COMPUTER VISION 2018 - 2019

>INTRODUCTION

UTRECHT UNIVERSITY

RONALD POPPE

OUTLINE

Logistics

Computer vision

- Applications
- Challenges

Recap of linear algebra

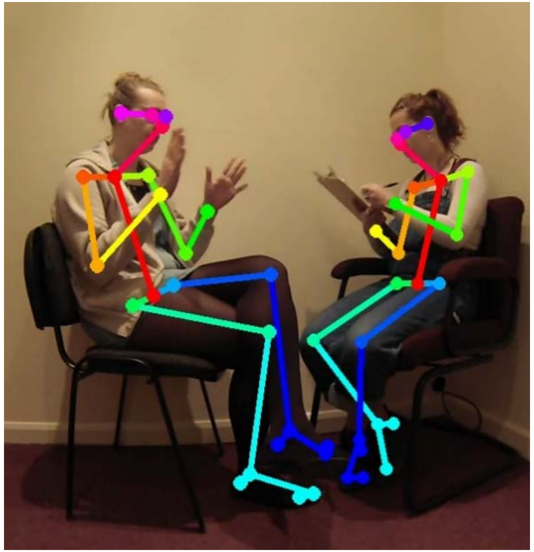
Topics in this course

LOGISTICS

YOUR LECTURER

Ronald Poppe

- Previously: University of Twente,
- Visiting: Delft University, Stanford University, Lancaster University



Working on:

- Analysis of humans in images and video
- Vision for interaction with machines and robots
- Detection of behavior patterns (lie detection)

Course coordinator and lecturer

YOUR LECTURER^2

Alexandros Stergiou

- Previously: University of Essex

Working on:

- Human interaction detection in video

Lecturer on (convolutional) neural networks

- Coordinating Assignments 4 and 5

YOUR LECTURER^3

Practicum assistant:

- Breixo Soliño Fernandez

Tasks within course:

- Supervision of walk-in sessions for Assignments 1-3
- Help/feedback on Assignments 1-3
- Grading of Assignments 1-3

COURSE INFO

Two lectures per week (weeks 6 – 14):

- Tuesday 13:15 – 15:00, BESTUURS-LIEREGG/BBG 209
- Thursday 11:00 – 12:45, RUPPERT-042/BBG 201

Information and feedback:

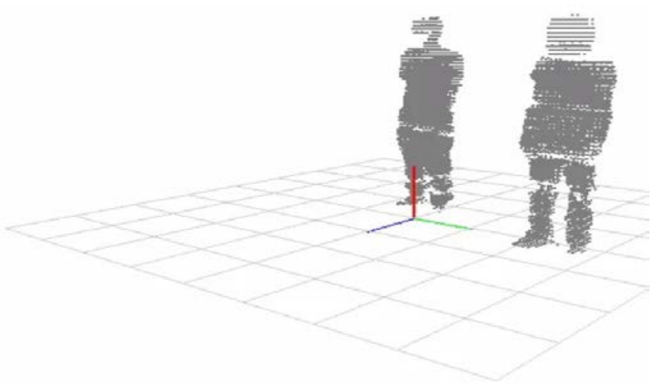
- Course website: http://www.cs.uu.nl/docs/vakken/mcv/
- News will appear on the website
- Ask questions straight away in the lecture (or break)
- Contact me via mail afterwards
- Use Slack: https://join.slack.com/t/infomcv2019/signup

COURSE MATERIALS

Reading/viewing material:

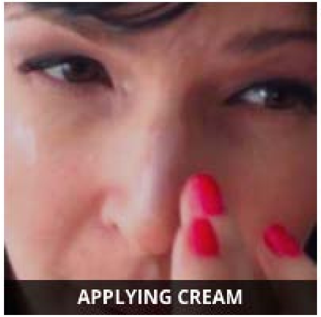
- R. Szeliski, “Computer Vision: Algorithms and Applications”, 2010. (Free to download)
- G. Bradski and A. Kaehler, “Learning OpenCV: Computer Vision With the OpenCV Library”, 2008.
- Links on website

ASSIGNMENTS



Five in total:

- Camera calibration (10%)
- Voxel-based 3D reconstruction (15%)
- Voxel-based tracking (25%)
- Data loading for NN training (5%)
- Action classification in video (45%)



A1-3: supervision by Breixo (b.solinofernandez@students.uu.nl)

A4-5: supervision by Alex (a.g.stergiou@uu.nl)

ASSIGNMENTS^2

Technical details:

- Use of OpenCV 3.2 and up, OpenCV 4.0 not recommended
- Framework is given (only for C++) for A1-3
- A1-3: Full support for C/C++. Limited support for Python, Java
- A4-5: Full support for Python

- Use of EmguCV C# wrapper is at your own risk (so no support)
- Assignments carried out in pairs
- If you can’t find a partner, let me know in the break or use Slack!

ASSIGNMENTS^3

Assignments 3 and, especially, 5 require significantly more work

- And also count more towards your final grade

Deadlines are strict!

- 0 - 24 hours late: 1 point deduction
- 24-48 hours late: 2 points deduction
- >48 hours late: submission is rejected
- Submission through online system (submit)

ASSIGNMENTS[4]

You all know this, but:

- No exchange of code with other groups
- No use of code of previous years
- No copying of (parts) of text in reports
- No fabrication of results
- Make sure you and your partner equally contribute

ASSIGNMENTS[5]

If you get stuck, just ask for help…!

- Visit assignment help sessions (3)
- Email student assistant
- Use Slack

If there are problems, inform me in time

- Don’t wait until (after) the deadline

GRADING

Assignments:

- Follow the steps and you’ll get an 8
- For missing/incorrect steps, points will be deducted
- For extra steps (or insights/testing), points will be added
- Different weight per assignment (bigger ones count more)

Practical assignments: 50%, Written exam: 50%

Retake only if weighted overall grade is >= 4

Final score must be at least 5.5 to pass (minimum for assignments/exam: 4)

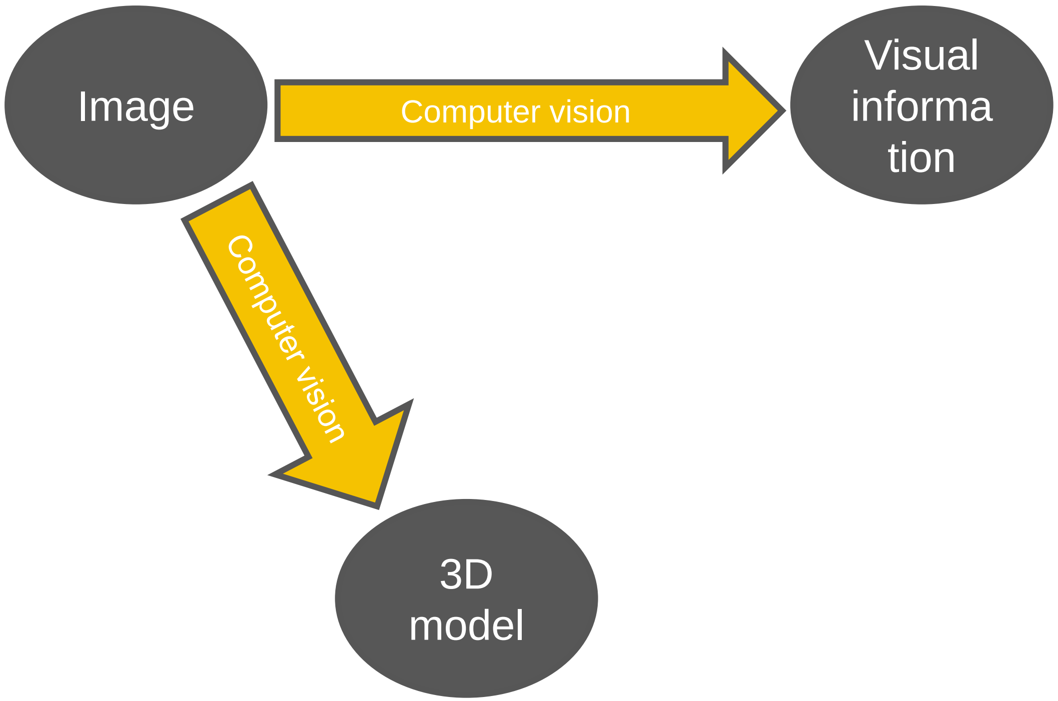
No assignment retake!

QUESTIONS SO FAR?

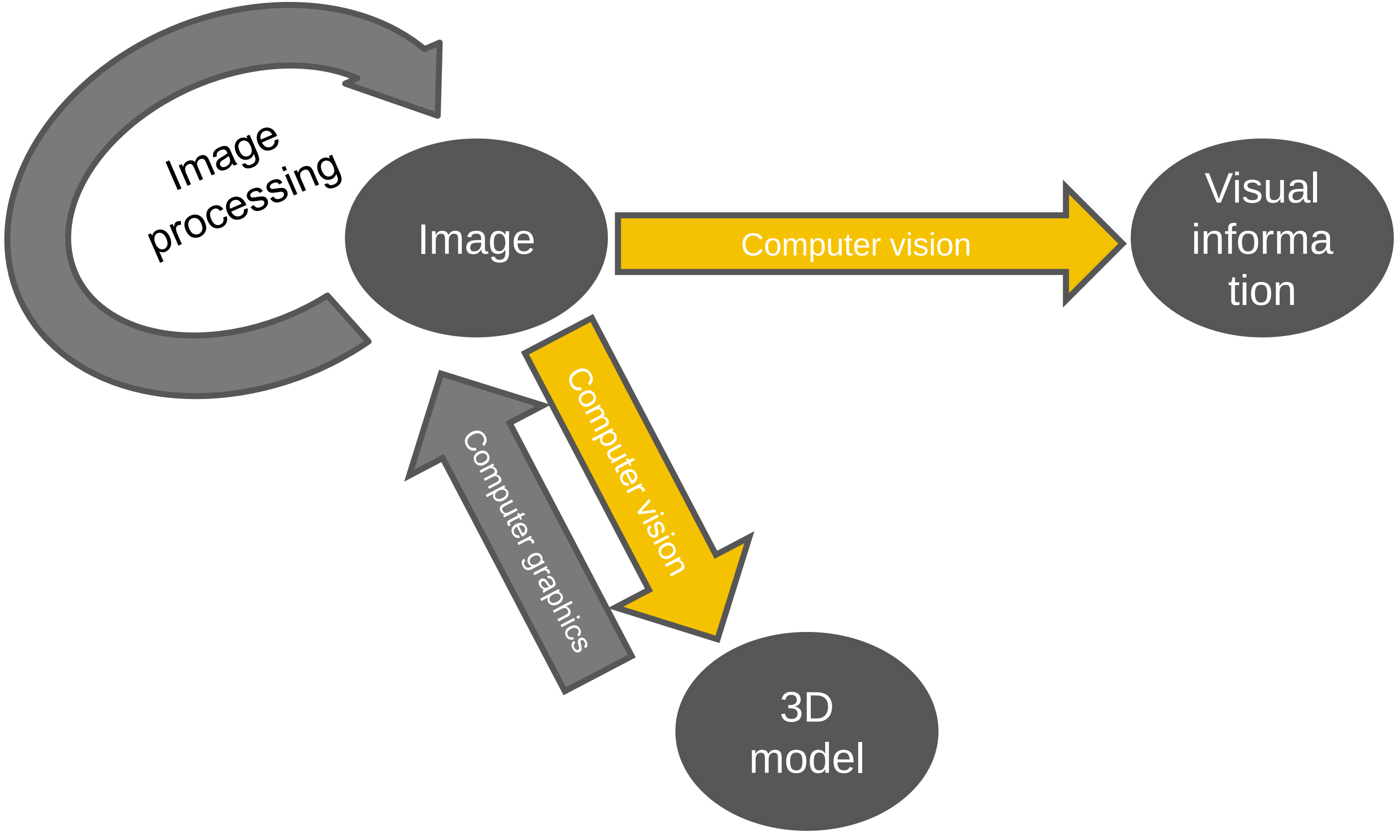
COMPUTER VISION

DEFINITION

Computer vision is the automatic analysis and interpretation of images.



RELATION TO OTHER FIELDS



EXAMPLE

Image processing

Computer vision

ID = “62-XH-FK”

Computer graphics

INPUT & OUTPUT

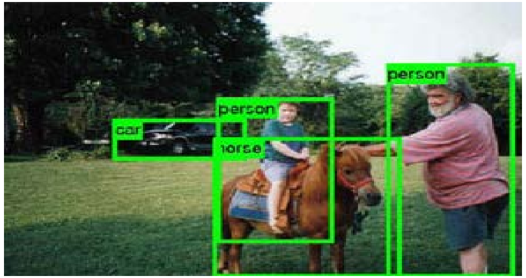
Input to computer vision algorithm:

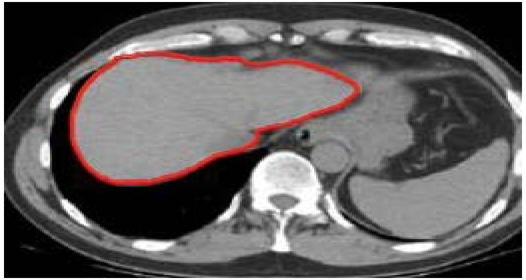
- Image (single or multiple)
- Video
- Depth images
- Additional information:
    - Bounding boxes
    - Segmentations
    - Labels
    - Calibration data
    - Etc.

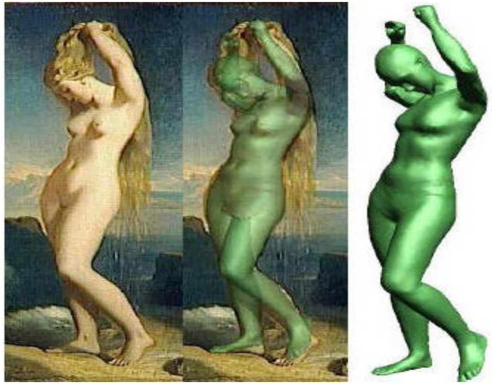
INPUT & OUTPUT^2

Output from a computer vision algorithm:

- Object label, location (object detection)
- Segmentation
- 3D shape
- Scene information (depth, lighting, etc.)
- Camera information (zoom, viewpoint)







https://www.youtube.com/watch?v=Kq-rB7YpYZ8

CHARACTERIZATION

Computer vision combines theory and engineering:

- Requires geometry for calibration and modeling
- Requires photometry to understand how colors are affected
- Requires statistics and probability theory to deal with data

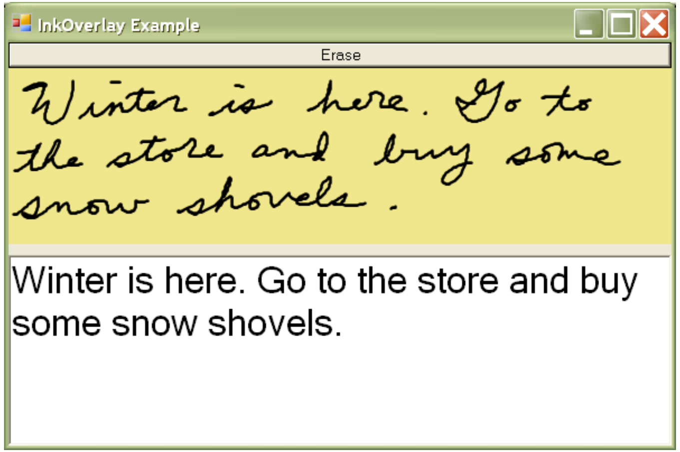
- Often approximations to keep things tractable
- Solutions are often non-trivial and the result of trial-and-error
- Creativity and a sharp eye are a plus

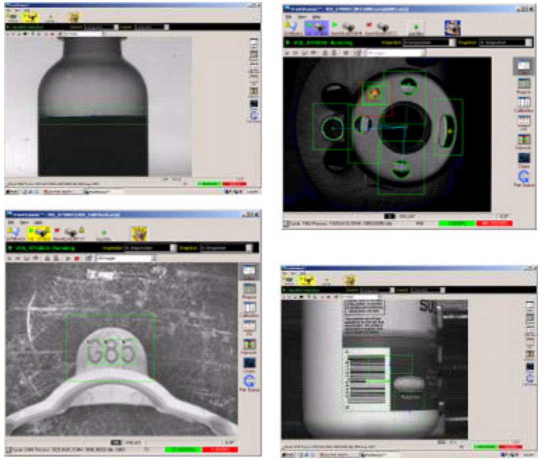
- It can be hard, but it’s fun… ☺

APPLICATIONS

PROCESS AUTOMATION

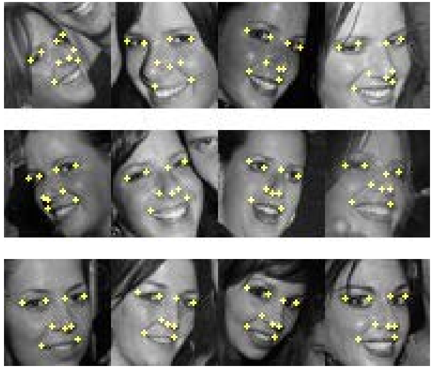
- License plate recognition
- Handwriting recognition/verification
- Product verification
- Rail track inspection

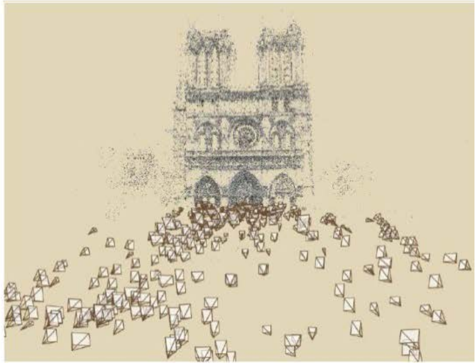




BIG DATA

- Face recognition in social networks
- Automatic tagging of images
- Finding duplicate movies
- Recognizing where photos have been taken
- Constructing the world

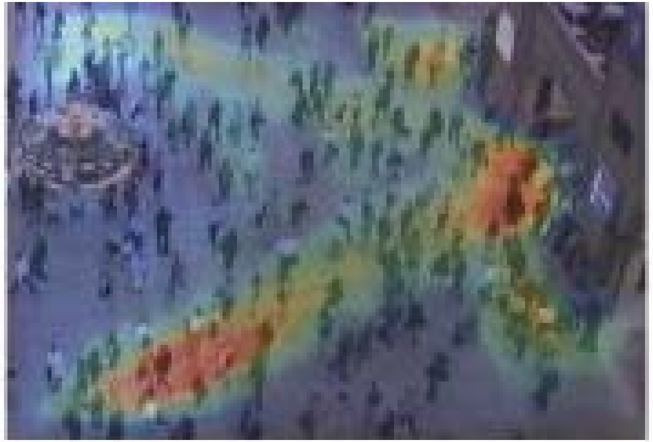


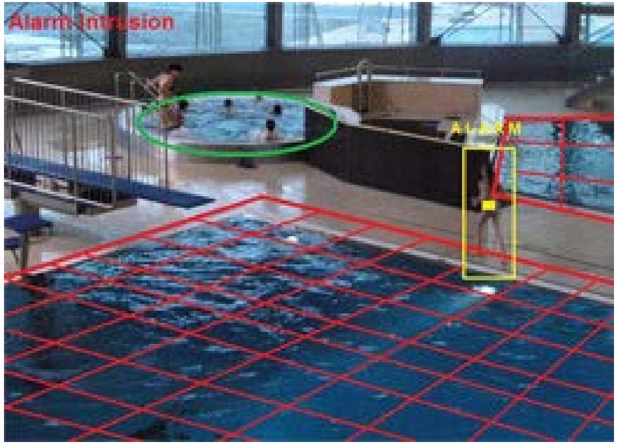


https://www.youtube.com/watch?v=VFdNQc3gVkM

SECURITY & SURVEILLANCE

- Finding left luggage
- Localizing threats
- Crowd control
- Monitoring of elderly homes
- Drowning warning systems



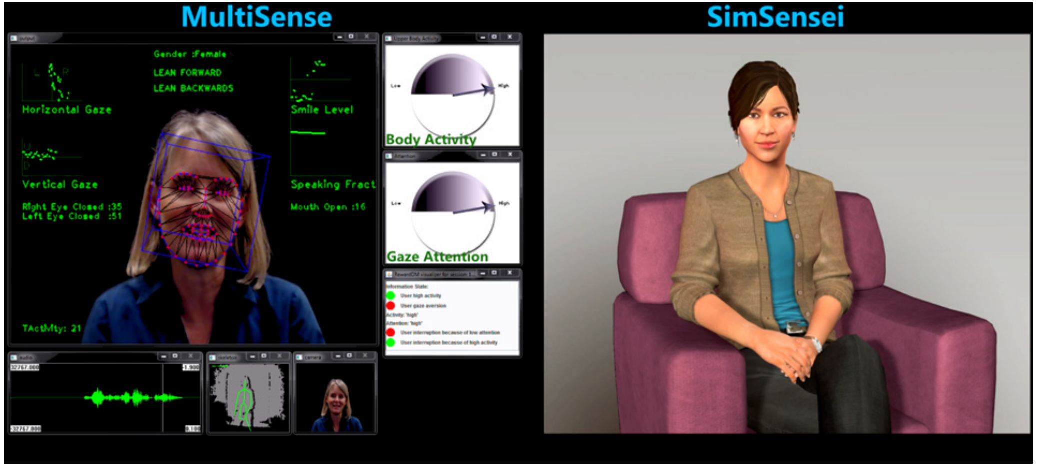


HUMAN-COMPUTER INTERACTION

- Gaming
- Social robots
- Gesture control

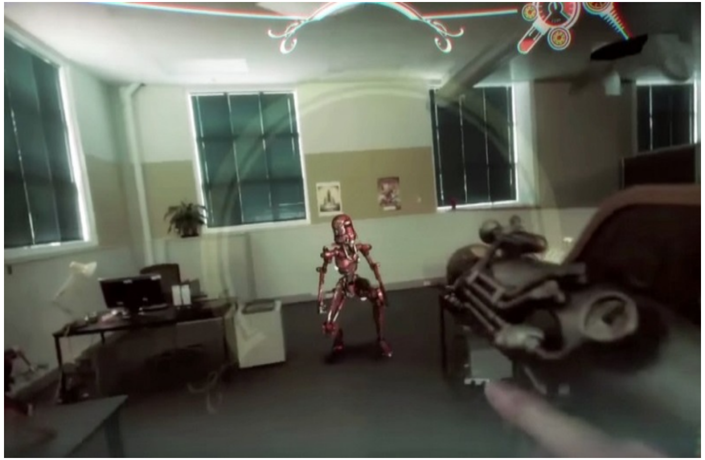
(SERIOUS) GAMING

- Body movement: Wii, Move, Kinect
- Facial expression: affect, control, playful chat (avateering)

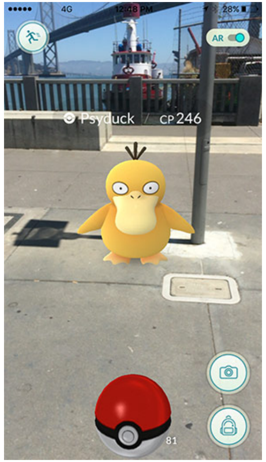



AUGMENTED REALITY

- Overlay of information
- Combination with virtual and real world
- Requires accurate estimation of camera position and orientation
- Requires localization of planes and objects

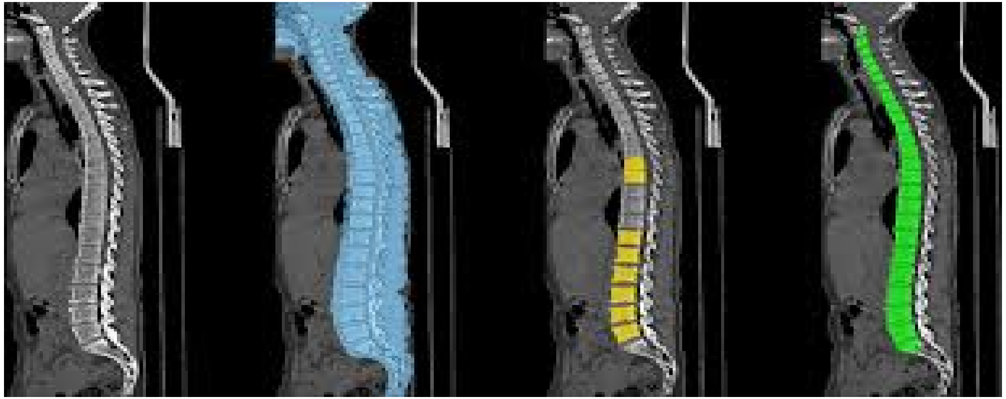


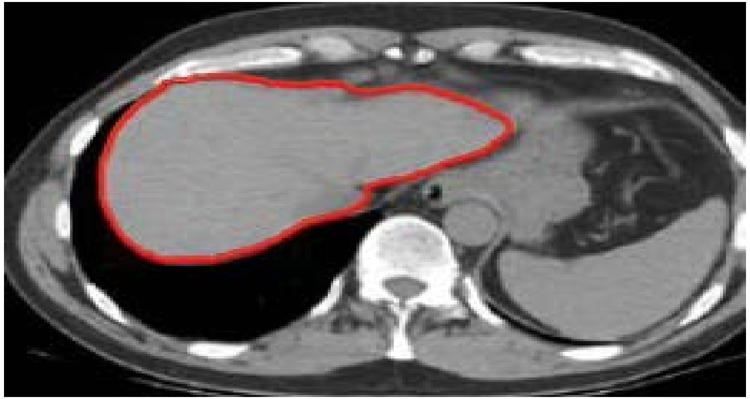
https://www.youtube.com/watch?v=kPMHcanq0xM



MEDICAL

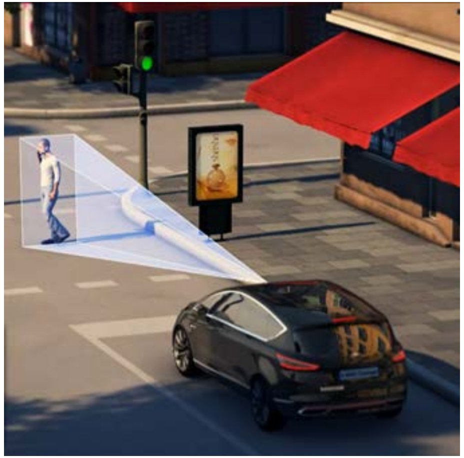
- Rehabilitation
- Visual assistance
- Anomaly detection (cancers, spots, etc.)
- Classification of diseases

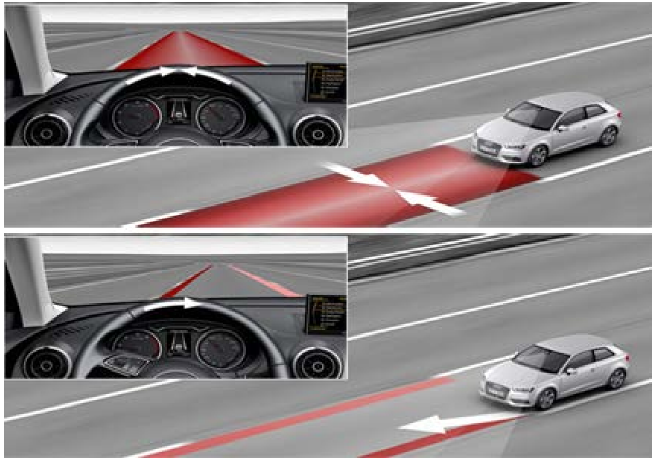




SMART CARS

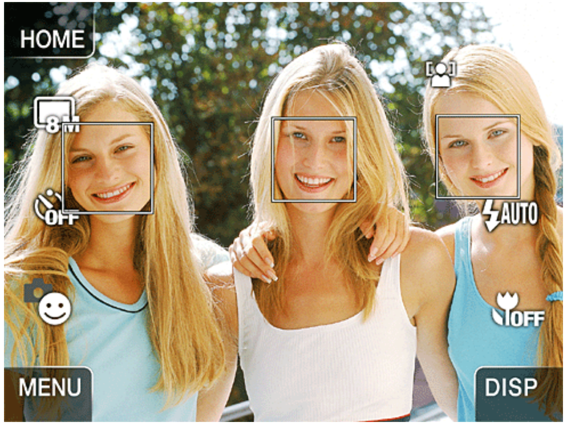
- Automatic collision avoidance
- Lane assist
- Driver monitoring
- Autonomous driving





FACE DETECTION

- Surveillance
- Consumer products
- Social networks



OPPORTUNITIES

Computer vision has many opportunities

- And there is a high demand on computer vision experts

You might want to continue in this direction after the course

- Limited number of MSc thesis projects available
- We WILL be looking at grades

CHALLENGES

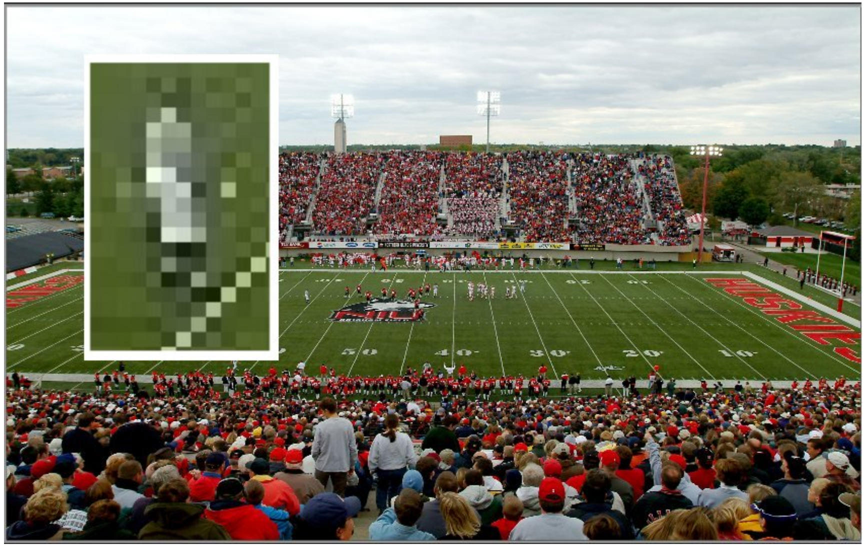
PROJECTION

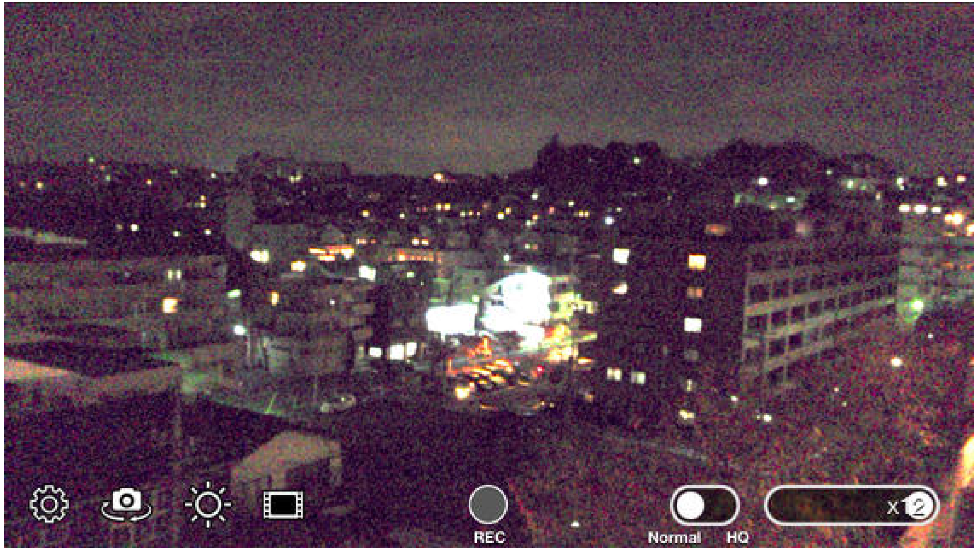
- Depth is lost
- Relative sizes
- Deformation



LOW RESOLUTION & NOISE

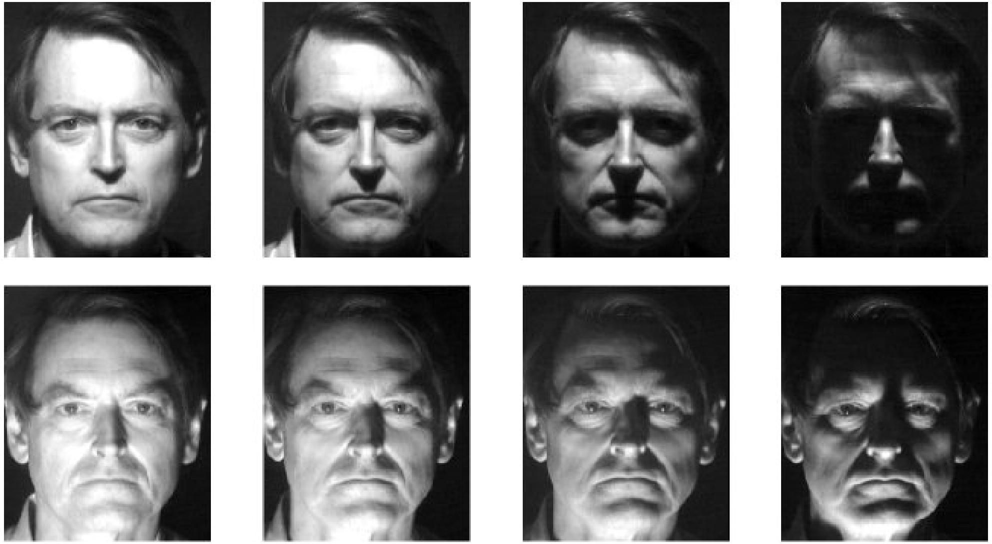
- Limited image resolution
- Limited dynamic sensor range
- Weather

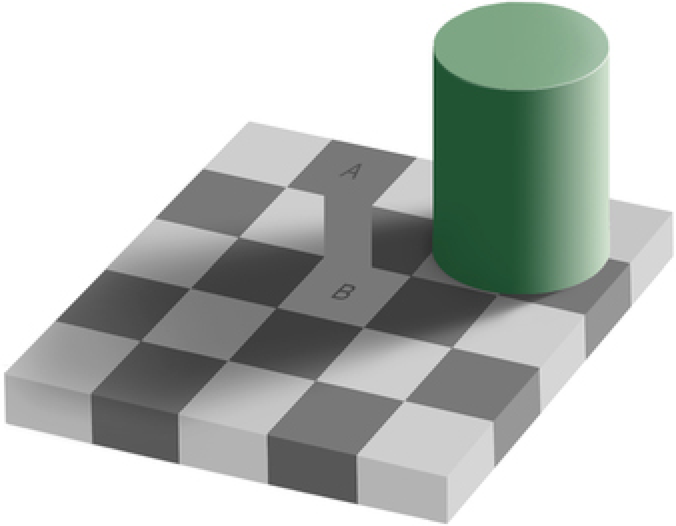




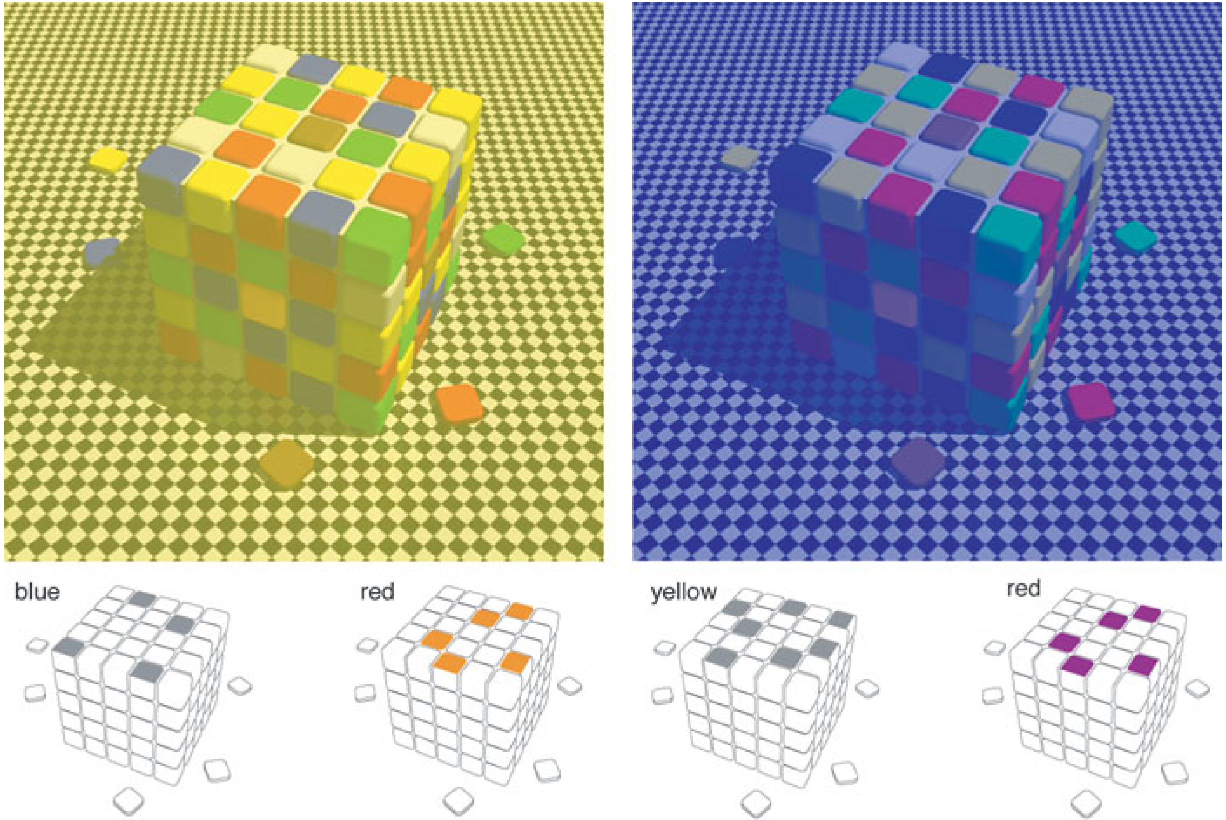
LIGHTING

- Directed light
- Amount of lighting
- Shadows





LIGHTING^2



CLUTTER AND OCCLUSION

- Distractions due to cluttered environment
- Incomplete observation due to partial occlusions

VARIATION IN SHAPE

- Concept (what makes a chair a chair?)
- Viewpoint

SIMILARITIES BETWEEN CLASSES

- Objects that you would like to distinguish can be very similar

CONTEXT

- Other objects
- Object use
- Prior knowlege
- Light


LINEAR ALGEBRA

NOTATION

We typically denote:

- Scalars: x (regular)
- Vectors: $\boldsymbol{x}$ (bold) → can be row or column vector

- Vectors have $|\boldsymbol{x}|$ elements, with $|\cdot|$ the norm
- Elements of vectors: $\boldsymbol{x}_i$ is a scalar ($1 \leq i \leq |\boldsymbol{x}|$)

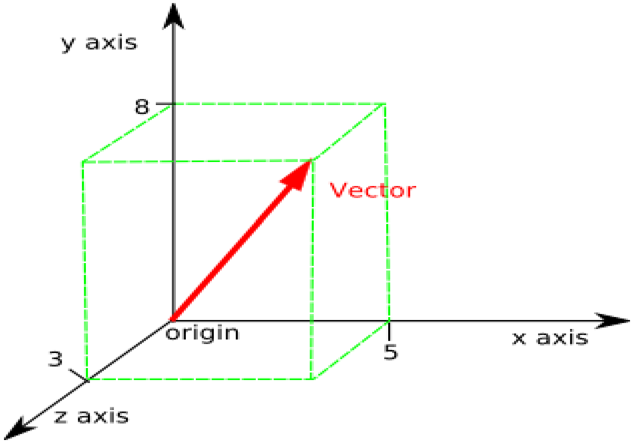
- Matrices: $\boldsymbol{A}$ (capital)

POINTS & MATRICES

A point in 2D space is denoted as vector: $\boldsymbol{x} = (x, y) = \begin{bmatrix} x \\ y \end{bmatrix}$.

- Units can be meters (m) or pixels (px).

A point in 3D space is denoted as vector: $\boldsymbol{x} = (x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$.



POINTS & MATRICES^2

Vectors can be transposed: $\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$**,** $\boldsymbol{x}^T = \begin{bmatrix} x & y & z \end{bmatrix}$

- $\boldsymbol{x}^{TT} = \boldsymbol{x}$

The augmented vector $\overline{\boldsymbol{x}}$**:** $\overline{\boldsymbol{x}} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ **or** $\overline{\boldsymbol{x}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$**.**

An $m \times n$ **matrix consists of** ***m*** **rows and** ***n*** **columns**

- $\boldsymbol{I}$ is the identity matrix: $\boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

ADDITION & MULTIPLICATION

Vectors can be added: $\boldsymbol{a} + \boldsymbol{b} = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$

Vectors can be multiplied by a scalar: $w\boldsymbol{a} = (wa_1, wa_2, wa_3)$

Vectors can be multiplied by a matrix:

$$\boldsymbol{Ra} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} r_{11}a_1 + r_{12}a_2 + r_{13}a_3 \\ r_{21}a_1 + r_{22}a_2 + r_{23}a_3 \\ r_{31}a_1 + r_{32}a_2 + r_{33}a_3 \end{bmatrix} \textbf{ (pre)}$$

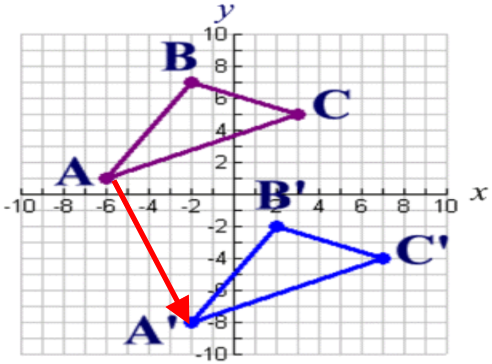
Or:

$$\boldsymbol{a}^T\boldsymbol{R} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}^T \times \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} r_{11}a_1 + r_{21}a_2 + r_{31}a_3 \\ r_{12}a_1 + r_{22}a_2 + r_{32}a_3 \\ r_{13}a_1 + r_{23}a_2 + r_{33}a_3 \end{bmatrix}^T \textbf{ (post)}$$

TRANSLATION

Given that we can add vectors, we can simply add a translation:

$$\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{t} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$



In augmented coordinates, we can use a multiplication:

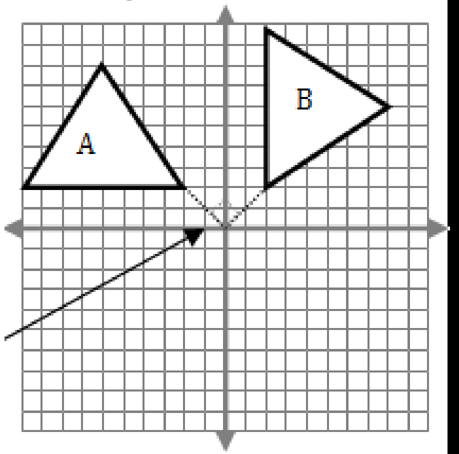
$$\bar{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \bar{\boldsymbol{x}} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \bar{\boldsymbol{x}}$$

With identity matrix *I*, translation vector *t* and zero vector *0*

ROTATION

We can rotate points around the origin (0,0), but more commonly, we combine a rotation with a translation:

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$$



Or in augmented coordinates:

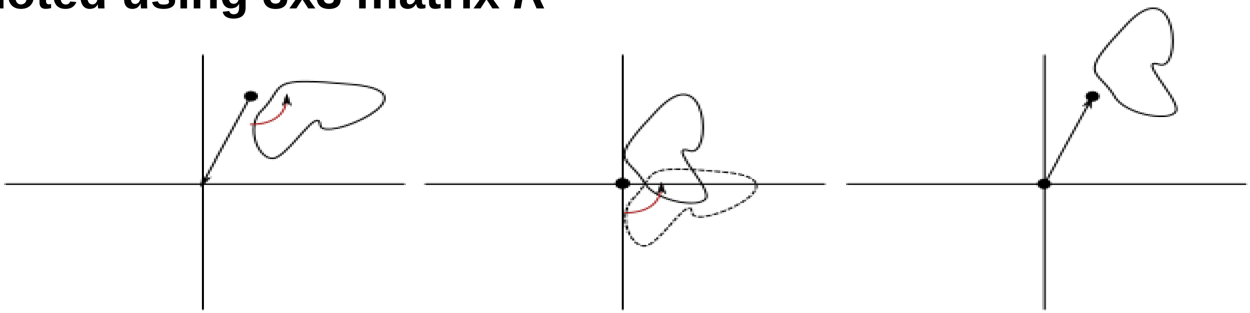
$$\bar{\boldsymbol{x}}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \bar{\boldsymbol{x}}$$

With *t* the translation vector and *R* the rotation matrix in 2D:

$$\boldsymbol{R} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$ **with θ the rotation angle**

ROTATION²

Rotation in 3D is less straightforward to understand, but can still be denoted using 3x3 matrix *R*



Remember we rotate around the origin

- To rotate a shape around an arbitrary point $\boldsymbol{c}$, we first subtract this point, perform the rotation and then add $\boldsymbol{c}$:
- $\boldsymbol{x}' = \boldsymbol{c} + \boldsymbol{R}(\boldsymbol{x} - \boldsymbol{c})$

THIS COURSE

LECTURES

Divided into two main topics:

- Multi-view reconstruction from silhouettes (4 lectures)
- Supervised learning and classification (7 lectures, 2 practical sessions)

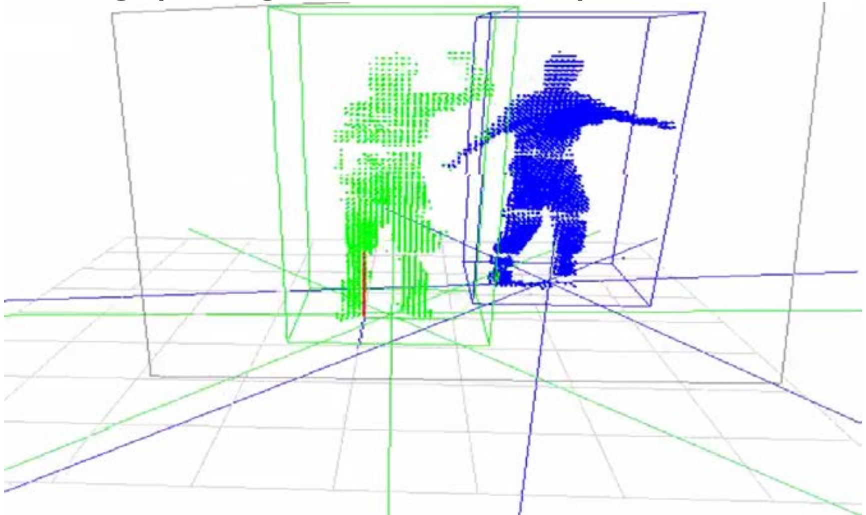
Theory and application

- Part low-level vision, part high-level vision
- Application in the assignments requires theory

LECTURES

Multi-view reconstruction from silhouettes (50%):

- Image Formation: Geometry and Radiometry (Assignment 1, 10%)
- Silhouette-based Volume Reconstruction (Assignment 2, 15%)
- Voxel-based Visual Tracking and Clustering (Assignment 3, 25%)



LECTURES²

Supervised learning and classification (50%):

- Training/test pipeline
- Working with TensorFlow/Keras (Assignment 4, 5%)
- Action classification in videos (Assignment 5, 45%)
- “Deep learning”

TASK

Install OpenCV (www.opencv.org) and compile it

- Version 3.2 or higher (Windows, Linux or Mac), not OpenCV 4.0
- Installation guide on the INFOMCV website

Start a new project that:

- Loads a jpeg file
- Draws a white 100x100 pixel rectangle in the middle
- Shows the image on screen and saves the image as jpeg file

You won’t be graded, but you’ll need this in Assignment 1.

NEXT LECTURE

Image Formation: Geometry and Radiometry

- How does a camera transform a 3D scene to a 2D image?
- How are colors recorded?
- Chapter 2 of Szeliski book

You'll need this for Assignment 1

- Deadline Sunday February 17, 23:00

Tuesday 13:15-15:00, BESTUURS-LIEREGG

QUESTIONS?