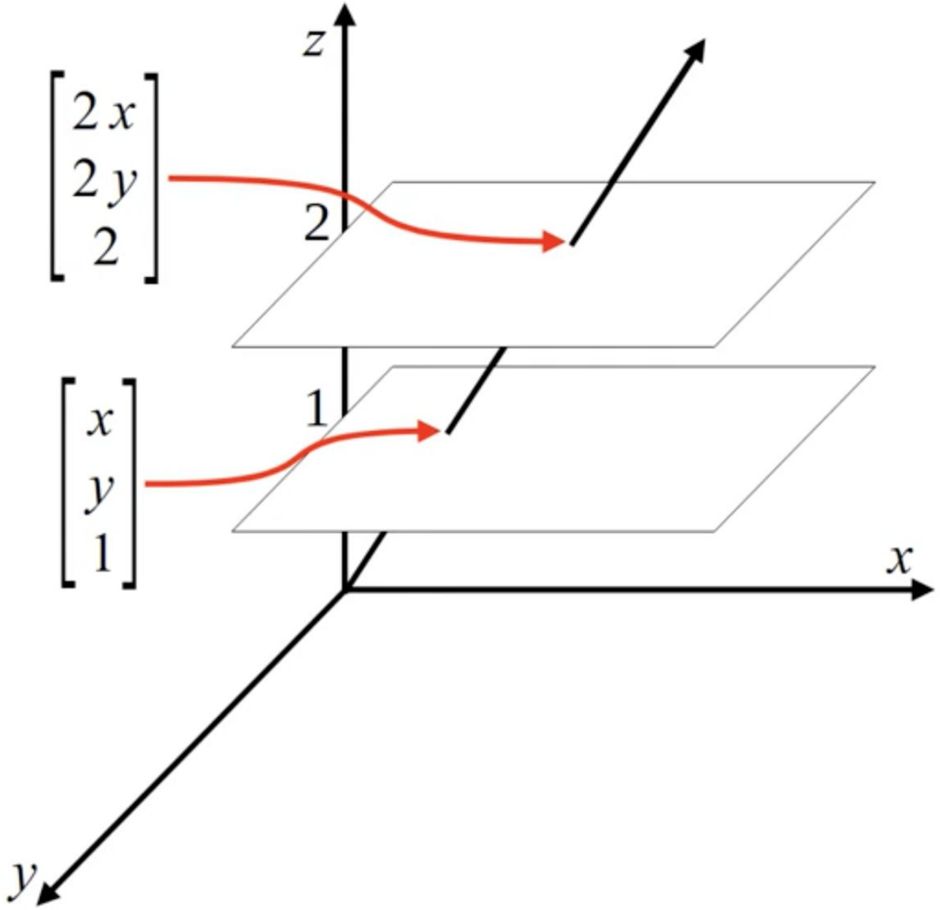
- heterogeneous → homogeneous $\begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

- homogeneous → heterogeneous $\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} x/w \\ y/w \end{bmatrix}$

- points differing only by scale are *equivalent*: $(x, y, w) \sim \lambda\,(x, y, w)$

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\mathbf{x}},$$

Homogeneous coordinates in 2D



In homogeneous coordinates, a point and its scaled versions are same

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} \qquad w \neq 0$$

Figure: https://www.youtube.com/@huseyin_ozde...

Everything is easier in Projective Space

2D Lines:

Representation: $l = (a, b, c)$

Equation: $ax + by + c = 0$

In homogeneous coordinates: $\bar{x}^T l = 0$

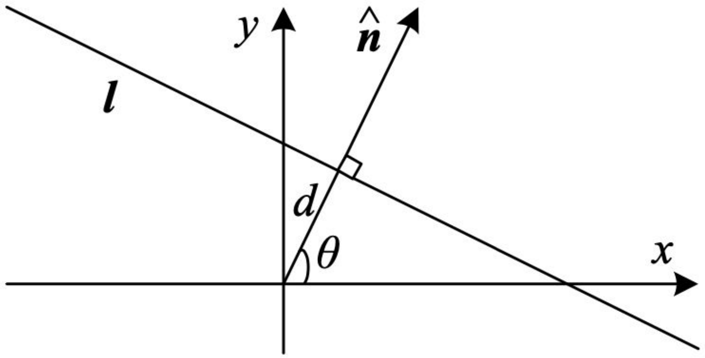
General idea: homogenous coordinates unlock the full power of linear algebra!

Everything is easier in Projective Space

2D Lines:

$$\tilde{\mathrm{x}}^{\mathrm{T}}\mathrm{l} = 0, \forall \tilde{\mathrm{x}} = (x, y, w) \in P^2$$

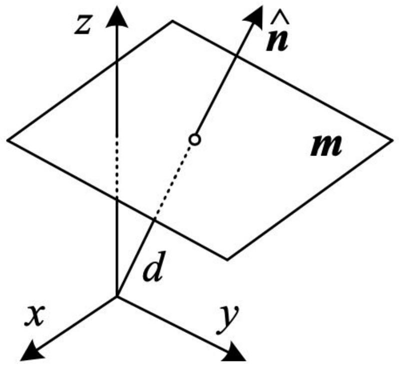
$$\mathbf{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\mathbf{n}}, d) \text{ with } \|\hat{\mathbf{n}}\| = 1$$



3D planes: same!

$$\tilde{\mathrm{x}}^{\mathrm{T}}\mathrm{m} = 0, \forall \tilde{\mathrm{x}} = (x, y, z, w) \in P^3$$

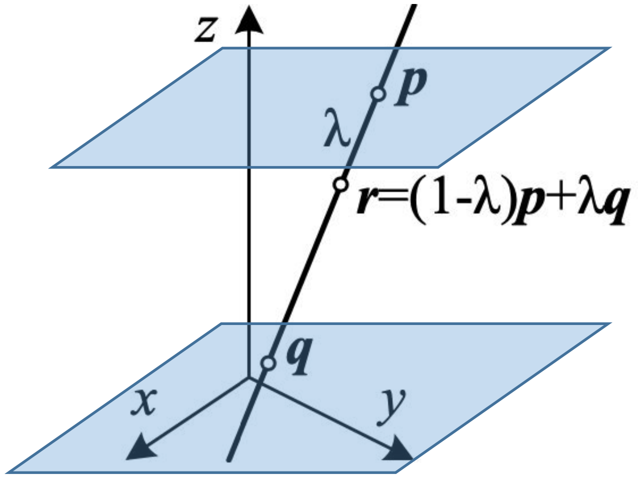
$$\mathbf{m} = (\hat{n}_x, \hat{n}_y, \hat{n}_z, d) = (\hat{\mathbf{n}}, d) \text{ with } \|\hat{\mathbf{n}}\| = 1$$



Lines in 3D

Two-point parametrization:

$$\mathbf{r} = (1-\lambda)\mathbf{p} + \lambda\mathbf{q} \qquad \tilde{\mathbf{r}} = \mu\tilde{\mathbf{p}} + \lambda\tilde{\mathbf{q}}$$



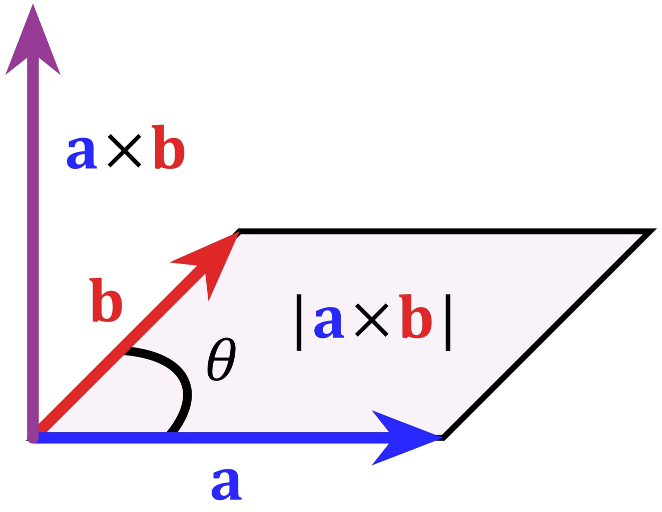
Two-plane parametrization:

coordinates (x_0, y_0) & (x_1, y_1) of intersection

with planes at $z = 0, 1$ (or other planes)

Cross-product quick reminder

$$\mathbf{a} \times \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin(\theta) \, \mathbf{n}$$



$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Benefits of Homogeneous Coordinates

- Line – Point duality:
  - line between two 2D points: $\tilde{\mathbf{l}} = \tilde{\mathbf{x}}_1 \times \tilde{\mathbf{x}}_2$
  - intersection of two 2D lines: $\tilde{\mathbf{x}} = \tilde{\mathbf{l}}_1 \times \tilde{\mathbf{l}}_2$
- Representation of Infinity:
  - points at infinity: $(x, y, 0)$; line at infinity: $(\mathbf{0,0,1})$
- Parallel & vertical lines are easy (take-home: intersect //)
- Makes 2D & 3D transformations linear!

Questions?

What will we learn today?

Why Geometric Vision Matters

Geometric Primitives in 2D & 3D

2D & 3D Transformations

The camera as a coordinate transformation

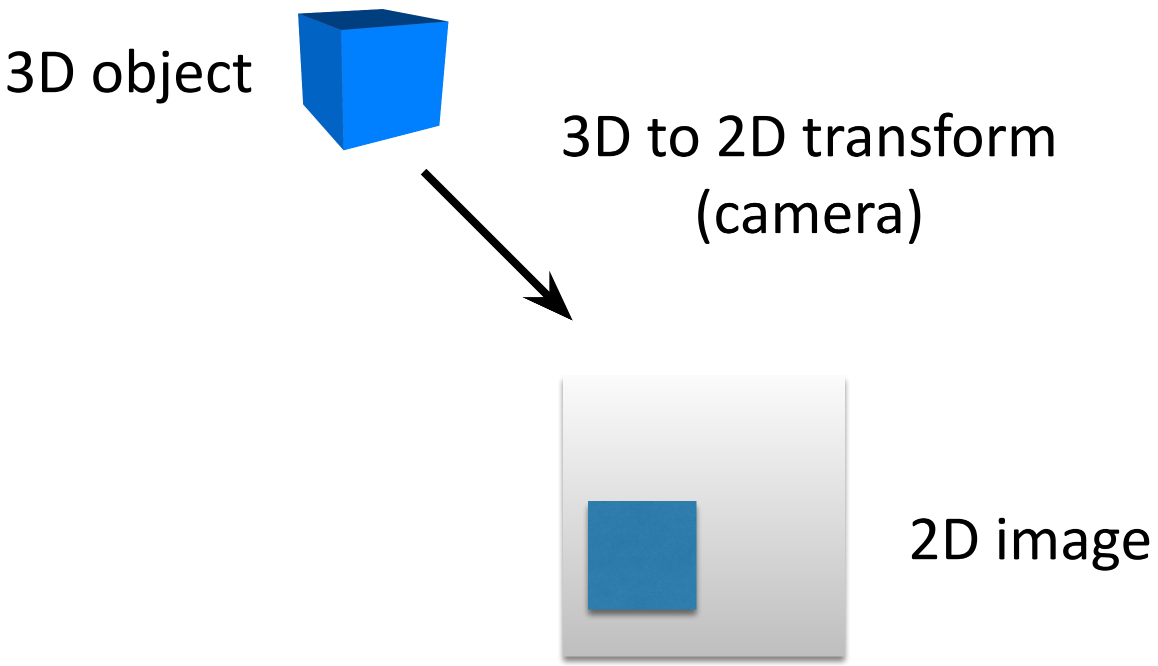
A camera is a mapping

from:

the 3D world

to:

a 2D image



Source: K. Kitani

The camera as a coordinate transformation

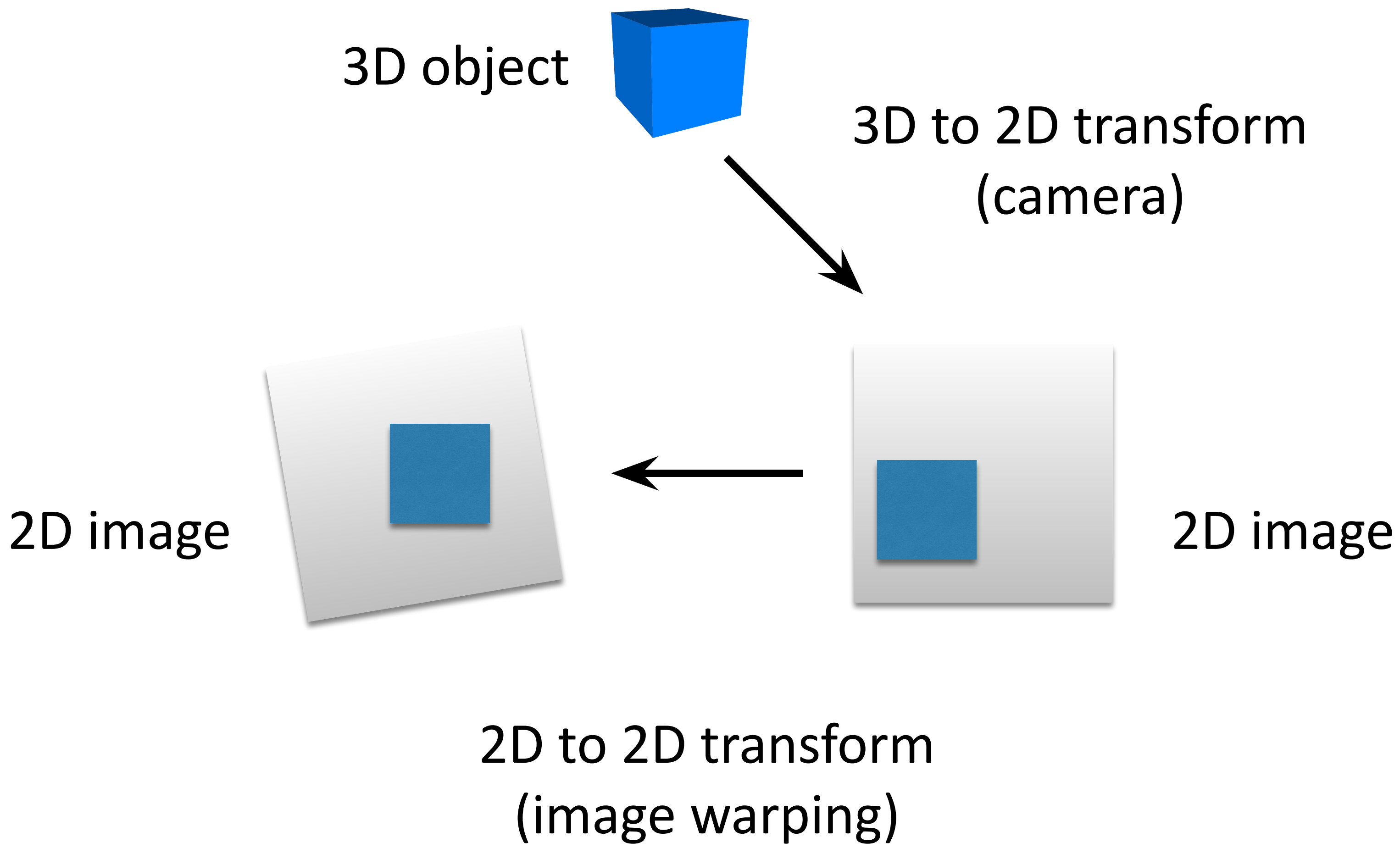
A camera is a mapping

from:

the 3D world

to:

a 2D image



Source: K. Kitani

Cameras and objects can move!

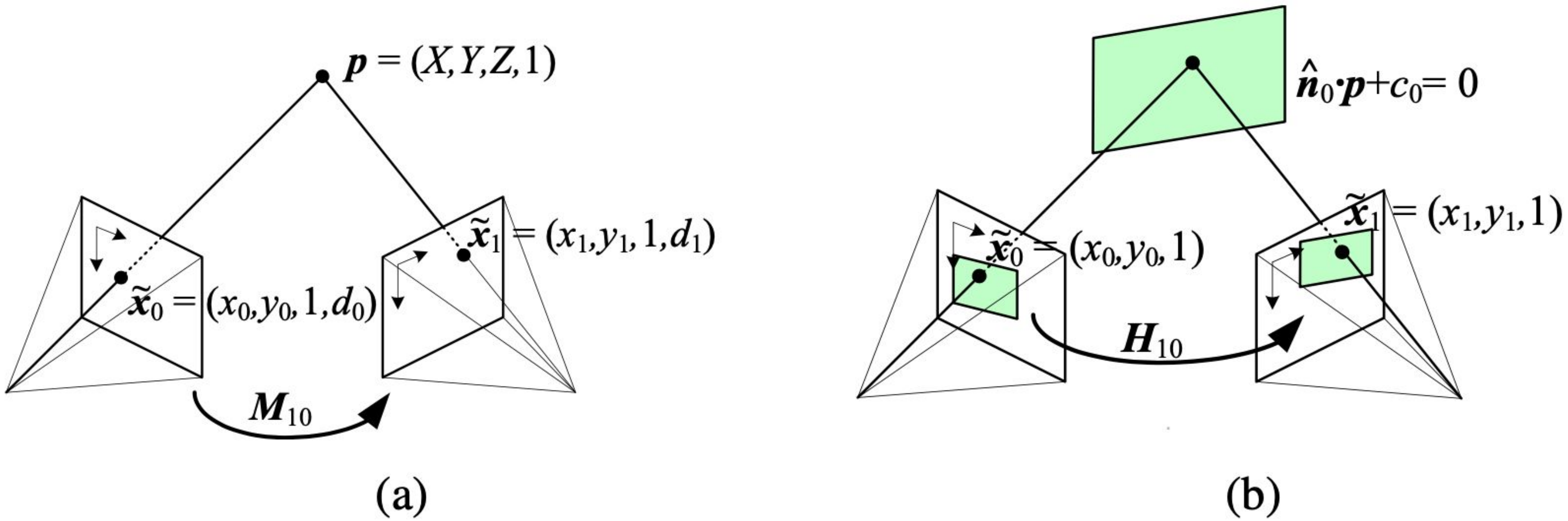


Figure 2.12 *A point is projected into two images: (a) relationship between the 3D point coordinate* $(X, Y, Z, 1)$ *and the 2D projected point* $(x, y, 1, d)$*; (b) planar homography induced by points all lying on a common plane* $\hat{\mathbf{n}}_0 \cdot \mathbf{p} + c_0 = 0$.

Figure: R. Szeliski

2D Transformations Zoo

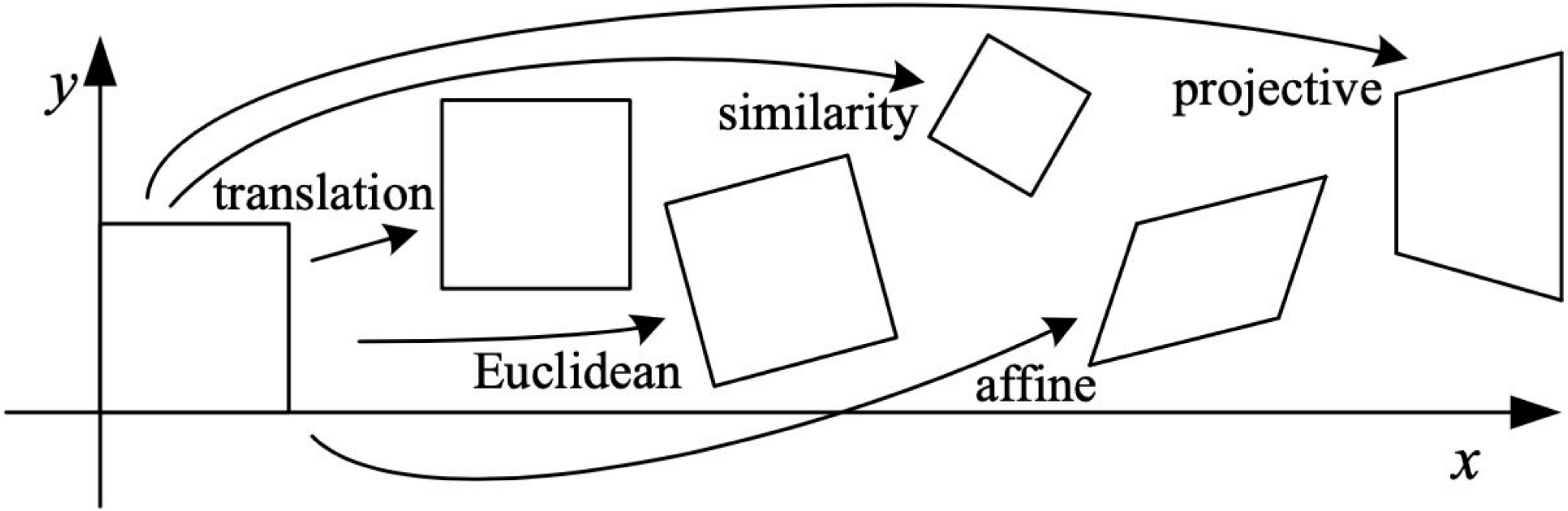


Figure: R. Szeliski

Transformation = Matrix Multiplication

Scale

$$\mathbf{M} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

Flip across y

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Rotate

$$\mathbf{M} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Flip across origin

$$\mathbf{M} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

Shear

$$\mathbf{M} = \begin{bmatrix} 1 & s_x \\ s_y & 1 \end{bmatrix}$$

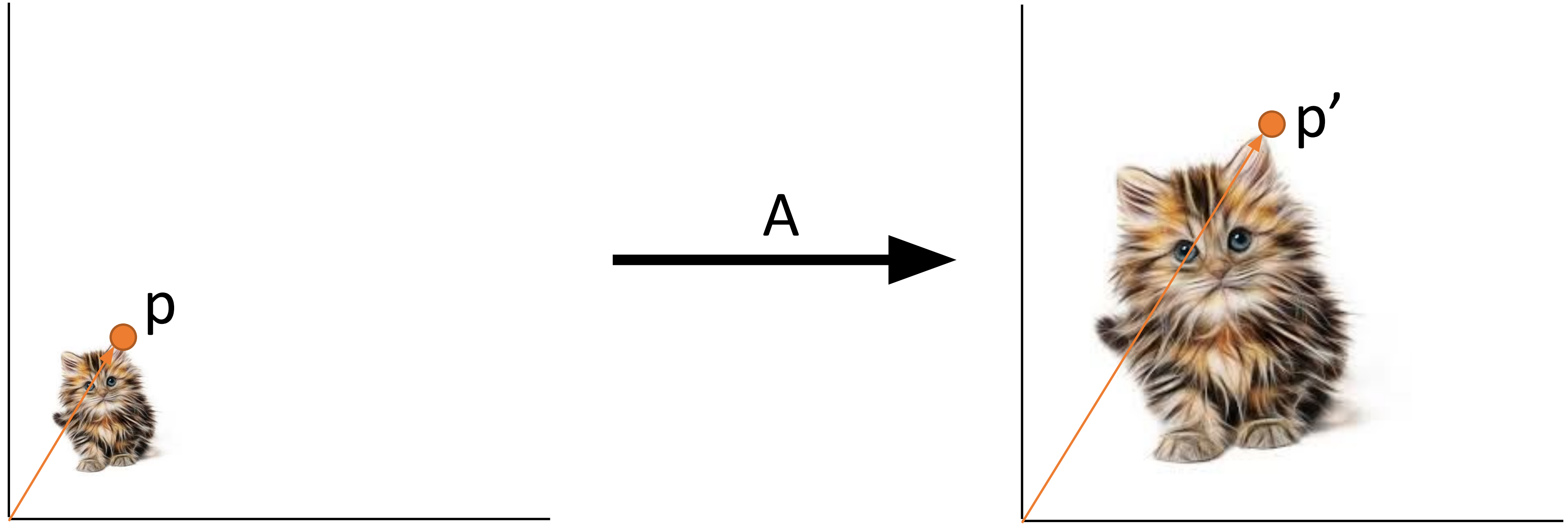
Identity

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Scaling

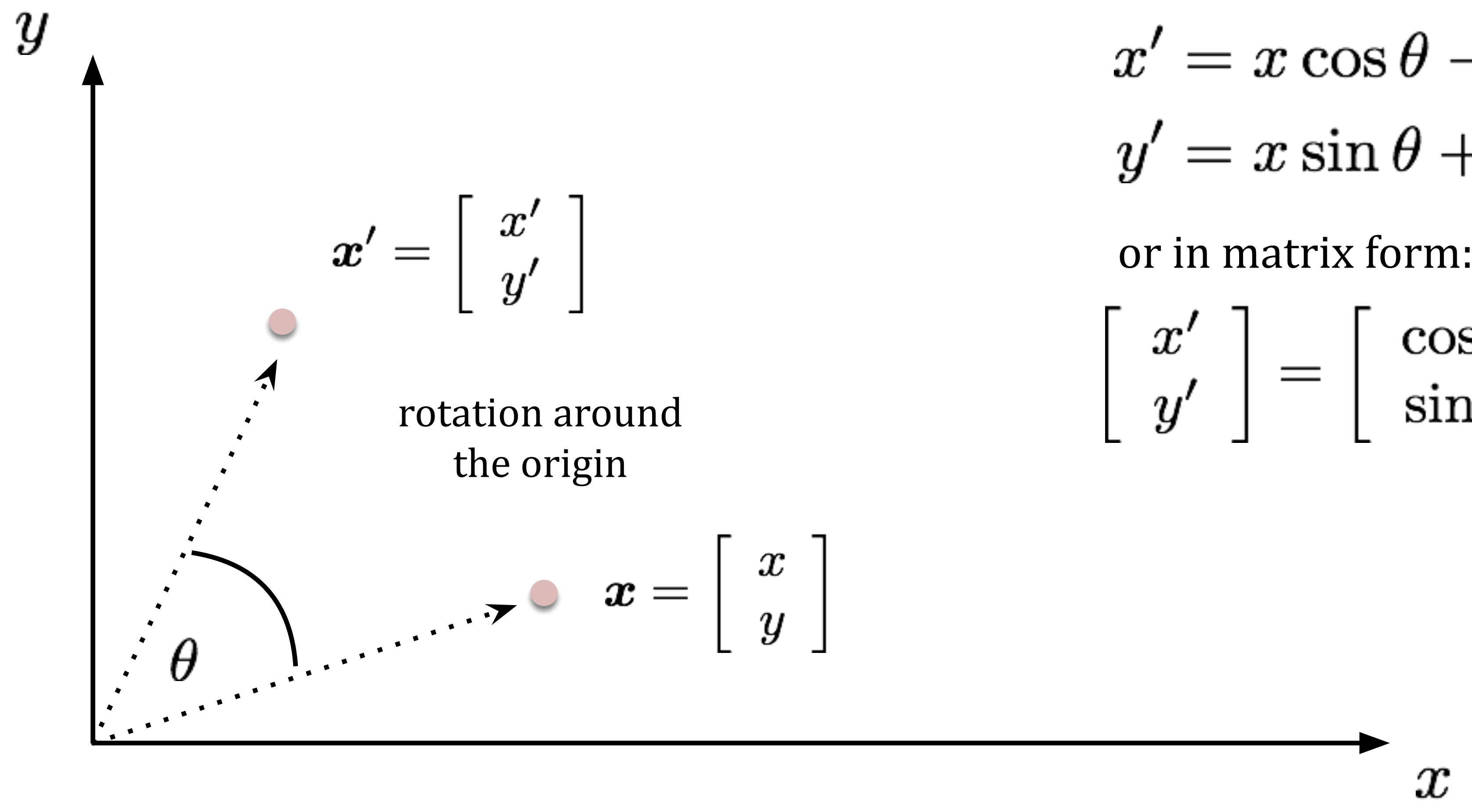
$$\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$$

A p p'



Slide: JC. Niebles

Rotation



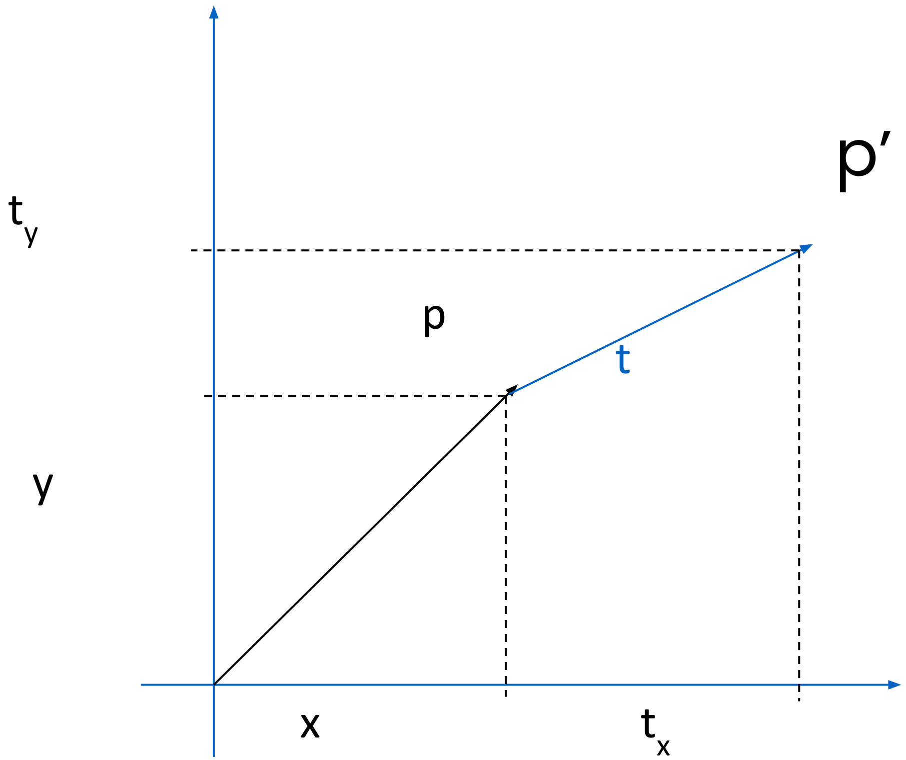
$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

or in matrix form:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Slide: K. Kitani

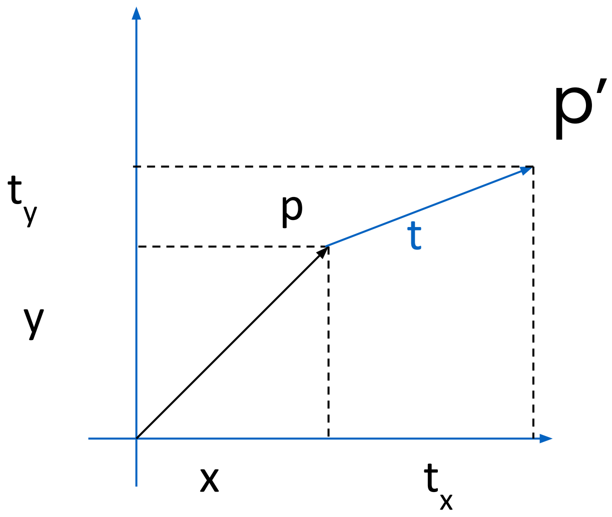
Translation



$$x' = x + t_x$$
$$y' = y + t_y$$

As a matrix?

Translation with homogeneous coordinates



$$p = \begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \rightarrow \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix}$$

$$p' = Tp$$

$$p' \rightarrow \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} p = Tp$$

Slide: JC. Niebles

2D Transformations with homogeneous coordinates

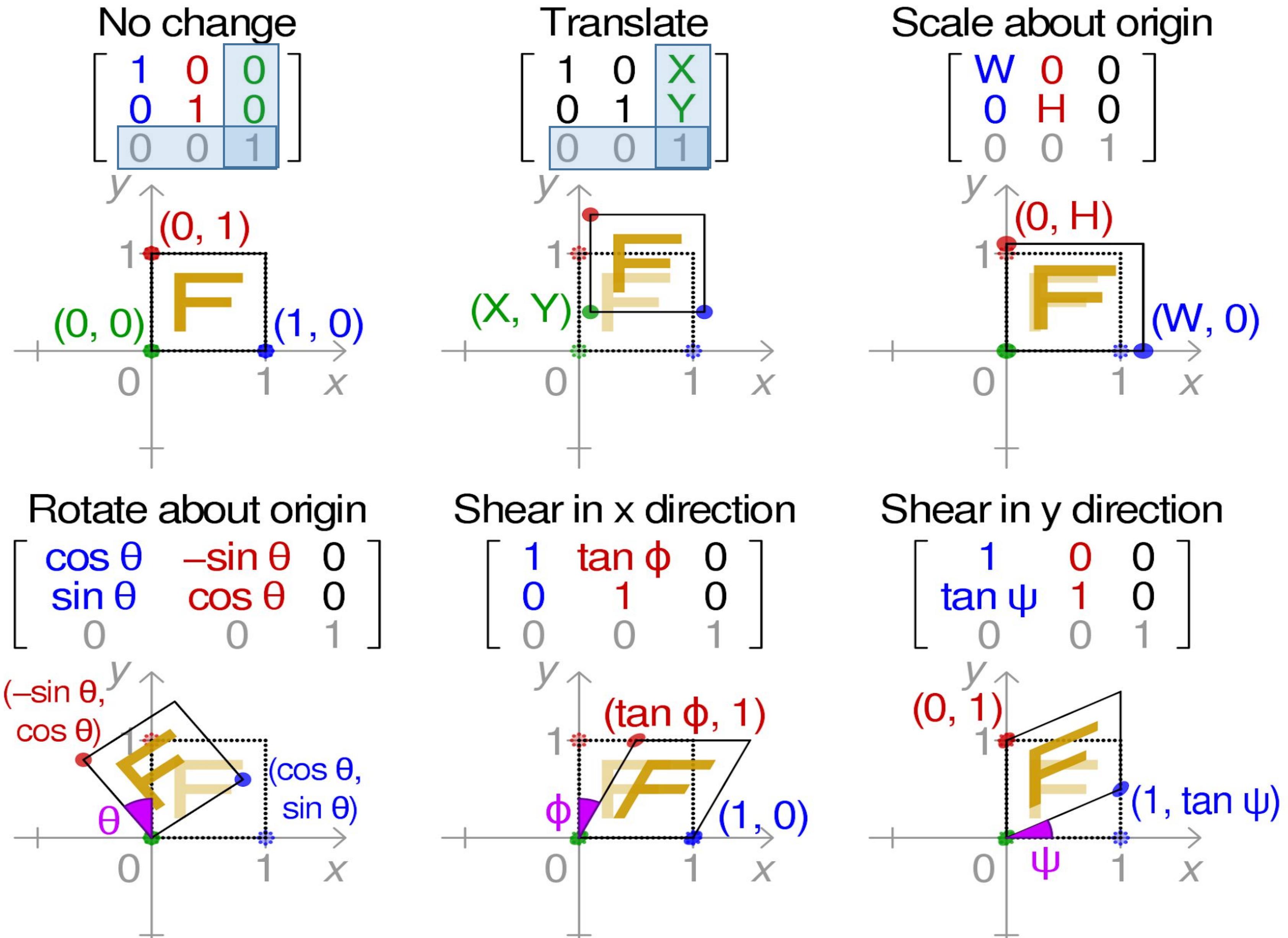


Figure: Wikipedia

Questions?

2D Transformations Zoo

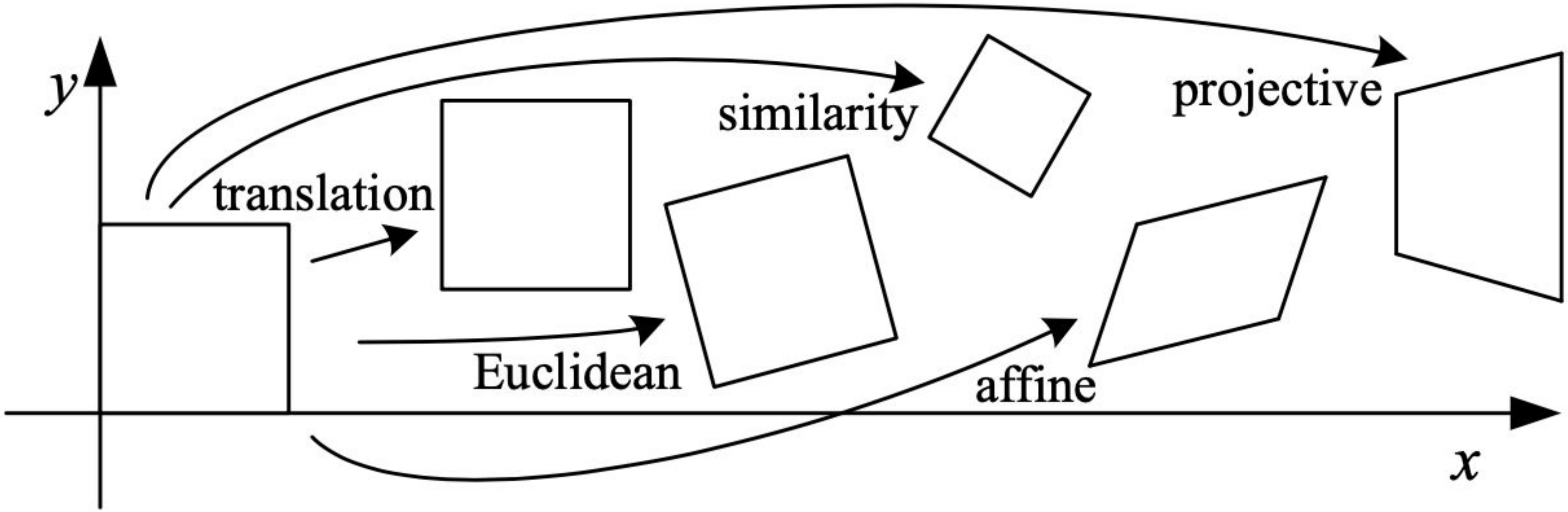


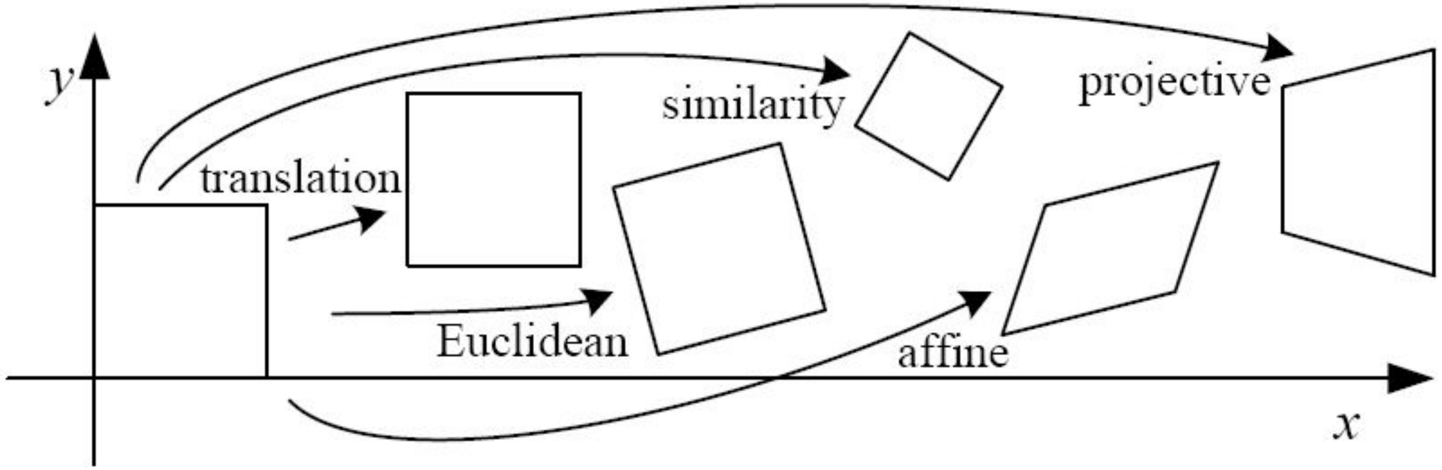
Figure: R. Szeliski

Euclidean / Rigid Transformation

Euclidean (rigid): rotation + translation

$$\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

How many degrees of freedom?

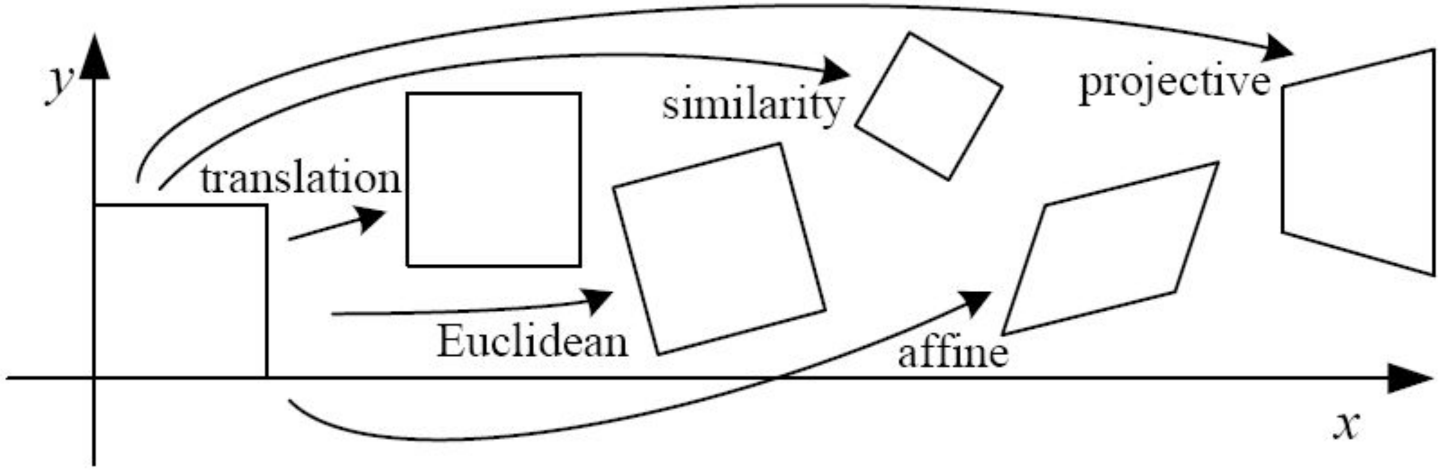


Similarity Transformation

Similarity: Scaling + rotation + translation

$$\begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

How many degrees of freedom?

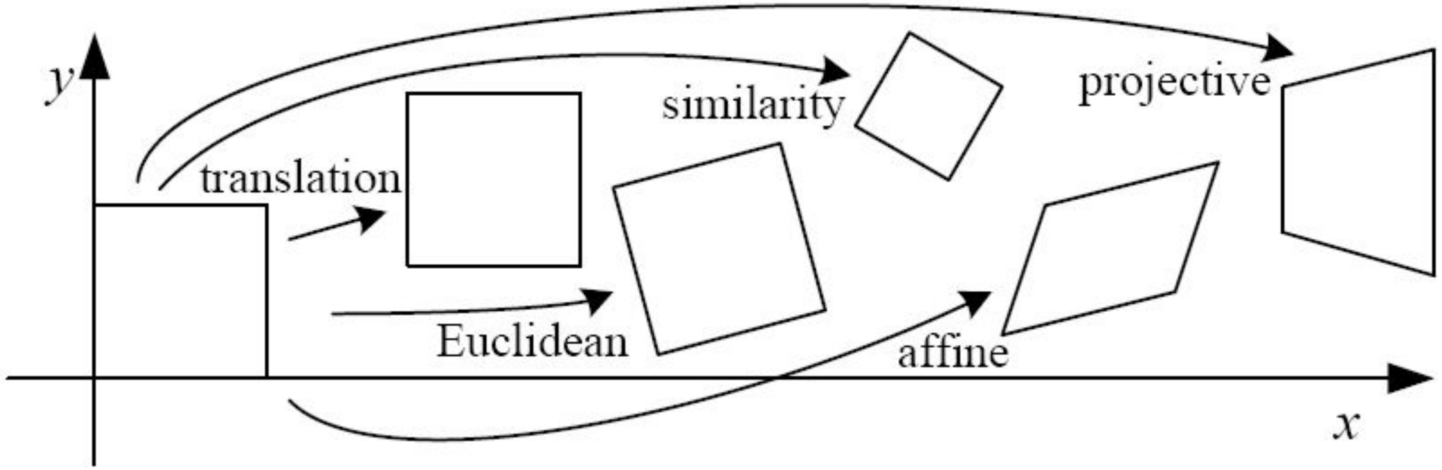


Similarity Transformation

Similarity: Scaling + rotation + translation

$$\begin{bmatrix} a & -b & t_x \\ b & a & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \alpha\cos\theta & -\alpha\sin\theta & b_0 \\ \alpha\sin\theta & \alpha\cos\theta & b_1 \\ 0 & 0 & 1 \end{bmatrix}$$

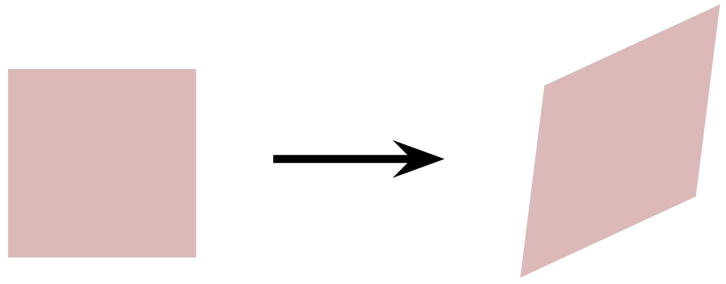
How many degrees of freedom?



Affine Transformation

Affine transformations are combinations of

- Arbitrary (4-DOF) linear transformations + translations



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Cartesian coordinates

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Homogeneous coordinates

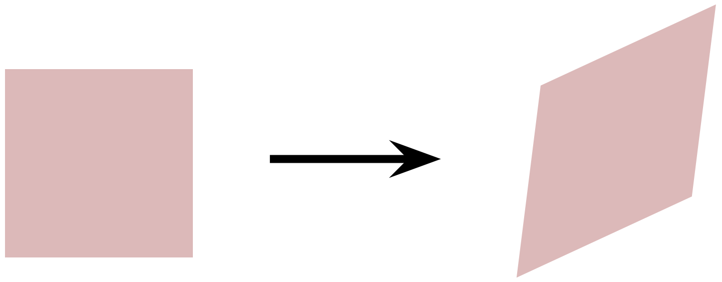
$$\longrightarrow \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Source: K. Kitani

Affine Transformation

Affine transformations are combinations of

- Arbitrary (4-DOF) linear transformations + translations



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} \\ A_{10} & A_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Cartesian coordinates

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

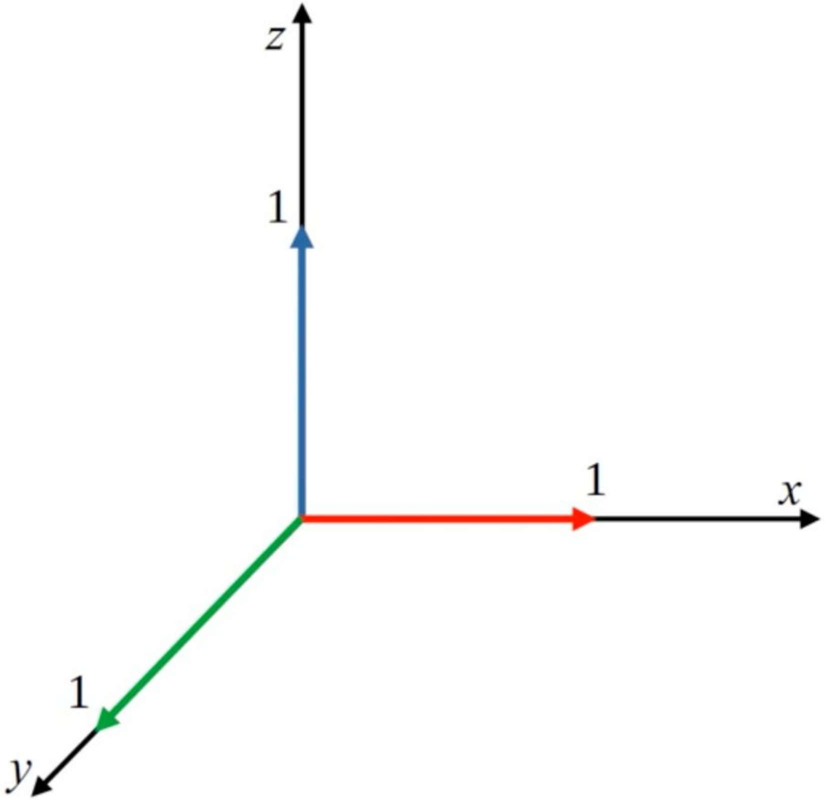
Homogeneous coordinates

$$\longrightarrow \quad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How many degrees of freedom?

Source: K. Kitani

Affine Transformation

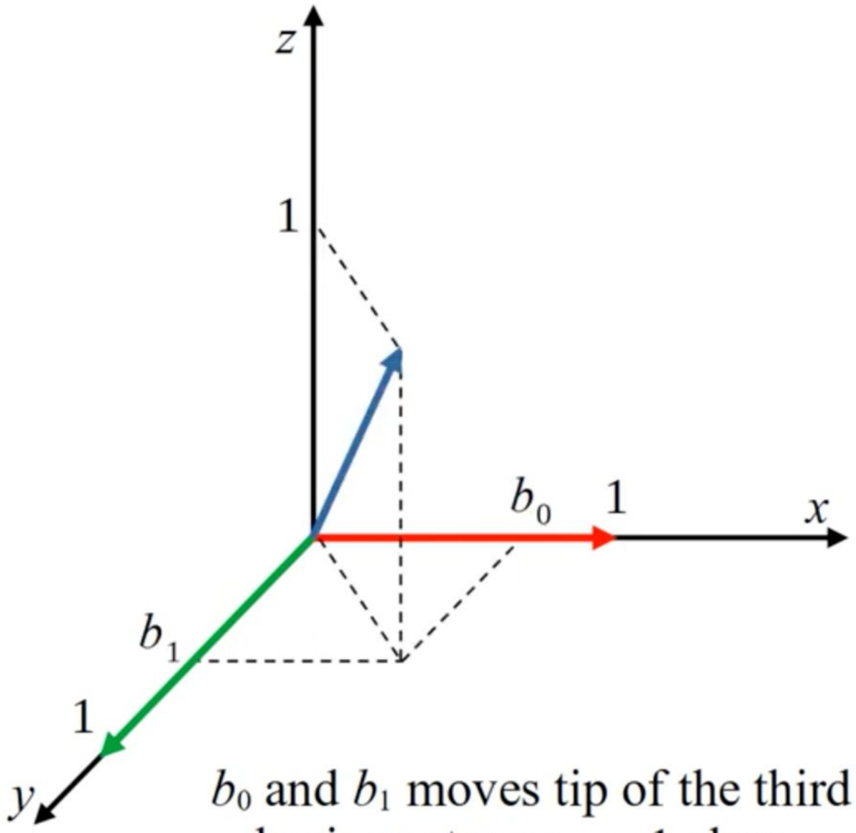


This matrix is a linear transformation matrix in 3D

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation



b_0 and b_1 moves tip of the third basis vector on $z=1$ plane

This matrix is a linear transformation matrix in 3D

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Then this column is the third basis vector of transformed vector space

And what b_0 and b_1 do is to change the orientation of that basis vector

Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation

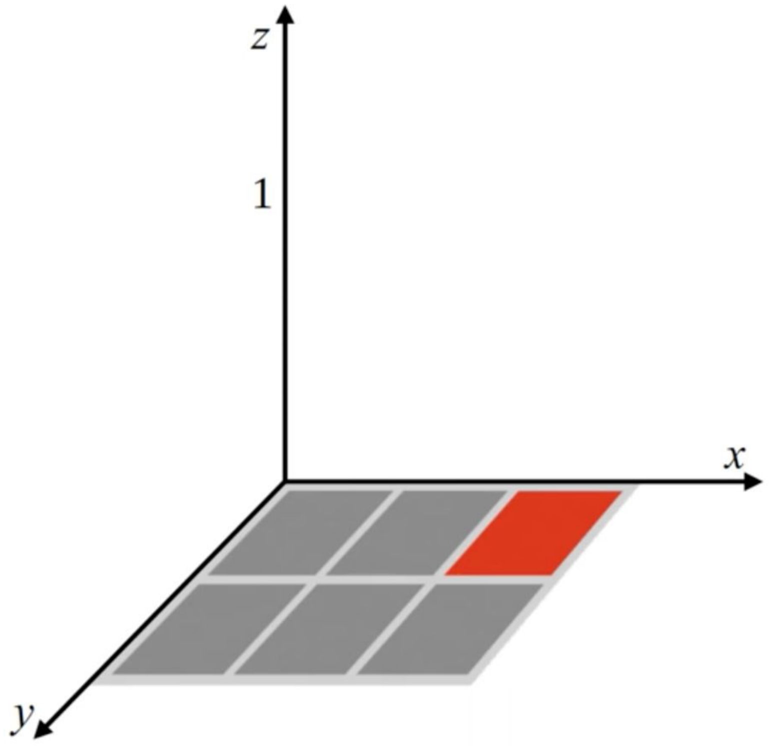


Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation

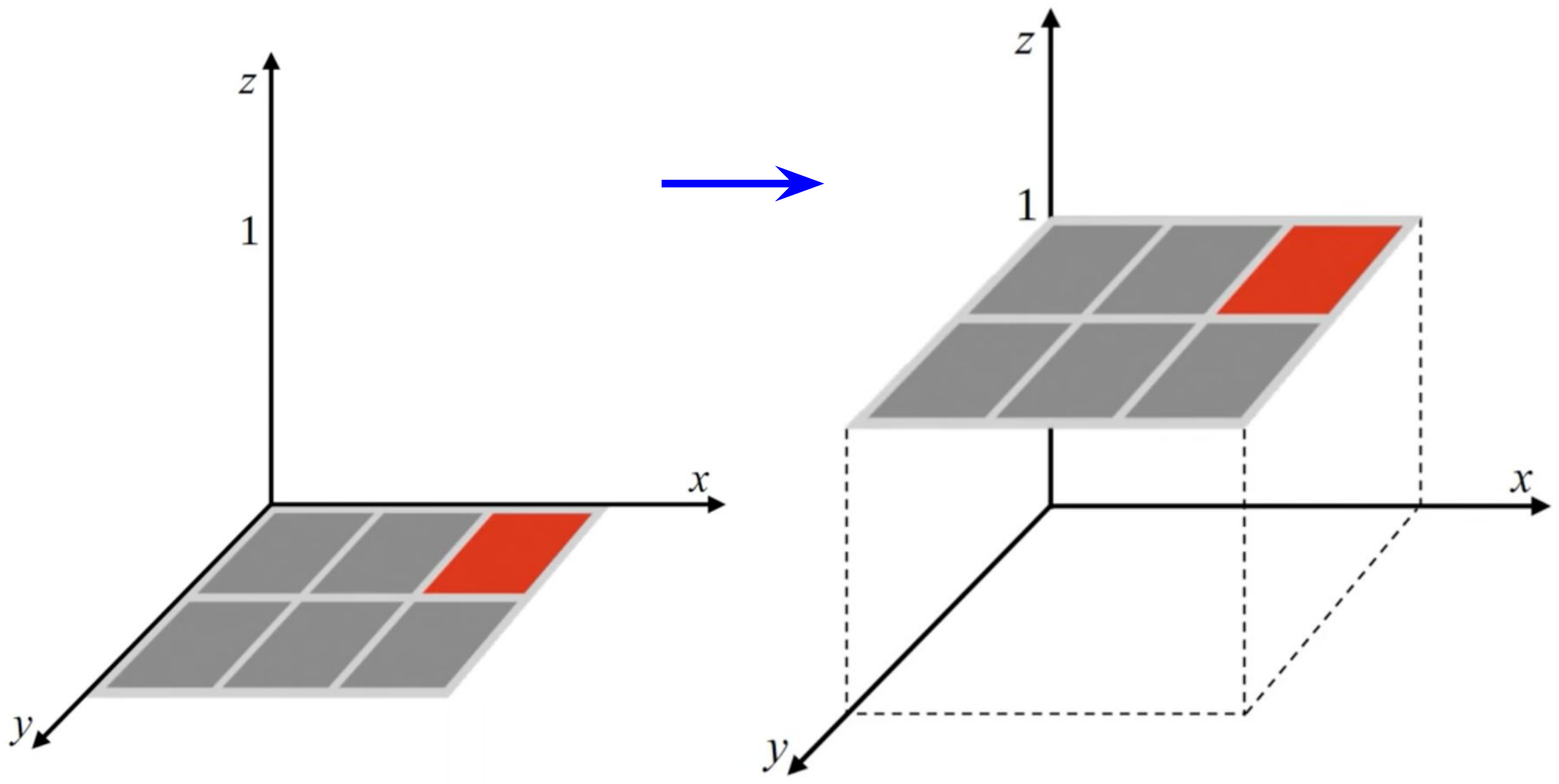


Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation

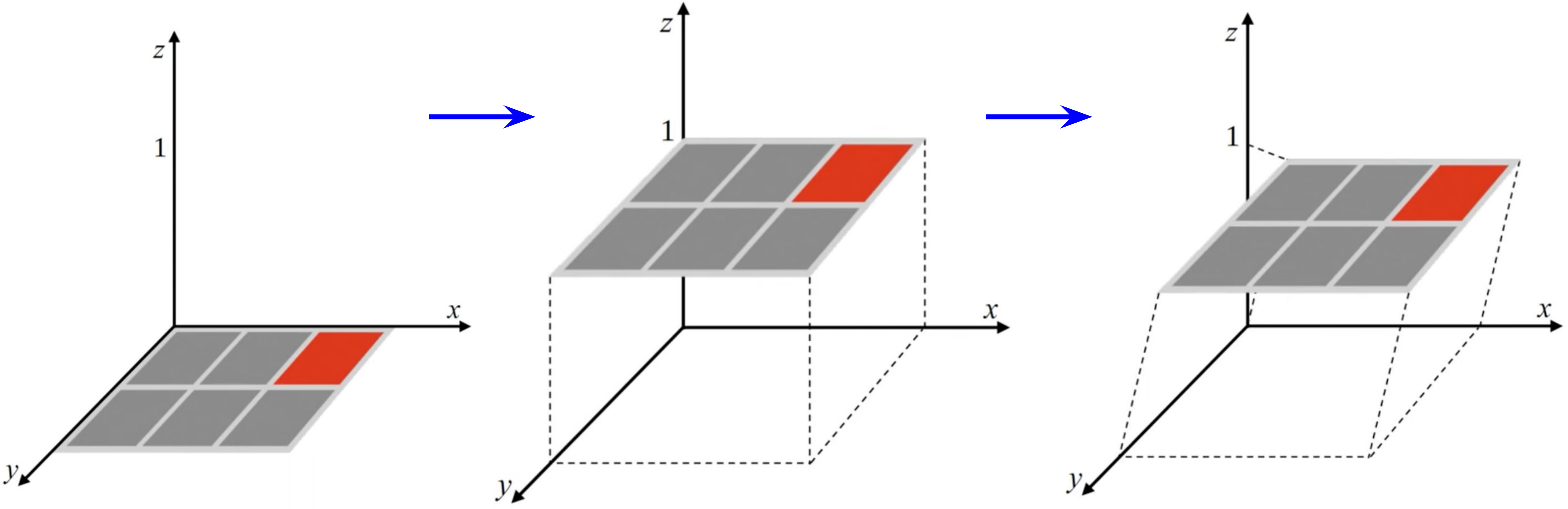


Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation

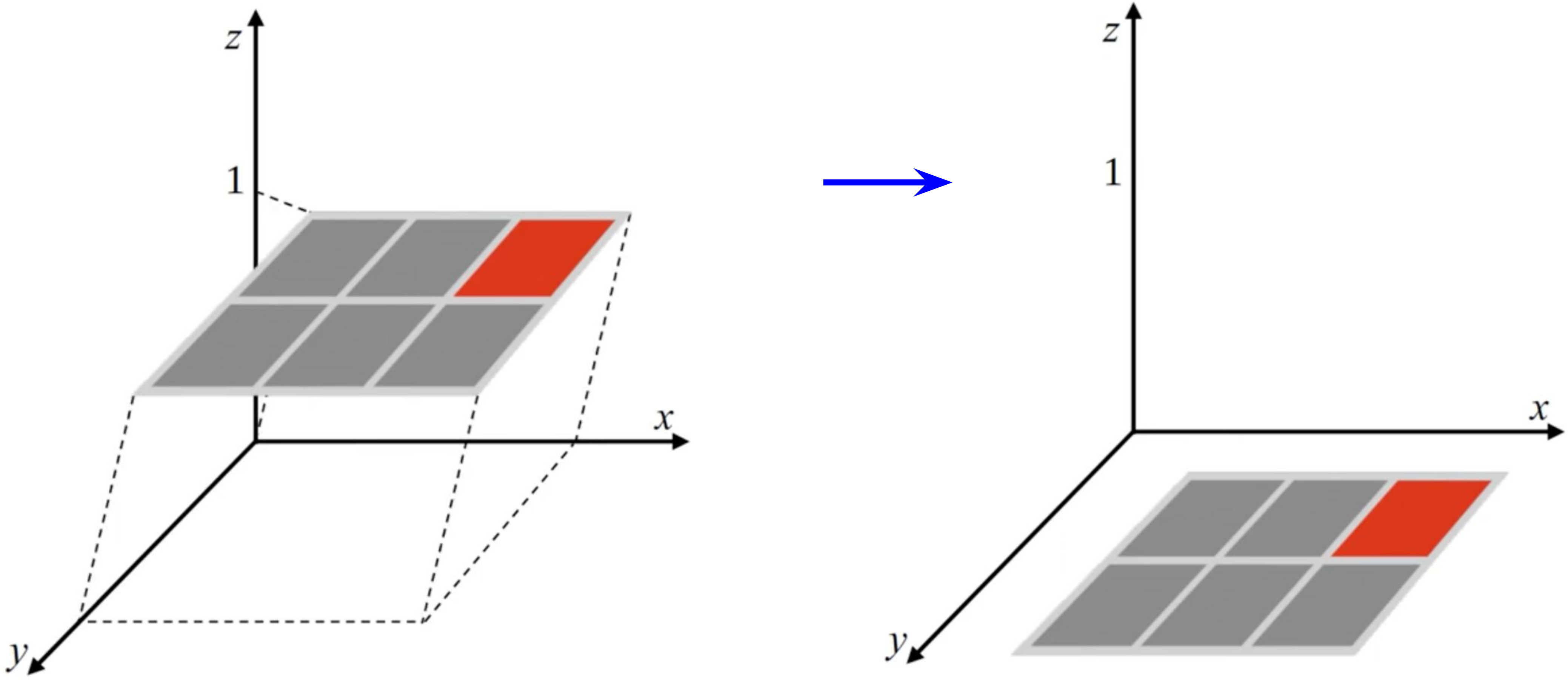
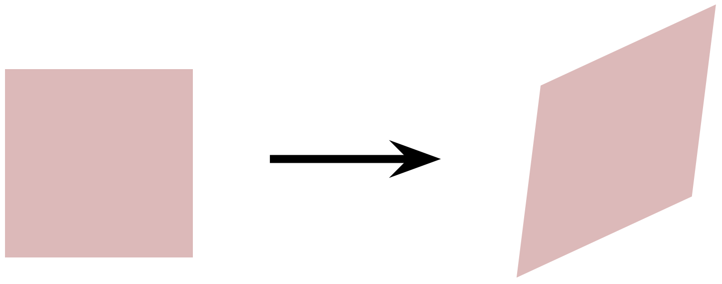


Figure: https://www.youtube.com/@huseyin_ozde...

Affine Transformation

Affine transformations are combinations of

- Arbitrary (4-DOF) linear transformations + translations



Properties of affine transformations:

- origin does not necessarily map to origin
- lines map to lines
- parallel lines map to parallel lines
- ratios are preserved

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Source: K. Kitani

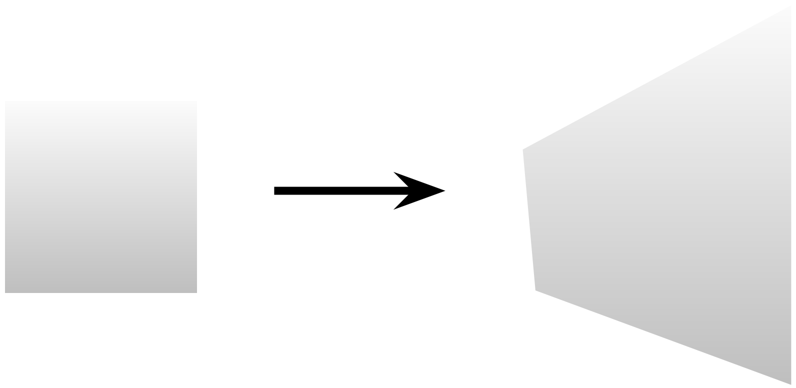
Projective Transformation (homography)

Projective transformations are combinations of

- Affine transformations + projective warps

$$w\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ h_0 & h_1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How many degrees of freedom?



Source: K. Kitani

Projective Transformation (homography)

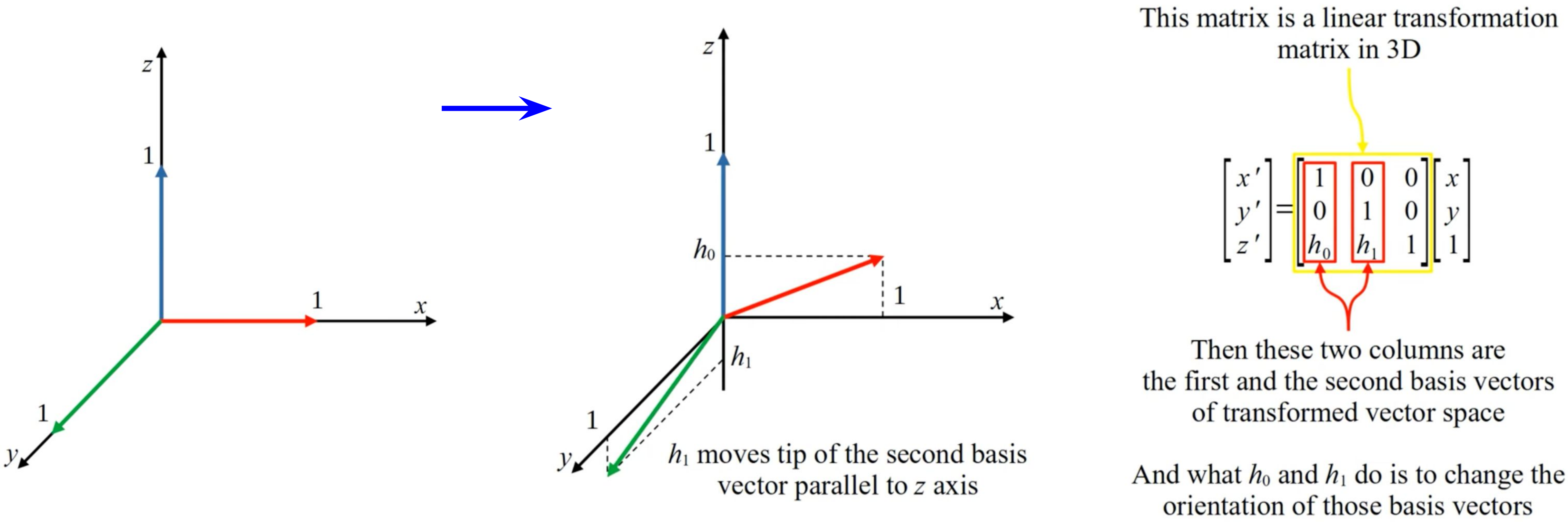
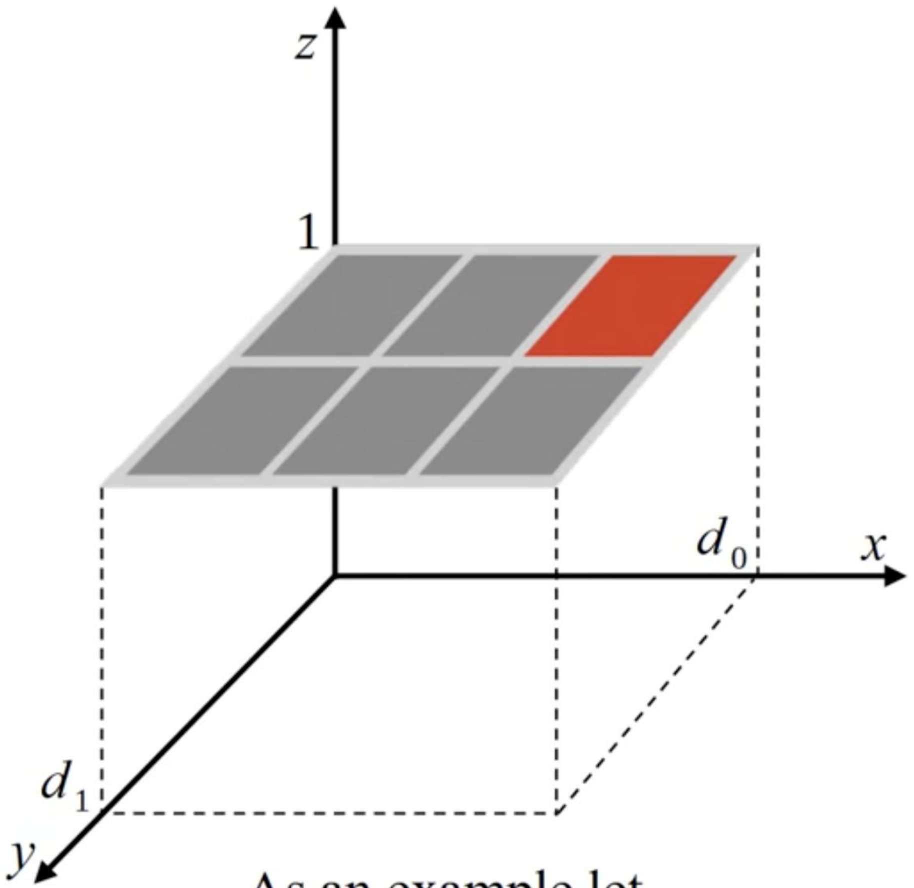


Figure: https://www.youtube.com/@huseyin_ozde...

Projective Transformation (homography)



As an example let
$h_0>0$, $h_1<0$ and $|h_0|>|h_1|$

Figure: https://www.youtube.com/@huseyin_ozde...

Projective Transformation (homography)

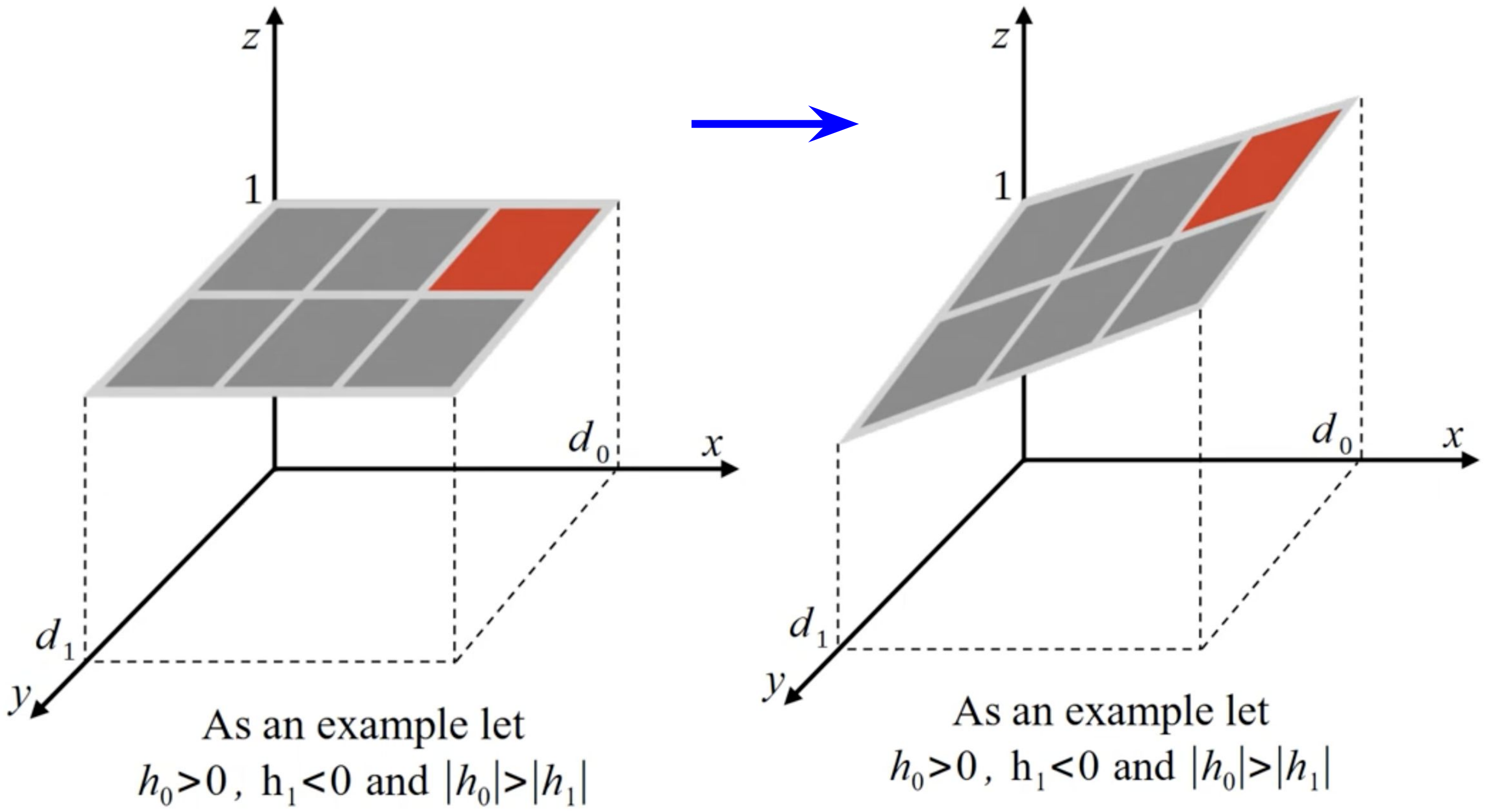


Figure: https://www.youtube.com/@huseyin_ozde...

Projective Transformation (homography)

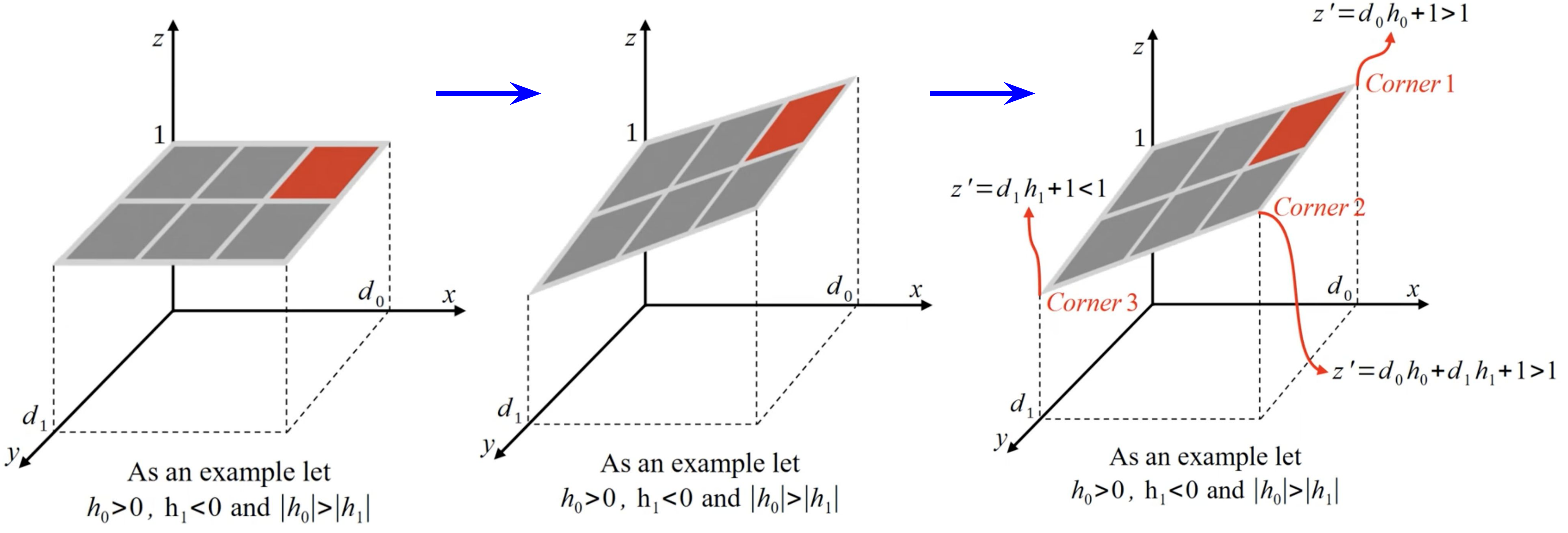


Figure: https://www.youtube.com/@huseyin_ozde...

Projective Transformation (homography)

When going back to Cartesian coordinates

$$\frac{x'}{z'}$$

$$\frac{y'}{z'}$$

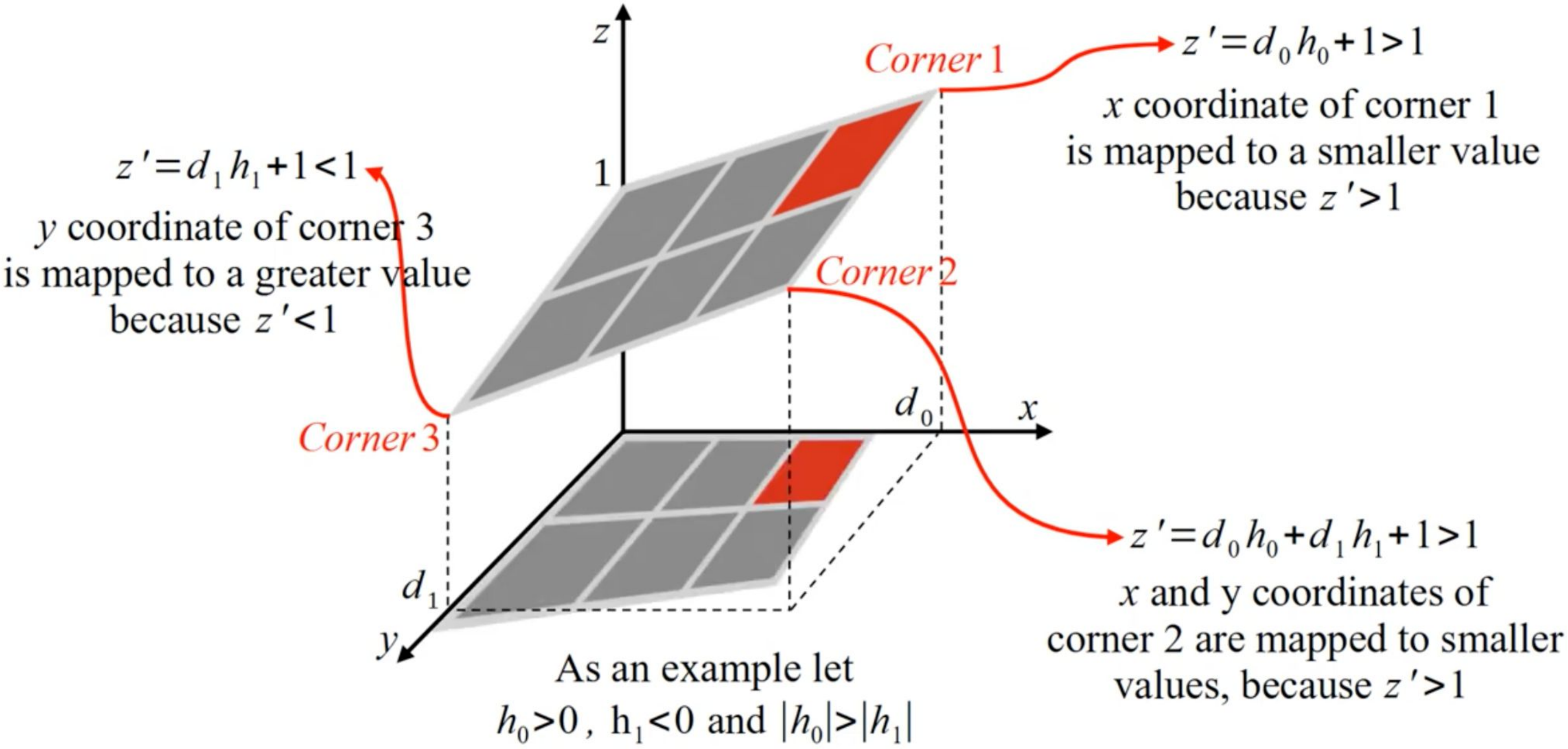
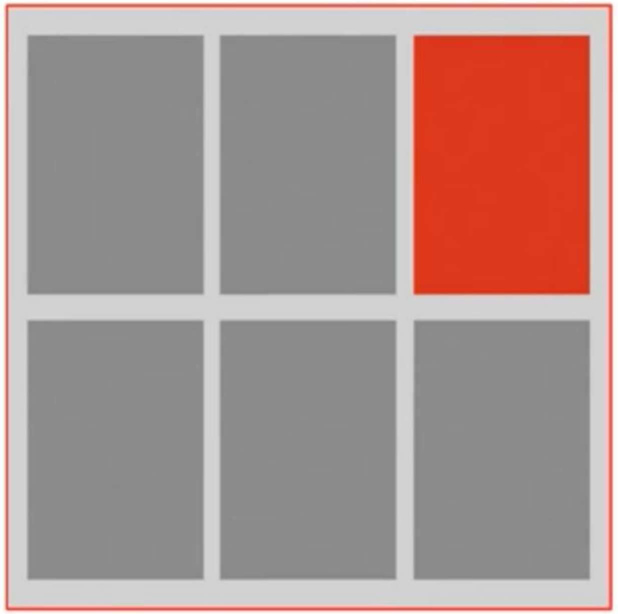


Figure: https://www.youtube.com/@huseyin_ozde...

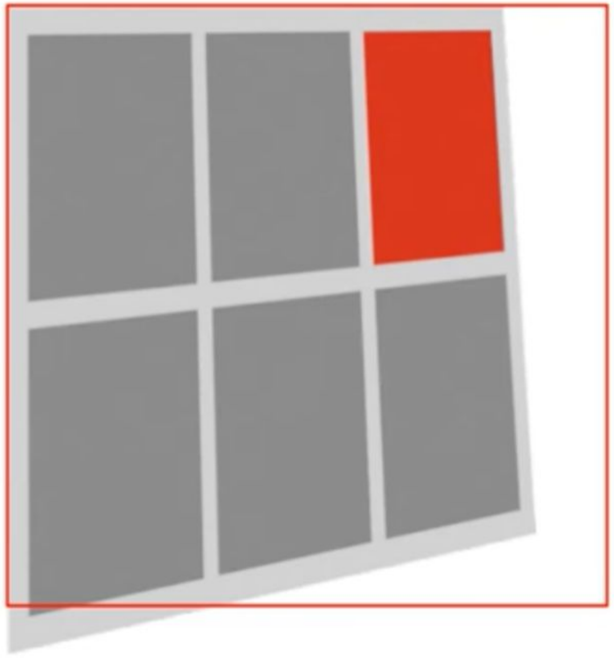
Projective Transformation (homography)



Original Image

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ h_0 & h_1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$h_0 > 0$, $h_1 < 0$ and $|h_0| > |h_1|$



Warped Image

Figure: https://www.youtube.com/@huseyin_ozde...

Projective Transformation (homography)

Projective transformations are combinations of

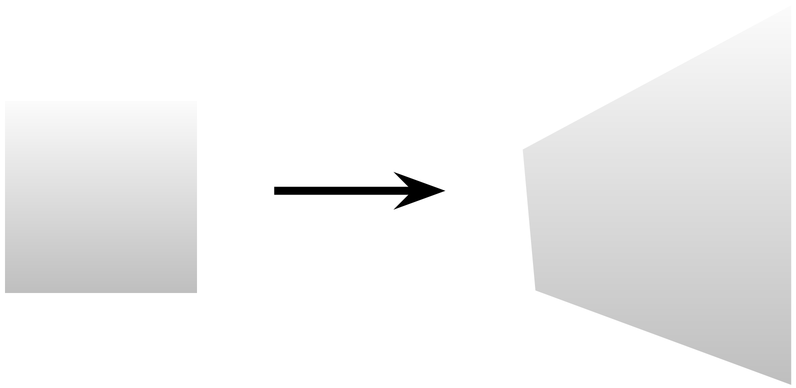
- Affine transformations + projective warps

$$w\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} A_{00} & A_{01} & b_0 \\ A_{10} & A_{11} & b_1 \\ h_0 & h_1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

How many degrees of freedom?

Properties of projective transformations:

- origin does not necessarily map to origin
- lines map to lines
- parallel lines do not necessarily map to parallel lines
- ratios are not necessarily preserved



Source: K. Kitani

Questions?

Composing Transformations

Transformations = Matrices => Composition by Multiplication!

$$p' = R_2 R_1 S p$$

In the example above, the result is equivalent to

$$p' = R_2(R_1(Sp))$$

Equivalent to multiply the matrices into single transformation matrix:

$$p' = (R_2 R_1 S)p$$

Order Matters! Transformations from ***right to left***.

Scaling & Translating != Translating & Scaling

$$p'' = TSp = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + t_x \\ s_y y + t_y \\ 1 \end{bmatrix}$$

$$p''' = STp = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & s_x t_x \\ 0 & s_y & s_y t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x x + s_x t_x \\ s_y y + s_y t_y \\ 1 \end{bmatrix}$$

Similarity: Translation + Rotation + Scaling

p'= (T R S) p

$$p' = TRSp = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} RS & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

This is the form of the general-purpose transformation matrix

2D Transforms = Matrix Multiplication

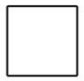
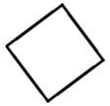
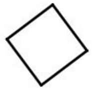
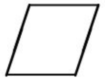
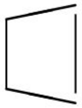
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2\times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times 3}$	3	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times 3}$	4	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2\times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3\times 3}$	8	straight lines	

Table 2.1 *Hierarchy of 2D coordinate transformations, listing the transformation name, its matrix form, the number of degrees of freedom, what geometric properties it preserves, and a mnemonic icon. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 2×3 matrices are extended with a third $[\mathbf{0}^T \; 1]$ row to form a full 3×3 matrix for homogeneous coordinate transformations.*

Figure: R. Szeliski

3D Transforms = Matrix Multiplication

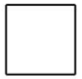
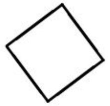
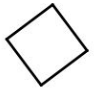
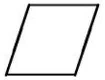
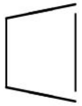
Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{3\times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4}$	6	lengths	
similarity	$\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{3\times 4}$	7	angles	
affine	$\begin{bmatrix} \mathbf{A} \end{bmatrix}_{3\times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{4\times 4}$	15	straight lines	

Table 2.2 *Hierarchy of 3D coordinate transformations. Each transformation also preserves the properties listed in the rows below it, i.e., similarity preserves not only angles but also parallelism and straight lines. The 3 × 4 matrices are extended with a fourth* $[\mathbf{0}^T \; 1]$ *row to form a full 4 × 4 matrix for homogeneous coordinate transformations. The mnemonic icons are drawn in 2D but are meant to suggest transformations occurring in a full 3D cube.*

Figure: R. Szeliski

What did we learn today?

- Geometry is essential to Computer Vision!

What did we learn today?

- Geometry is essential to Computer Vision!
- Geometric Primitives in 2D & 3D
    - Homogeneous coordinates, points, lines, and planes in 2D & 3D

What did we learn today?

- Geometry is essential to Computer Vision!
- Geometric Primitives in 2D & 3D
  - Homogeneous coordinates, points, lines, and planes in 2D & 3D
- 2D & 3D Transformations
  - scaling, translation, rotation, rigid, similarity, affine, homography

Questions?

Appendix

3D Rotations: SO(3) representations

Euler Angles: yaw, pitch, roll (α, β, γ)
→ compose $R(\gamma)R(\beta)R(\alpha)$ (order, axes!)

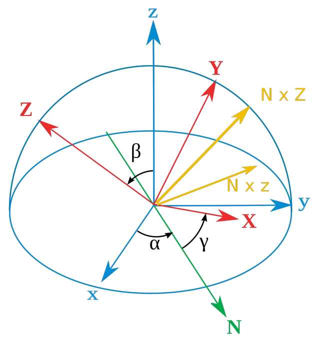
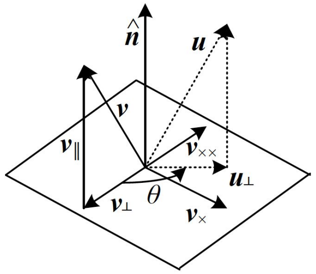


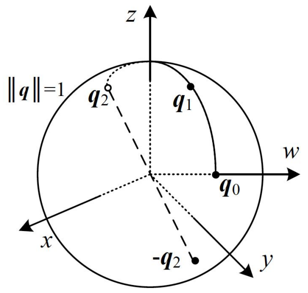
Figure: Wikipedia

Axis-angle: $(\hat{n}, \theta)$ or $\omega = \theta\hat{n}$
→ matrix via Rodrigues formula (simple for small θ)

$$\mathbf{R}(\hat{\mathbf{n}}, \theta) = \mathbf{I} + \sin\theta[\hat{\mathbf{n}}]_\times + (1 - \cos\theta)[\hat{\mathbf{n}}]_\times^2 \approx \mathbf{I} + [\theta\hat{\mathbf{n}}]_\times$$



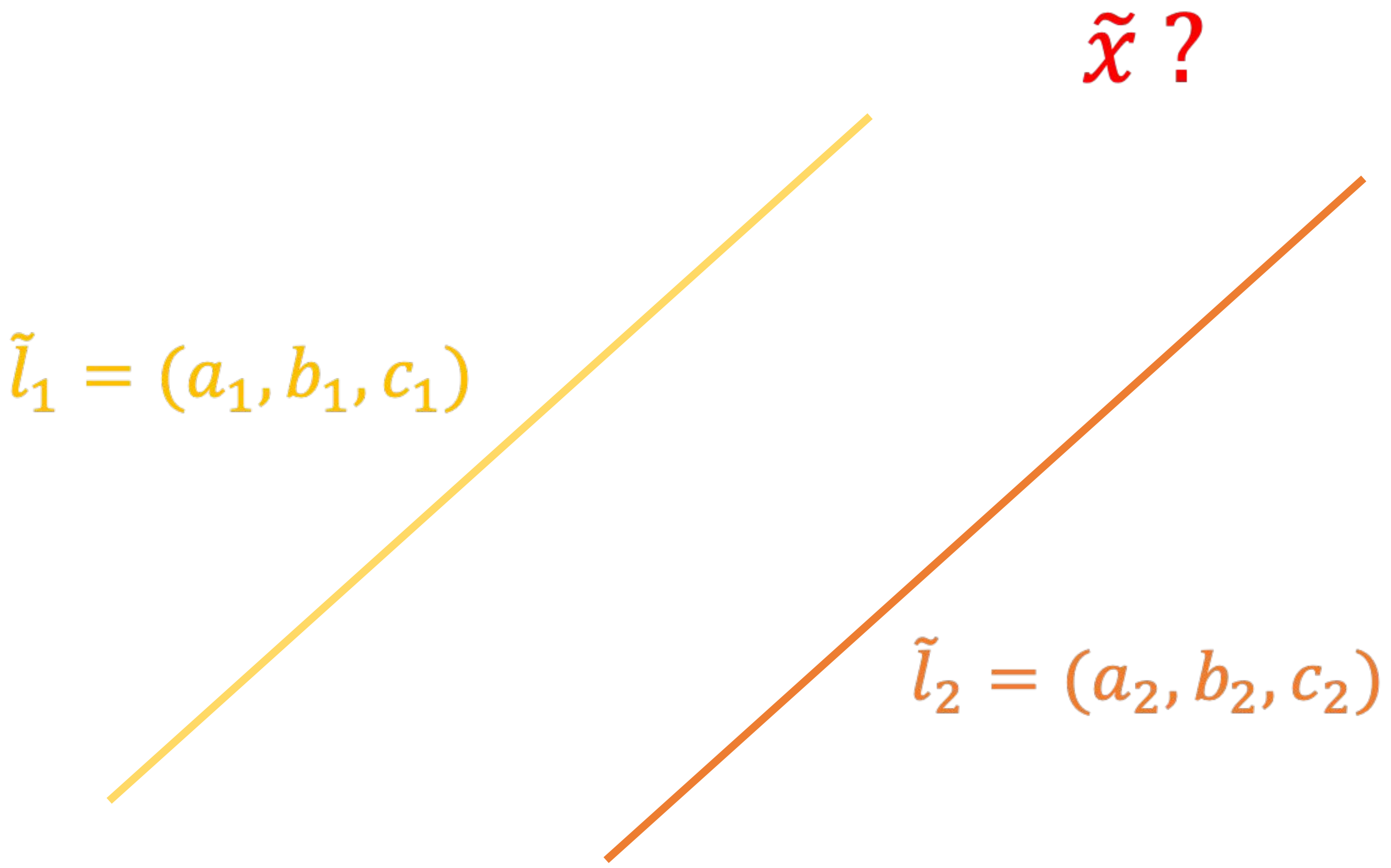
Unit Quaternions: $q = (x, y, z, w) = (\sin\frac{\theta}{2}\,\hat{n}, \cos\frac{\theta}{2})$, $\|q\| = 1$
→ continuous, nice algebraic properties, matrix via Rodrigues



$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - zw) & 2(xz + yw) \\ 2(xy + zw) & 1 - 2(x^2 + z^2) & 2(yz - xw) \\ 2(xz - yw) & 2(yz + xw) & 1 - 2(x^2 + y^2) \end{bmatrix}$$

See Szeliski 2.1.3 for more details

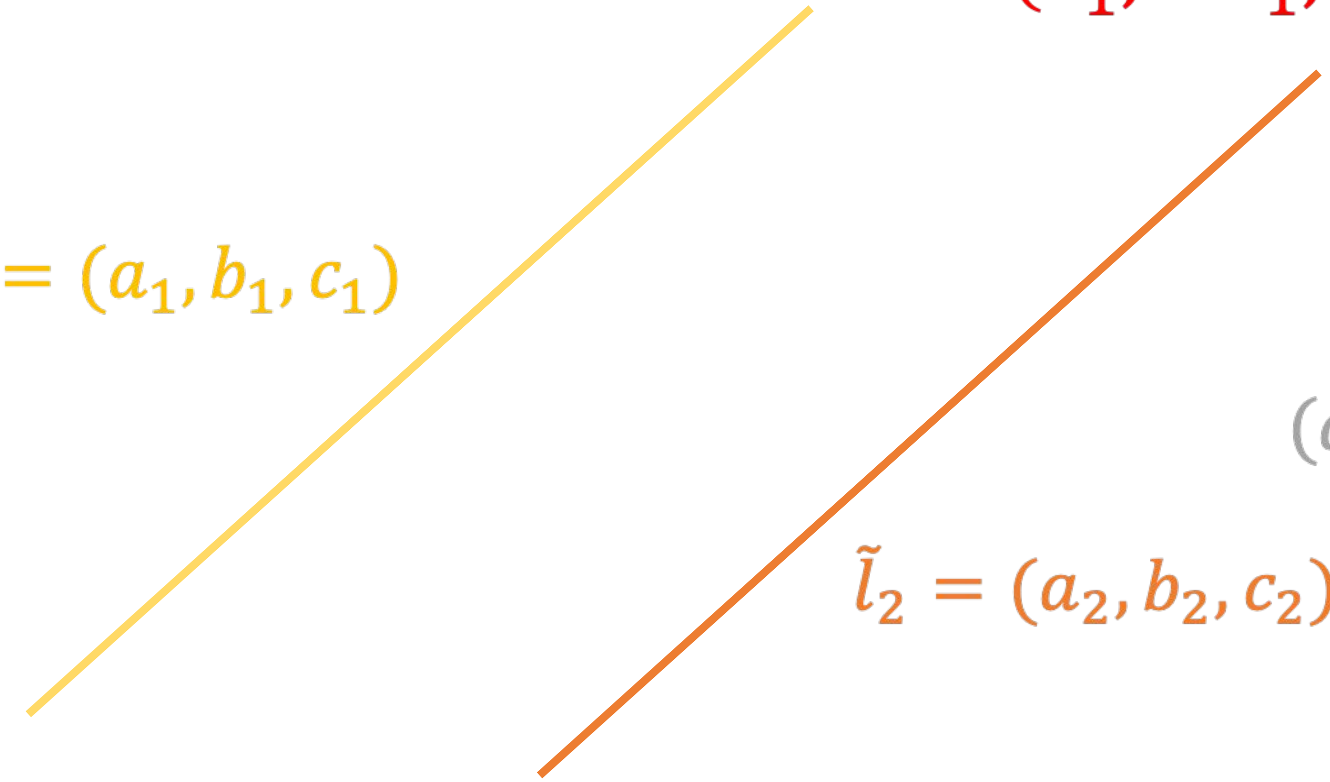
Intersecting Parallel Lines



Intersecting Parallel Lines

$$\tilde{x} = \tilde{l}_1 \times \tilde{l}_2$$

$$\tilde{x} \sim (b_1, -a_1, 0)$$



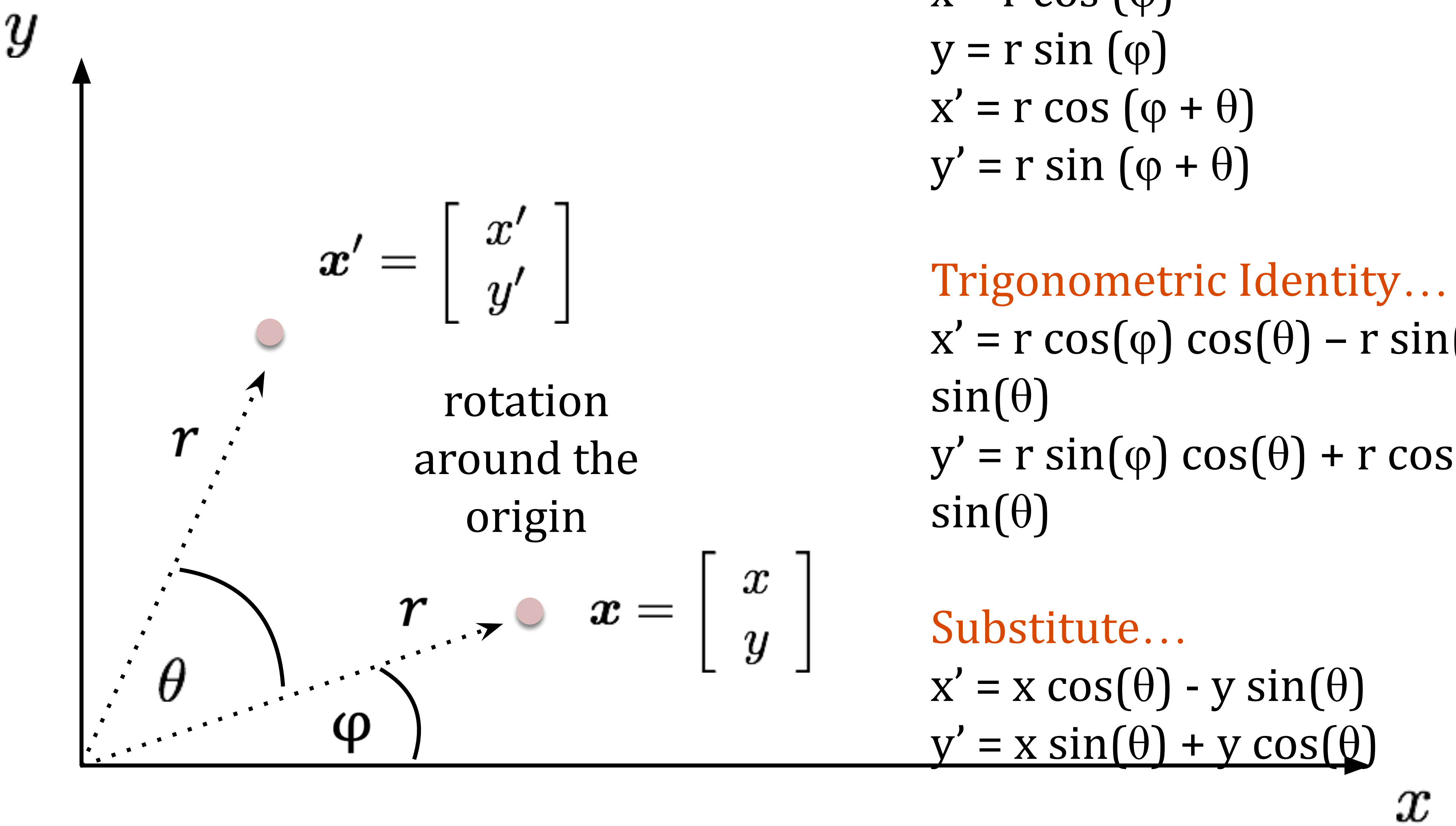
$$\tilde{l}_1 = (a_1, b_1, c_1)$$

$$-\frac{a_1}{b_1} = -\frac{a_2}{b_2}$$

$$(a_2, b_2) = w(a_1, b_1)$$

$$\tilde{l}_2 = (a_2, b_2, c_2)$$

2D planar transformations



Polar coordinates…

x = r cos (φ)
y = r sin (φ)
x' = r cos (φ + θ)
y' = r sin (φ + θ)

Trigonometric Identity…

x' = r cos(φ) cos(θ) – r sin(φ) sin(θ)
y' = r sin(φ) cos(θ) + r cos(φ) sin(θ)

Substitute…

x' = x cos(θ) - y sin(θ)
y' = x sin(θ) + y cos(θ)