DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Vision and Image Processing: Introductory lecture

Søren Ingvor Olsen

The teachers

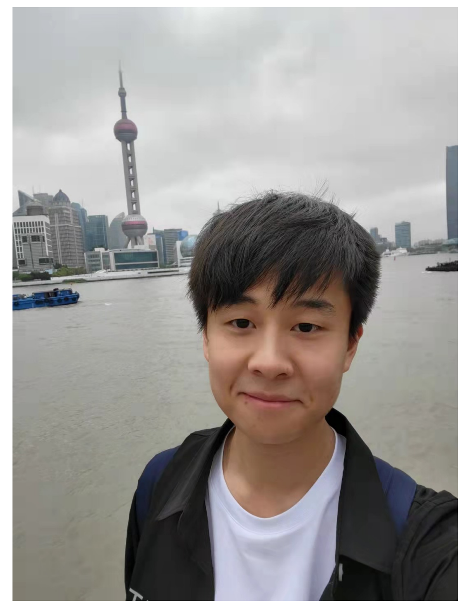
- Søren Ingvor Olsen, ingvor@di.ku.dk, DIKU Image Group (course responsible)
- Francois Bernard Lauze, francois@di.ku.dk, DIKU Image Group

Teaching Assistants

- TA1 : Seyednavid Mohammadifoumani
- TA2 : Steffen Czolbe
- TA3 : Peidi Xu


Steffen


Navid


Peidi

CoVid19 update

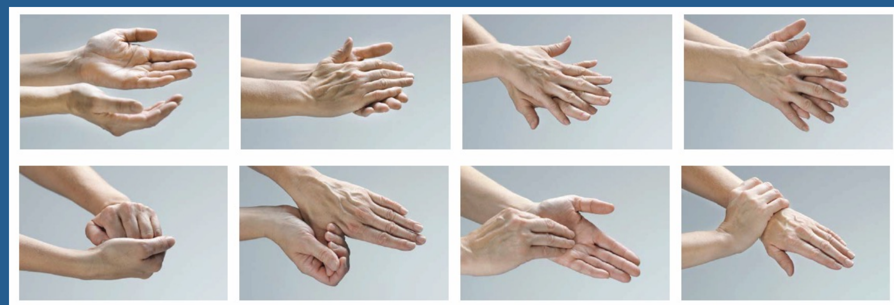
- The infection level is increasing rapidly
- There is NO restrictions at KU (yet)
- You are NOT required to show Corona passport

- Please take care, wash your hands
- Use disinfectant
- Stay home if you are ill, and take a PCR-test
- If infected inform the people you have been close to

Corona – guidelines

Brug af håndsprit

Hand disinfectant

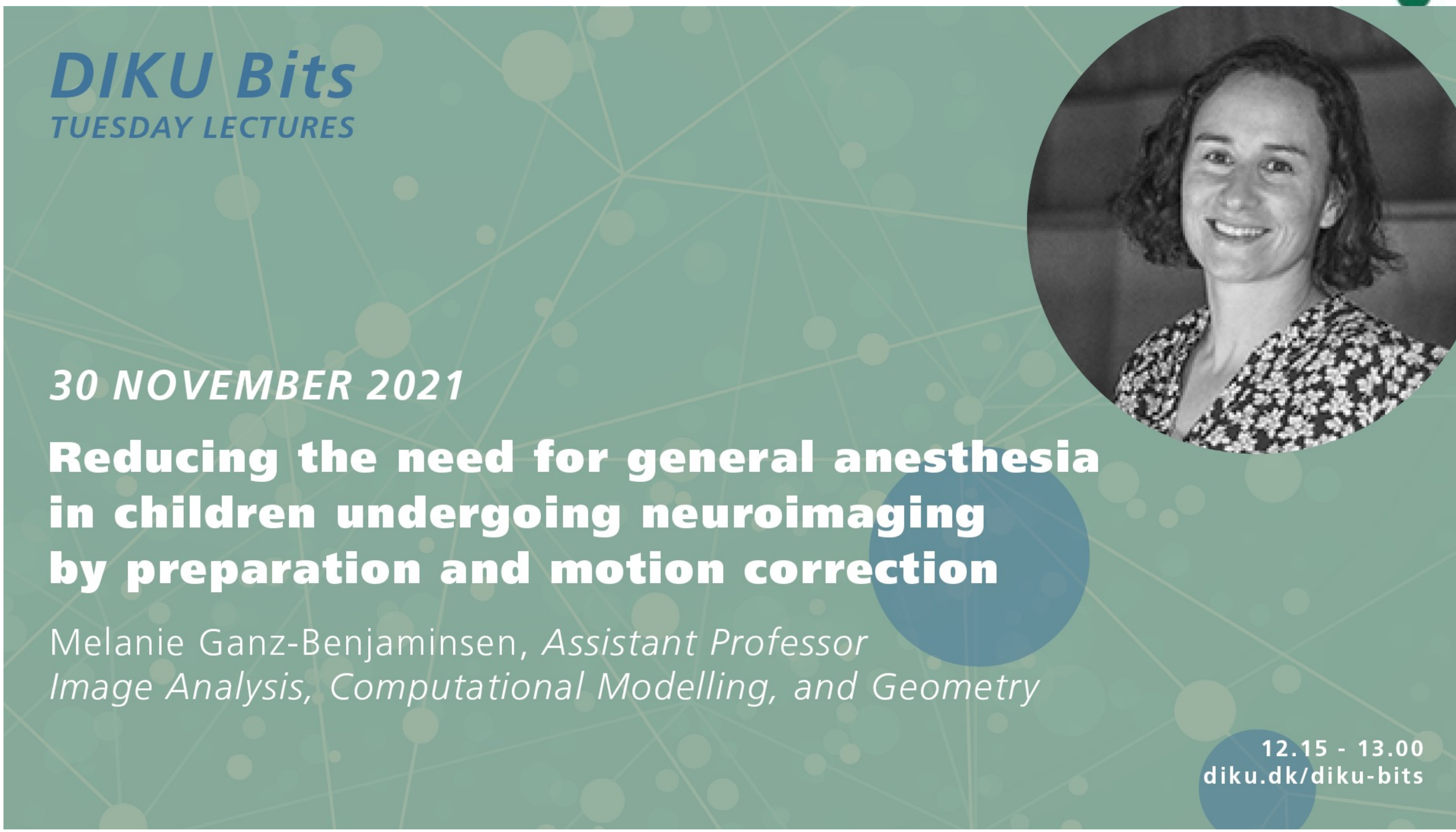


- Skal bruges på rene og tørre hænder
- *Er hænderne våde, mindskes effekten*
- Skal gnides ind i ca. 30 sekunder.
- Brug de samme greb som ved håndvask.

- Must be used on clean and dry hands
- *Wet hands reduce the effect*
- Rub in for approx. 30 seconds.
- Use the same procedures as for hand washing.

UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

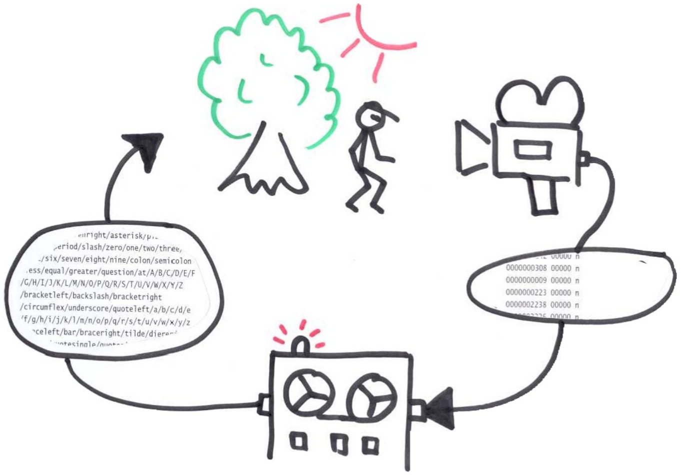
Announcement



What is this course about?

Computer Vision

Image Procesing



Making computers see

Definitions of Computer Vision

- **Wikipedia:** An interdisciplinary field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

- **Techopedia:** Computer vision is a field of computer science that works on enabling computers to see, identify and process images in the same way that human vision does, and then provide appropriate output. It is like imparting human intelligence and instincts to a computer. In reality though, it is a difficult task to enable computers to recognize images of different objects.

- **Science**Daily: Computer vision is the science and technology of machines that see.

Tentative plan for lectures

Week	Content
47	Filtering, Linear algebra
48	Image formation, Surface reflection
49	Features, Descriptors, Matching
50	Photometric Stereo
51	Camera models, Homographies, CBIR
1	Stereo, Color Image Analysis
2	Segmentation
3	Convolutional Neural Nets

Schedule: When and where?

Lectures:

- Mondays 15:15 – 17:00 in Lille Aud., DIKU
- Wednesdays 10:15 – 12:00 Lille Aud., DIKU
- Monday 29/11: Lecture moved to AKB Aud 1

Exercises Wednesdays 13:15-15 all at NBI, Jagtvej 155:

- Team 1: 3.I, room 164
- Team 2: 3.H, room 142
- Team 3: 3.I, room 080

You will work individually and in groups with mandatory assignments. Remember to bring your laptop. Also remember paper and pencil.

Mandatory assignments

The course includes 6 mandatory assignments:

- A mix of theoretical and practical problems
- You have 1-2 weeks to solve each assignment
- The first assignment must be made individually. The remaining in groups
- Weekly on-line questionnaires to test your knowledge
- You can get help during the exercise class
- You are encouraged to use the discussion forum

Students Prerequisites

- This course is offered on the M.Sc. educations in It & Cognition and Computer Science.

We assume that you know:

- Programming at a basic level (either Python, Matlab, or C/C++) plus basic Linear Algebra and Statistics.

Be aware:

- You are a mixed crowd with different backgrounds!
- You might find some parts trivial and other parts difficult.

Relation to other DIKU courses

- **Signal and image processing (SIP)**, Q3
  - Extremely useful for CS-students, but not a requirement for this course. It is necessary for e.g. ATIA and for thesis work within IP & CV
- **Machine learning (ML)**, Q2
  - Mandatory course for CS-students
- **Introduction to Data Science (IDS)**, Q3
  - ML-course for non-CS-students
- **Medical image analysis (MIA)**, Q1
  - Related topic
- **Numerical optimization (NO)**, Q3
  - Foundation in optimization techniques
- **Advanced topics in machine learning (ATML)**, Q1
  - CS-Thesis preparation course
- **Advanced topics in Image Analysis (ATIA)**, Q1
  - CS-Thesis preparation course
- **Elements of Machine Learning** (bachelor), Q3

How do I pass this course?

- You have to pass **all** assignments in order to pass this course.
- Assignment 1 must be solved individually, but we encourage you to discuss it with your fellow students.
- In case you do not pass an assignment you will be given a second (but not a third...) chance to submit a new solution **assuming that you have made a SERIOUS attempt the first time**. This does **not** apply to the the last assignment.
- Final grading for the course is: Pass / Fail based on the assignments.

How much time should I spend on this course?

- KU expect that you to use 23 hours / week for a 7.5 ECTS course.
- How should I spend my time:
  - Lectures and exercise classes = 4 + 2 = 6 hours/wk
  - Preparation and assignment = 23 – 6 = 17 hours/wk

- We recommend that you prepare by reading the current weeks material and doing some research on your own, ideally prior to each lecture (approx. 8 hours/wk)
- Work on the assignment at home (approx. 9 hours/wk)

Course material

- We will use the textbook:
  - D. Forsyth and J. Ponce: *Computer Vision - A Modern Approach*. Pearson, 2.ed, 2012.
- Additional material will be made available in Absalon under the Course material menu item.

Literature for this week

Reading material for this week:

- Forsyth and Ponce: 1.1
- Tutorial on Linear Algebra
- (Forsyth and Ponce: 4.1)

- Please check the course plan at Absalon for the literature to next lectures.

➢ Relevant software

- Matlab is available from KUnet software library for installation on your laptop

- You can do well with Python using:
  - Numpy, Scipy, Scikit-image, Scikit-learn etc.
  - You will (probably) also need OpenCV
  - We recommend Python 3.7 or later. OpenCV is included in recent Anaconda Python packages.

- If you prefer C / C++, we recommend the libraries:
  - OpenCV
  - VLFeat

How to get help

- We use Absalon (access via your KUnet account)
  - You will find latest lecture plan
  - Links to lecture slides (usually after the lecture)
  - Course material (reading material)
  - Exercise material, and assignment upload
  - Mandatory questionnaires (when made public)
  - Links to additional material (reading, programming, etc.)
  - Discussion board in Absalon for course related topics

- Talk with the TAs at class or, if that fails, per e-mail.

Avoid Plagiarism

- You are not allowed to copy/steal other peoples work
- Copying from the textbook, from the net, or from a classmate is considered plagiarism.
- Plagiarism will be revealed, and students will have their study suspended or they will be expelled from the university.
- Please look at: https://en.wikipedia.org/wiki/Plagiarism

Examples of plagiarism

- Submitting someone's work as their own.
- Taking passages from their own previous work without adding citations (self-plagiarism).
- Re-writing someone's work without properly citing sources.
- Using quotations but not citing the source.
- Interweaving various sources together in the work without citing.
- Citing some, but not all, passages that should be cited.
- Melding together cited and uncited sections of the piece.
- Providing proper citations but failing to change the structure and wording of the borrowed ideas enough (close paraphrasing).
- Inaccurately citing a source.
- Relying too heavily on other people's work, failing to bring original thought into the text.

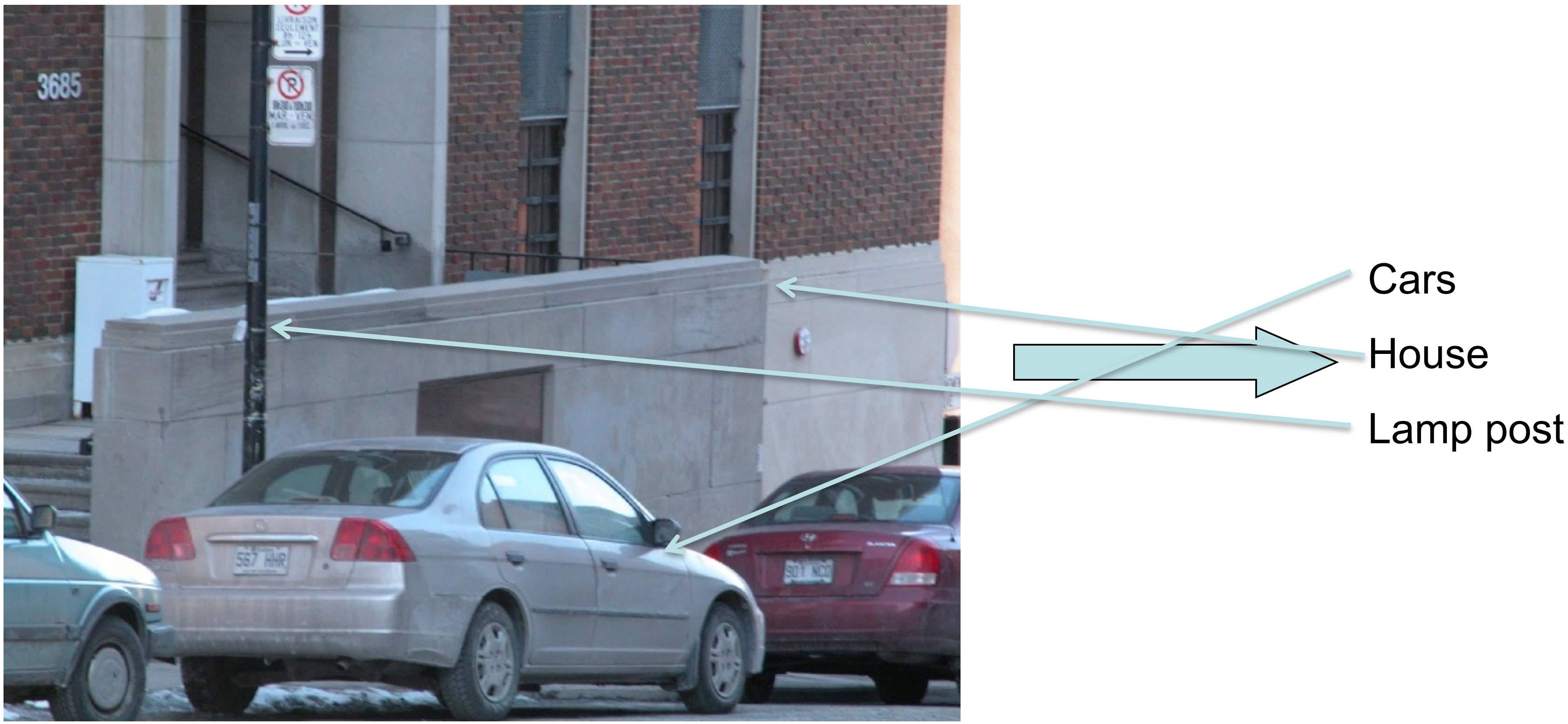
Questions ?

What do you see in this image?

- Low level cues: Texture, shading, shadows, occlusion boundaries, etc.
- High level interpretation: Objects, foreground-background, 3D perception, etc.
- And we did not include motion!

Example: Object recognition and detection

What is in this image? And where?

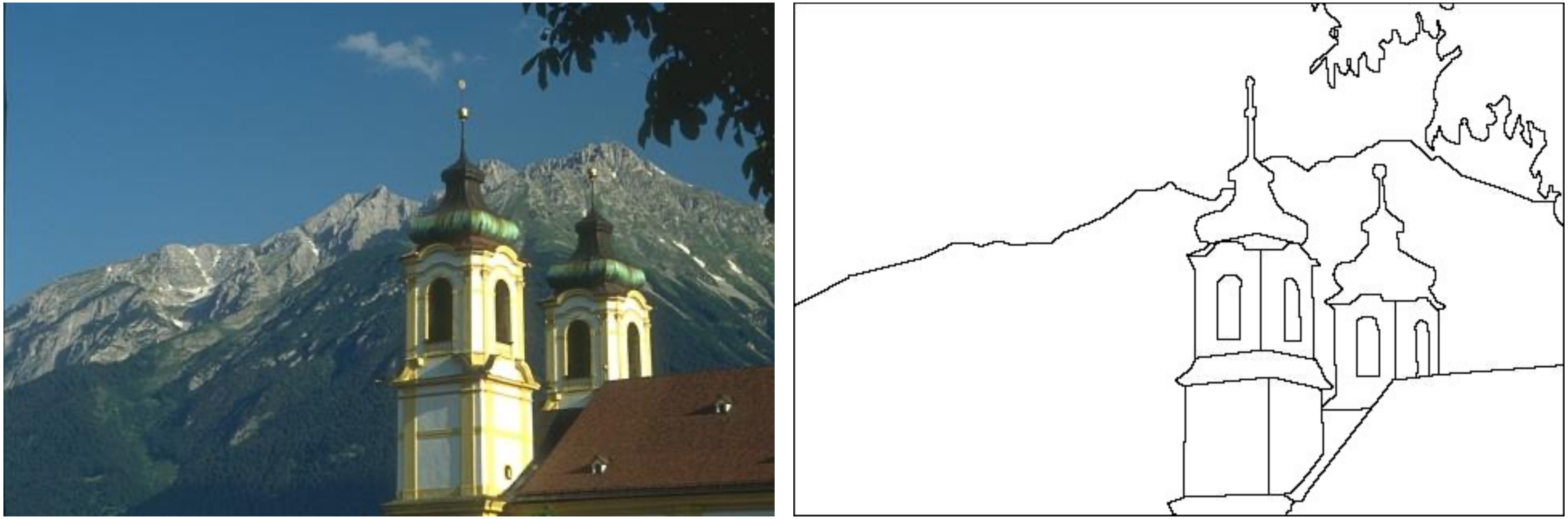


Example: Content-Based Image Retrieval
Please return all image that are similar to my query image

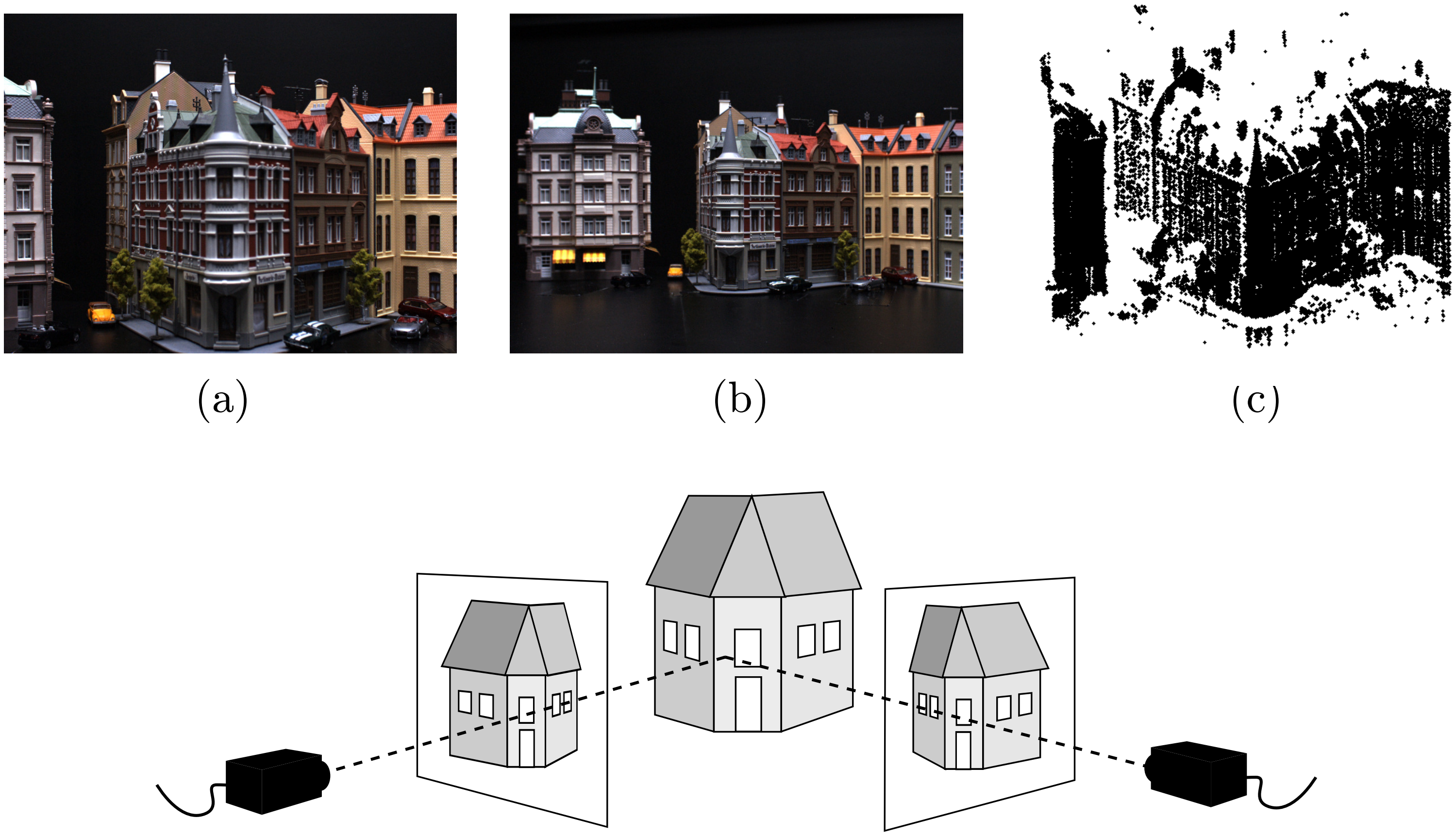
Query image

Search result

Example: Image Segmentation
Information on object boundaries and parts



Example: 3D reconstruction
Stereo or multi-view

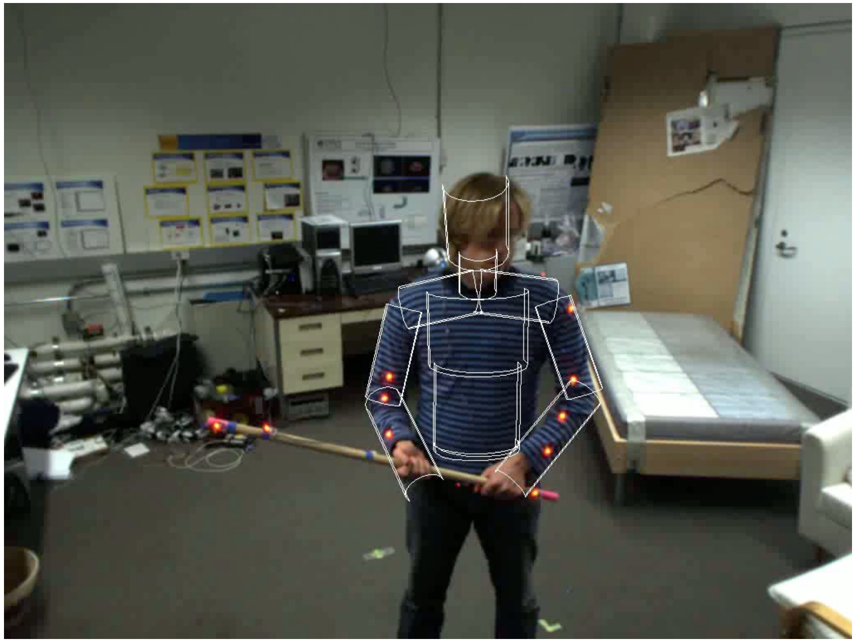


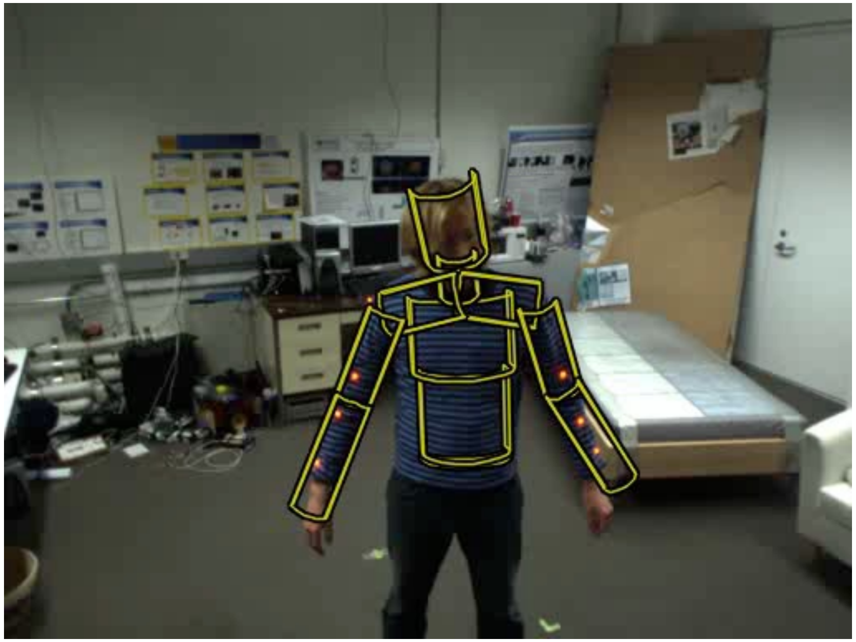
(a) (b) (c)

(d)

Example: 2D tracking of objects
Estimate an objects 2D trajectory in the image over time

Example: 3D articulated human tracking
Estimate 3D pose over time

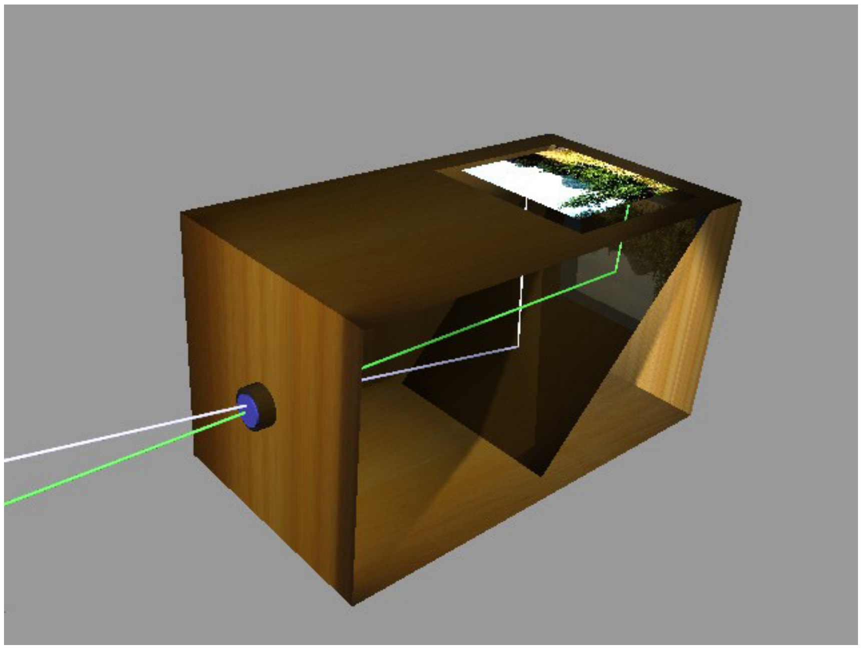




Lets start from the bottom and go up

Our input data consists of images – so what is an image?

Camera Obscura



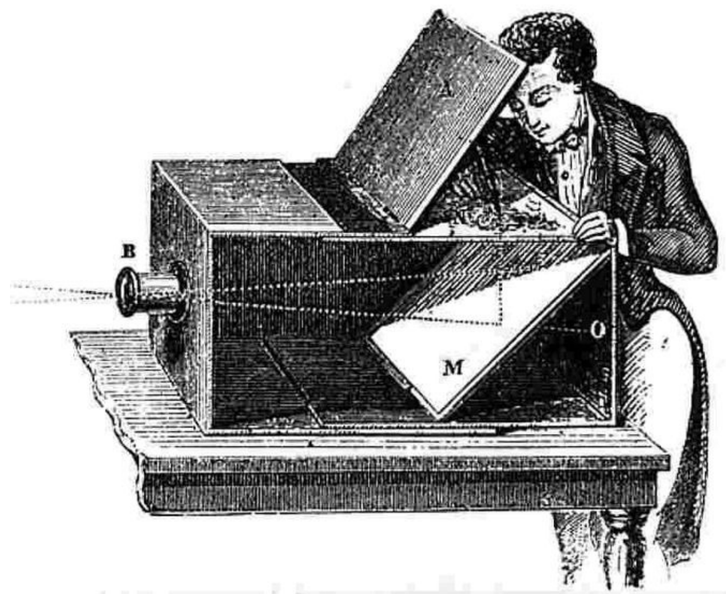


Image formation: The pinhole camera

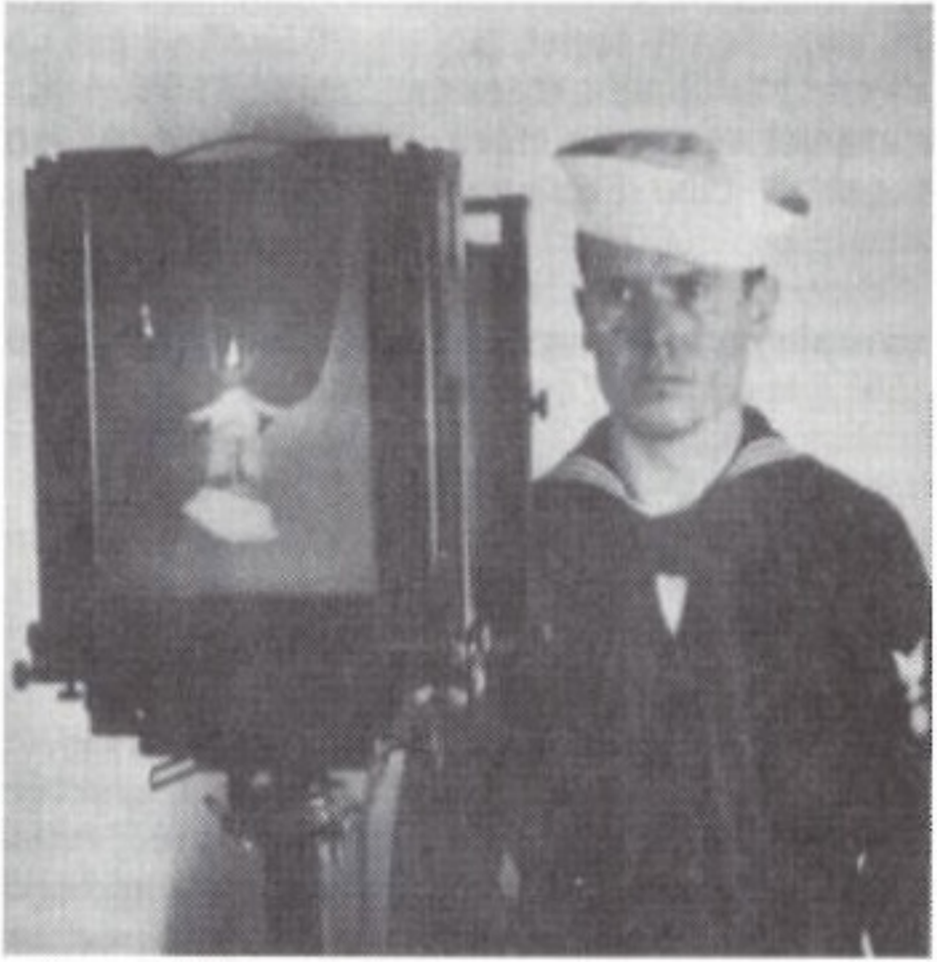
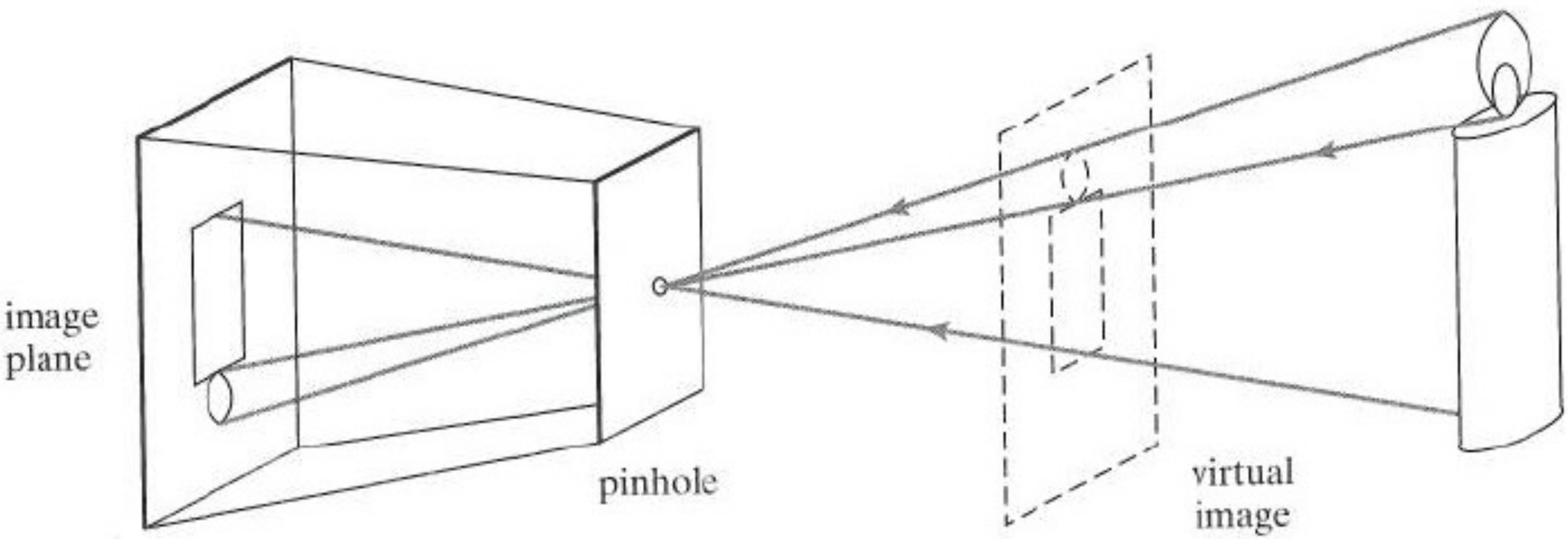


Image formation:
Camera obscura / pinhole camera



Notice: An image is a perspective projection of the 3D world onto a plane.

Image formation:
Some effects of perspective projection

Objects far away appear smaller than close by objects

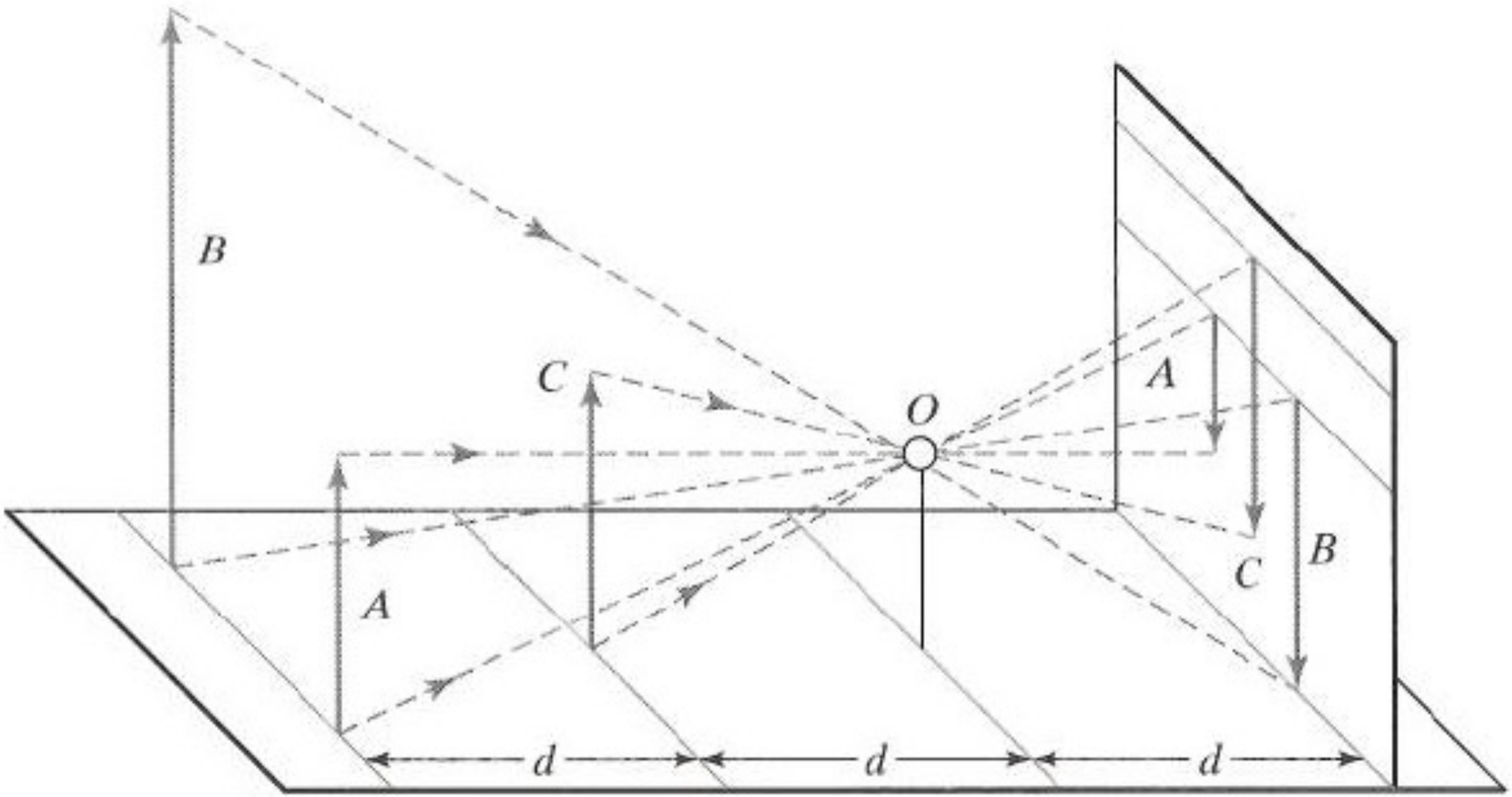


Image formation:
Some effects of perspective projection

Parallel lines appear to cross in the image plane (vanishing point)

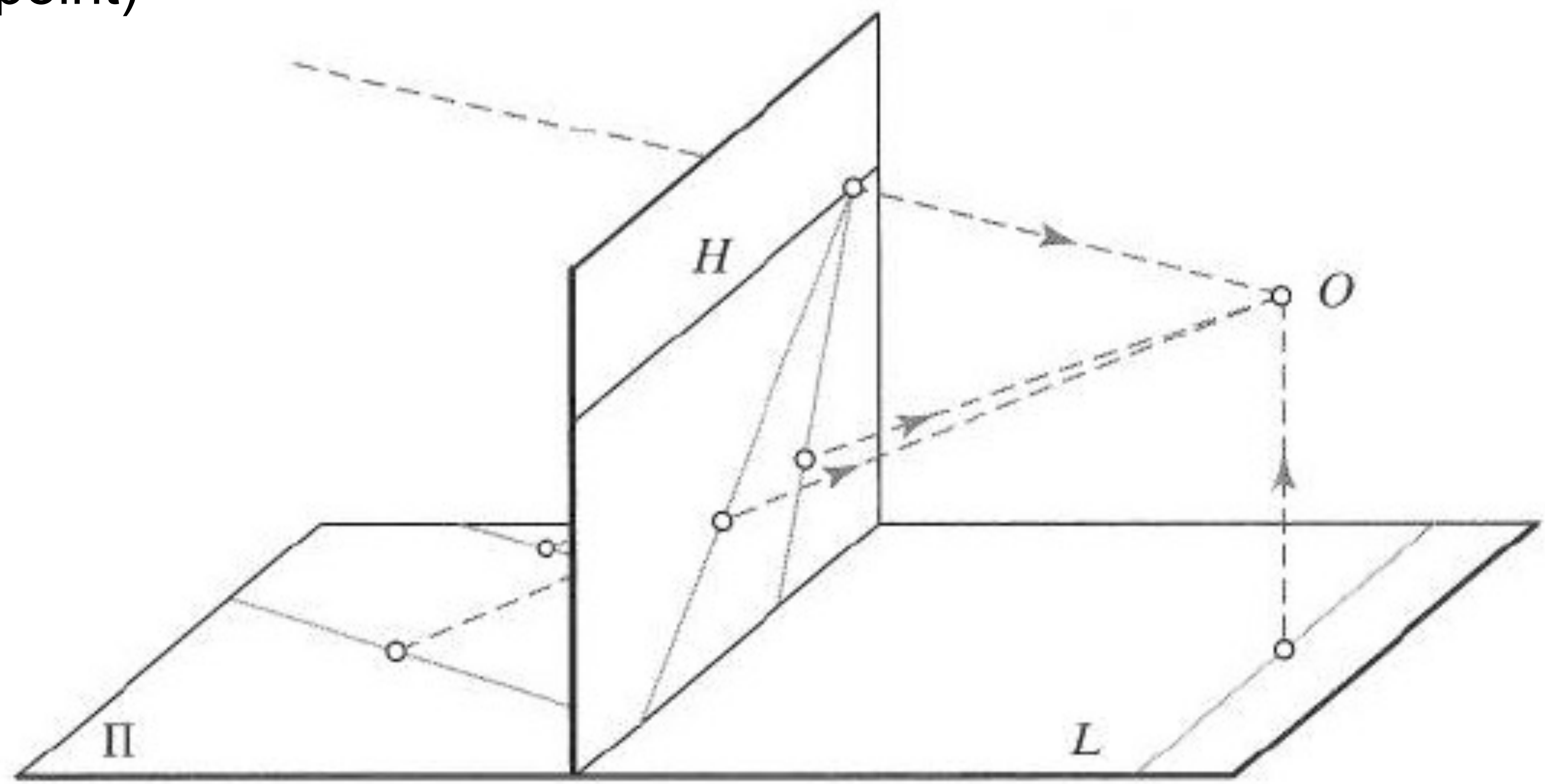
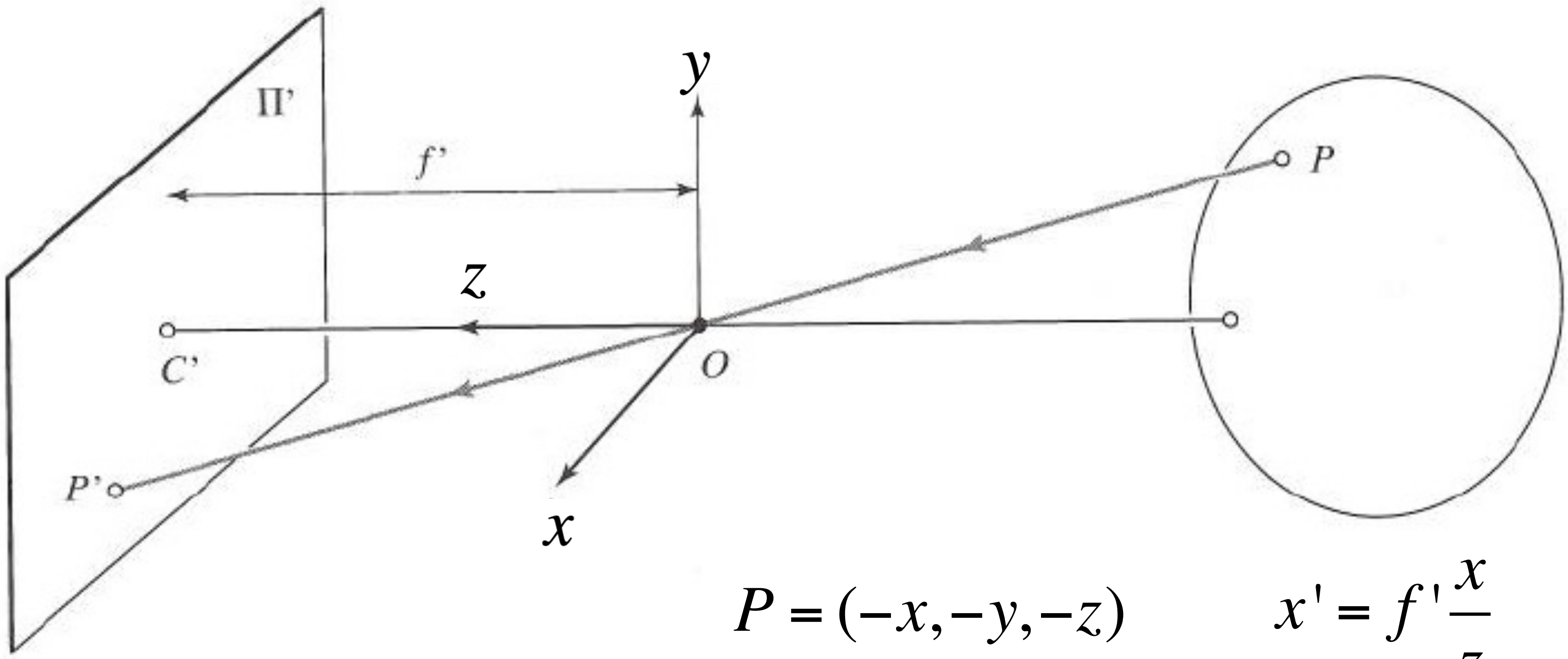


Image formation:
The pinhole camera model



$$P = (-x, -y, -z)$$

$$P' = (x', y', z')$$

$$C' = (0, 0, f')$$

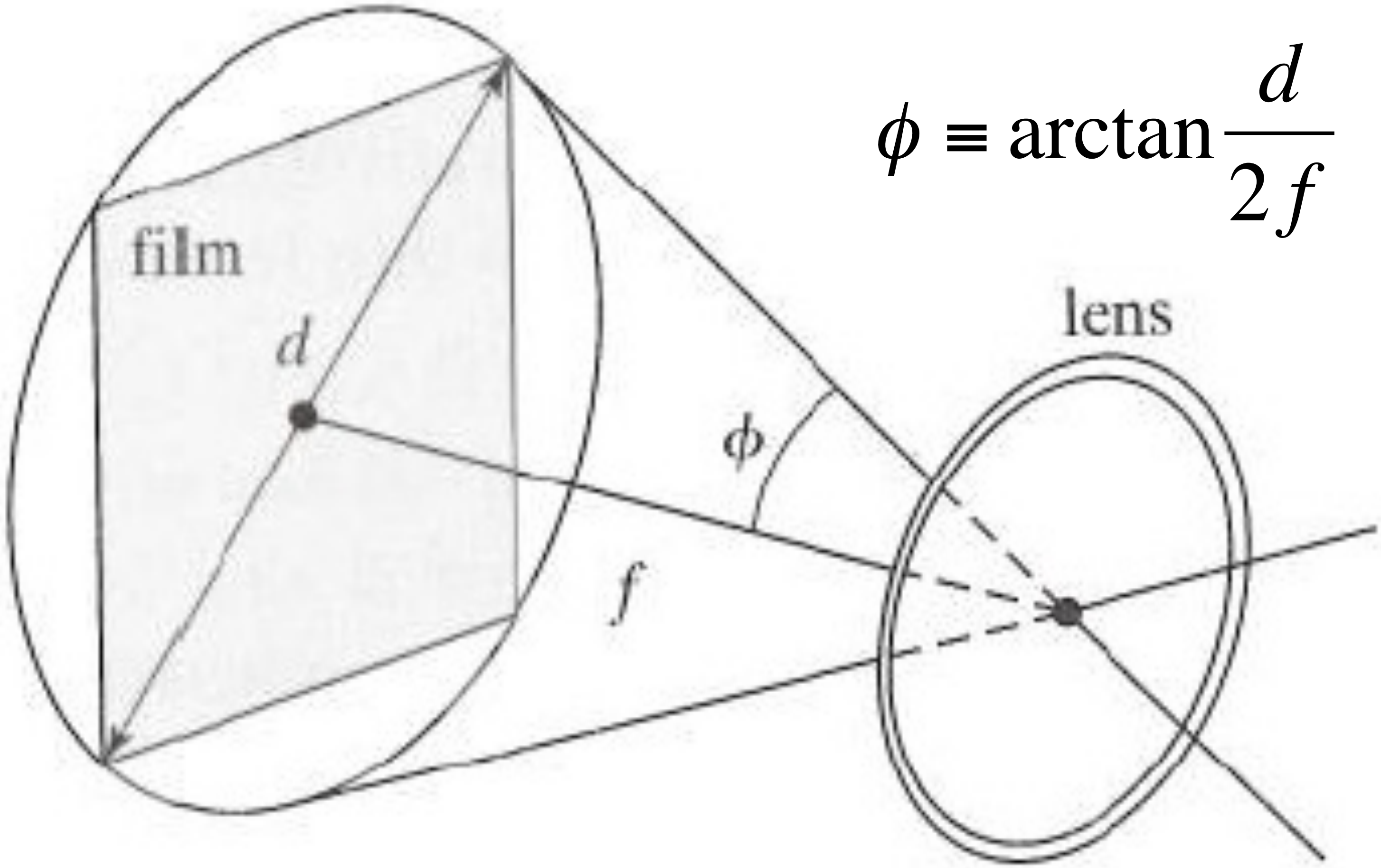
$$x' = f'\frac{x}{z}$$

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

$$z' = f'$$

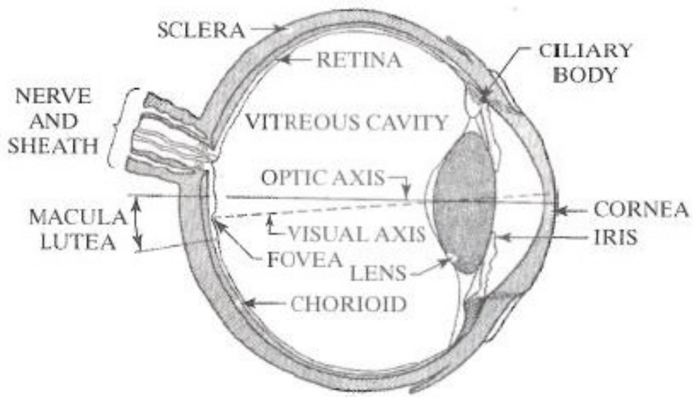
O : optical center / pinhole
k : optical axis
C' : image center/principal point
f' : focal length

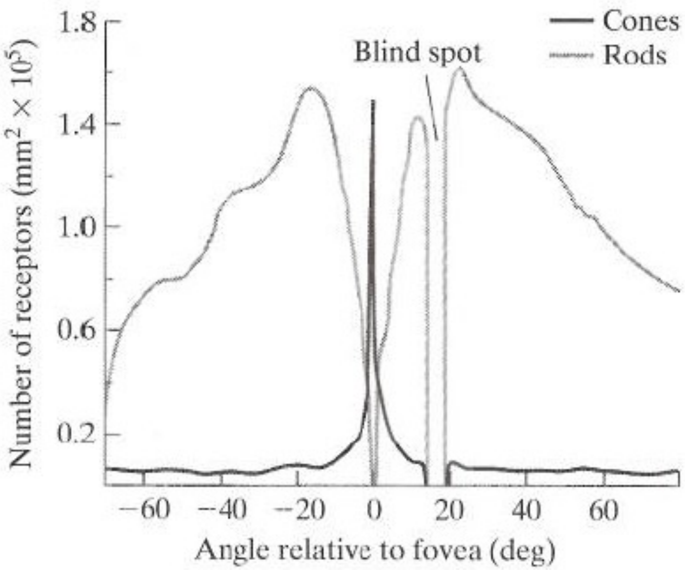
Image formation:
Field of view of the camera is 2ϕ



$$\phi \equiv \arctan\frac{d}{2f}$$

Image formation:
The human eye – a stereo camera





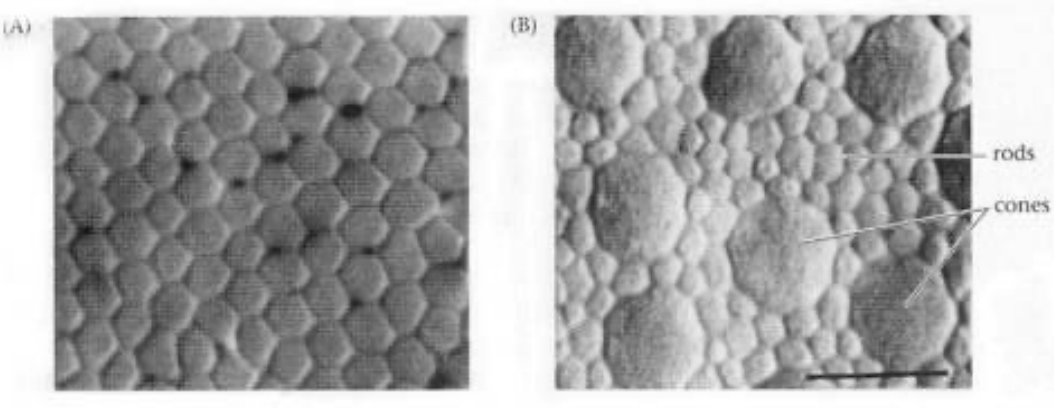


Image formation: The digital camera

- Most cameras use charge coupled device (CCD) chip

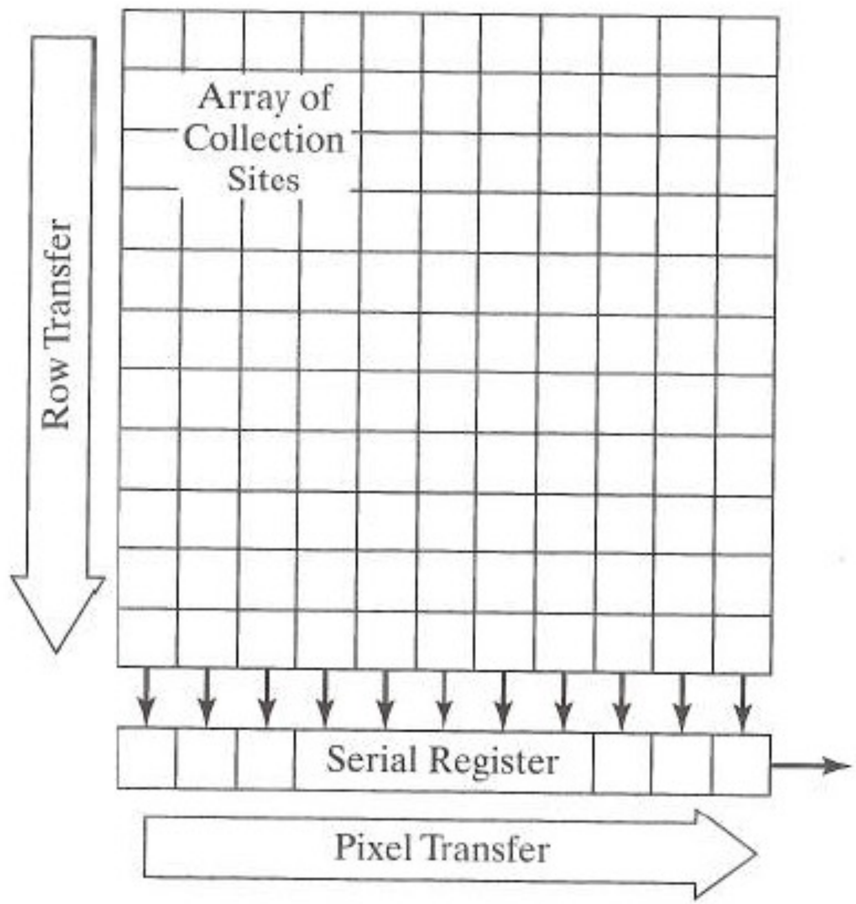
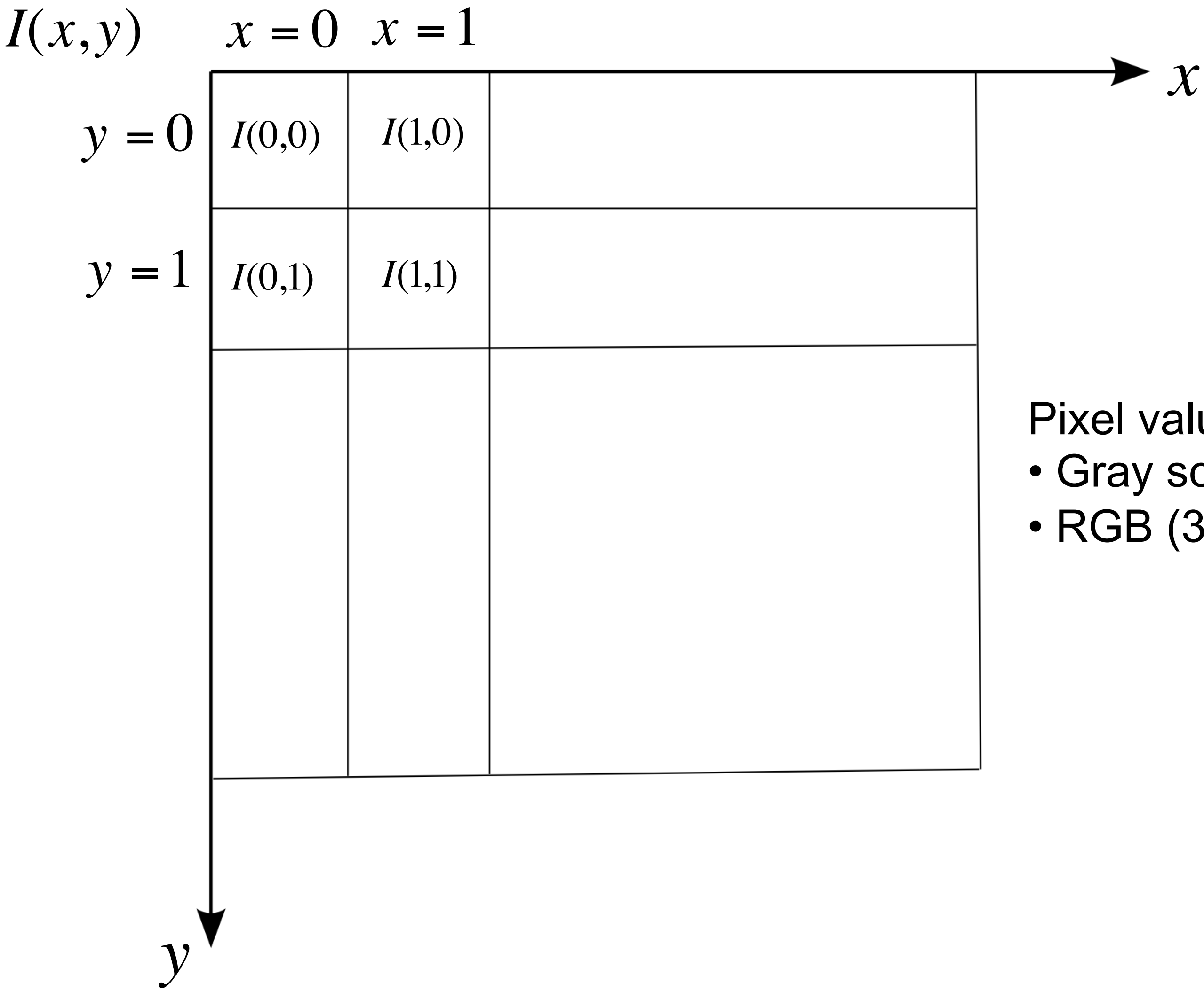
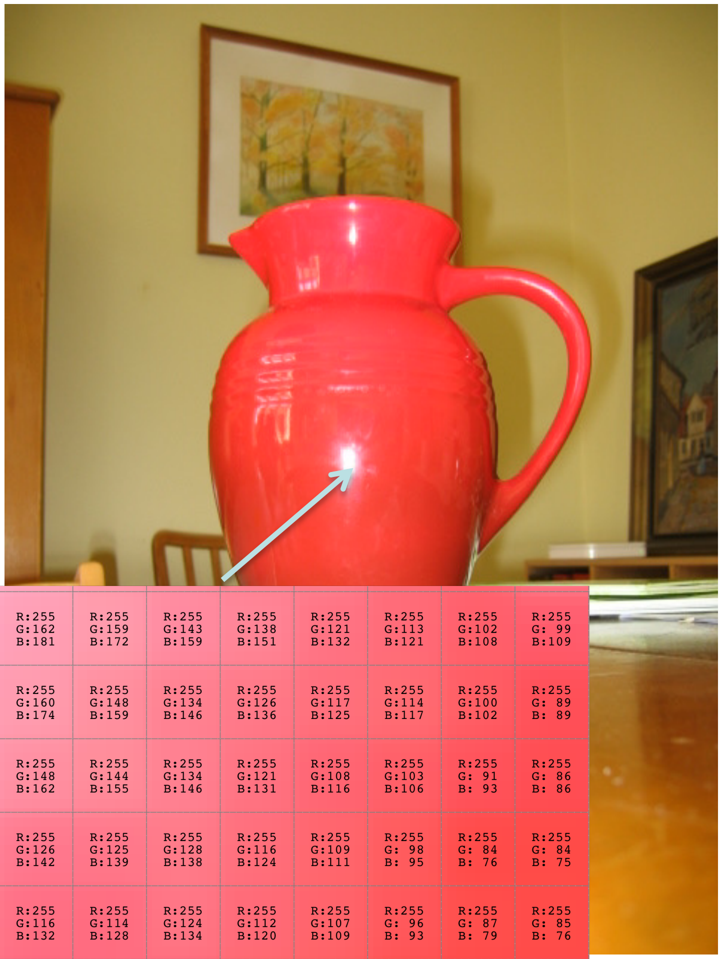


Image representation:
An array of pixels (picture elements)

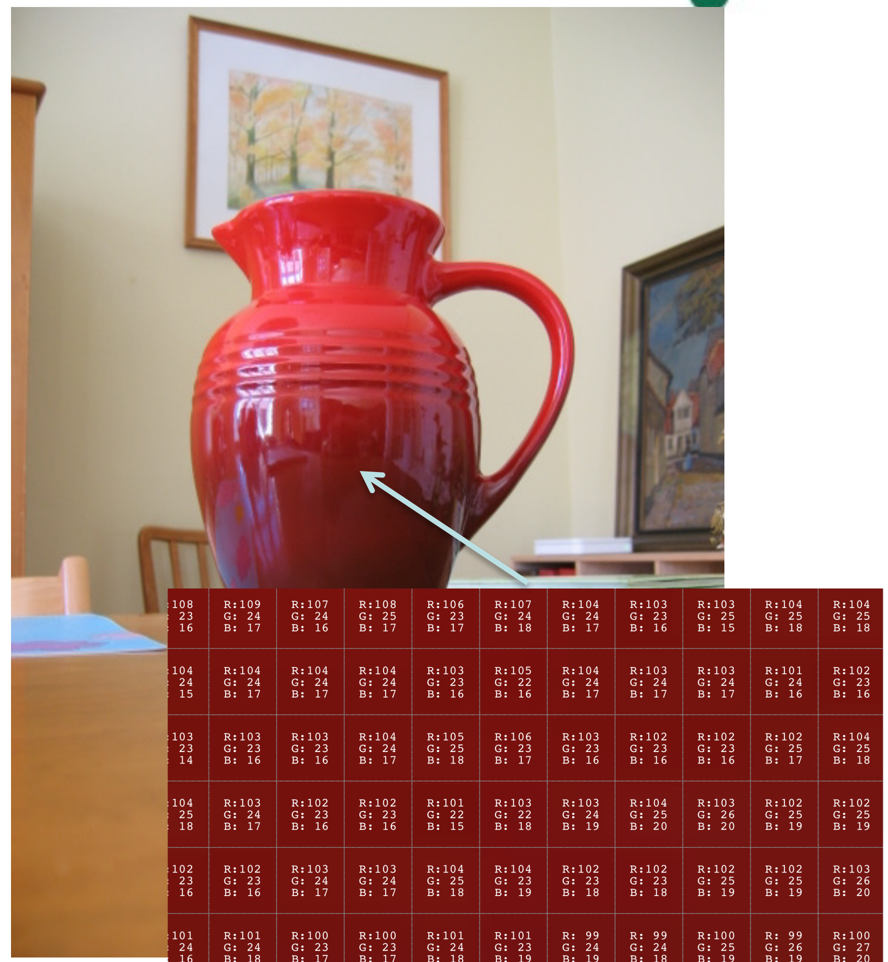


Pixel values:
- Gray scale (1 integer)
- RGB (3 integers vector)

Image representation:
An array of pixels (picture elements)



R:255 G:213 B:227	R:255 G:170 B:194	R:255 G:162 B:181	R:255 G:159 B:172	R:255 G:143 B:159	R:255 G:138 B:151	R:255 G:121 B:132	R:255 G:113 B:121	R:255 G:102 B:108	R:255 G: 99 B:109
R:255 G:190 B:206	R:255 G:162 B:181	R:255 G:160 B:174	R:255 G:148 B:159	R:255 G:134 B:146	R:255 G:126 B:136	R:255 G:117 B:125	R:255 G:114 B:117	R:255 G:100 B:102	R:255 G: 89 B: 89
R:255 G:187 B:203	R:255 G:158 B:177	R:255 G:148 B:162	R:255 G:144 B:155	R:255 G:134 B:146	R:255 G:121 B:131	R:255 G:108 B:116	R:255 G:103 B:106	R:255 G: 91 B: 93	R:255 G: 86 B: 86
R:255 G:165 B:180	R:255 G:139 B:154	R:255 G:126 B:142	R:255 G:125 B:139	R:255 G:128 B:138	R:255 G:116 B:124	R:255 G:109 B:111	R:255 G: 98 B: 95	R:255 G: 84 B: 76	R:255 G: 84 B: 75
R:255 G:148 B:163	R:255 G:130 B:145	R:255 G:116 B:132	R:255 G:114 B:128	R:255 G:124 B:134	R:255 G:112 B:120	R:255 G:107 B:109	R:255 G: 96 B: 93	R:255 G: 87 B: 79	R:255 G: 85 B: 76
R:255 G:131 B:147	R:255 G:121 B:134	R:255 G:111 B:125	R:255 G:114 B:126	R:255 G:113 B:123	R:255 G:110 B:108	R:255 G:104 B: 97	R:255 G: 87 B: 77	R:255 G: 83 B: 76	R:255 G: 82 B: 78
R:255 G:129 B:145	R:255 G:119 B:132	R:255 G:113 B:127	R:255 G:112 B:124	R:255 G:108 B:118	R:255 G:103 B:101	R:255 G: 94 B: 87	R:255 G: 85 B: 75	R:255 G: 84 B: 77	R:255 G: 77 B: 73

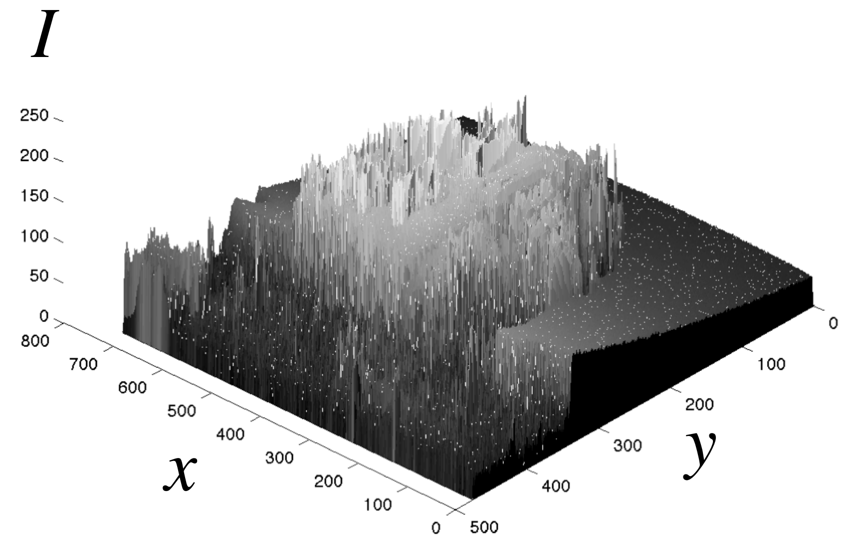


108 23 16	R:109 G: 24 B: 17	R:107 G: 24 B: 16	R:108 G: 25 B: 17	R:106 G: 23 B: 17	R:107 G: 24 B: 18	R:104 G: 24 B: 17	R:103 G: 23 B: 16	R:103 G: 25 B: 15	R:104 G: 25 B: 18	R:104 G: 25 B: 18
104 24 15	R:104 G: 24 B: 17	R:104 G: 24 B: 17	R:104 G: 24 B: 17	R:103 G: 23 B: 16	R:105 G: 22 B: 16	R:104 G: 24 B: 17	R:103 G: 24 B: 17	R:103 G: 24 B: 17	R:101 G: 24 B: 16	R:102 G: 23 B: 16
103 23 14	R:103 G: 23 B: 16	R:103 G: 23 B: 16	R:104 G: 24 B: 17	R:105 G: 25 B: 18	R:106 G: 23 B: 17	R:103 G: 23 B: 16	R:102 G: 23 B: 16	R:102 G: 23 B: 16	R:102 G: 25 B: 17	R:104 G: 25 B: 18
104 25 18	R:103 G: 24 B: 17	R:102 G: 23 B: 16	R:102 G: 23 B: 16	R:101 G: 22 B: 15	R:103 G: 22 B: 18	R:103 G: 24 B: 19	R:104 G: 25 B: 20	R:103 G: 26 B: 20	R:102 G: 25 B: 19	R:102 G: 25 B: 19
102 23 16	R:102 G: 23 B: 16	R:103 G: 24 B: 17	R:103 G: 24 B: 17	R:104 G: 25 B: 18	R:104 G: 23 B: 19	R:102 G: 23 B: 18	R:102 G: 23 B: 18	R:102 G: 25 B: 19	R:102 G: 25 B: 19	R:103 G: 26 B: 20
101 24 16	R:101 G: 24 B: 18	R:100 G: 23 B: 17	R:100 G: 23 B: 17	R:101 G: 24 B: 18	R:101 G: 23 B: 19	R: 99 G: 24 B: 19	R: 99 G: 24 B: 18	R:100 G: 25 B: 19	R: 99 G: 26 B: 19	R:100 G: 27 B: 20
102 25 17	R:102 G: 25 B: 19	R:101 G: 24 B: 18	R:101 G: 24 B: 18	R:101 G: 24 B: 18	R:100 G: 22 B: 18	R: 99 G: 24 B: 19	R:100 G: 25 B: 19	R:100 G: 25 B: 19	R: 98 G: 25 B: 18	R: 98 G: 25 B: 18
100 23 15	R: 98 G: 23 B: 17	R: 98 G: 23 B: 17	R:101 G: 24 B: 18	R:102 G: 25 B: 19	R: 98 G: 25 B: 19	R: 98 G: 25 B: 19	R: 97 G: 24 B: 18	R: 98 G: 25 B: 19	R: 97 G: 26 B: 20	R: 98 G: 27 B: 21

Viewing images as functions

$$I(x,y)$$

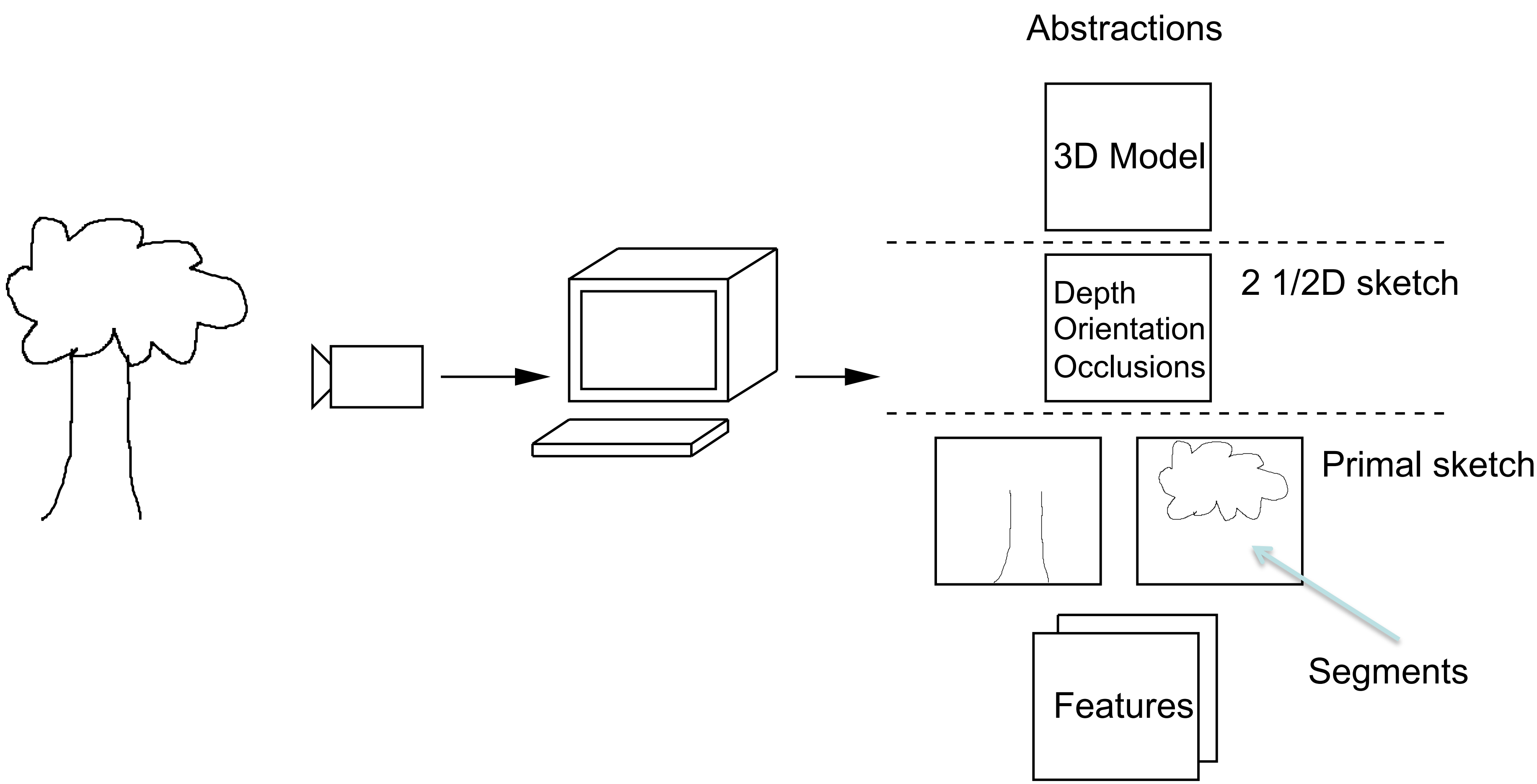
$$I(x,y)$$



A bunch of numbers in a table is not enough!

Analysis and abstraction is needed

Image representation:
Marr's layers of abstraction (1982)



Mandatory and individual assignment 1:

Training in

- Linear Algebra (vectors, matrices etc)
- Differential calculus (derivatives)
- Python programming

- You should know everything of this in advance, but you may have forgot or may have not operative training.

Mandatory assignment 2: Filtering and edge detection

- Get acquainted with basic filtering
- Extraction of a primitive feature – edges.
- Assignment text available at Absalon from next Monday

Filtering and edges

Basic image processing

Linear position invariant filtering (aka convolutions) is the most frequent image processing operation in both classical vision and recent neural networks.

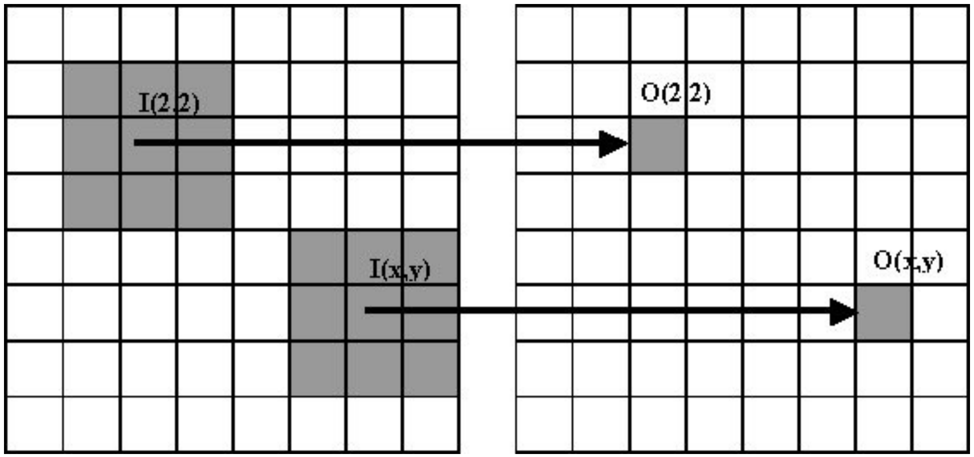
What is filtering good for ?

1. Noise reduction
2. Elimination of image quantization effects and conversion from integers to floats
3. Anti aliasing when up/down-sampling
4. Estimation of image derivatives
5. Enhancement of specific structures
6. Detection of image features such as edges and corners

Filtrering

- Almost all filtering is local, i.e. the filter response in a pixel is computed based on an image patch centered at the pixel.

- Almost all filters are linear and position invariant, implying that the filtering may be described by a convolution.



Convolutions

- Filtering by discrete convolution
  - A filter is defined by a filter kernel *h*(*x,y*)
  - Filtering the image *I*(*x,y*) with the kernel *h*(*x,y*) is defined as

$$R(x,y) = \sum_{u,v} I(u,v)h(x-u,y-v) \equiv I(x,y) * h(x,y)$$

- Consider filter kernels as images
  - They do not have to be of the same size as the image
  - Filtering slides the mirrored filter kernel across the image and compute the product sum at each location
  - The result R(x,y) is called the filter response
  - *R(x,y)* is an image of same size as the original image *I*

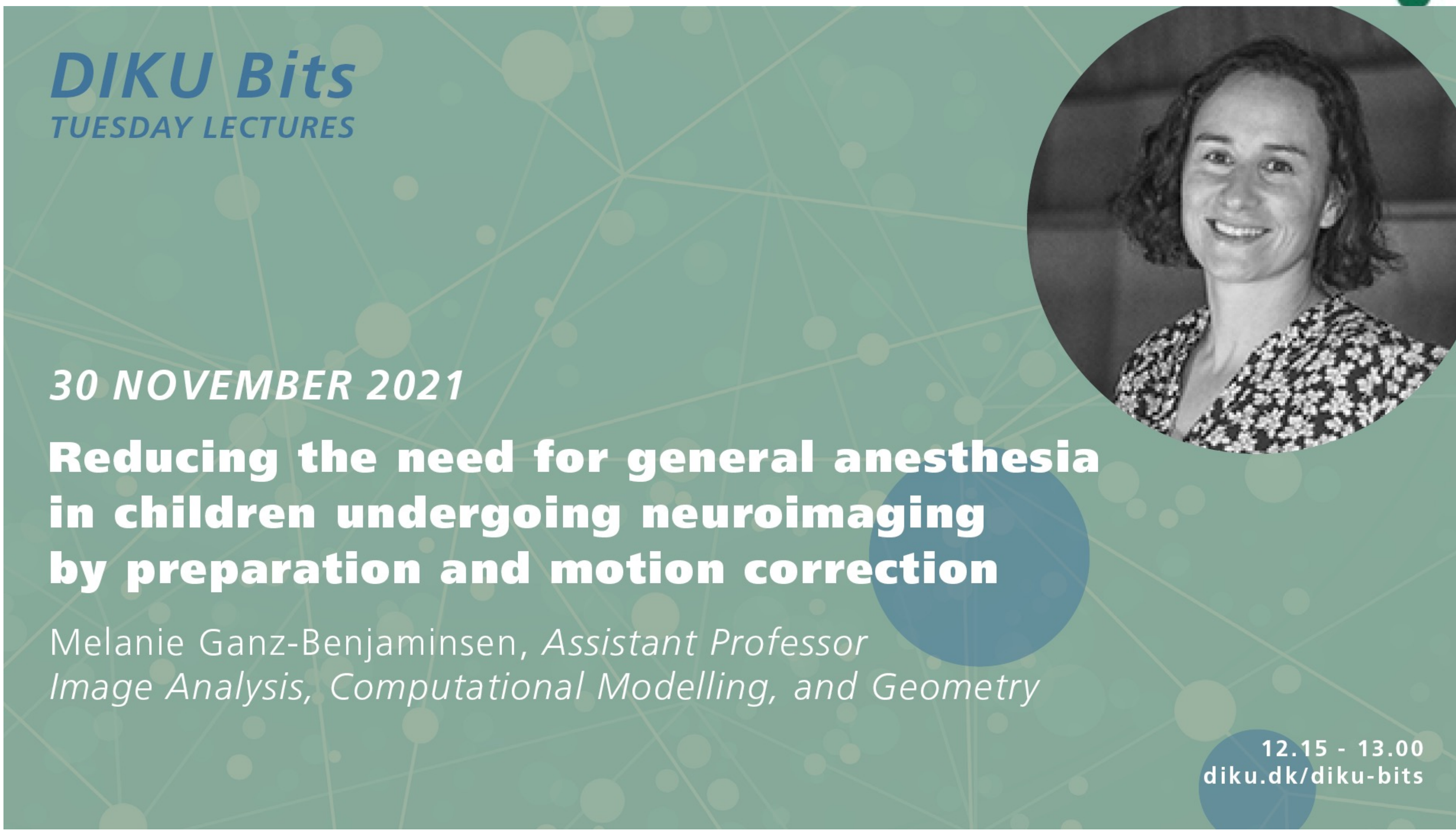
Summary

- Examples of computer vision problems
- Image formation:
  - Pinhole camera model
  - The human eye is a "camera"
- Image representation:
  - At the bottom: Digital images as an array of pixels
  - Pixel values are difficult to interpret
- Intro to filtering by convolutions

Wednesday, you will refresh your knowledge on linear algebra and learn new tricks enabling non-trivial computer vision

Remember paper, pencil and laptop for your TA-session

Repeated announcement



Color, Shading and Shadows also concerns us

(a)

(b)

(c)

(d)