

Computer Vision

DATA.ML.300, 5 study credits

Esa Rahtu
Computing Sciences Unit, Tampere University

Plan for today

- Background
 - What is computer vision?
 - Why to study computer vision
- Overview of the course
- Lecture 1: Image formation

A few words about me

Esa Rahtu

Assistant Prof. Computer vision

- PhD, University of Oulu 2007
- Professor at TAU since 2017
- Working with computer vision since 2003
- Visiting scholar at University of Oxford (2011-2012)
- Co-founder and CTO at IndoorAtlas Ltd 2012 ->
- Recent projects etc. <http://esa.rahtu.fi>



Motivation - what is computer vision?

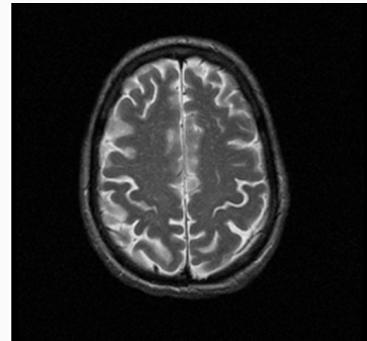
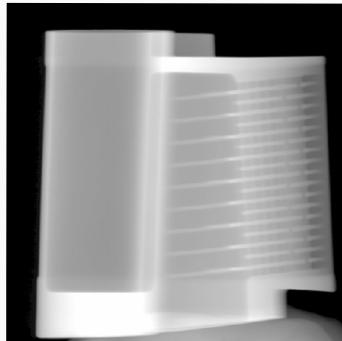
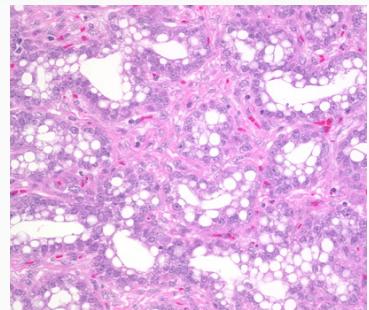
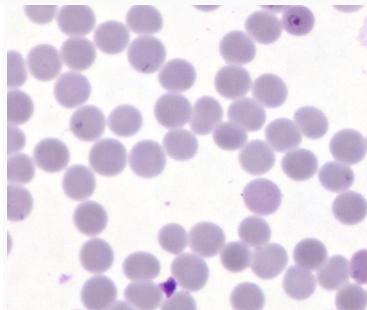
Make computers understand images

- What kind of scene?
- Where are the cars?
- How far are the buildings?
- Where are the cars going?
-



Many data modalities

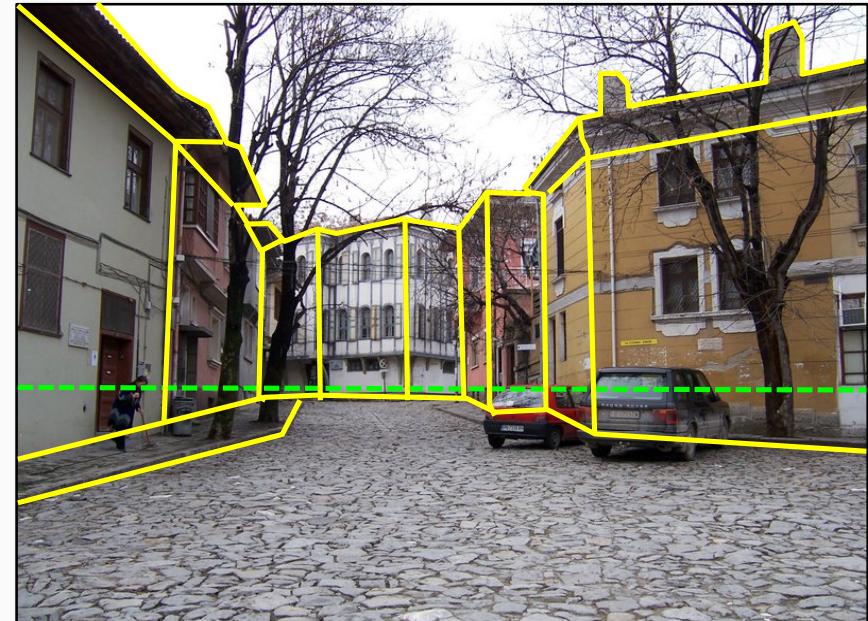
- 2D or 3D still images
- Video frames
- X-ray
- Ultra-sound
- Microscope
-



What kind of information can be extracted?



Semantic information



Geometric information

What do we have here?



... seems pretty easy...

Wrong! Very hard big data problem...

- Hardware perspective:
 - RGB stereo images with 30 frames per second -> 100s MB/s data stream.
 - Non-trivial processing per each byte.
 - Massive image collections.
- Mathematical perspective
 - Information is highly implicit or lost by reprojection
 - 2D -> 3D mapping is ill-posed and ill-conditioned -> need to use constraints

Wrong! Very hard big data problem...

- Artificial intelligence perspective
 - Images have uneven information content
 - Computational visual semantics is hard (what does visual stuff mean exactly?)
 - If we have limited time, what is the important visual stuff right now?

Still a massive challenge - if we want genuine autonomy.

Natural vision is hard problem

- Humans see effortlessly

Natural vision is hard problem

- Humans see effortlessly, but... it is very hard work for our brains!
 - There are 10^{11} neutrons involved.
 - Years of evolution generated hardwired priors.

So why bother?

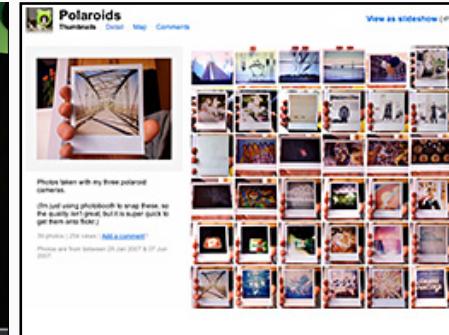
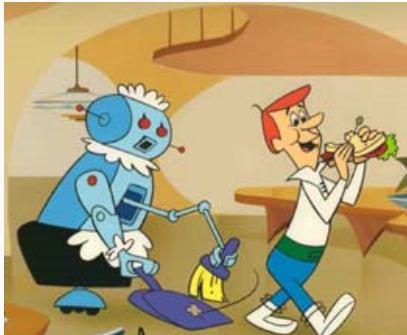
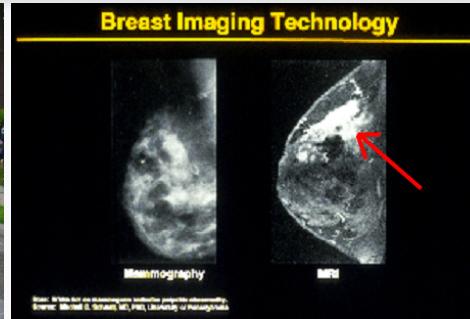
What are the advantages?

Why it matters?

- Engineering point of view - Computer Vision helps to solve many practical problems: business potential
- Scientific point of view - Human kind of visual system is one of the grand challenges of Artificial Intelligence (AI)
 - AI itself is a grand challenge of computing

Why it matters?

- Safety
- Health
- Security
- Fun
- Access
- ...



OK, got it, but I don't need it ... or do I?

- Actually you are already surrounded by devices using computer vision
- Imagine what can be done with already installed cameras!



Example: Wii remote hacks

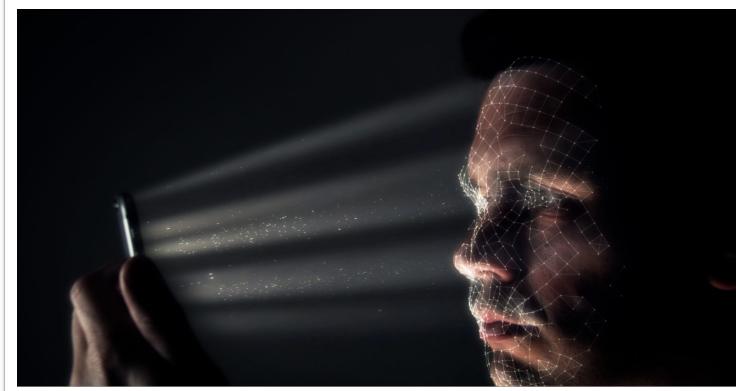
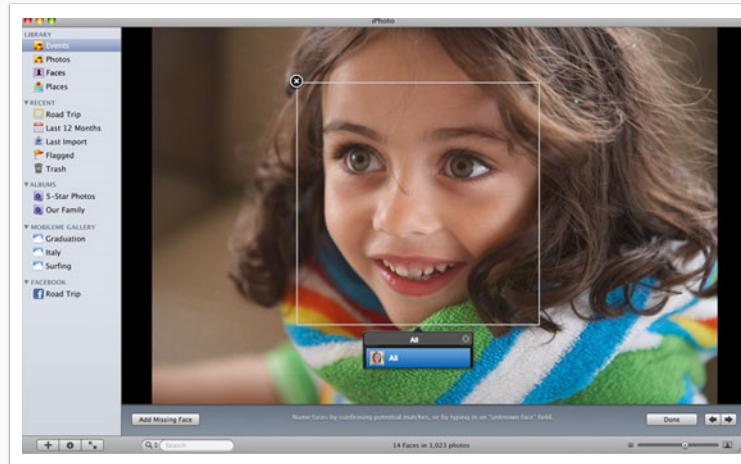


Motivation - Success stories

Recognizing “simple” patterns



Face recognition



The Smile Shutter flow

Imagine a camera smart enough to catch every smile! In Smile Shutter Mode, your Cyber-shot® camera can automatically trip the shutter at just the right instant to catch the perfect expression.

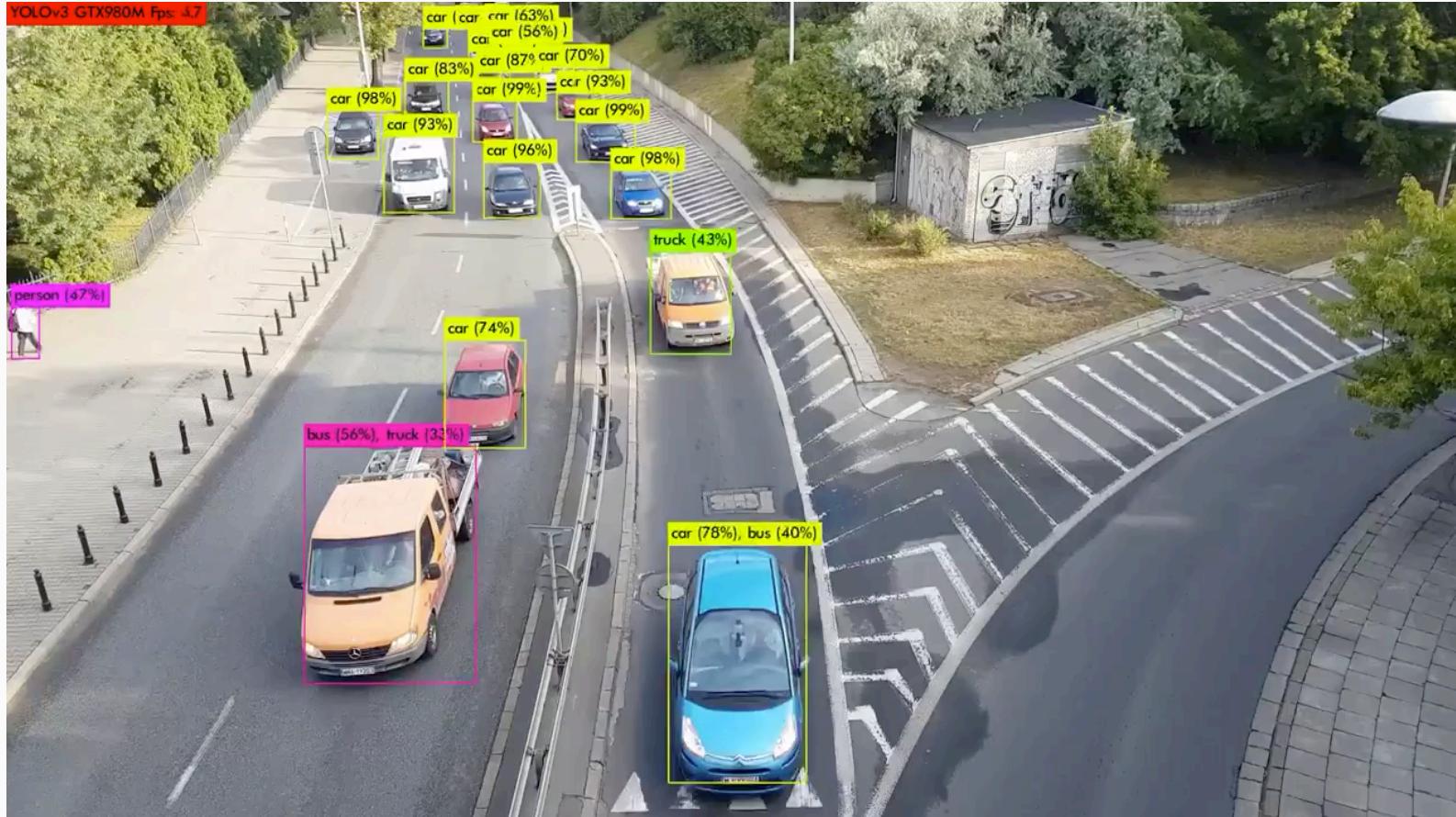
A grid of 10 small images showing a boy smiling. Each image has a yellow circle in the top-left corner and a green rectangular box around his face, indicating the camera detected a smile. Below each image is the text 'Smile Captured!'.

Application: Funny Nikon ads

“Nikon S60 detects up to 12 faces.”



Object detection and recognition



Semantic segmentation



Reconstruction: 3D from photo collections

Colosseum, Rome, Italy



San Marco Square, Venice, Italy

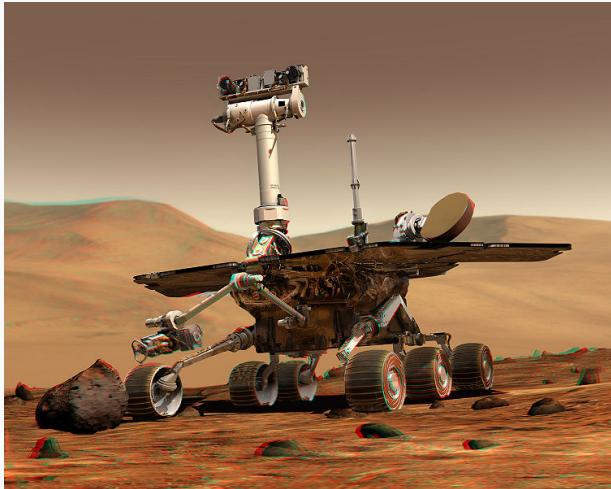


The Visual Turing test for Scene Reconstruction, Shan, Adams, Curless, Furukawa, Seitz, in 3DV 2013. [YouTube video](#).

A recent commercial 3D reconstruction system

Acute3D
Technology preview
Aerial and street-level imagery fusion

Robotics



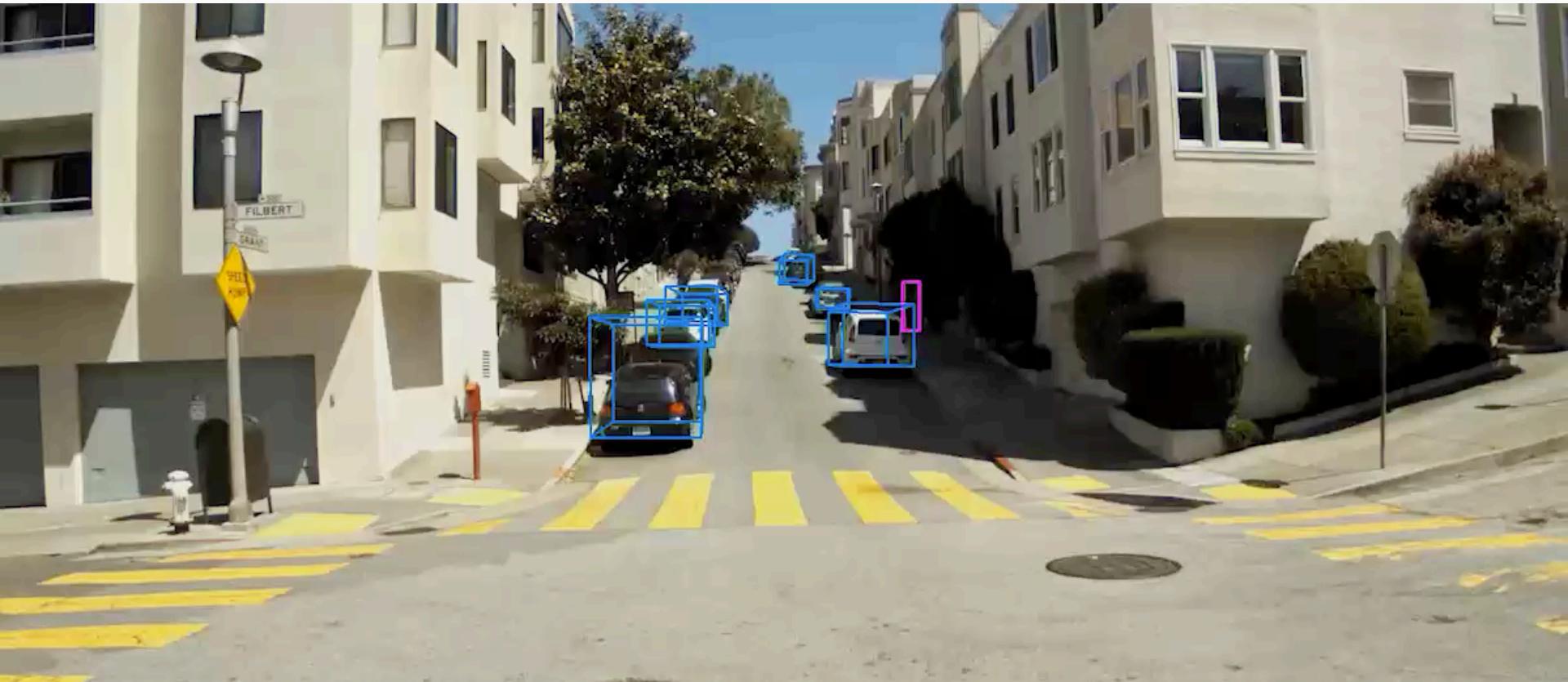
NASA's Mars Rover
See "[Computer Vision on Mars](#)"



Robocup
See www.robocup.org

[STAIRS](#) at Stanford
Saxena et al. 2008

Self-driving cars



Visual odometry and SLAM



<https://www.youtube.com/embed/J8MJRuJBW3I?rel=0>

Solin A, Cortes S, Rahtu E, Kannala J, PIVO: Probabilistic Inertial-Visual Odometry for occlusion-robust navigation, WACV 2018

Augmented Reality (AR) and Virtual Reality (VR)



Image generation



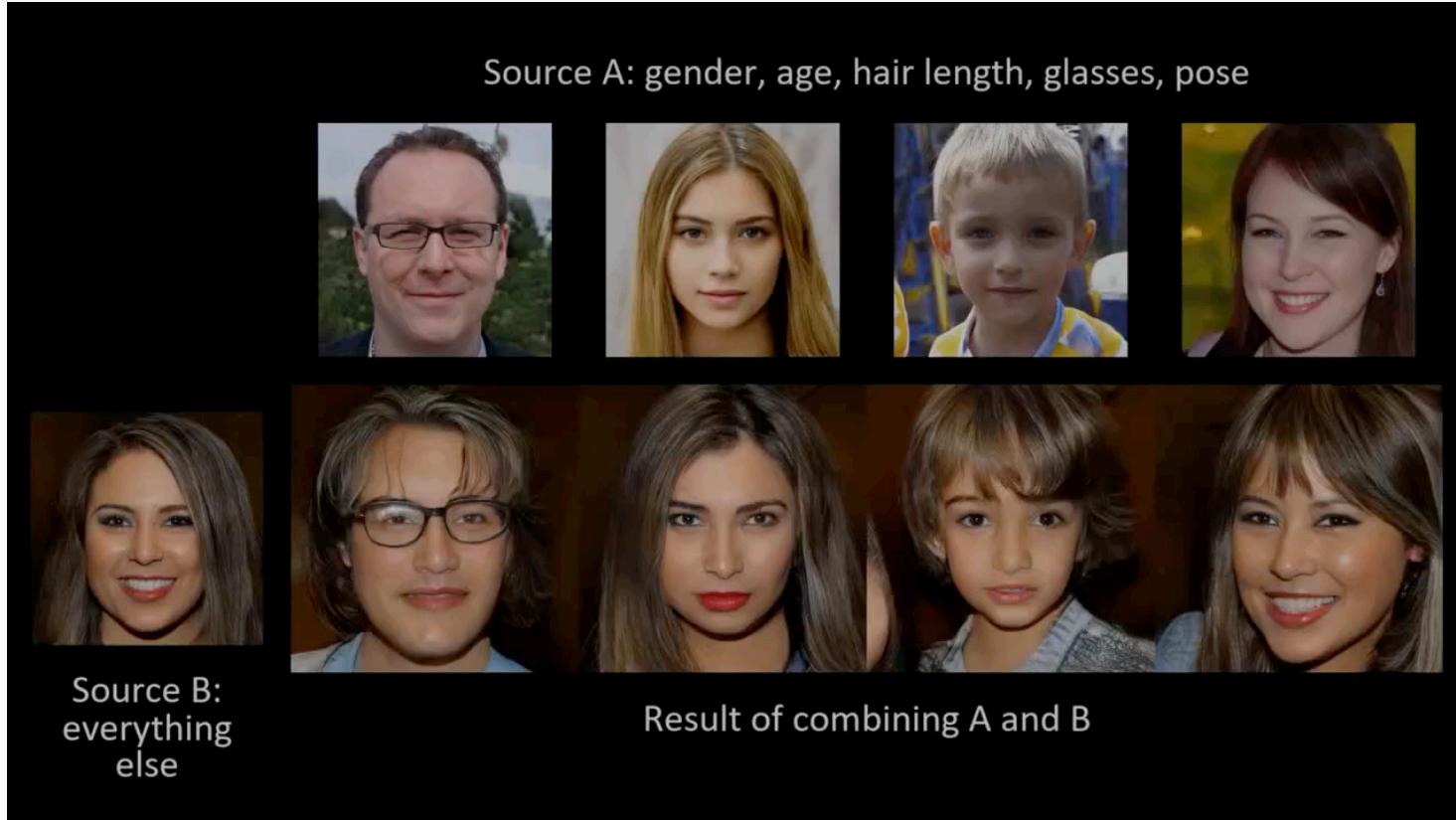
Image manipulation

Animating Faces

A single model animates all images given only a single source image



Image generation

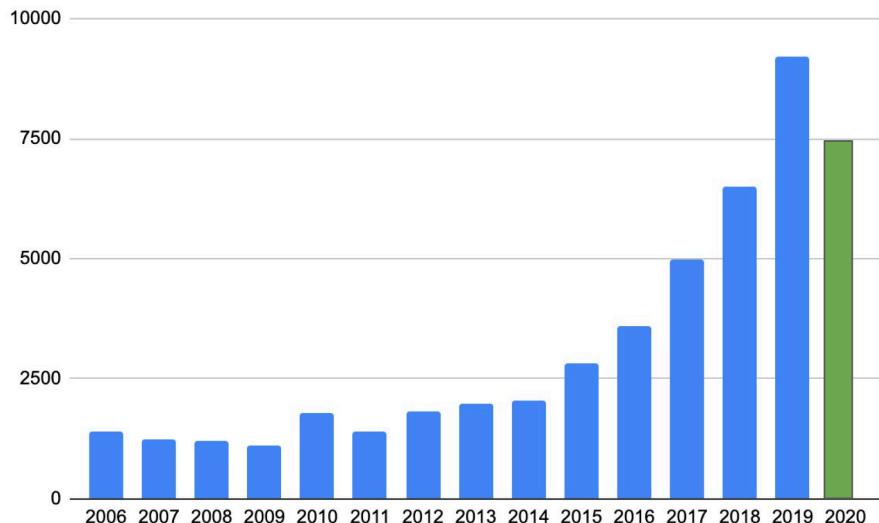


Current state of the affairs

- Many of the previous examples are less than 5 years old!
- Many new application to appear in the next 5 years
- Strong open source culture
 - Many recent state-of-the-art methods are freely available
 - See papers from top conferences like CVPR, ECCV, ICCV, and NIPS

Rapidly growing area

Attendees



Submission and accepted



IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Rapidly growing area

	Publication	<u>h5-index</u>	<u>h5-median</u>
1.	Nature	<u>362</u>	542
2.	The New England Journal of Medicine	<u>358</u>	602
3.	Science	<u>345</u>	497
4.	The Lancet	<u>278</u>	417
5.	Chemical Society reviews	<u>256</u>	366
6.	Cell	<u>244</u>	366
7.	Nature Communications	<u>240</u>	318
8.	Chemical Reviews	<u>239</u>	373
9.	Journal of the American Chemical Society	<u>236</u>	309
10.	Advanced Materials	<u>235</u>	336
11.	Proceedings of the National Academy of Sciences	<u>226</u>	291
12.	Angewandte Chemie International Edition	<u>213</u>	295
13.	JAMA	<u>209</u>	309
14.	Nucleic Acids Research	<u>208</u>	392
15.	ACS Nano	<u>199</u>	279
16.	Physical Review Letters	<u>197</u>	286
17.	Energy and Environmental Science	<u>196</u>	330
18.	Journal of Clinical Oncology	<u>196</u>	279
19.	Nano Letters	<u>194</u>	281
20.	IEEE Conference on Computer Vision and Pattern Recognition, CVPR	<u>188</u>	302

2018

Rapidly growing area

2020

Publication	<u>h5-index</u>
1. Nature	<u>368</u>
2. The New England Journal of Medicine	<u>352</u>
3. Science	<u>338</u>
4. The Lancet	<u>282</u>
5. Chemical Reviews	<u>266</u>
6. Nature Communications	<u>260</u>
7. Advanced Materials	<u>252</u>
8. Chemical Society reviews	<u>251</u>
9. Cell	<u>250</u>
10. IEEE/CVF Conference on Computer Vision and Pattern Recognition	<u>240</u>
11. Journal of the American Chemical Society	<u>236</u>
12. Proceedings of the National Academy of Sciences	<u>227</u>
13. Angewandte Chemie International Edition	<u>221</u>
14. Nucleic Acids Research	<u>217</u>
15. JAMA	<u>211</u>
16. Physical Review Letters	<u>211</u>
17. Energy & Environmental Science	<u>207</u>
18. ACS Nano	<u>203</u>
19. Journal of Clinical Oncology	<u>191</u>
20. Nano Letters	<u>188</u>

Rapidly growing area

Publication	<u>h5-index</u>	<u>h5-median</u>
1. Nature	<u>376</u>	552
2. The New England Journal of Medicine	<u>365</u>	639
3. Science	<u>356</u>	526
4. The Lancet	<u>301</u>	493
5. IEEE/CVF Conference on Computer Vision and Pattern Recognition	<u>299</u>	509
6. Advanced Materials	<u>273</u>	369
7. Nature Communications	<u>273</u>	366
8. Cell	<u>269</u>	417
9. Chemical Reviews	<u>267</u>	438
10. Chemical Society reviews	<u>240</u>	368
11. Journal of the American Chemical Society	<u>236</u>	324
12. Angewandte Chemie	<u>229</u>	316
13. Proceedings of the National Academy of Sciences	<u>228</u>	299
14. JAMA	<u>220</u>	337
15. Nucleic Acids Research	<u>219</u>	475

2021

Rapidly growing area - substantial commercial interest

Diamond Sponsors



Platinum Sponsors



Gold Sponsors



Silver Sponsors



Plenty of job opportunities

- Companies are looking for computer vision and deep learning experts.
- Big Internet players are investing heavily (Apple, Google, Facebook, Microsoft, Baidu, Tencent, ...) as well as car industry (Tesla, BMW,...)
- Strong imaging ecosystem at Tampere and elsewhere in Finland



Specifics of this course

What will you learn on this course?

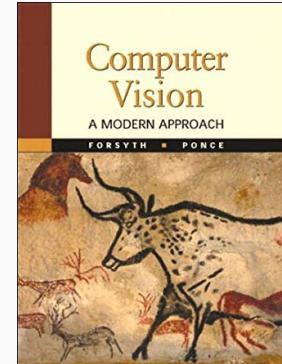
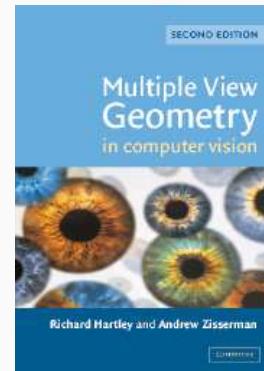
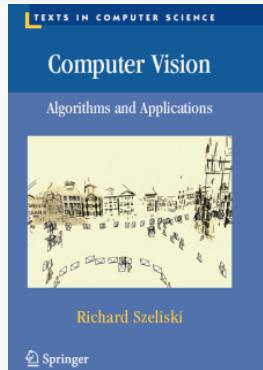
- Fundamentals of projective geometry and camera models
- Basics of 3D reconstruction and multi view stereo
- Basics of optical flow and object tracking
- Basics of image classification and CNNs
- Basics of object detection

What will you NOT learn on this course?

- Software packages
 - PyTorch, TensorFlow, Keras, Caffe, etc.
 - We have simple exercise with Python, and Matlab though
- Deep learning tricks
 - Tweaking architectures, loss functions, etc.
- All the bells and whistles in state-of-the-art systems
 - We concentrate on the basic concepts (get them right and the rest is easier for you)

Useful textbooks

- Szeliski: Computer Vision
 - [Full-copy freely available](#)
- Hartley & Zisserman: Multiple View Geometry in Computer Vision
 - Available as e-book via library
- Forsyth & Ponce: Computer Vision
 - [Full-copy freely available](#)



Crew and website

- Instructor: Esa Rahtu (<http://esa.rahtu.fi>)
- Teaching assistants:
 - Leading TA: Tiitus Hiltunen (first.last@tuni.fi)
- Course website at Moodle
 - Lecture slides (videos as well, I hope...), Exercise materials, Grading
- Course Slack workspace (link in Moodle)
 - Exercises, announcements, questions, discussion, and more

Organization

- Duration of the course is 7 weeks
- Each week covers one (sub)topic
 - Builds on previously learned content
 - Lectures, exercises, and solution session
- To pass the course
 - Final exam + laboratory exercises

Weekly schedule

Time	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-09:00	Lecture 1		Excercise group 1	Excercise group 4	
09:00-10:00	Remote		Remote	Remote	
10:00-11:00			Excercise group 2	Excercise group 5	
11:00-12:00			Remote	Remote	
12:00-13:00		Lecture 2	Excercise group 3	Excercise group 6	
13:00-14:00		Remote	Remote	Remote	
14:00-15:00					Solution session
15:00-16:00					Remote

Laboratory exercises

- Exercises contain pen&paper and programming tasks
 - Matlab or Python (recommended)
- Solve on your own time or during exercise session (6 groups)
- Answers returned via Moodle
- Solution session on Fridays (solutions not available on website)
- Note! Answers can be still returned after solutions session (less points)

Grading

- Laboratory exercises:
 - 7 lab exercises that each give max 4 points if returned in time (before the solution session starts) and max 1 point if returned late.
 - Minimum 10 points and non-zero points from 6 lab exercises
- Final exam:
 - Max 30 points
 - Minimum 10 points

Grading

- Theoretical maximum points will be:
 $30 \text{ (exam)} + 7 \times 4 \text{ (exercises)} = 58$
- Grade will be determined according to following linear scale:

Grade	Min. points
1	29
2	35
3	41
4	47
5	53

Weekly topics

- Week 1: Perspective camera and image filtering
- Week 2: Image classification and retrieval
- Week 3: Object class detection
- Week 4: Local features and least squares fitting
- Week 5: Planar transforms and tracking
- Week 6: Camera calibration and two-view stereo
- Week 7: Multi-view stereo and structure from motion

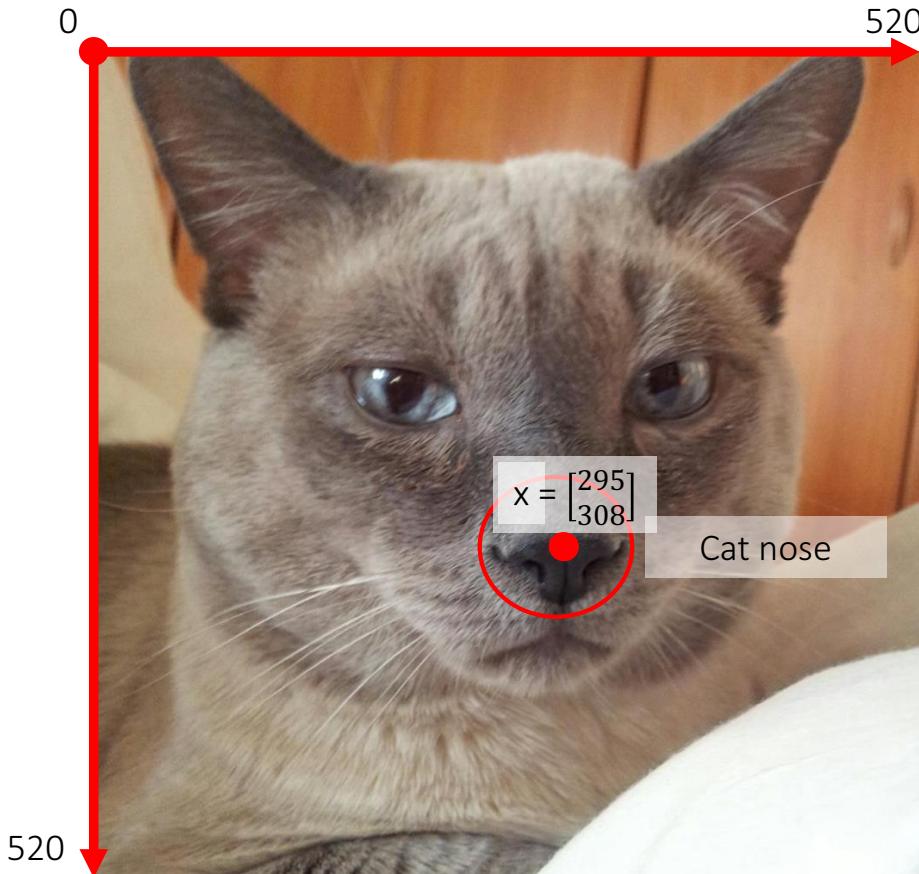
Questions at this point (use Slack)?

Lecture 1: Camera model

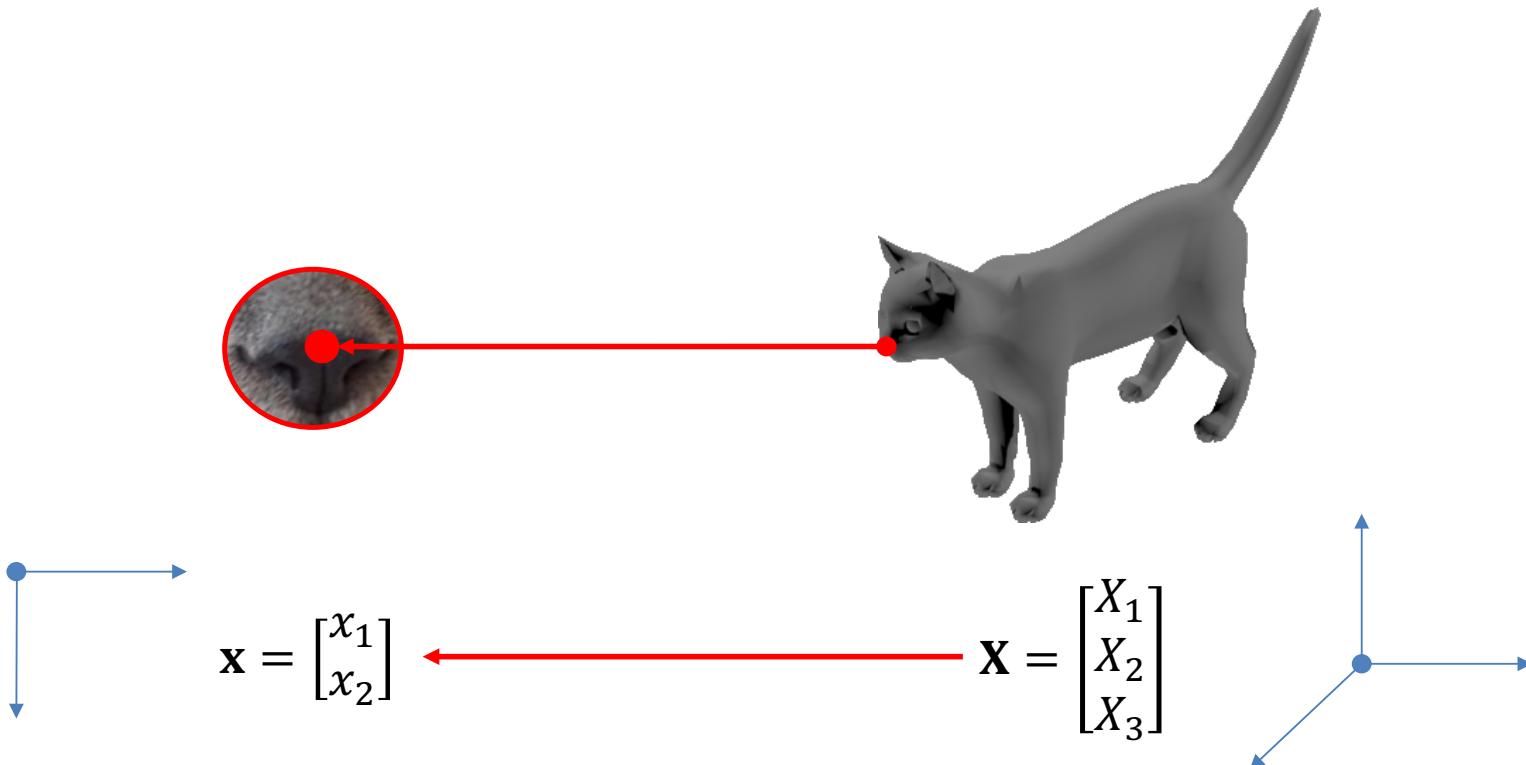
Relevant reading

- Chapters 2,3, and 6 in [Hartley & Zisserman]
 - Comprehensive presentation of the core content
- Chapter 2 in [Szeliski]
 - Broader overview of the image formation

This is (a picture of) a cat

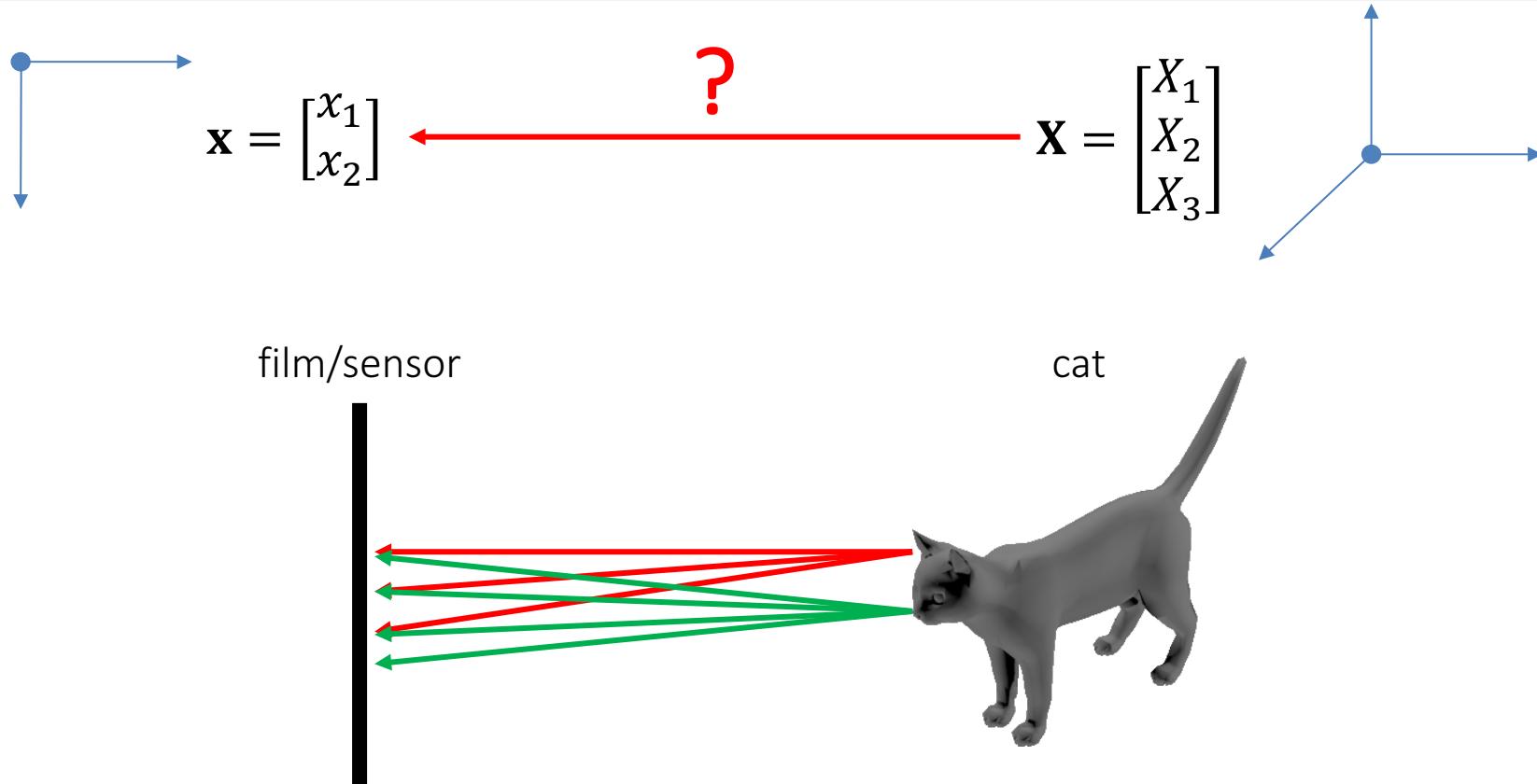


Cat lives in a 3D world



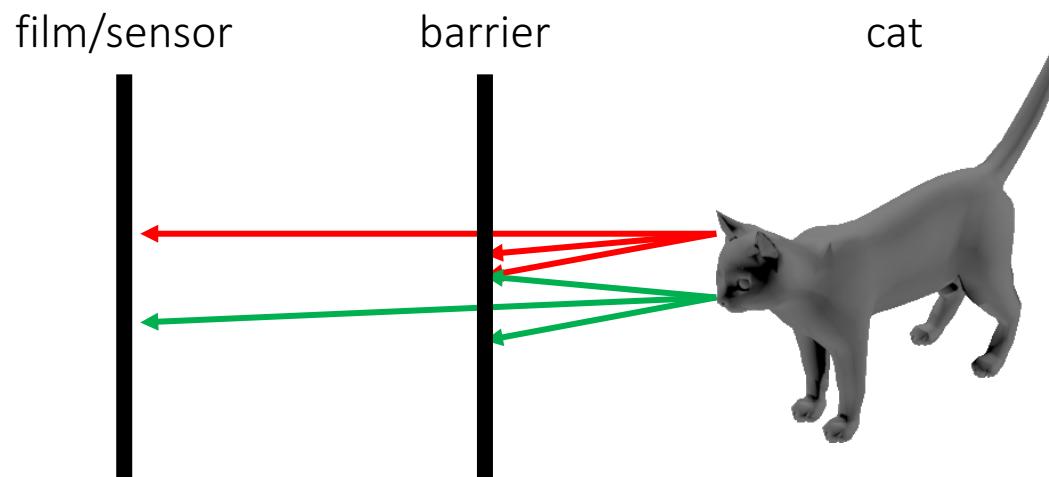
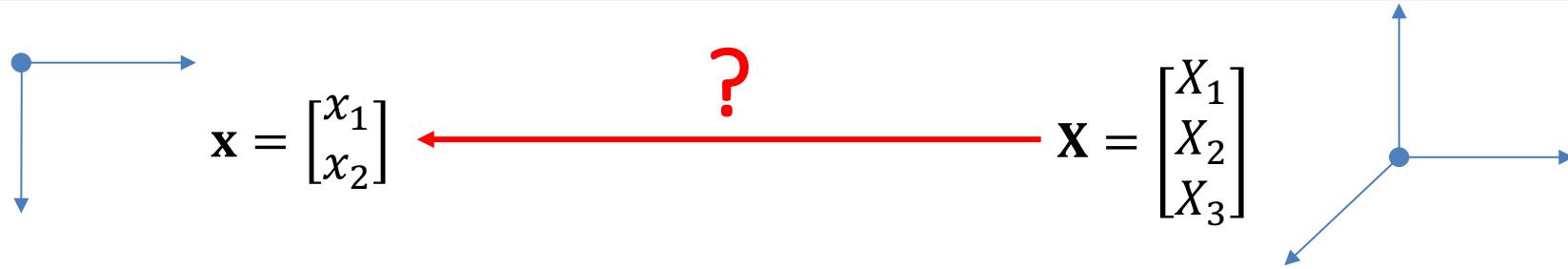
The point \mathbf{X} in world space projects to the point \mathbf{x} in image space.

Going from X in 3D to x in 2D



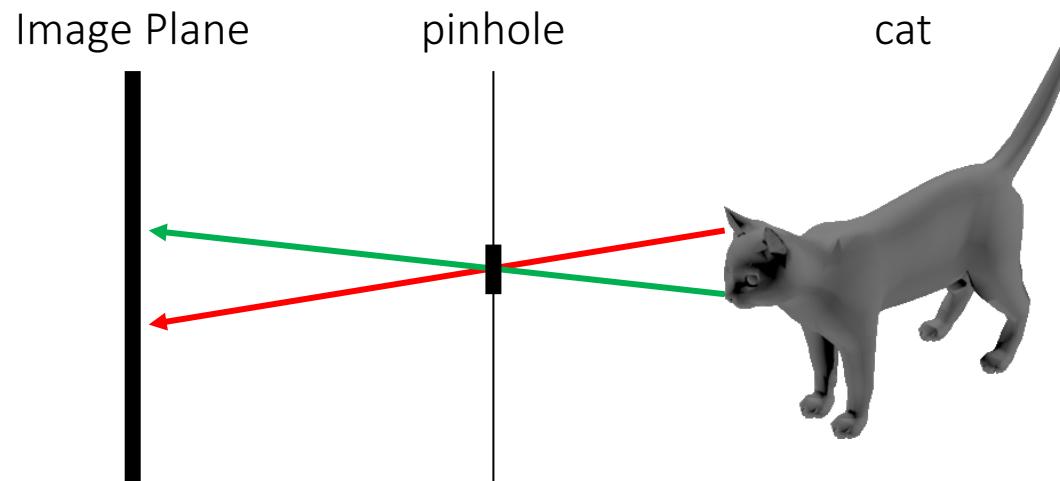
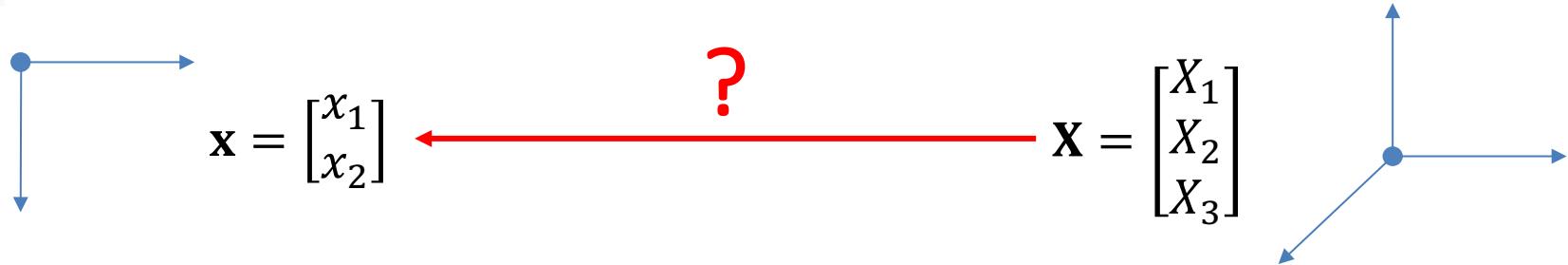
The output would be blurry if film just exposed to the cat.

Going from X in 3D to x in 2D



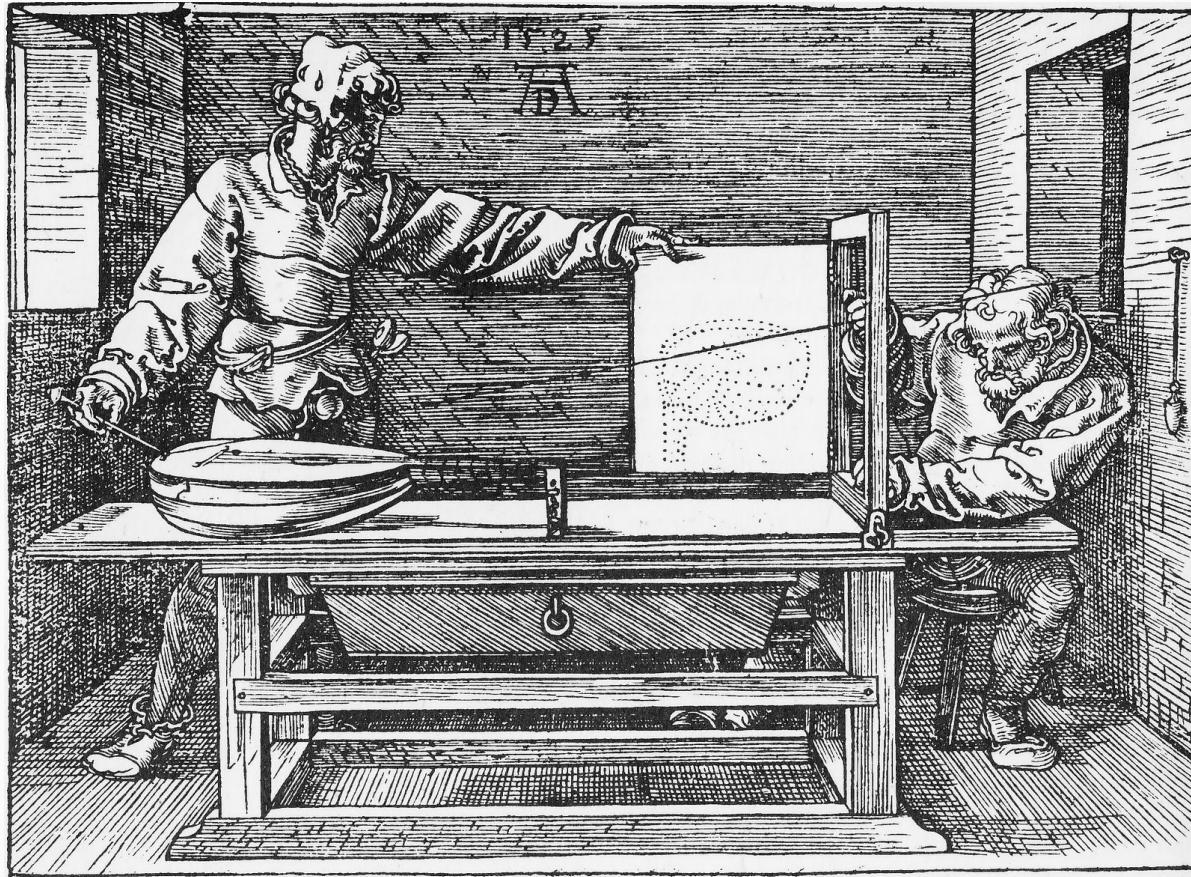
Blur reduced, looks good

Pinhole camera

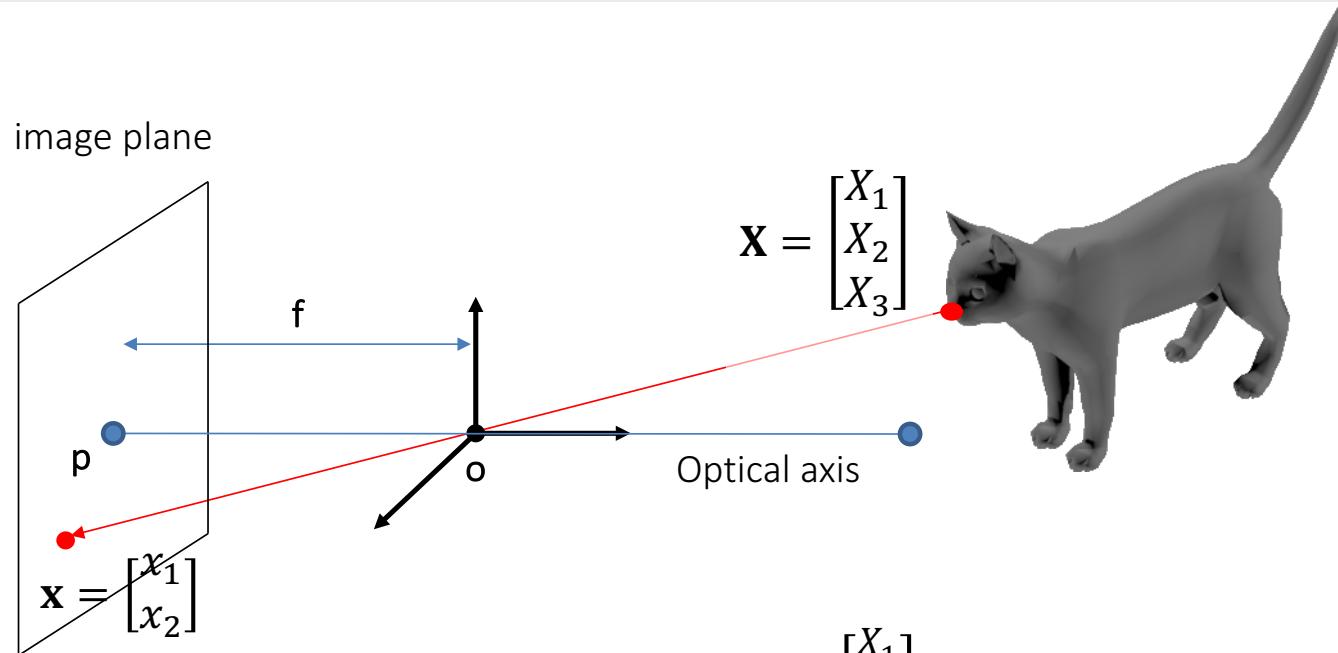


All rays passing through a single point (center of projection)

Pinhole camera



Pinhole camera



f – focal length

o – camera origin

p – principal point

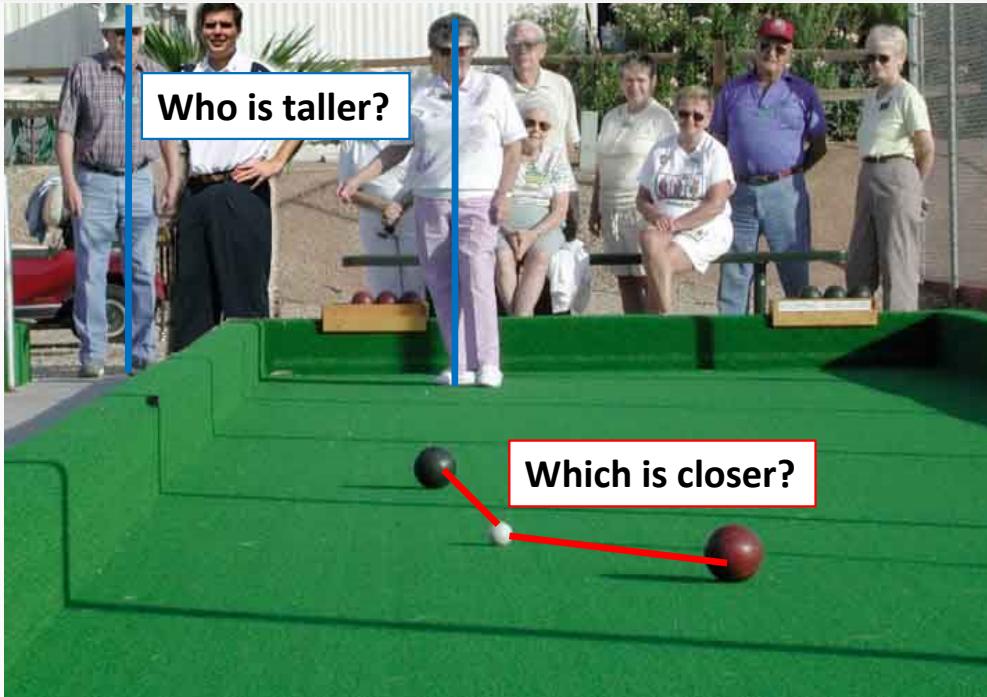
The 3D point $\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$ is imaged into $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f \frac{X_1}{X_3} \\ f \frac{X_2}{X_3} \end{bmatrix}$$

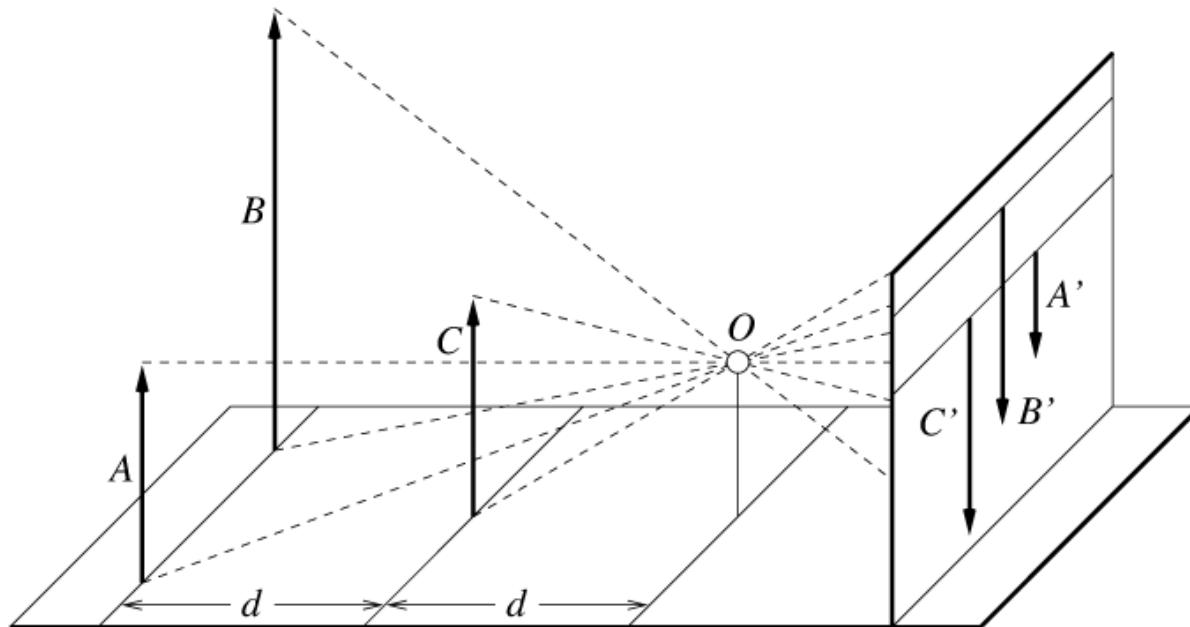
What happens in the projection?

- Projection from 3D to 2D -> information is lost
- What properties are preserved?
- What properties are not preserved?

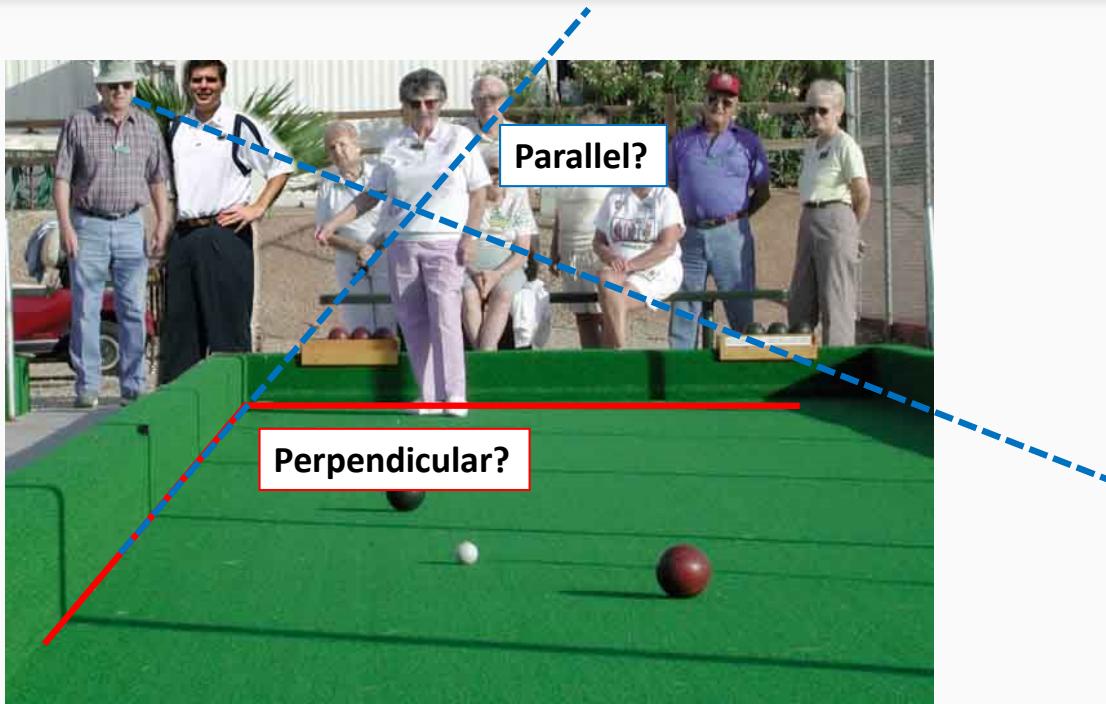
Projective geometry - what is lost?



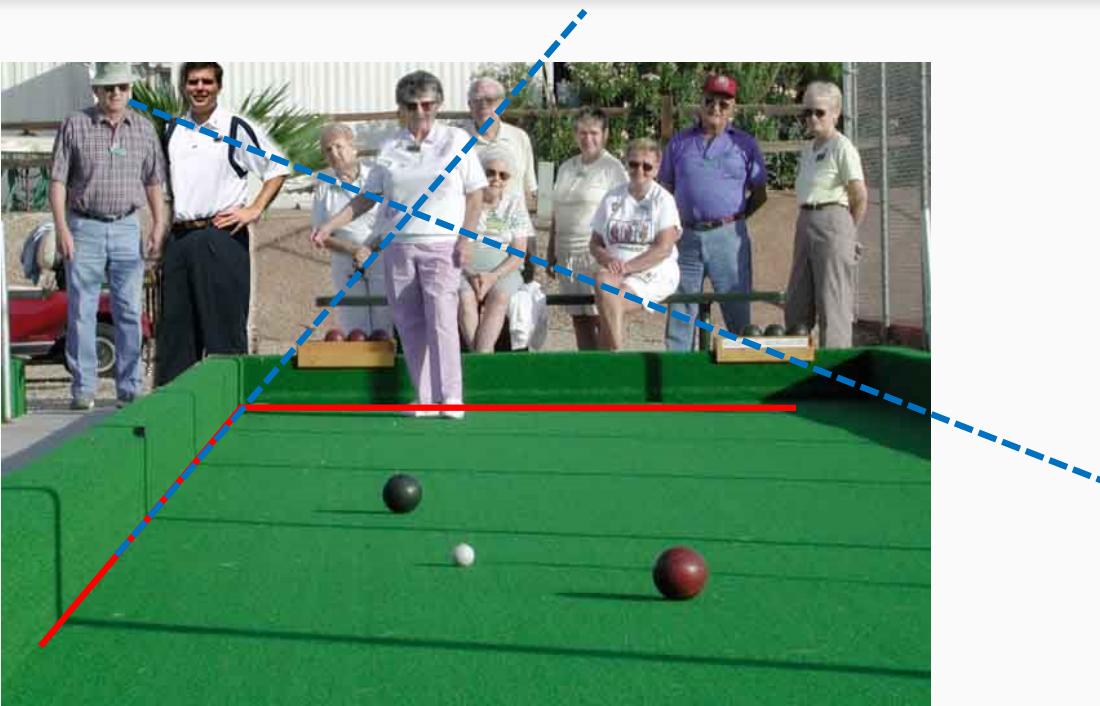
Length is not preserved



Angles are not preserved



Straight lines are still straight

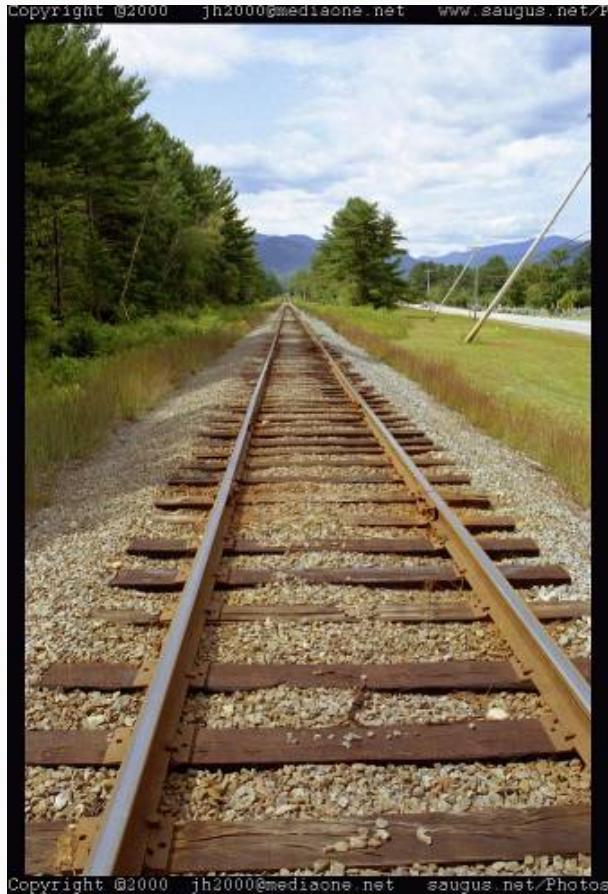


What happens in the projection?

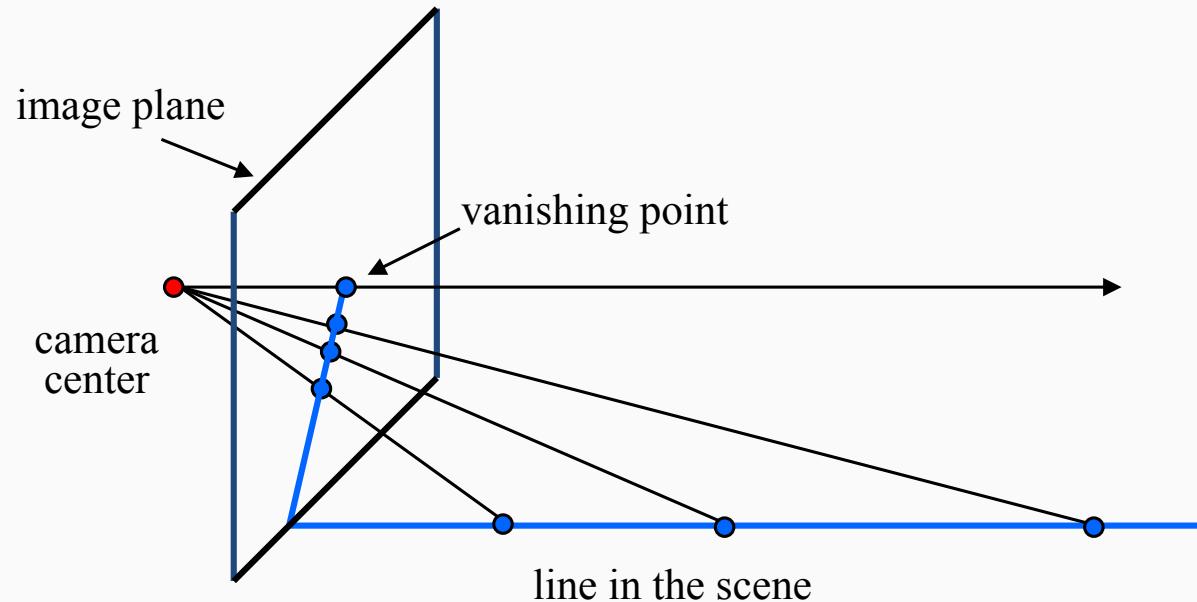
- Projection from 3D to 2D -> information is lost
- What properties are preserved?
 - Straight lines
 - Incidence
- What properties are not preserved?
 - Angles
 - Lengths

Vanishing points and lines

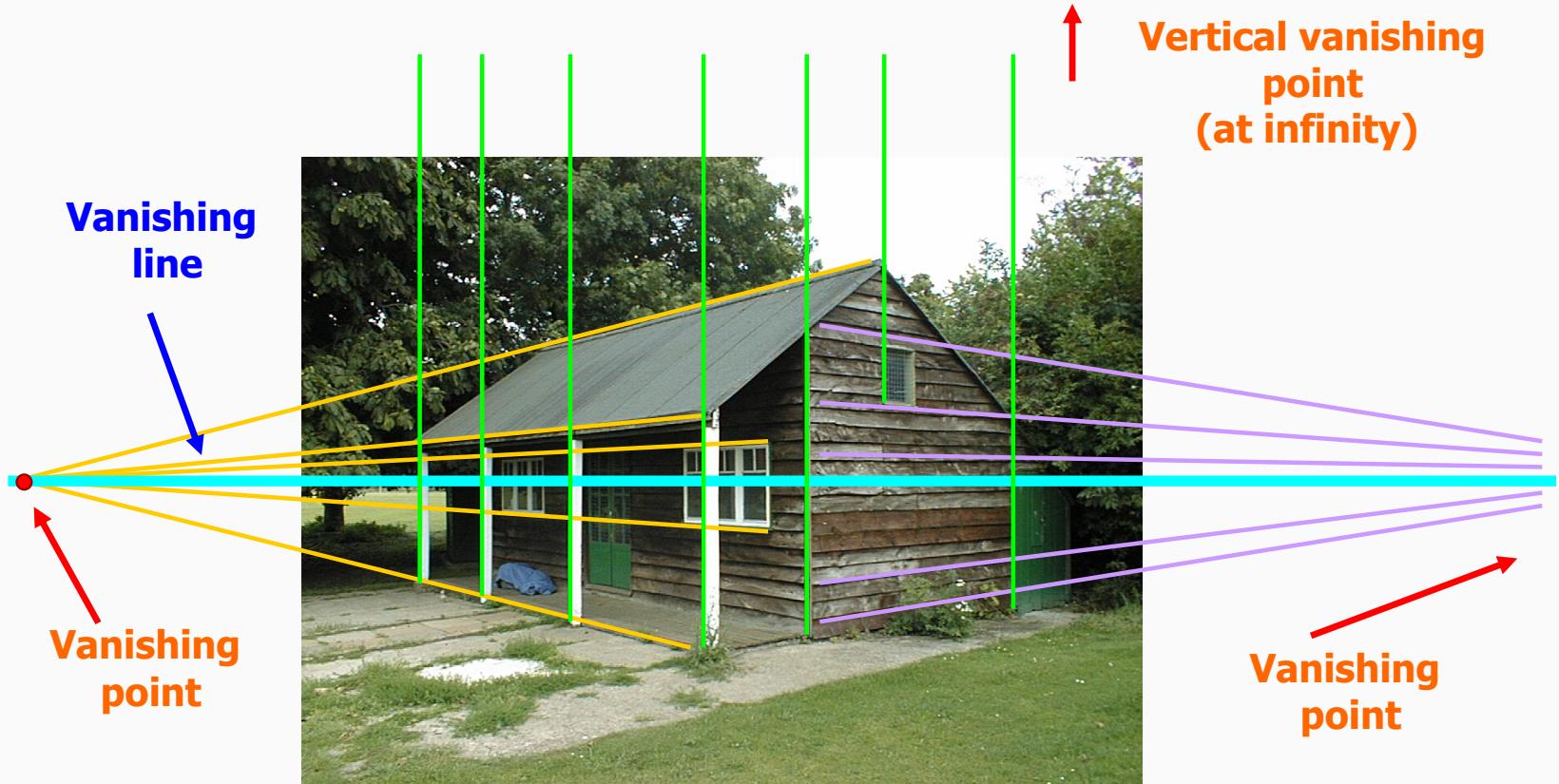
- Parallel lines in the world intersect at a “vanishing point”



Constructing the vanishing point of a line



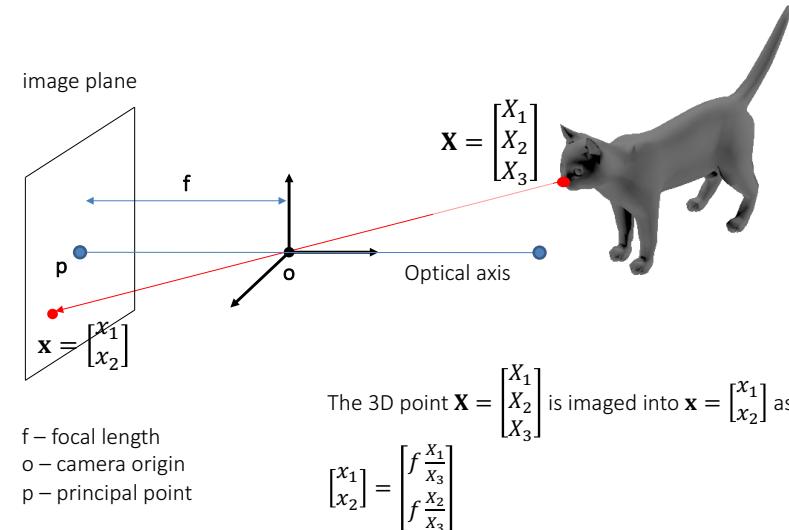
Vanishing points and lines



All parallel lines will have the same vanishing point.

Homogenous coordinates

- The projection $x = fX/x_3$ is non linear!
- Can be made linear using homogenous coordinates
- Homogenous coordinates allow for transforms to be concatenated easily



Homogenous coordinates

Conversion to homogenous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Conversion from homogenous coordinates

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w) \quad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Invariance to scaling

$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

E.g. [1,2,3] is the same as [3,6,9] and both represent the **same** inhomogeneous point [0.33,0.66].

Basic geometry in homogenous coordinates

- Line equation: $ax+by+c=0$
- A pixel p in homogenous coordinates:
- Line is given by cross product of two points
- Intersection of two lines is given by cross product of the lines

$$line_i = \begin{bmatrix} a_i \\ b_i \\ c_i \\ u_i \end{bmatrix}$$

$$p_i = \begin{bmatrix} v_i \\ 1 \end{bmatrix}$$

$$line_{ij} = p_i \times p_j$$

$$q_{ij} = line_i \times line_j$$

3D Euclidean transformation

- Cat moves through 3D space
- The movement of the nose can be described using an Euclidean Transform

$$\mathbf{X}'_{3 \times 1} = R_{3 \times 3} \mathbf{X}_{3 \times 1} + \mathbf{t}_{3 \times 1}$$

$\begin{pmatrix} \cos \beta \cos \gamma & -\sin \beta & \cos \beta \\ \sin \beta \cos \gamma & \cos \beta & -\sin \beta \sin \gamma \\ \sin \gamma & 0 & \cos \gamma \end{pmatrix}$ { rotation } { translation }



Building the 3D rotation matrix R

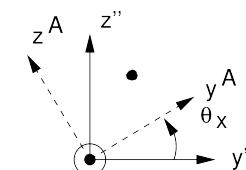
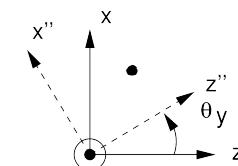
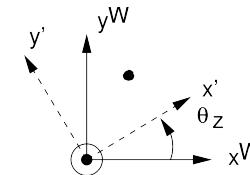
- R can be build from various representations (Euler ang., quaternion)
- Euler angles represent the rotation using three parameters, one for each axis

$$X' = R_z X^W = \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} X^W$$

$$X'' = R_y X' = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} X'$$

$$X^A = R_x X'' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \pm \sin \theta_x \\ 0 & \mp \sin \theta_x & \cos \theta_x \end{bmatrix} X''$$

$$\mathbf{R}^{CW} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$



Order matters!

3D Euclidean transformation

- Concatenation of successive transforms is a mess!

$$\mathbf{X}_1 = \mathbf{R}_1 \mathbf{X} + \mathbf{t}_1$$

$$\mathbf{X}_2 = \mathbf{R}_2 \mathbf{X}_1 + \mathbf{t}_2$$

$$= \mathbf{R}_2(\mathbf{R}_1 \mathbf{X} + \mathbf{t}_1) + \mathbf{t}_2 = (\mathbf{R}_2 \mathbf{R}_1) \mathbf{X} + (\mathbf{R}_2 \mathbf{t}_1 + \mathbf{t}_2)$$

Homogenous coordinates save the day!

- Replace 3D points $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ with homogenous versions $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$
- The Euclidean transform becomes
$$\begin{bmatrix} \mathbf{X}' \\ 1 \end{bmatrix} = \mathbf{E} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}$$
- Transformation can now be concatenated by matrix multiplication

$$\begin{bmatrix} \mathbf{X}_1 \\ 1 \end{bmatrix} = \mathbf{E}_{10} \begin{bmatrix} \mathbf{X}_0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} \mathbf{X}_2 \\ 1 \end{bmatrix} = \mathbf{E}_{21} \begin{bmatrix} \mathbf{X}_1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{X}_2 \\ 1 \end{bmatrix} = \mathbf{E}_{21} \mathbf{E}_{10} \begin{bmatrix} \mathbf{X}_0 \\ 1 \end{bmatrix}$$

More 3D-3D and 2D-2D transformations

Projective (15 dof):

$$\begin{bmatrix} X'_1 \\ X'_2 \\ X'_3 \\ X'_4 \end{bmatrix} = [\mathbf{P}_{4 \times 4}] \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}$$

Affine (12 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{3 \times 3} & \mathbf{t}_3 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Similarity (7 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} S\mathbf{R}_{3 \times 3} & \mathbf{t}_3 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Euclidean (6 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t}_3 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Projective (aka Homography, 8 dof):

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = [H_{3 \times 3}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Affine (6 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{2 \times 2} & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Similarity (4 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} S\mathbf{R}_{2 \times 2} & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Euclidean (3 dof):

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{2 \times 2} & \mathbf{t}_2 \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

Examples of 2D-2D transforms



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

Euclidean
3 DoF



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

Similarity
4 DoF



$$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

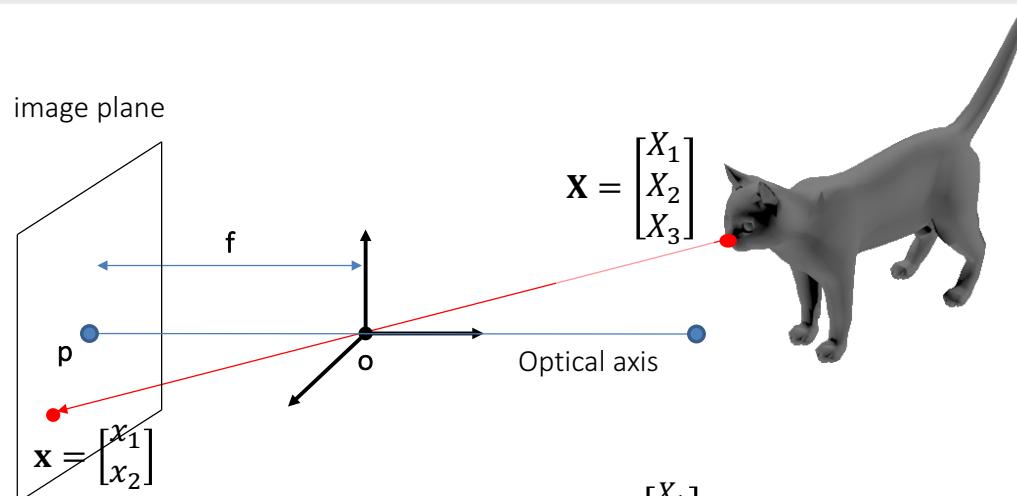
Affine
6 DoF



$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$

Projective
8 DoF

Perspective transformation (3D-2D)



f – focal length

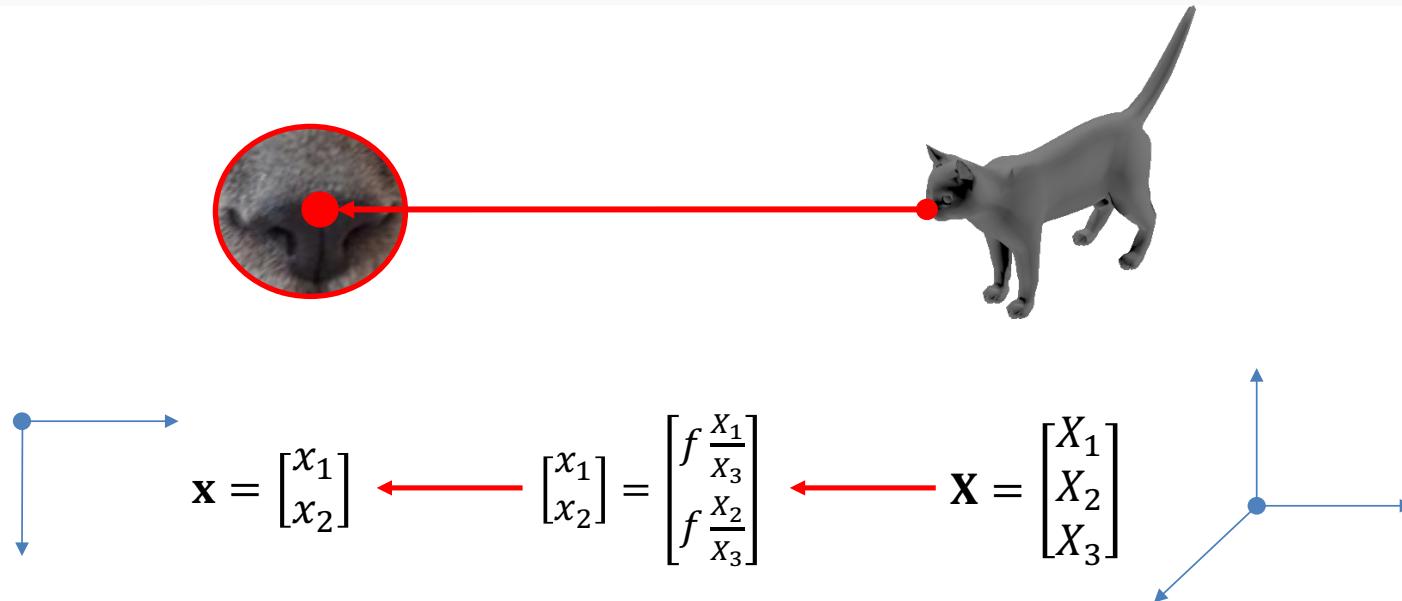
o – camera origin

p – principal point

The 3D point $\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$ is imaged into $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} f \frac{X_1}{X_3} \\ f \frac{X_2}{X_3} \end{bmatrix}$$

Perspective using homogenous coordinates



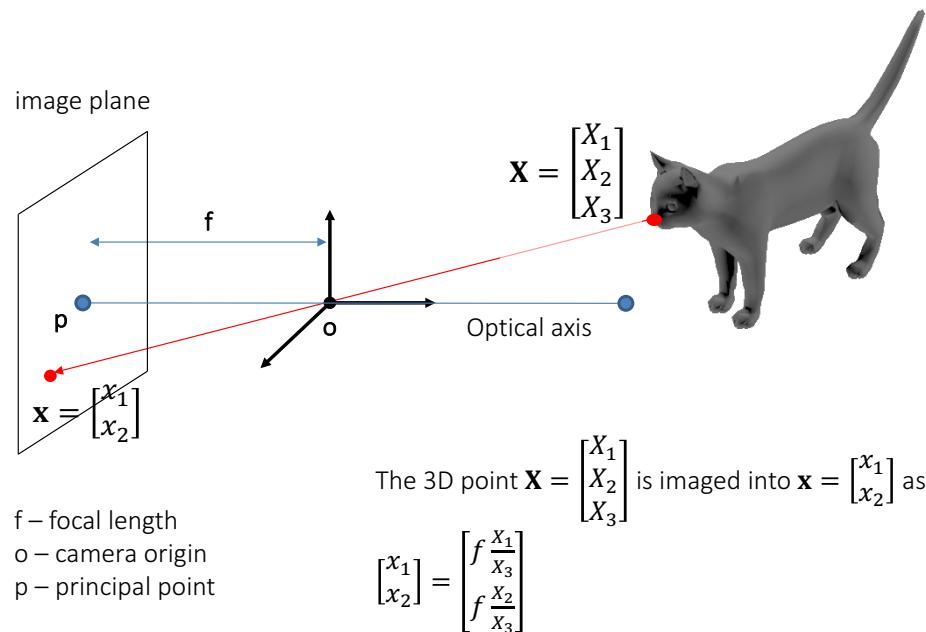
$$\lambda \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix} \rightarrow \begin{aligned} \lambda x_1 &= fX_1 & x_1 &= f \frac{X_1}{X_3} \\ \lambda x_2 &= fX_2 & x_2 &= f \frac{X_2}{X_3} \\ \lambda &= X_3 \end{aligned}$$

Perspective using homogenous coordinates

Image Point	Projection Matrix	World Point
$\lambda \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{bmatrix}$		

Wait! Our setup has several assumptions

- Camera at world origin
- Camera aligned with world coordinates
- Ideal pinhole camera



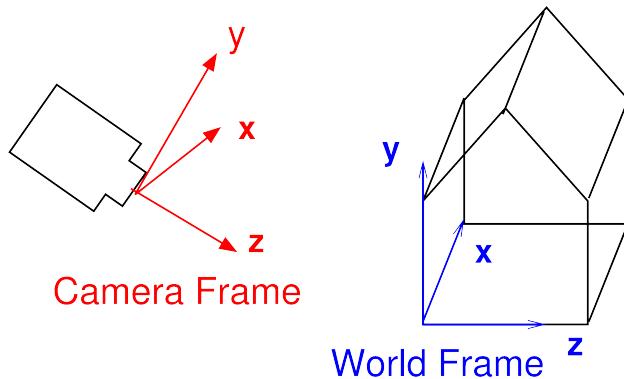
Removing the initial assumptions

- It is useful to split the overall projection matrix into three parts:
 - A part that depends on the internals of the camera (intrinsic)
 - A vanilla projection matrix
 - An Euclidean transformation between the world and camera frames (extrinsic)
- Assume first that the world is aligned with camera coordinates
-> the extrinsic camera matrix is an identity

Image Point	Camera's Intrinsic Calibration	Projection matrix (vanilla)	Camera's Extrinsic Calibration	World Point
$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	\mathbf{x}

More realistic setting - camera pose

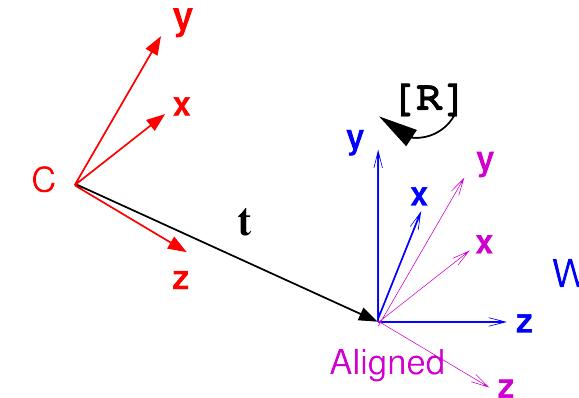
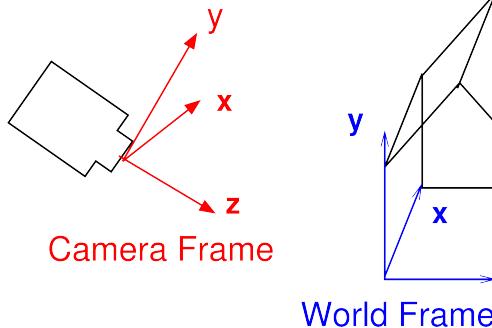
- Assume the camera is translated and rotated with respect to the world



The camera pose

- The non-ideal camera pose can be taken into account by first rotating and translating points from world frame to the camera frame

$$\begin{bmatrix} \mathbf{x}^c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^w \\ 1 \end{bmatrix}$$



The intrinsic parameters

- Describe the hardware properties of a real camera
 - The image plane might be skewed
 - Central axis of the lens might not line up with the optical axis
 - The pixels might not be square
 - Lens distortion

$$K = \begin{bmatrix} f & 0 & 0 \\ 0 & \gamma f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{u_0}{f} \\ 0 & 1 & \frac{v_0}{\gamma f} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f & sf & u_o \\ 0 & \gamma f & v_o \\ 0 & 0 & 1 \end{bmatrix}$$

different scaling
on x and y
 γ is the aspect
ratio.

Origin offset,
 (u_o, v_o) is the
principal point.

s accounts for
skew

Summary of steps from scene to image

- Move the scene point $(\mathbf{X}^w, 1)^T$ into camera coordinate system by 4x4 (extrinsic) Euclidean transformation:

$$\begin{bmatrix} \mathbf{x}^c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^w \\ 1 \end{bmatrix}$$

- Project into ideal camera via the vanilla perspective transformation

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = [\mathbf{I} | \mathbf{0}] \begin{bmatrix} \mathbf{x}^c \\ 1 \end{bmatrix}$$

- Map the ideal image into the real image using intrinsic matrix

$$\begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix}$$

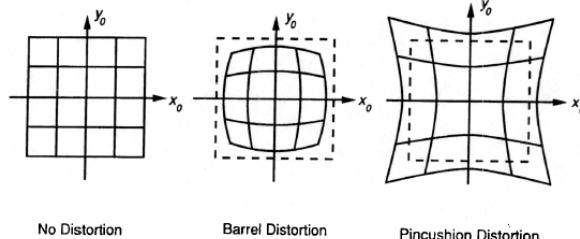
Camera projection matrix P

Image Point	Camera's Intrinsic Calibration	Projection matrix (vanilla)	Camera's Extrinsic Calibration	World Point
$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} f & sf & u_0 \\ 0 & \gamma f & v_o \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	\mathbf{x}

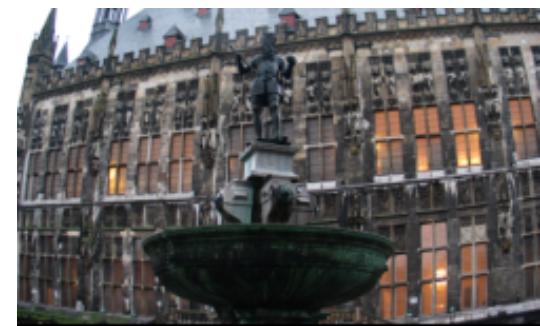
$$\mathbf{P}_{3 \times 4} = K[R|\mathbf{t}]$$

Beyond pinholes: Radial distortion

- Common in wide-angle lenses
- Creates non-linear terms in projection
- Usually handled by solving non-linear terms and then correcting the image



Original

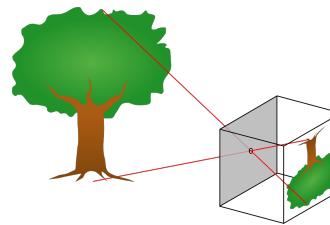


Corrected



Things to remember

- Pinhole camera model
- Homogenous coordinates
- Camera projection matrix



$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{K}[\mathbf{R} \quad \mathbf{t}] \mathbf{X}$$

That's it folks!