DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF COPENHAGEN

Vision and Image Processing: Introductory lecture

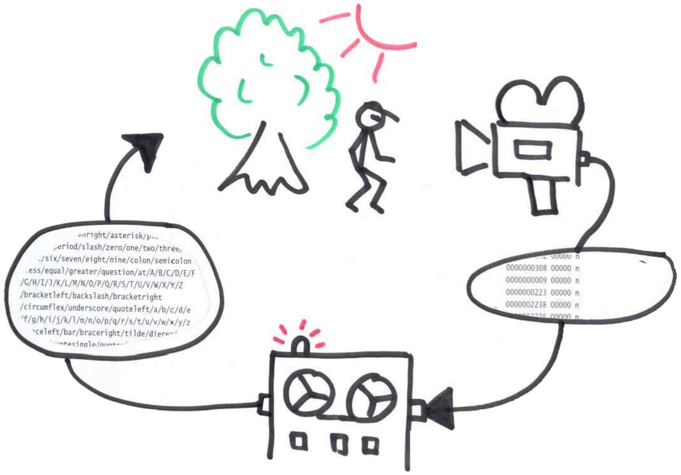
Søren Ingvor Olsen

The teachers

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

What is this course about?

Computer Vision



Making computers see

Definitions of Computer Vision

- **Kims definition:** The design of algorithms for interpreting visual data by mimicking the human visual perceptual system.

Alternatives:

- **McGraw-Hill Science & Technology Dict.:** The use of digital computer techniques to extract, characterize, and interpret information in visual images of a three-dimensional world.
- **Wikipedia:** As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner.
- **thefreedictionary.com:** Field of robotics in which programs attempt to identify objects represented in digitized images provided by video cameras, thus enabling robots to "see."

Tentative plan for lectures

Week	Content
47	Filtering, Linear algebra
48	Image processing, Features
49	Descriptors, Basic statistics
50	Content-Based Image Retrieval (CBIR), Segmentation
51	Cameras, projections, Image formation, Photometric stereo
1	Stereo Vision, 3D reconstructions
2	Tracking

Schedule: When and where?

Lectures:

- Mondays 13:15 – 15:00 in Large Auditorium [NEXT] (Nørre allé 53)

- Wednesdays 10:15 – 12:00 in Auditorium 2 [HCØ] (Universitetsparken 5)

Exercises:

- Room A107 (team 1) and A110 (team 2) at HCØ. TA will multiplex. You will work individually and in groups with mandatory assignments. Remember to bring your laptop.

Mandatory assignments

The course includes 4 mandatory assignments:

- A mix of theoretical and practical problems
- You have about two weeks to solve each of them
- The necessary theory will be presented at lectures
- The solutions must be made individually and some in groups
- You can get help during the lectures and 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++)

Be aware:

• You are a mixed crowd with different backgrounds!
• There might be parts you will find trivial and other parts you won't.

Relation to other DIKU courses

- **Signal and image processing**, Q3
  - Extremely useful for CS-students, but not a requirement for this course
- **Machine learning**, Q2
  - Excellent companion course for CS-students
- **Data analysis methods**, Q3
  - New ML-course for non-CS-students
- **Medical image analysis**, Q1
  - Related topic
- **Numerical optimization**, Q3
  - Foundation in optimization techniques
- **Advanced topics in Machine learning** , Q1
  - CS-Thesis preparation course
- **Advanced topics in Image Analysis**, Q2
  - CS-Thesis preparation course

How do I pass this course?

- You have to pass the 4 mandatory assignments in order to pass this course.
- Assignment 1 must be solved individually, but we encourage you to discuss it with your fellow students.
- Other assignments are either group or individual.
- In case you do not pass an assignment the first time you will be given a second chance to submit a new solution **assuming that you have made a SERIOUS attempt the first time**.
- 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. Approx. 46 hours/wk for full time study.

- 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 a mix of research papers and chapters from selected books.

- All material will be made available in Absalon under the Course material menu item.
- Supplementary good books (available at the book shop)

1. D. Forsyth and J. Ponce: Computer Vision - A Modern Approach. Pearson, 2.ed, 2012.
2. E. Solem: Programming Computer Vision with Python, O'Reilly, 2012.

Literature for today

Reading material for today:

- Forsyth and Ponce: Ch. 1

Additional material:

- David Marr: Vision. W. H. Freeman & Company, 1982

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

➢ Relevant software

- Matlab is available from KUnet software library for installation on your laptop
- You can do well with Python using:
  - Numpy, Scipy, etc.
  - You will also need openCV
- If you prefer C / C++ (or Python), 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
  - Links to additional material (reading, programming, etc.)

- Discussion board in Absalon for course related topics

- Talk with the TA/teachers at class, per e-mail, or try to catch us at our offices

Enough about the formalities!

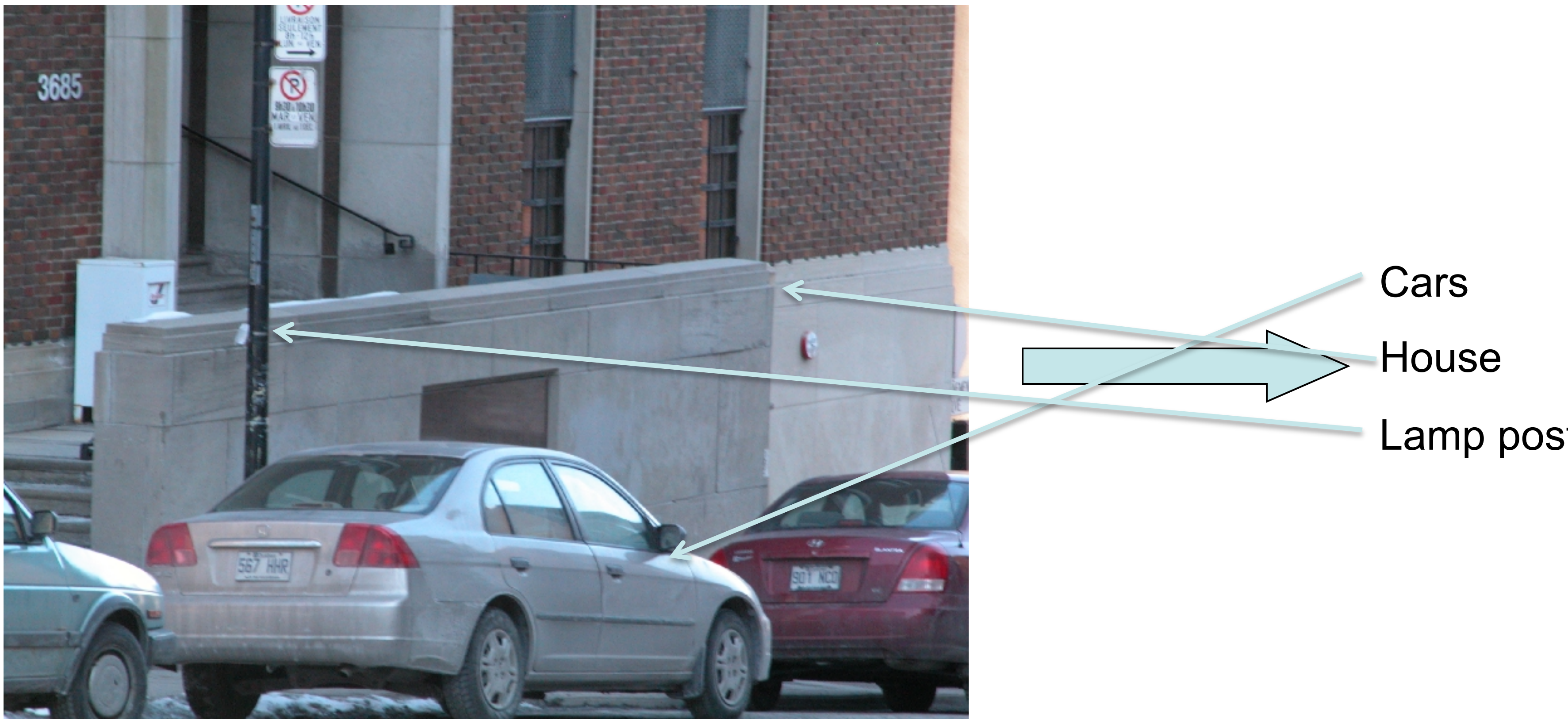
What do you see in this image?

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

A couple of examples of interesting problems in Computer Vision

Example: Object recognition and detection

What is in this image? And where?

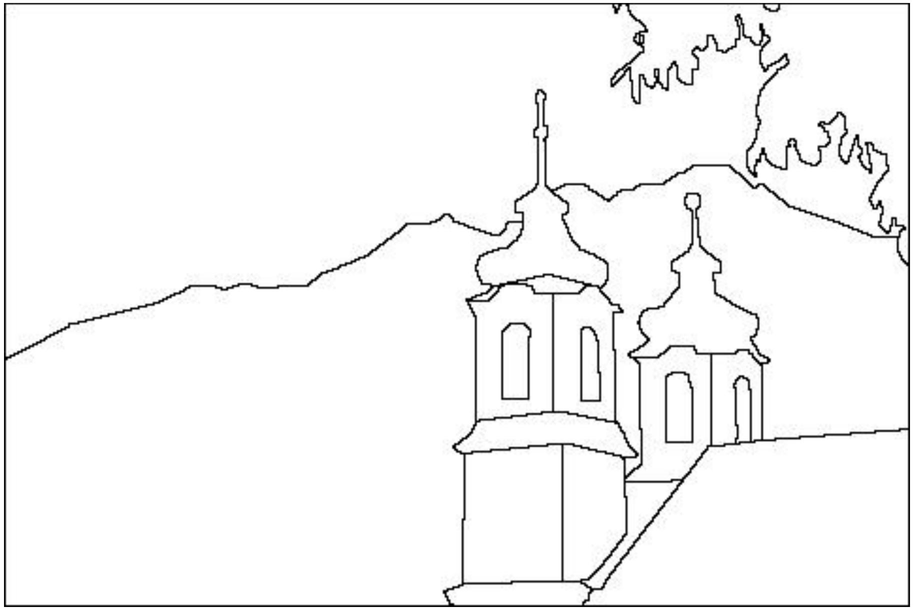


Example: Content-Based Image Retrieval
Please return all image that are similar to my query image

Search result

Query image

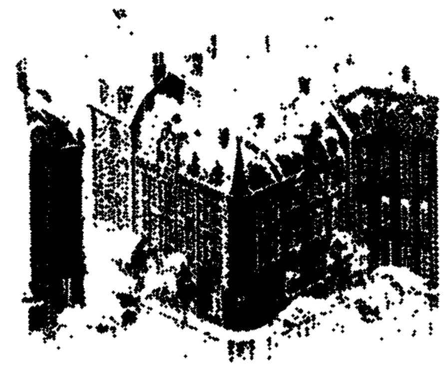
Example: Image Segmentation
Information on object boundaries and parts



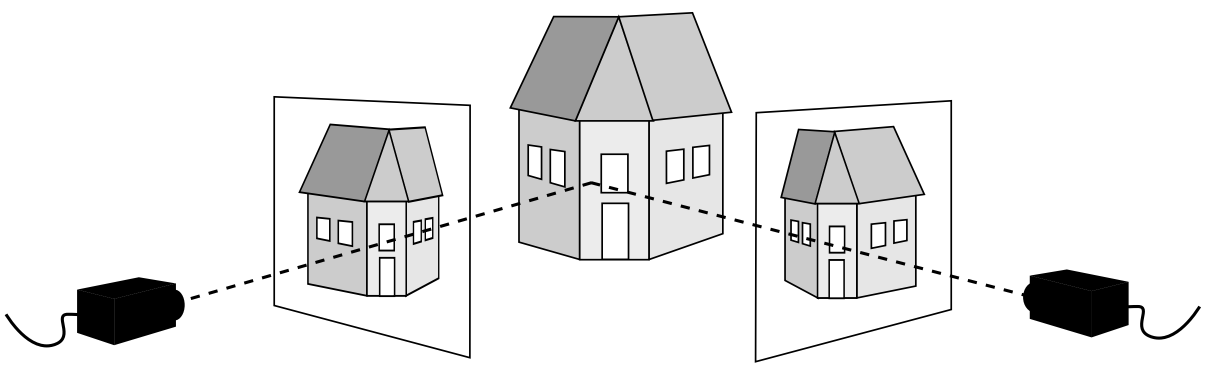
Example: 3D reconstruction
Stereo or multi-view

(a)

(b)



(c)

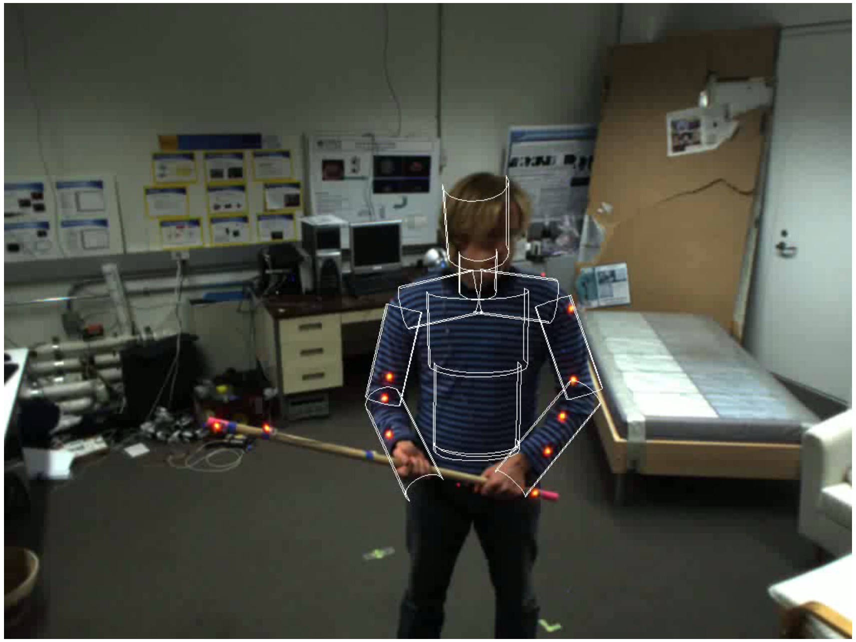


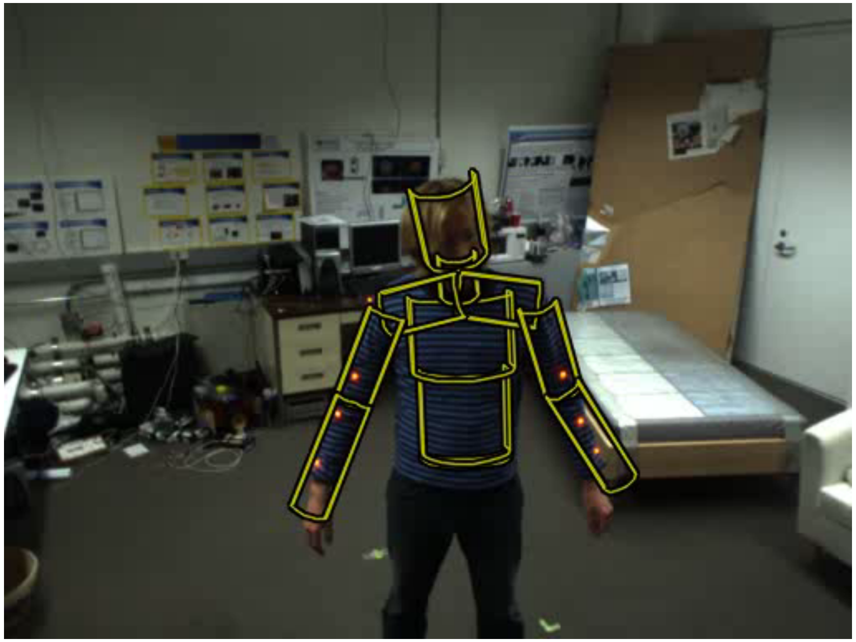
(d)

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?

Image formation:
Camera obscura / pinhole camera

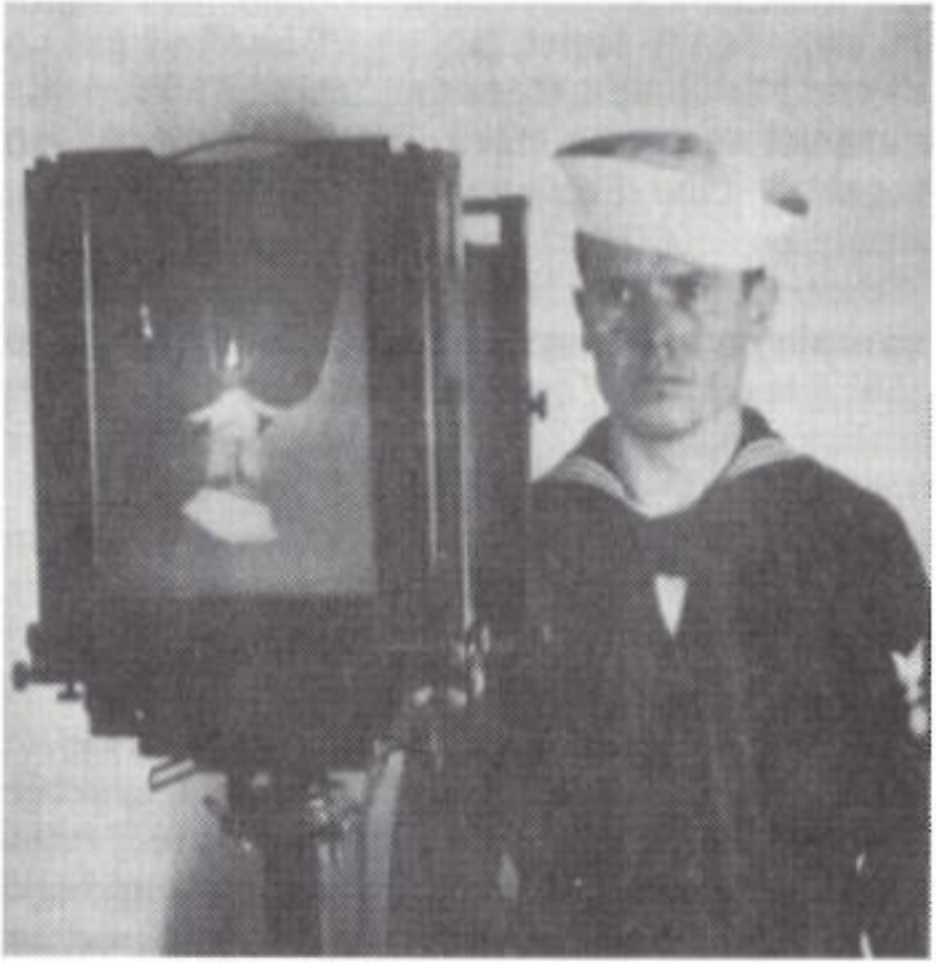
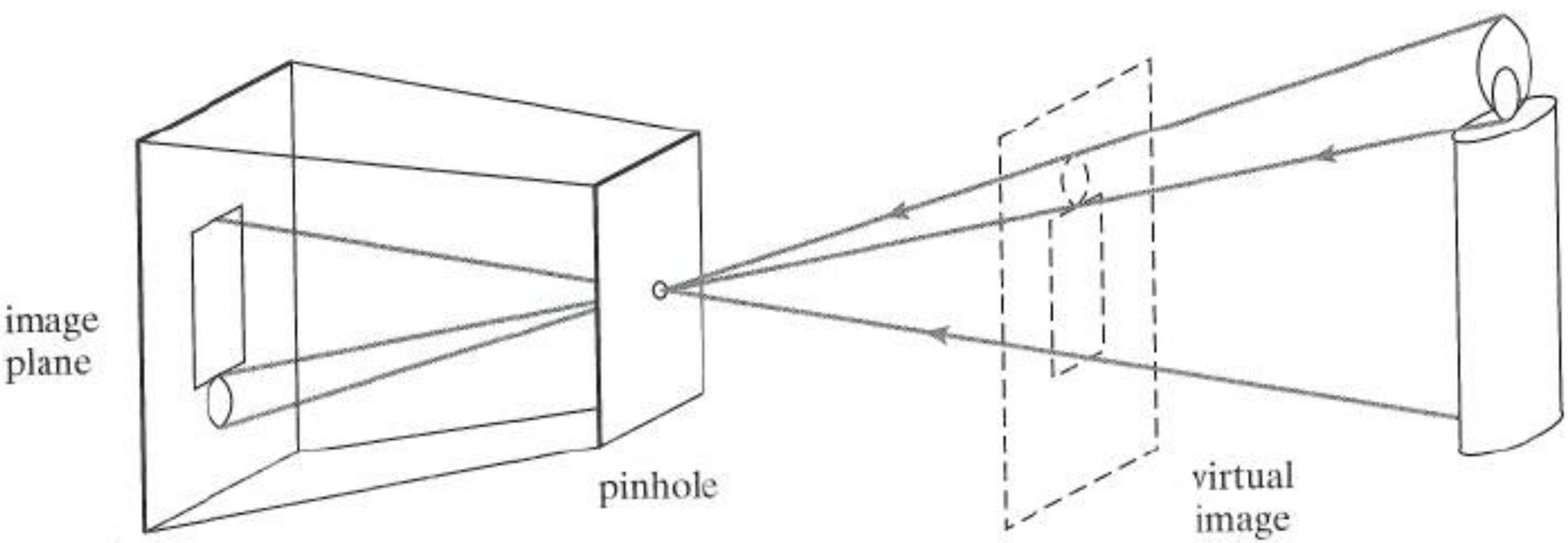


Image formation:
Camera obscura / pinhole camera



Notice: An image is a 2D projection of the 3D world through a perspective projection

Image formation:
Some effects of perspective projection

Objects far away appear smaller than close by objects

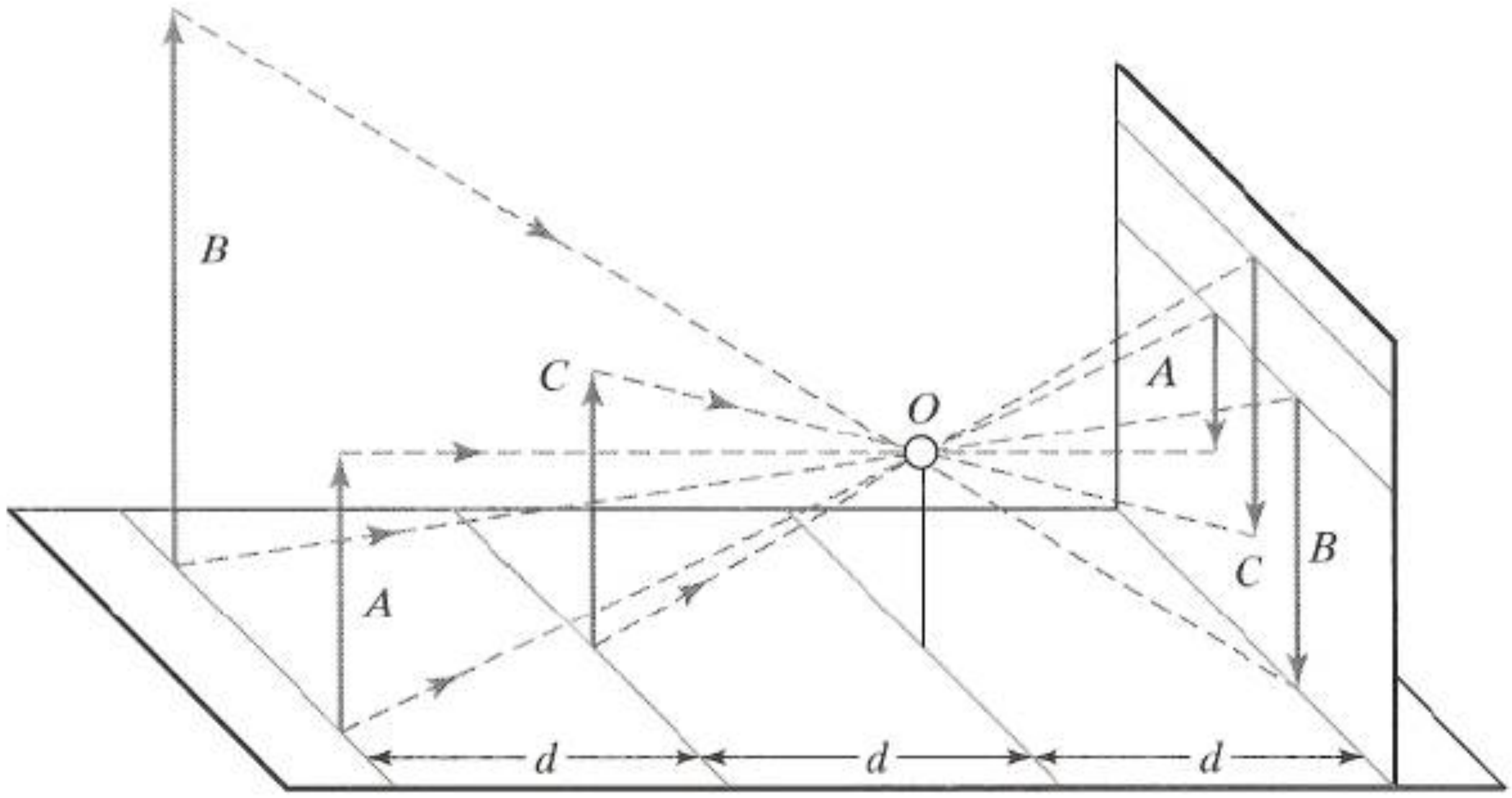


Image formation:
Some effects of perspective projection

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

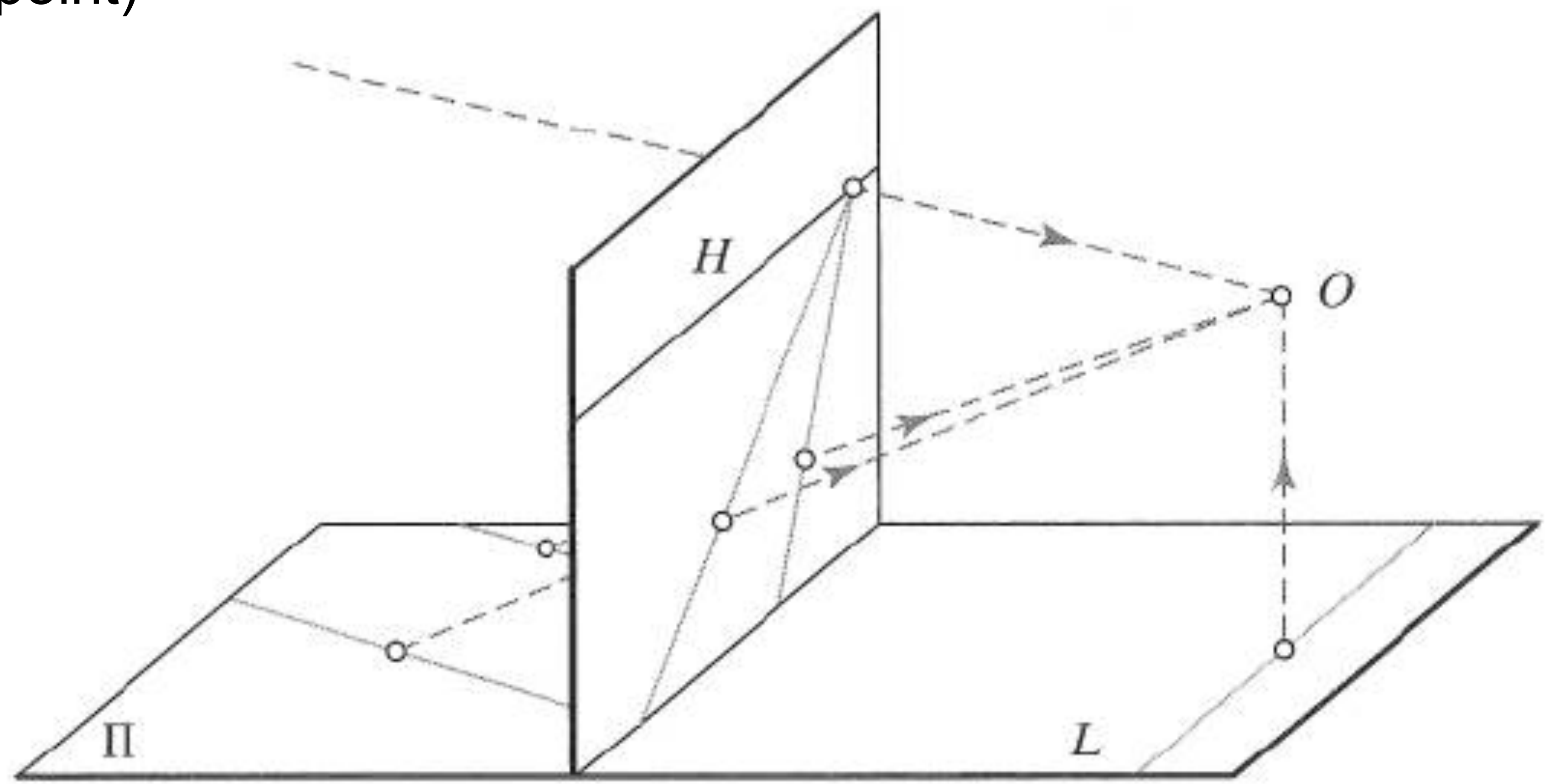
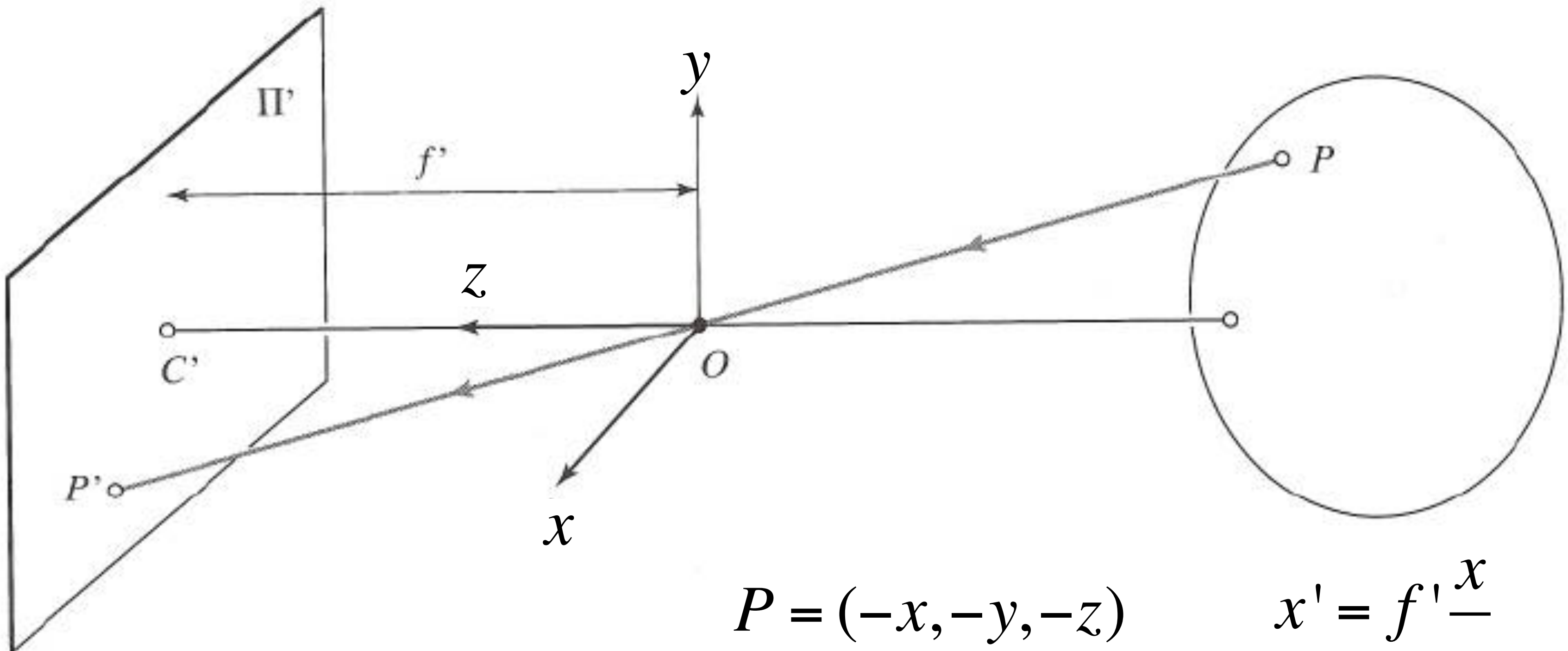


Image formation:
The pinhole camera model



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

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

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

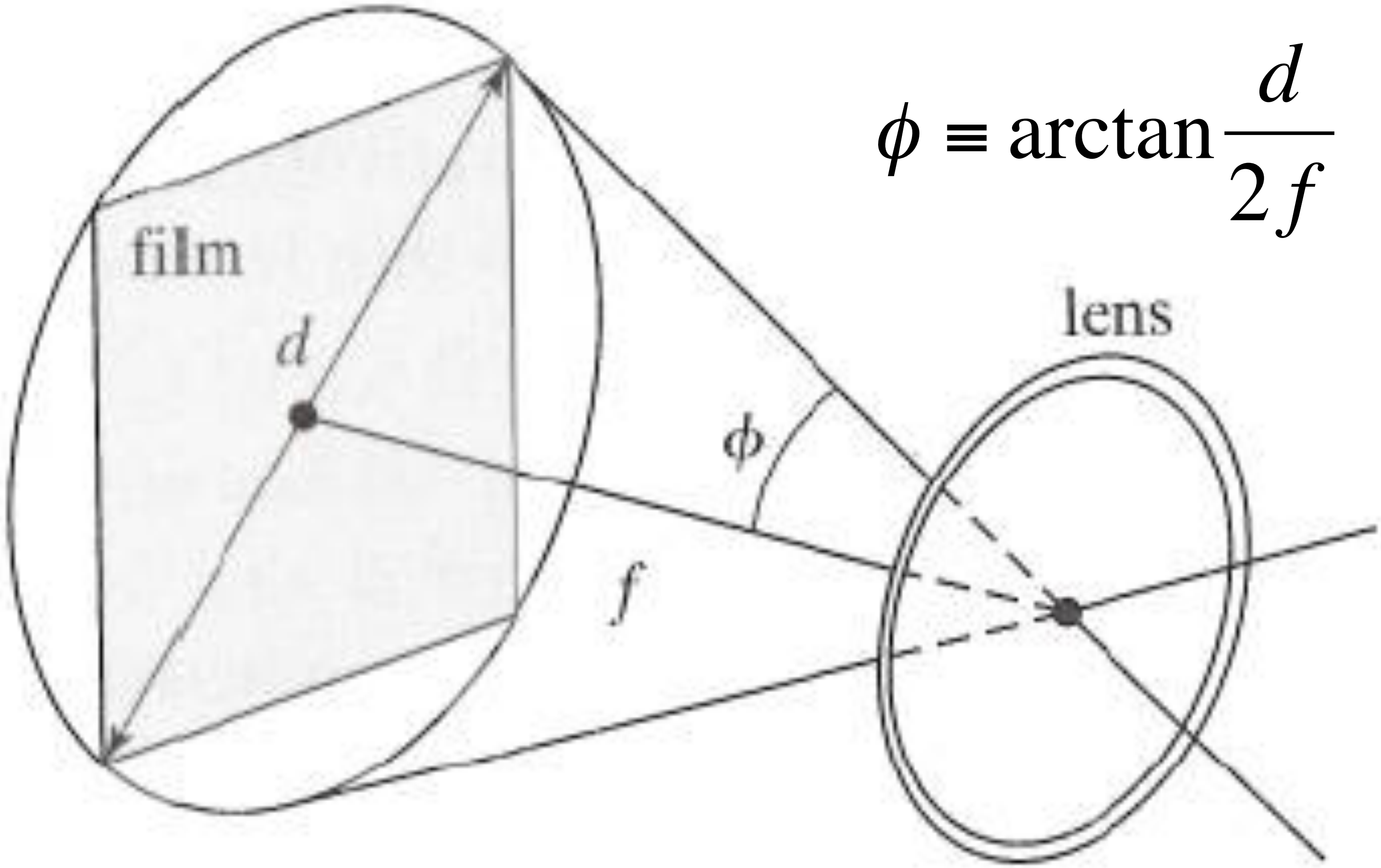
$$x' = f'\frac{x}{z}$$

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

$$z' = f'$$

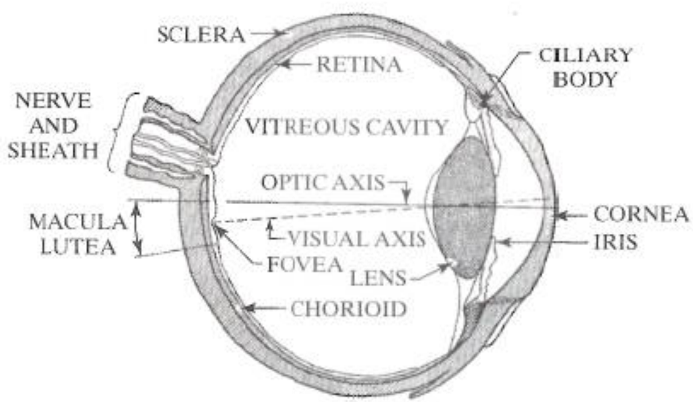
O : optical center / pinhole
k : optical axis
C' : image center
f' : focal length

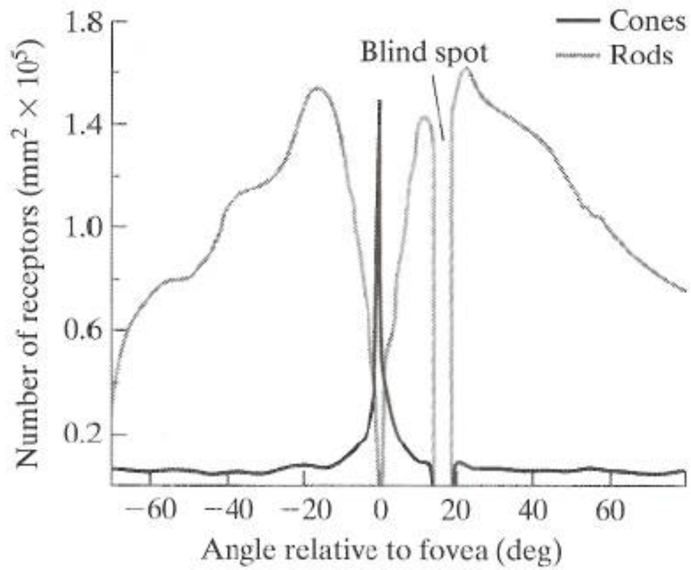
Image formation:
Field of view of the camera is 2ϕ



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

Image formation:
The human eye – a stereo camera





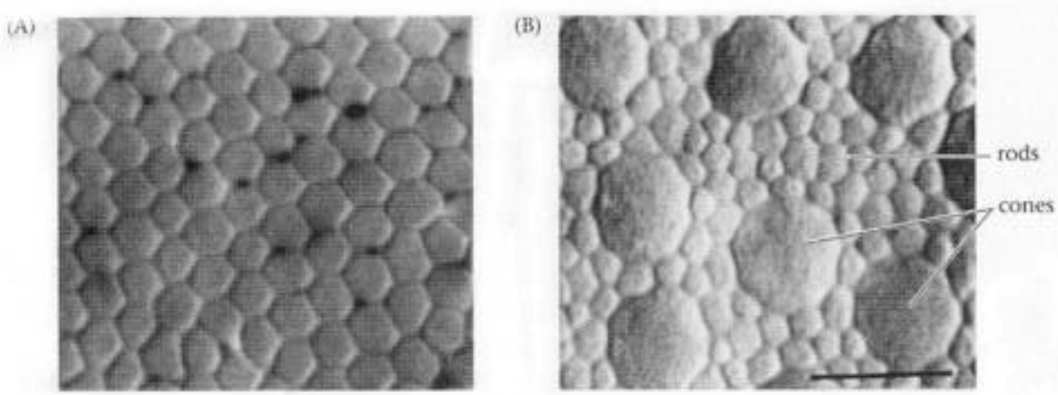


Image formation:
The digital camera

- Most cameras use charge coupled device (CCD) chip

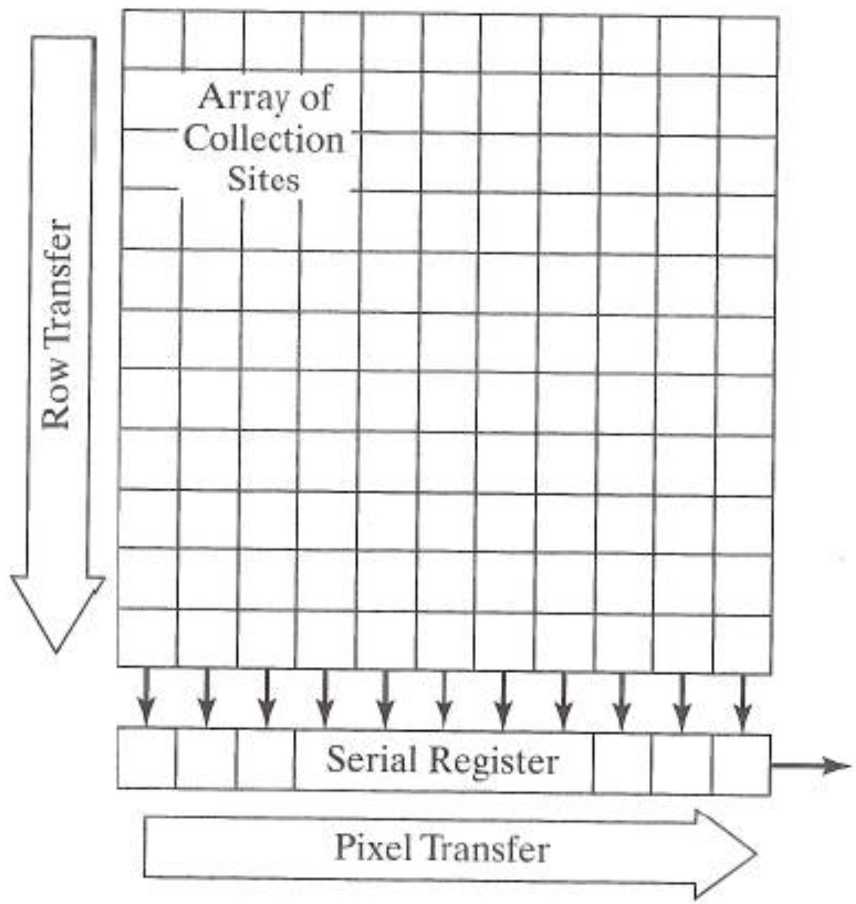
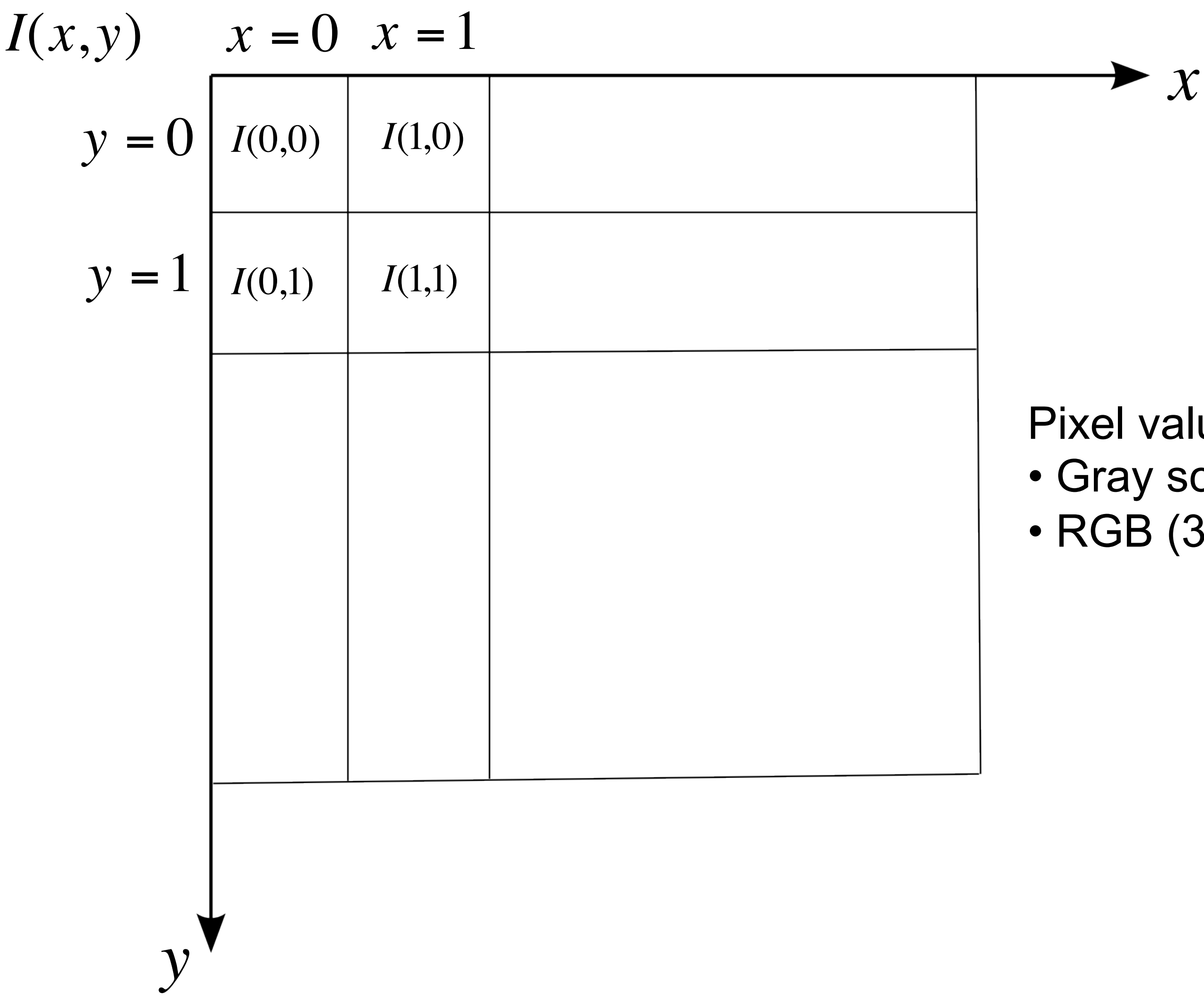


Image representation:
An array of pixels (picture elements)

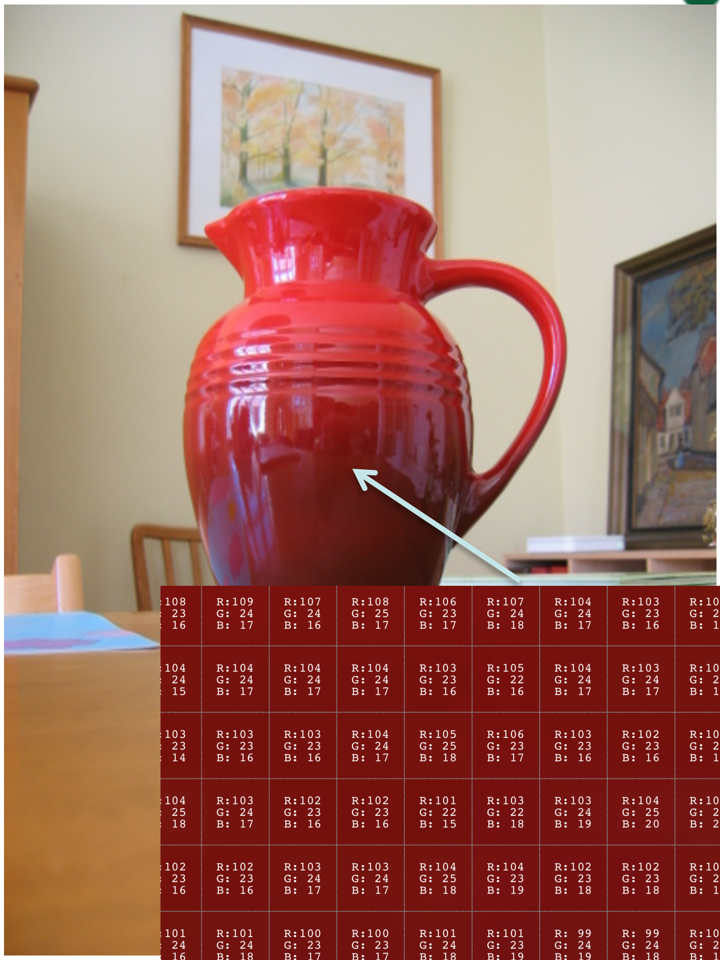


Pixel values:

- 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

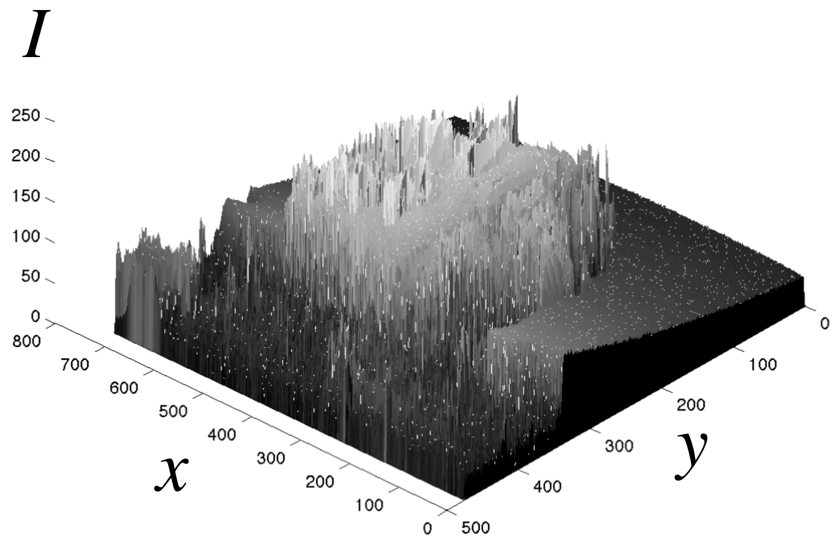


:108 : 23 : 16	R:109 G: 24 B: 17	R:107 G: 24 B: 16	R:108 G: 25 B: 17	R:106 G: 23 B: 17	R:107 G: 24 B: 18	R:104 G: 24 B: 17	R:103 G: 23 B: 16	R:103 G: 25 B: 15	R:104 G: 25 B: 18	R:104 G: 25 B: 18
:104 : 24 : 15	R:104 G: 24 B: 17	R:104 G: 24 B: 17	R:104 G: 24 B: 17	R:103 G: 23 B: 16	R:105 G: 22 B: 16	R:104 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

$I(x,y)$

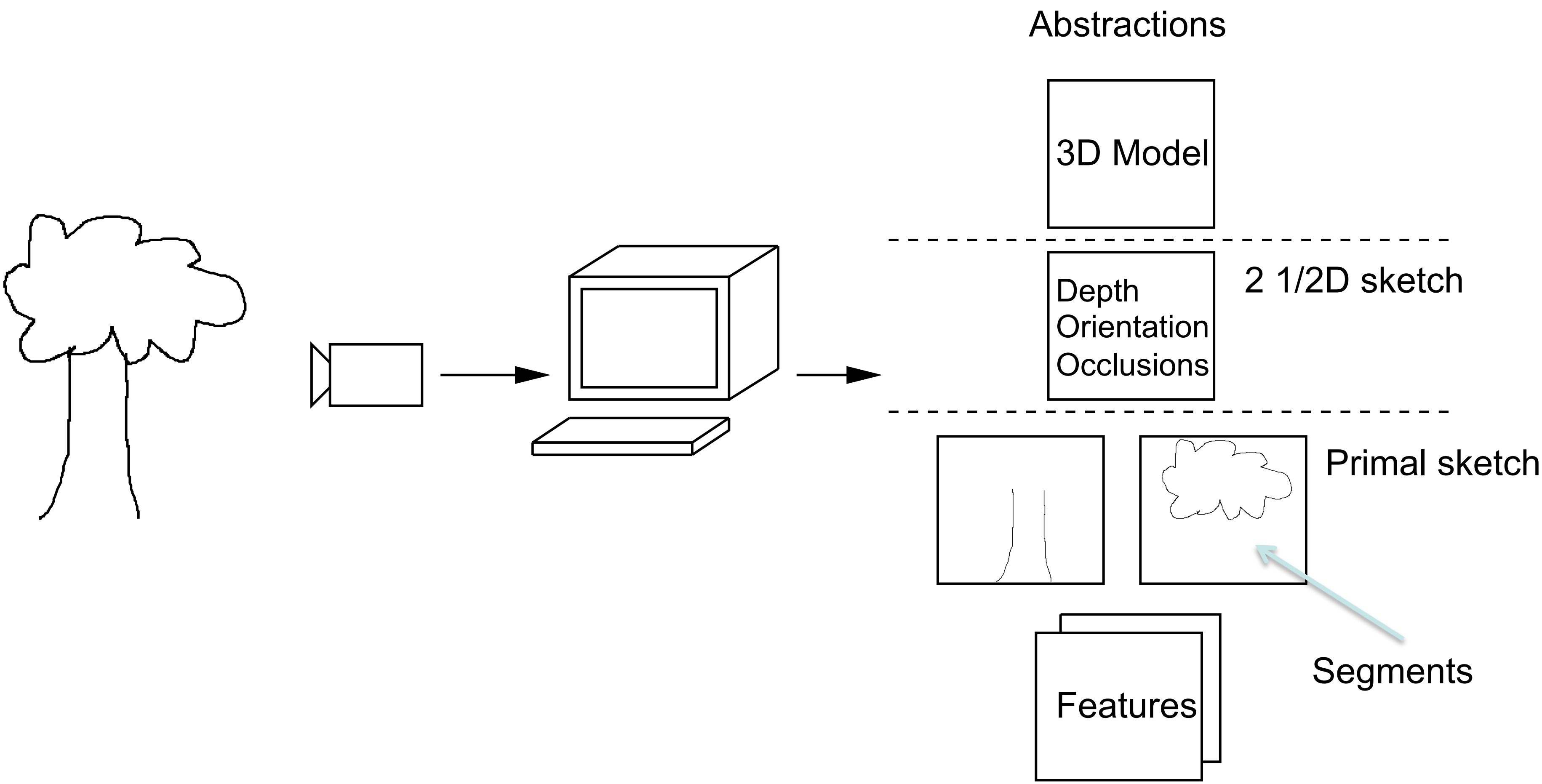
$I(x,y)$



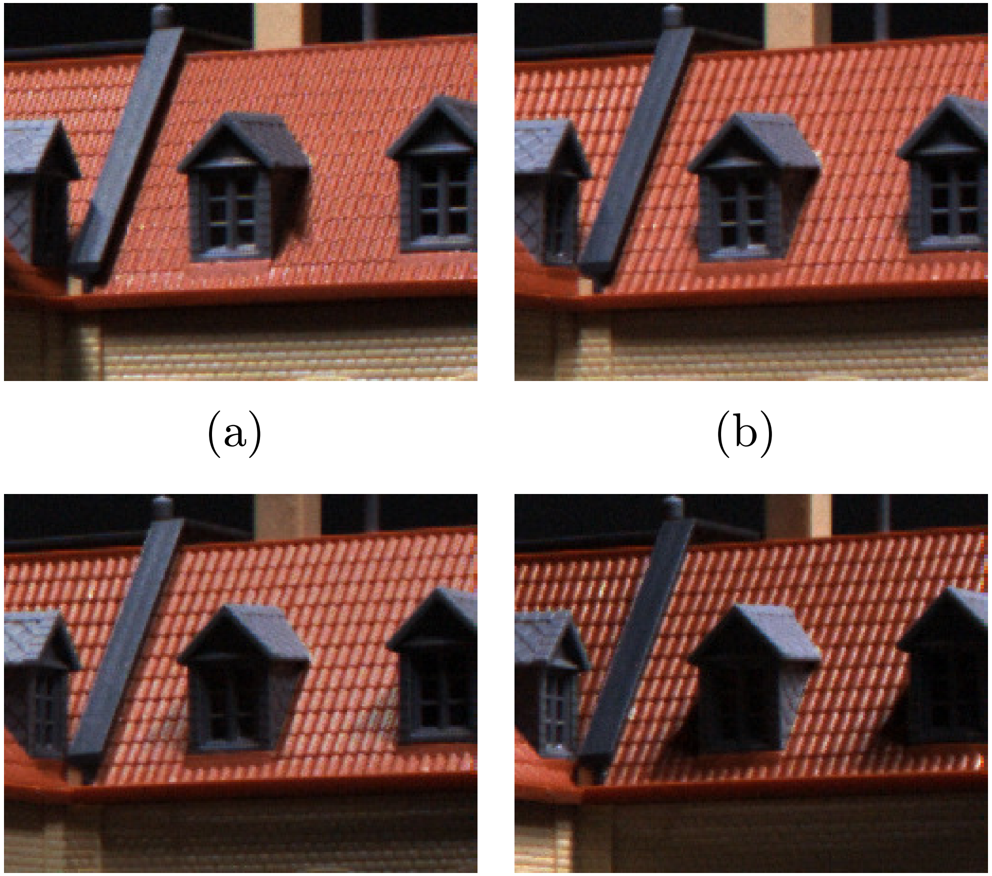
A bunch of numbers in a table is not enough!

Analysis and abstraction is needed

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



Color, Shading and Shadows also concerns us



(a) (b)

(c) (d)

Mandatory assignment 1: Filtering and edge detection

- Get acquainted with basic filtering
- Extraction of most primitive features – edges.
- Assignment text available at Absalon

Summary

- Examples of computer vision problems
- Image formation:
  - Pinhole camera model
  - We skip models for lenses
  - The human eye is a “camera”
- Image representation:
  - At the bottom: Digital images as an array of pixels
  - Marr’s layers of abstraction

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

Filtering and edges

Basic image processing

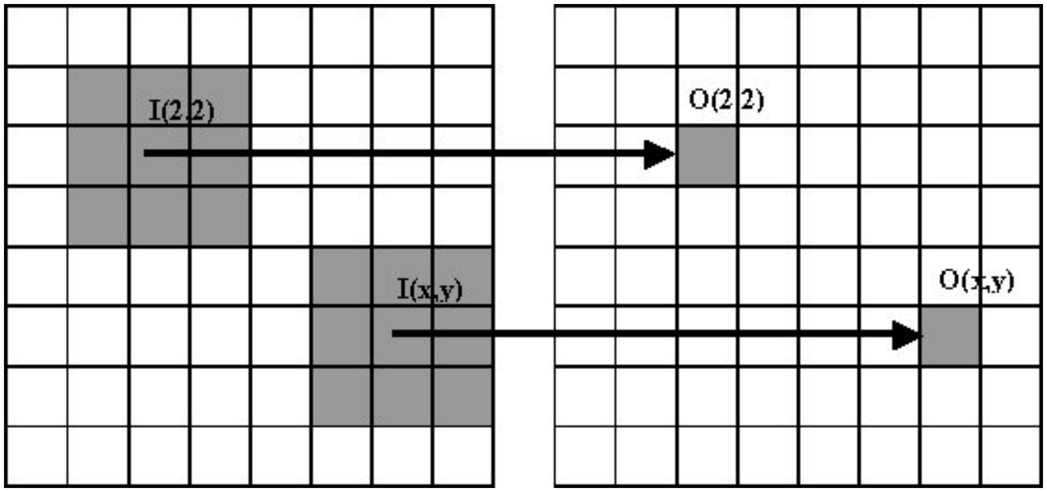
What is filtering good for ?

1. Noise reduction
2. Elimination of image quantization effects and conversion from integers to floats
3. Anti aliasing when up/down-scaling
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 reverse 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*