



---

# TSBB15 Computer Vision

Examiner:  
Per-Erik Forssén



# Today's lecture

---

- General course information
- Course aims and structure  
methods, theory, exercises, projects
- Applications
- Research  
Examples from CVL research projects
- Examples from CDIO student projects
- Methodology



# What's new?

---

- Main novelty this year is that all labs have been revised to be Python only, and to exploit Python fully.  
(last year we also had a Matlab option, and the Python code was mimicking Matlab.)
  - + Python encourages good programming practises.
  - + Widely used in industry and research
  - + Useful for TSBB11, TSBB17, master's thesis etc.
  - + open-source, no license server/key hassle.
- You may not know Python (yet)



# What's new?

---

Also a couple of minor changes:

- Preparation time for labs is explicitly scheduled.
- Minor updates to the projects
- New lecturers and TAs :-)



# Course Information

---

- **Website:** <http://www.cvl.isy.liu.se/education/undergraduate/tsbb15>
  - Updated continuously during the course
  - Schedule, lab sheets, pointers to code, old exams etc.
  - Lecture slides (after each lecture).



# Course Information

- **Website:** <http://www.cvl.isy.liu.se/education/undergraduate/tsbb15>
- **Lectures**



Michael Felsberg



Per-Erik Forssén

+ two special guests



# Course Information

- **Website:** <http://www.cvl.isy.liu.se/education/undergraduate/tsbb15>

- **Lectures**



Michael Felsberg



Per-Erik Forssén

+ two special guests

- **Computer Exercises and Project Guides**



Gustav Häger  
CE + PG



Anderson Tavares  
CE



Zahra Gharaee  
CE



Karl Holmquist  
CE+PG

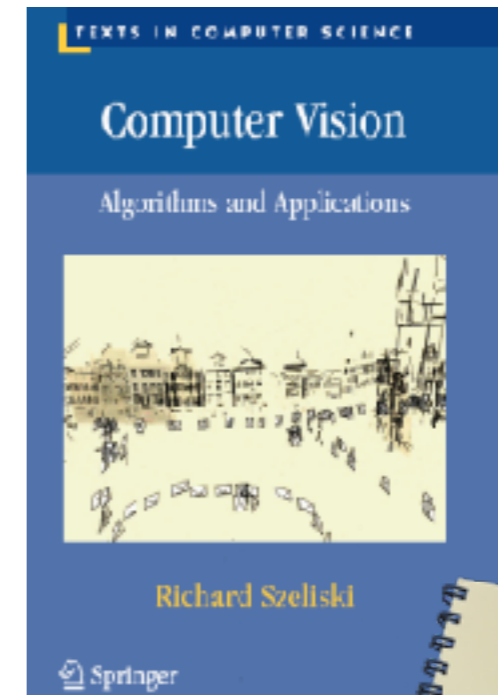


Felix Järemo Lawin  
PG



# Course Book

- Main book: Computer Vision— Algorithms and Applications, By Richard Szeliski 2011. Electronic version - see webpage.
- For geometry we will also use the IREG compendium by Klas Nordberg.
- Other written sources are linked in the Materials column on the webpage.







# Video Clips

- The lazy way to prepare for a seminar. There are links to recommended clips on the course webpage.

YouTube video player showing a handwritten lecture on the Gaussian Mixture Model (GMM). The video title is "(ML 16.6) Gaussian mixture model (Mixture of Gaussians)". The video content includes the text: "Gaussian Mixture Model (GMM) latent var.", "Z ∈ {z1, ..., zM}, P(Z=zk) = αk", "Given Z=zk, X ~ N(μk, Ck)", "where μ1, ..., μM ∈ R^d and C1, ..., CM are d x d cov. matrices.", and a diagram showing two overlapping Gaussian distributions in 2D space. The video has 87,482 views and 416 likes.

YouTube video player showing a video titled "Scale space" with the subtitle "Linear scale spaces". The video is uploaded by "ScaleSpace - 2-Linear Scale Spaces" and has 42 views. The video content includes a play button and the text "Scale space" and "Linear scale spaces".



# Course Aims

---

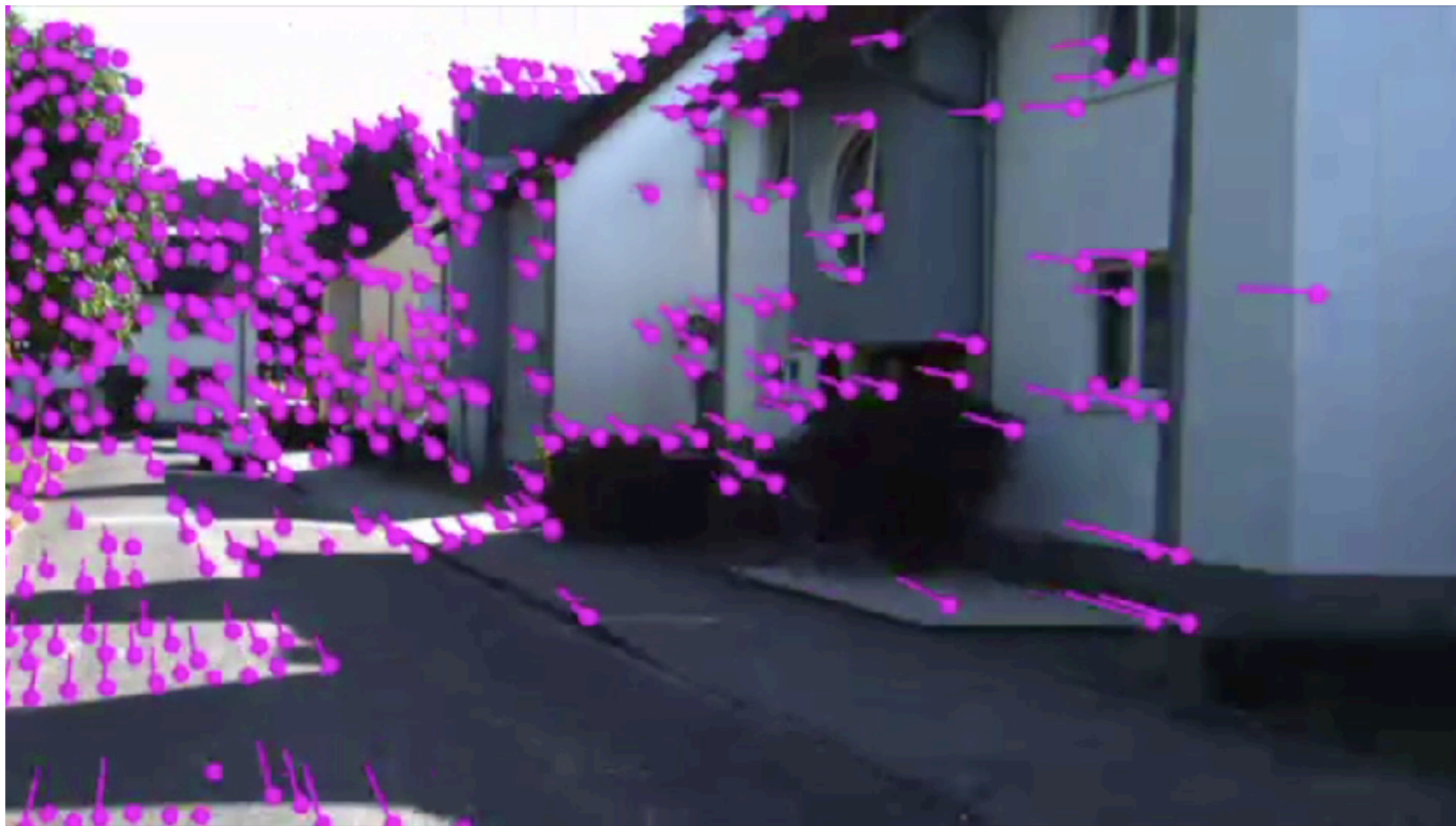
- From the LiTH Study Guide:

After having passed this course, the student should be able to **describe problems and algorithms** for the following basic computer vision and image processing tasks:

- tracking of image regions, triangulation from stereo images, estimation of optical flow, detection of several image features, matching of image features, graph and tree structures and other image representations, generative image models, segmentation of image regions, enhancement of images, debugging and visualisation



# Tracking





# Motion Estimation





# Object Tracking



Image credit: Zheng Wu, Boston University: <http://cs-people.bu.edu/wuzheng/>

- Project 1: Tracking using background models.



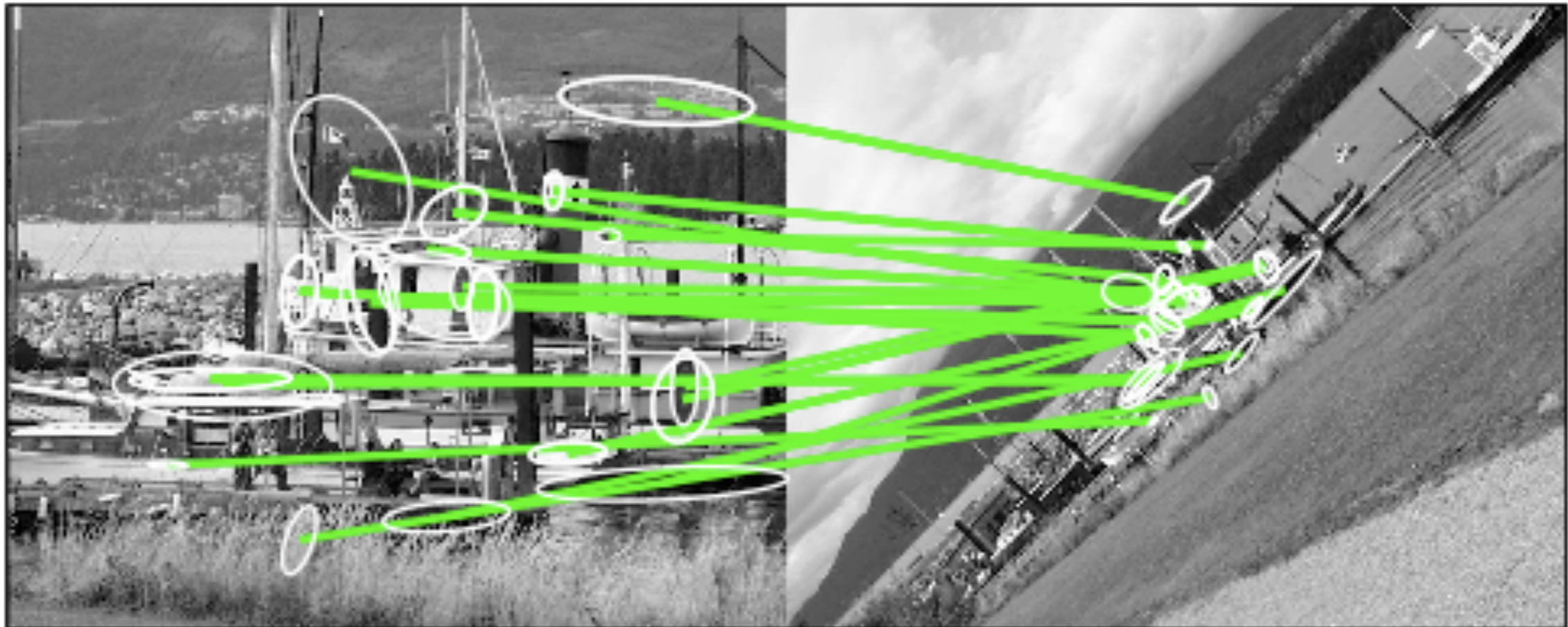
# Feature Matching



Find out which image regions are corresponding



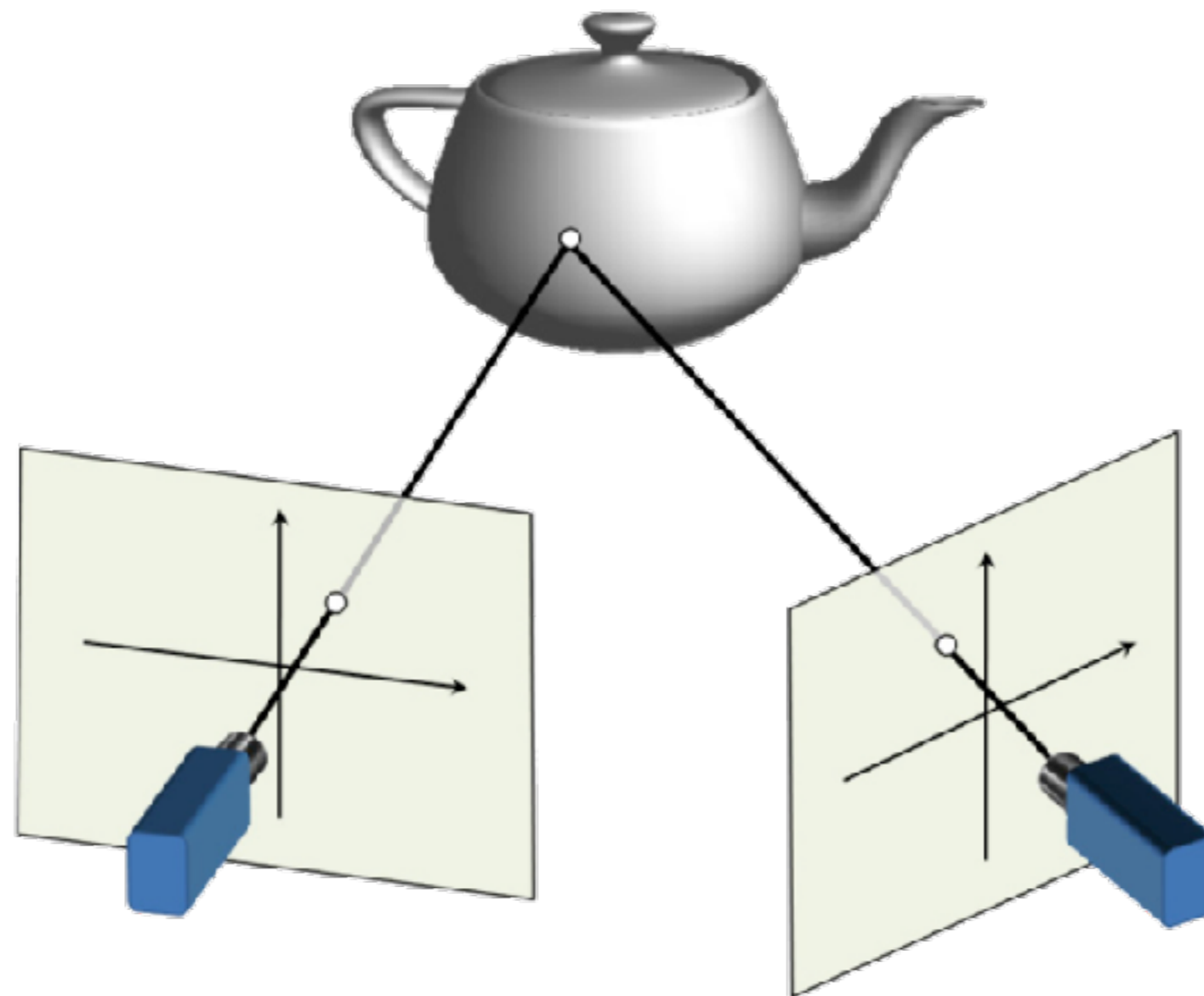
# Feature Matching



Find out which image regions are corresponding (LE6)



# Triangulation



T. Moons et al. FTCCGV 2008





# Triangulation

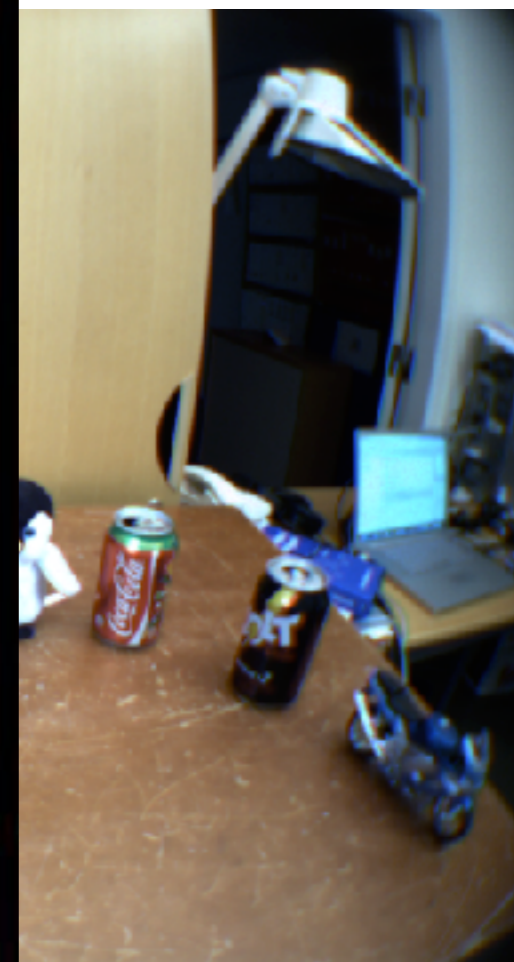


Left view

Right view



# Triangulation



Disparity



# Structure from motion

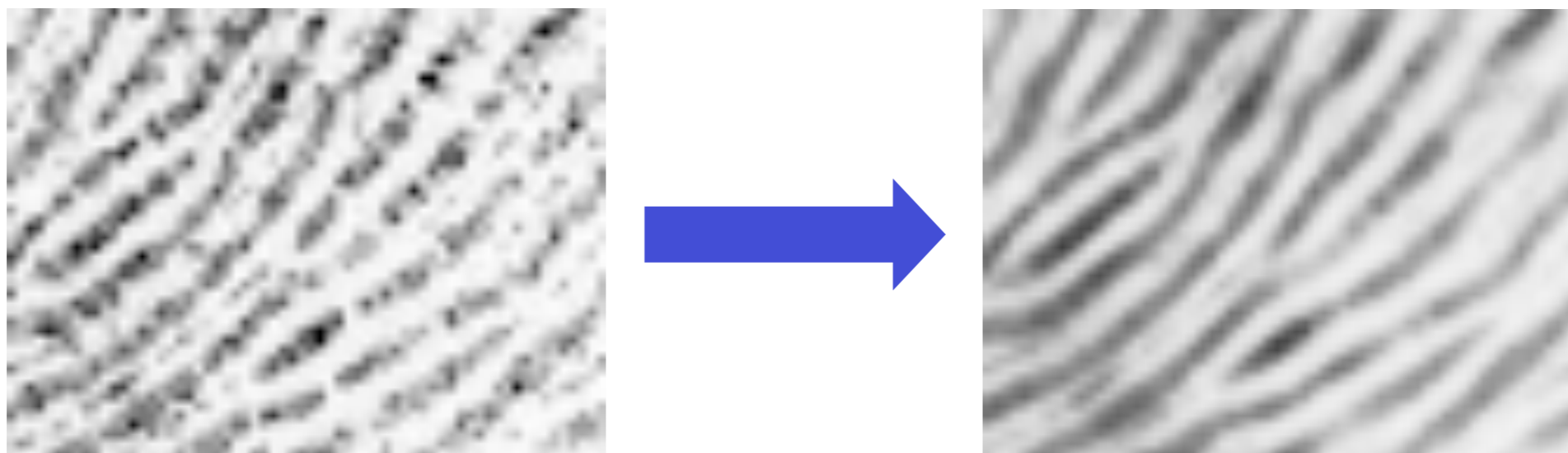


Image credit: Bundler home page <http://www.cs.cornell.edu/~snaveily/bundler/>

- Project 2: Start with two views, and successively add more.
- Finally perform simultaneous refinement of 3D point positions and camera positions (Bundle adjustment).

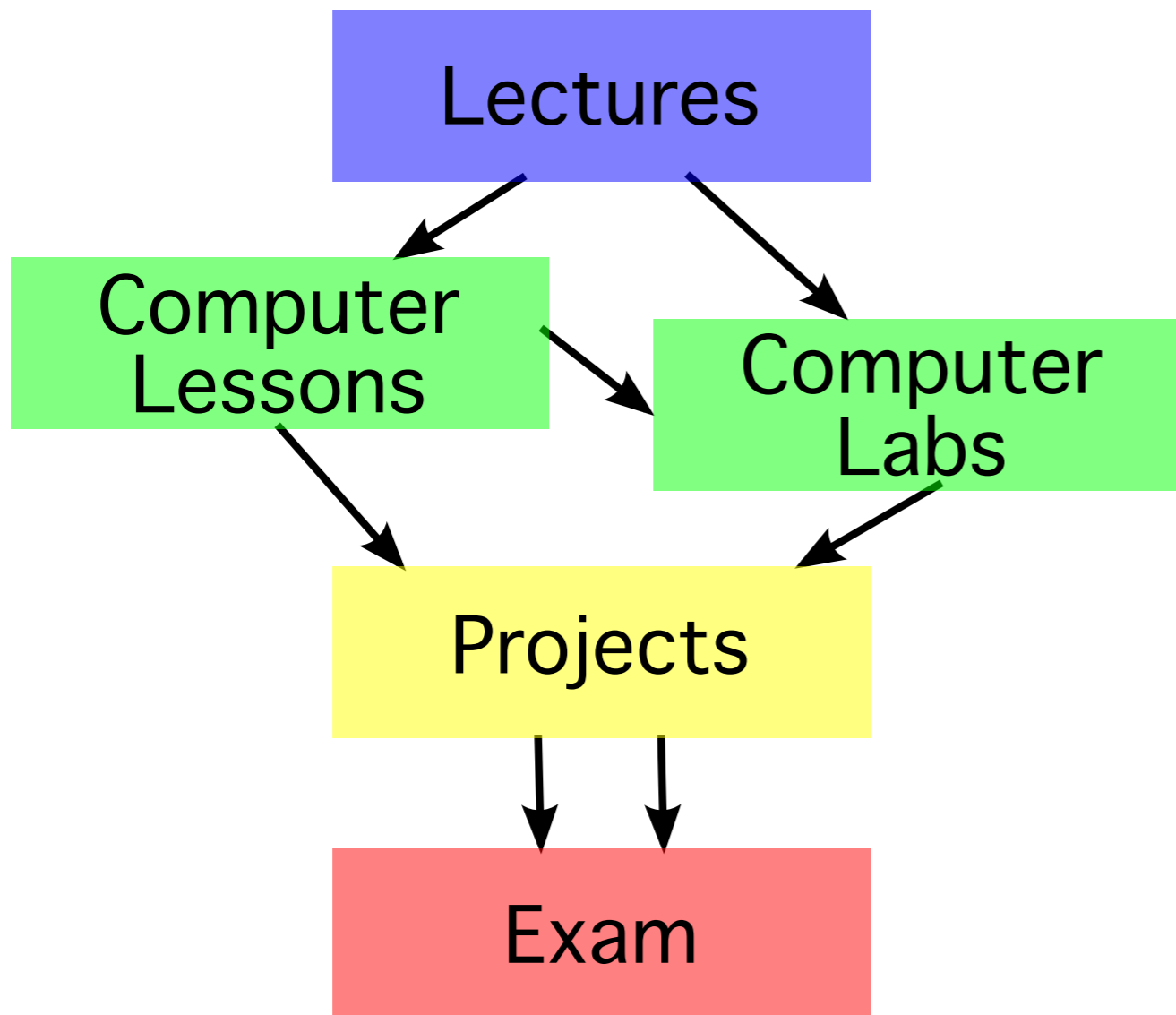


# Image Denoising, Enhancement and Inpainting





# Course contents



- Theory and two levels of practise:
- **Computer exercises** are designed to be useful for completing the projects. If you prepare well and understand them, the projects will be easier.
- **Projects** will give you plenty of time to try things out in practise, and gain better understanding for the exam.



# Course contents

---

- **16 lectures** (2h each)  
New theory, and project introductions
- **Two computer lessons** (2h each)  
How to use Python for computer vision. Hints for the exercises.
- **Four computer exercises** (4h each)  
Practical experience of critical skills needed for the projects.
- **Written mid-term exam** (4h)  
Voluntary, but very useful
- **Written exam** (4h)  
Only half if you do well on the mid-term exam
- **Two group projects** (50% of course)



# Warning: Labs are hard!

---

- The computer labs are more difficult than in other courses.
- The schedule has 2h booked preparation time per lab. Usually more is needed. An estimate of the preparation time is provided in each lab sheet. This is **assuming that you have already done all parts of the computer lessons** and understand them fully.
- Labs are however very useful for the projects, for future project courses, masters thesis projects etc.



# How do I pass the course?

---

- **Pass the four computer exercises**  
come prepared (spend at least the 2h booked in the lab)
- **Pass the two group projects**  
code in svn, report, oral presentation
- **Pass the written exam**  
midterm exam can give you bonus points





# What grade will I get?

---

- If you have passed the course, the grade is a weighted average of the project grades and the exam grade:

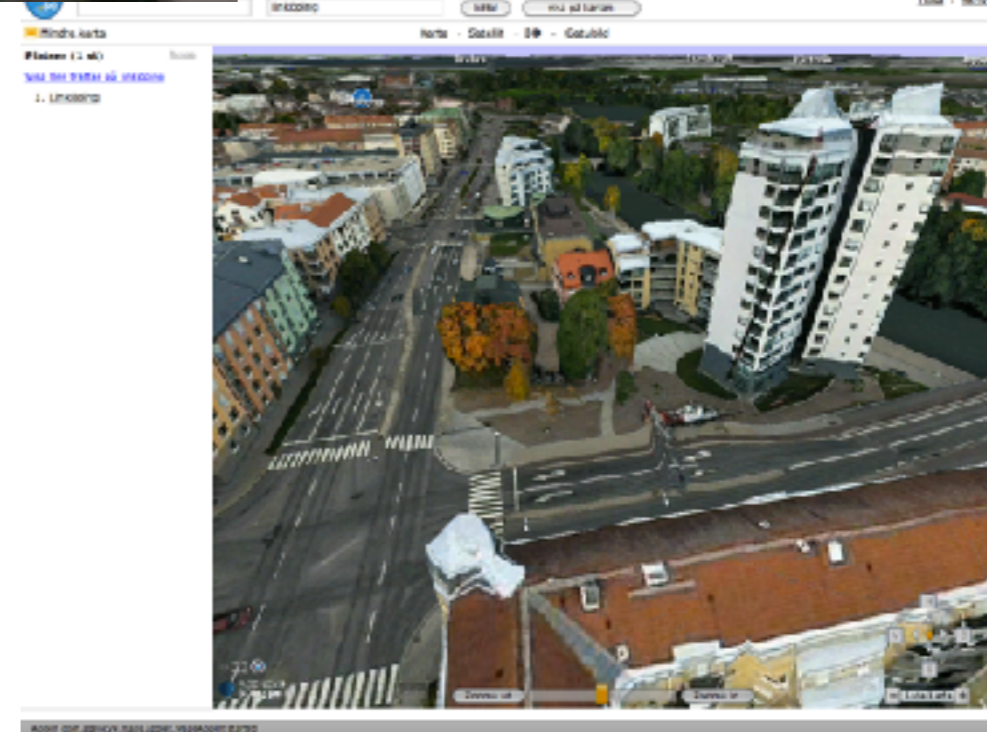
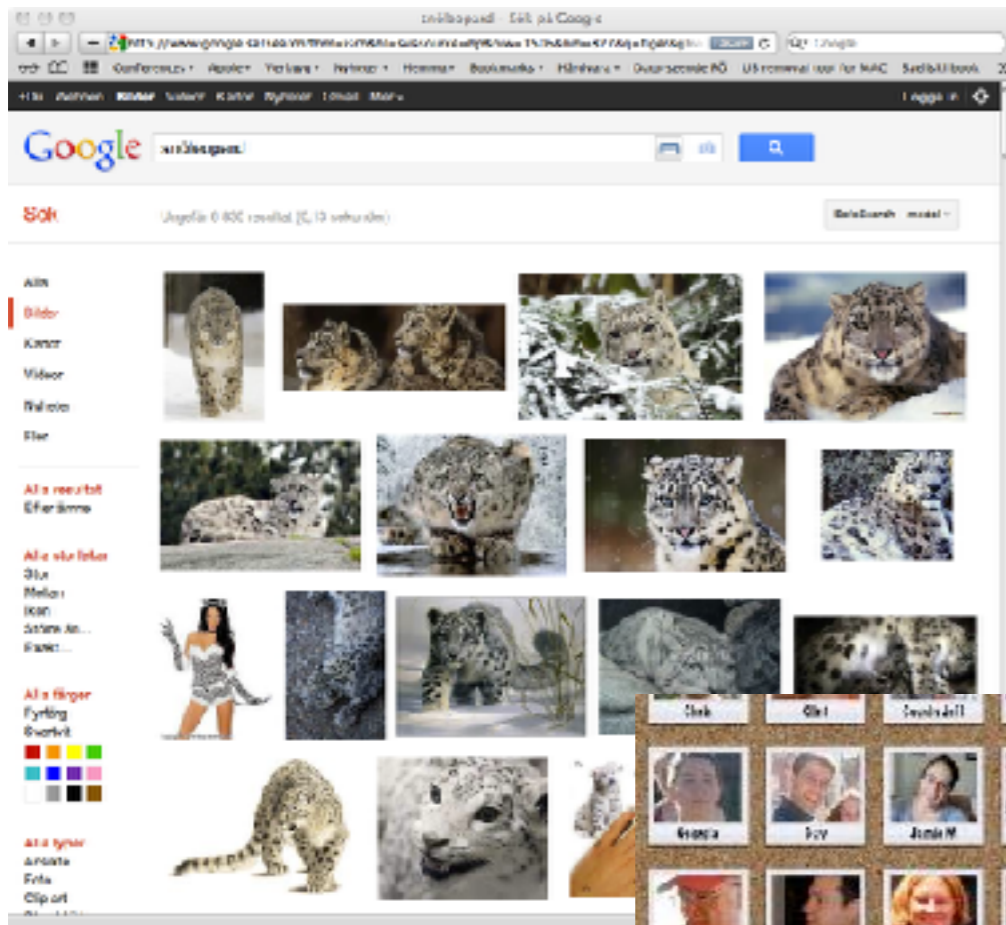
```
total_grade = np.round(  
    0.48*exam_grade  
    + 0.26*project1_grade  
    + 0.26*project2_grade)
```

- Project grade 4 if all requirements are satisfied  
Project grade 3 if minor errors  
Project grade 5 if an individual extra assignment is completed



# Applications

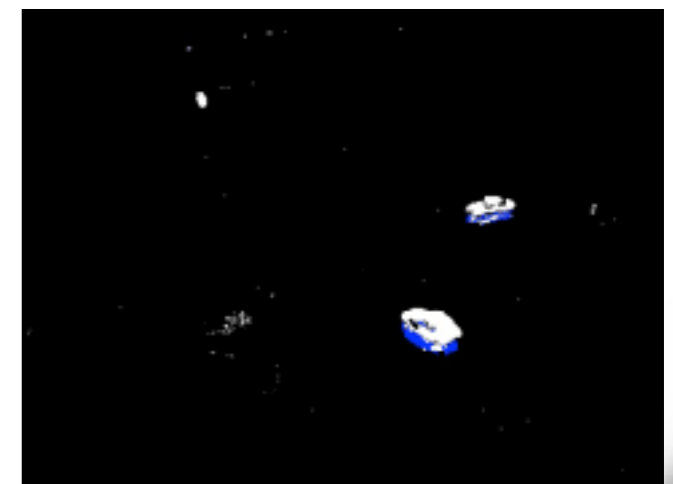
- Google, Apple (w. C3 and Polar rose), Facebook, MS





# Applications

- Hollywood, Zenuity (Volvo+Autoliv) Bosch/BMW/Google/Tesla, IVSS (e.g. Vägverket)





# MATRIS (2004-2007)

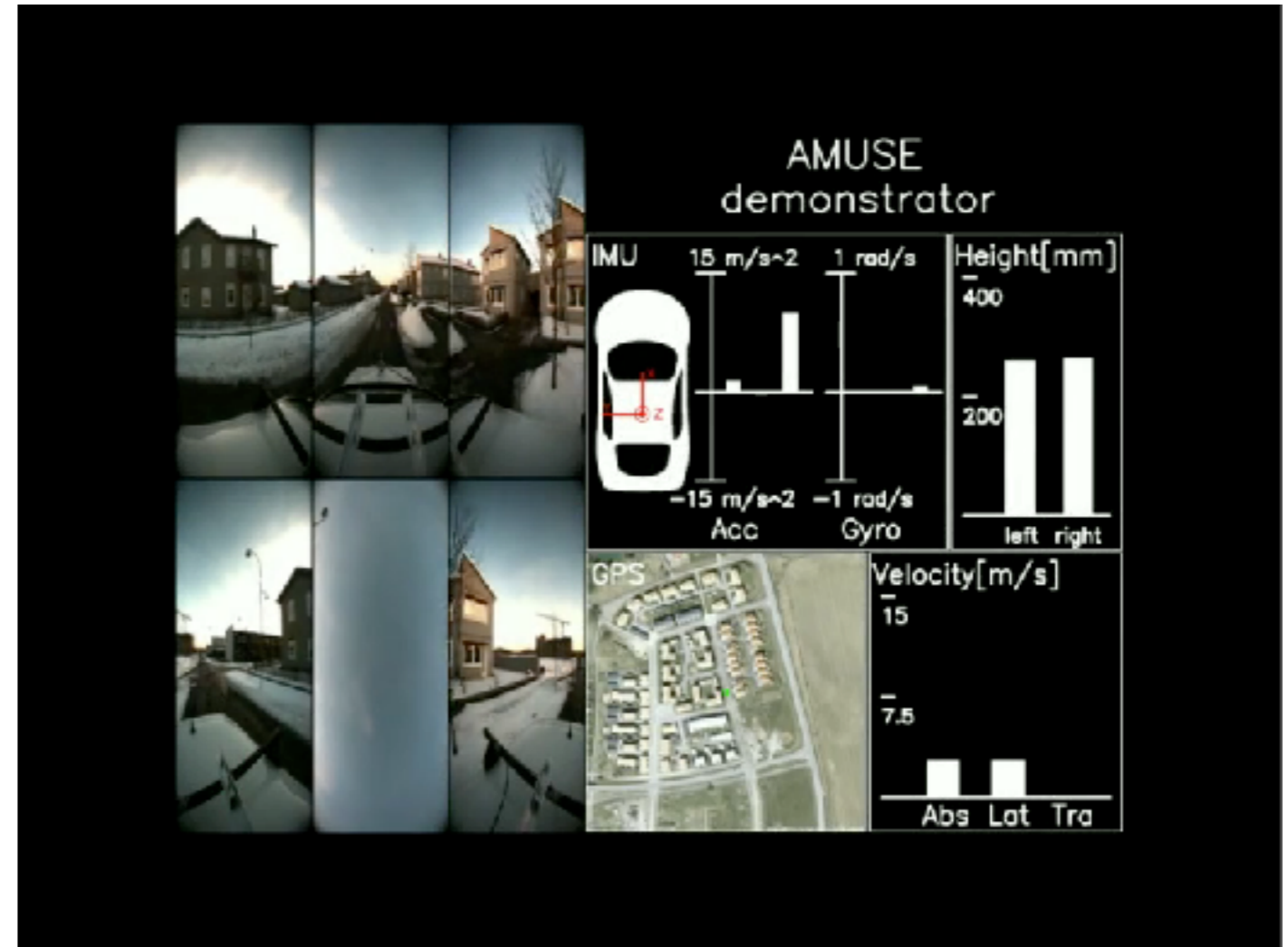


Markerless Real-Time Tracking for  
Augmented Reality Image Synthesis



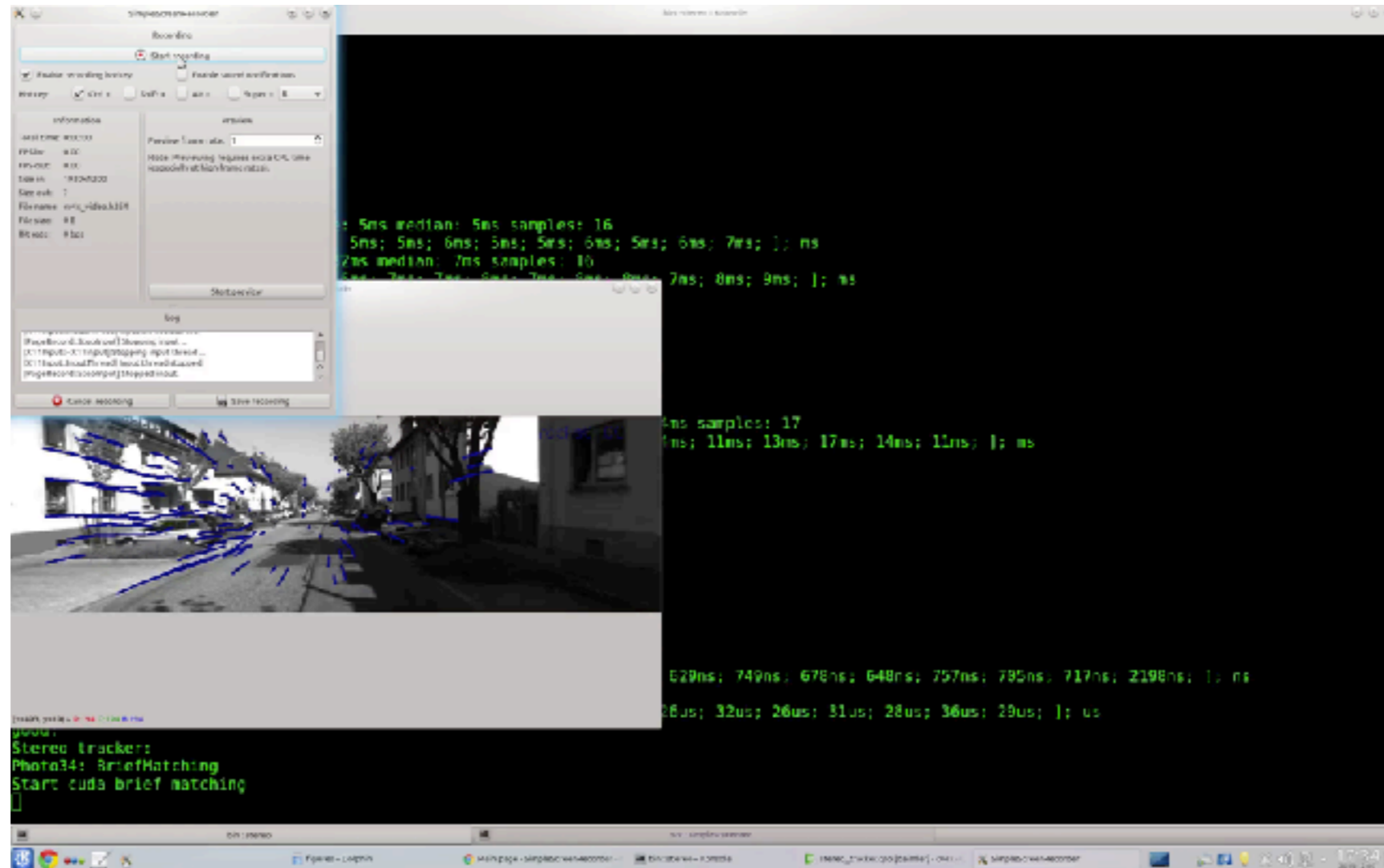
# Visual perception in cars

- EU-project DIPLECS (2007-2010)
- VINNOVA IQmatic (2013-2016)
- Daimler collaboration





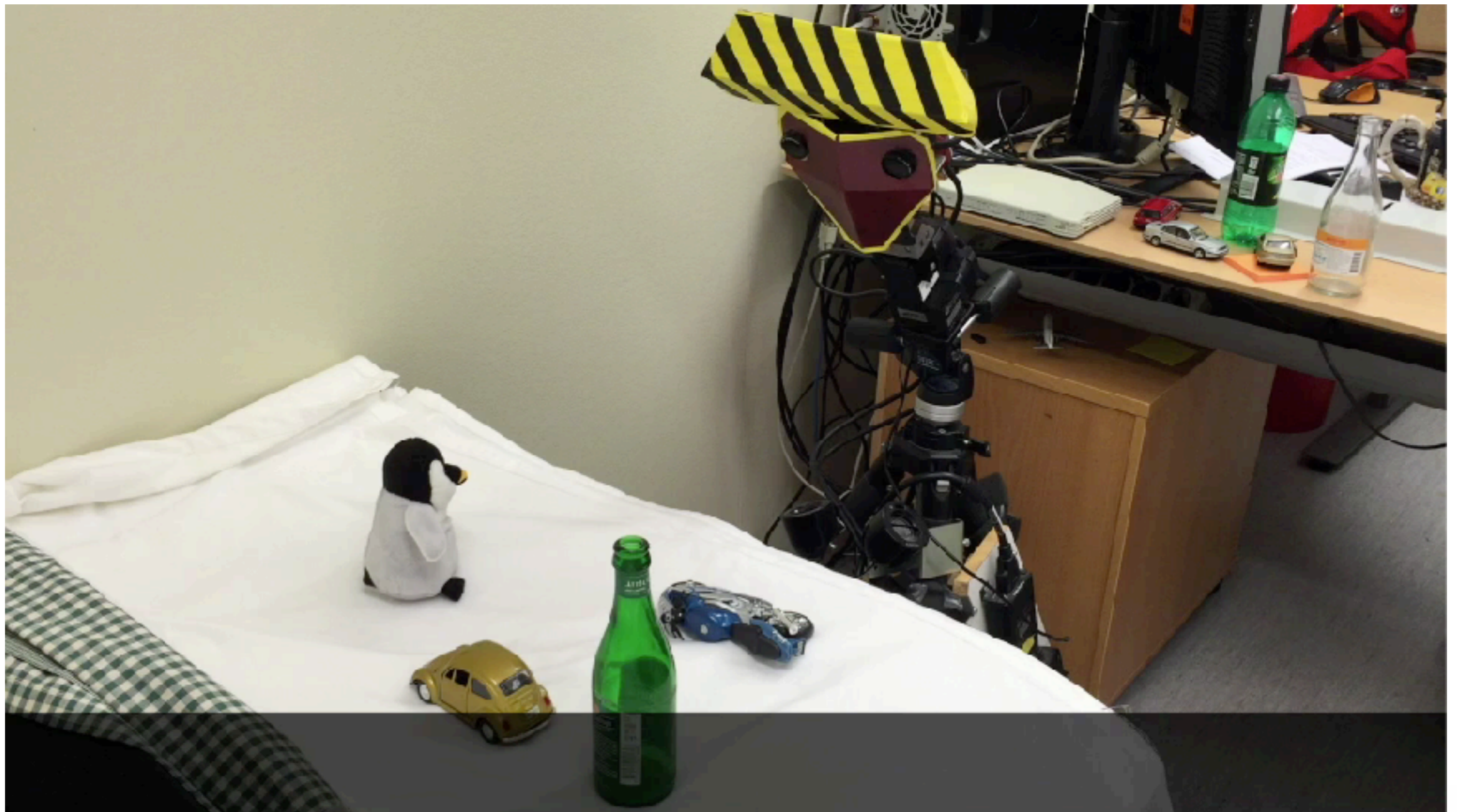
# Visual perception in cars



- Realtime SfM for egomotion from IQmatic project.



# Embodied Object Recognition 2009-2014





# Rolling Shutter Correction 2009-

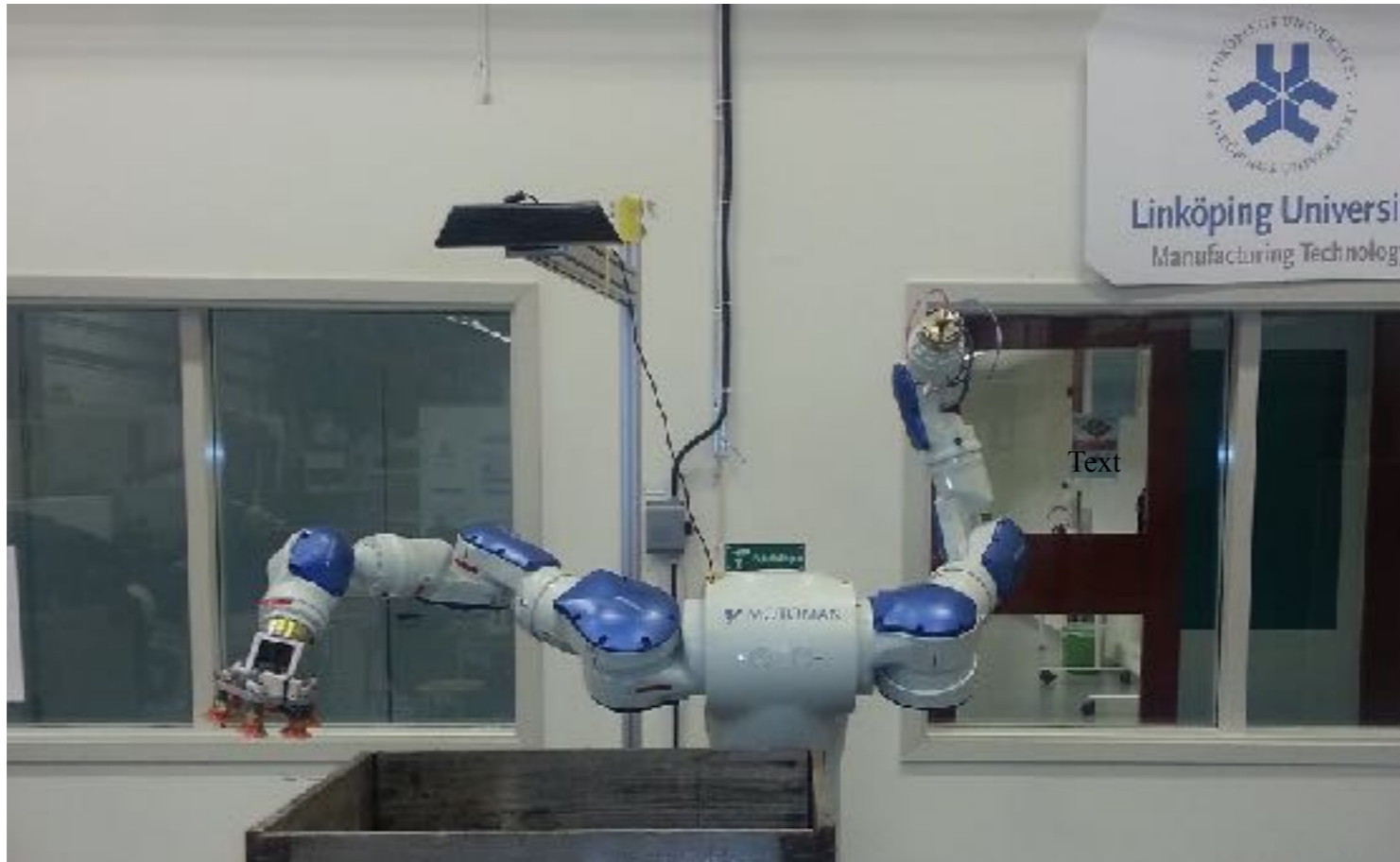


- LCMM (Learnable Camera Motion Models) project





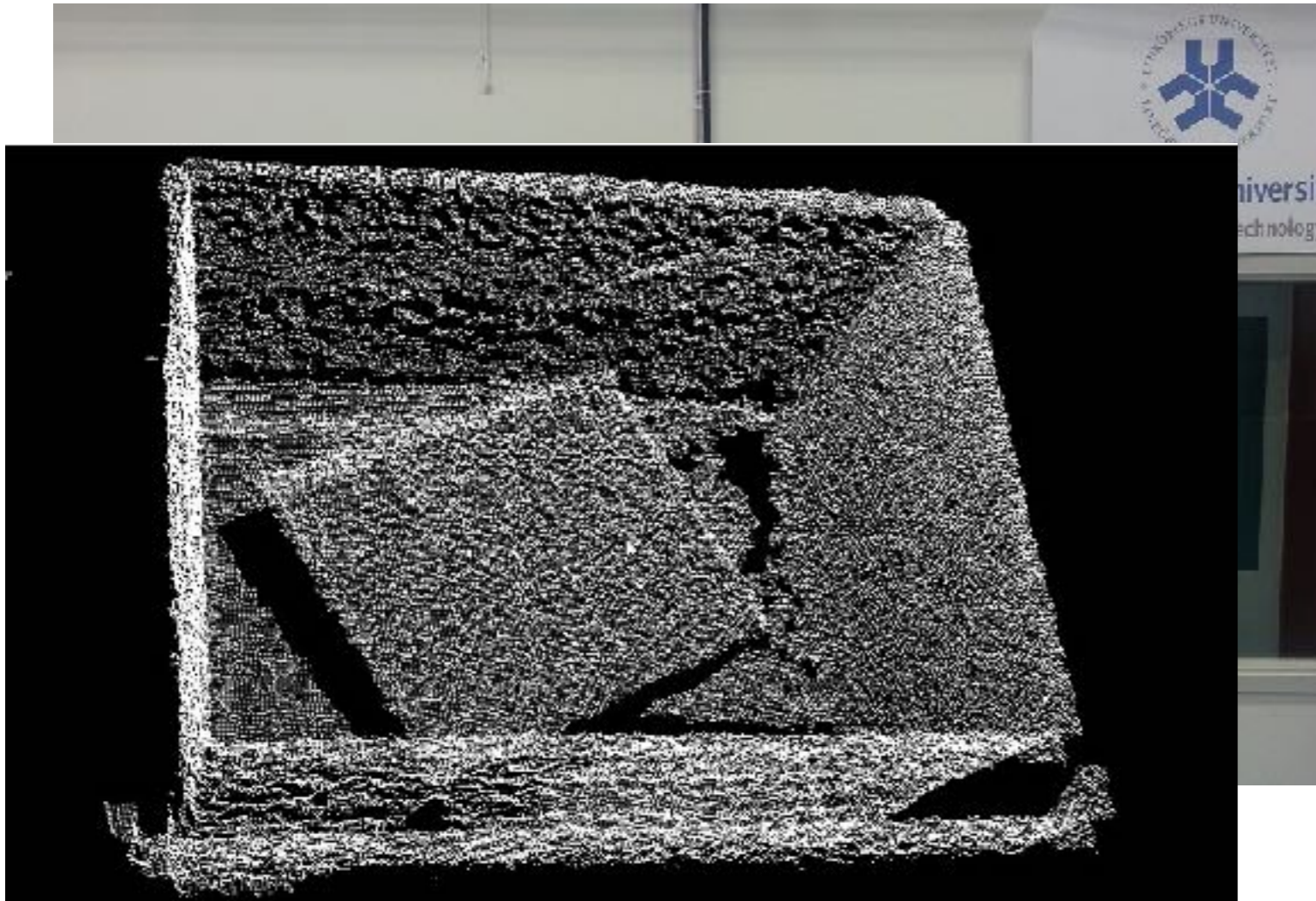
# CDIO Project Example (2012)



- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.



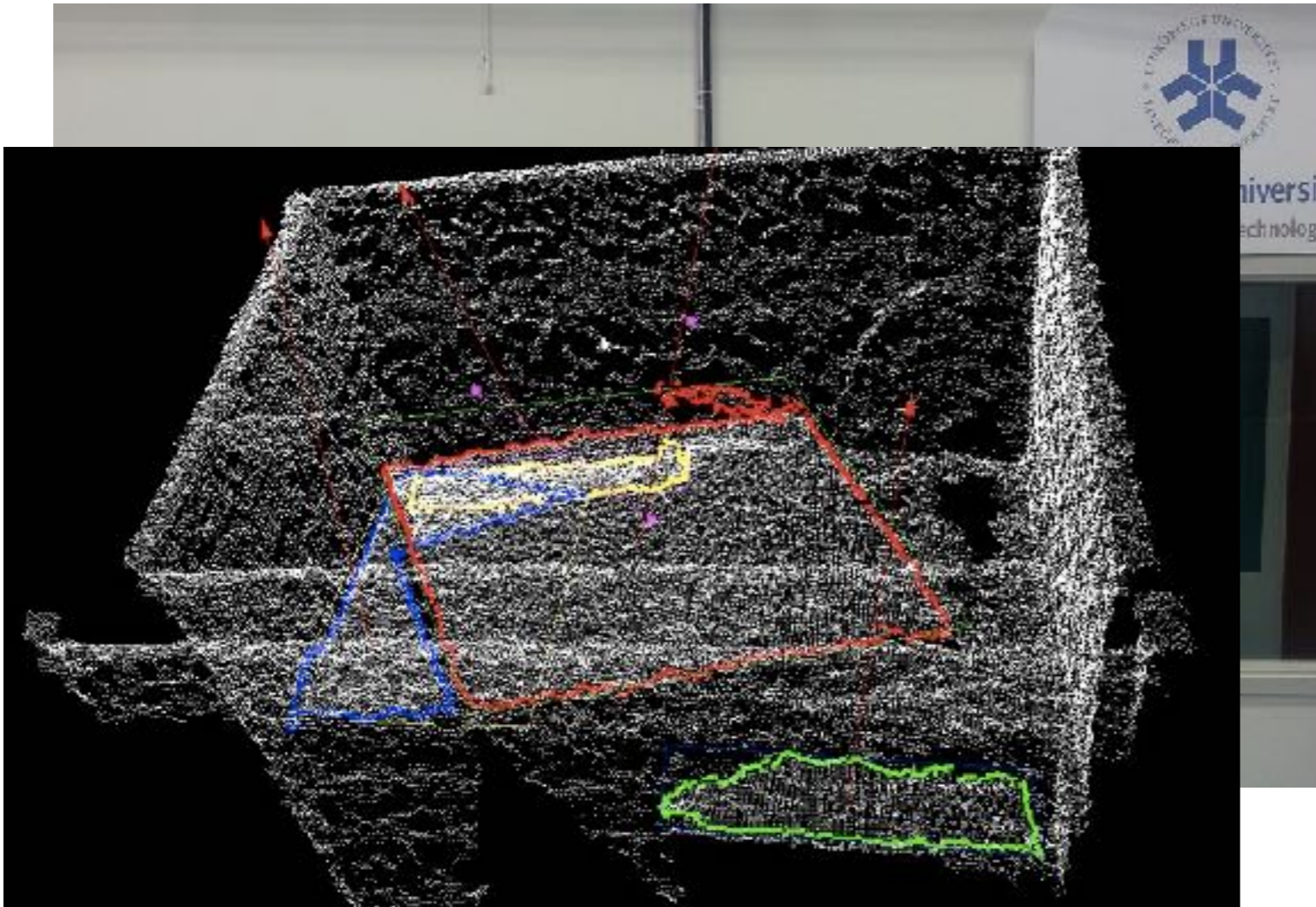
# CDIO Project Example (2012)



- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.



# CDIO Project Example (2012)

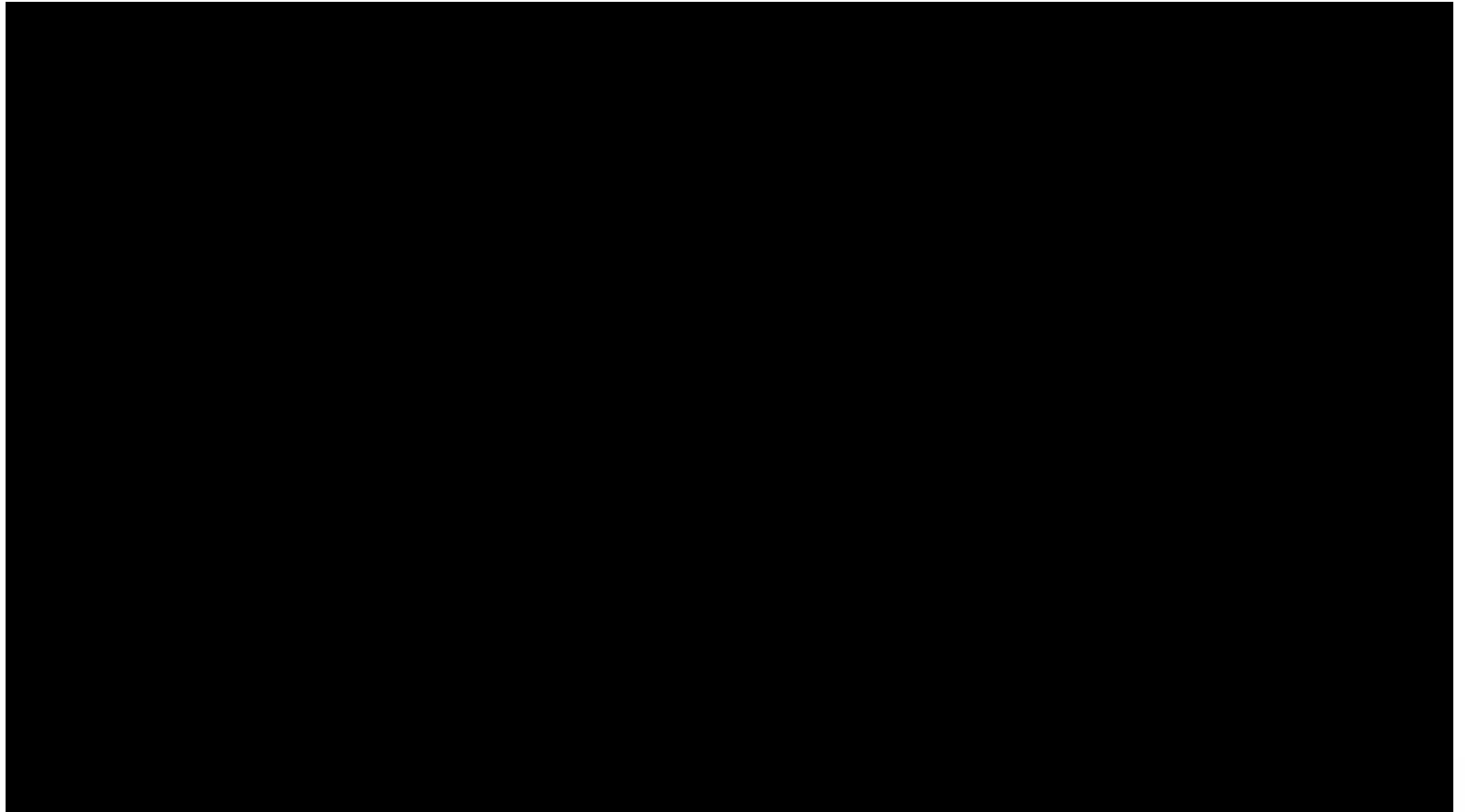


- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.



# CDIO Project Example (2012)

---





# CDIO Project Example (2012)



- Robot Dog  
the robot should be able to recognize and to follow a specific person.



# CDIO Project Example (2012)





# Part II: Methodology

---

- Local models
- Features (detector+descriptor)
- Representations
- Estimation



# Local models

- Here is an image that we want to analyze:



- How do we begin?





# Local models

- Global approach:  $\mathbf{I}(\mathbf{x}) \quad \mathbf{x} \in \Omega \subset \mathbb{Z}^2$

stack all pixels in a vector:

$$\mathbf{v} = [\mathbf{I}(\mathbf{x}_1) \dots \mathbf{I}(\mathbf{x}_k) \dots \mathbf{I}(\mathbf{x}_K)] \quad \mathbf{x}_k \in \Omega$$





# Local models

- Global approach:  $\mathbf{I}(\mathbf{x}) \quad \mathbf{x} \in \Omega \subset \mathbb{Z}^2$

stack all pixels in a vector:

$$\mathbf{v} = [\mathbf{I}(\mathbf{x}_1) \dots \mathbf{I}(\mathbf{x}_k) \dots \mathbf{I}(\mathbf{x}_K)] \quad \mathbf{x}_k \in \Omega$$



- A local model instead looks at limited neighbourhoods.
- These may be approximated by a simple model. (e.g. the signal is locally 1D)
- Each model tells us a little bit about the image.





# Locality

- **Local appearance**  
describe the appearance for relatively small image regions
- **Local processing**  
processing of images is done locally, i.e. the result at a certain image point depends only on the input data in a neighborhood of the point.  
⇒ Suitable for implementation on graphics cards.



# Locality

- This is also how the brain does it (LE9).
- Locality in a resolution pyramid (LE2).
- Convolutional Neural Networks etc.

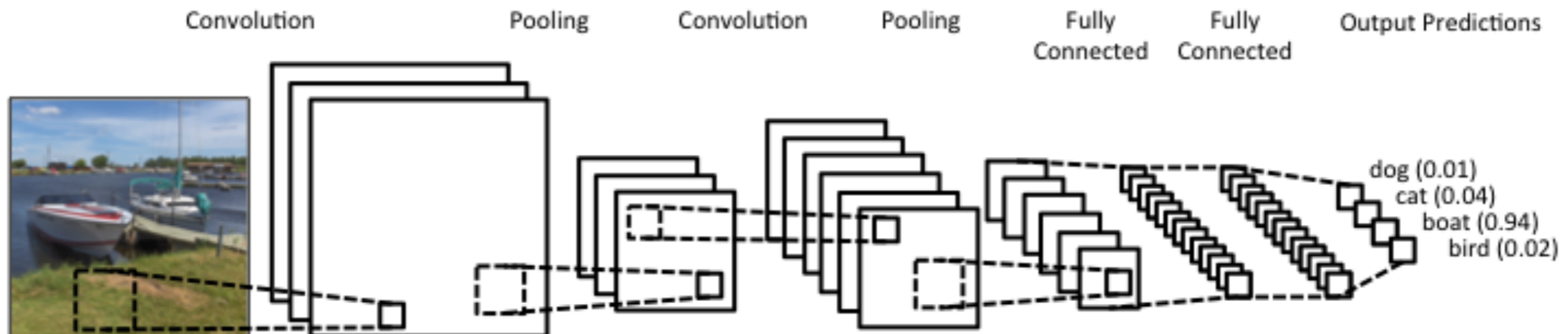
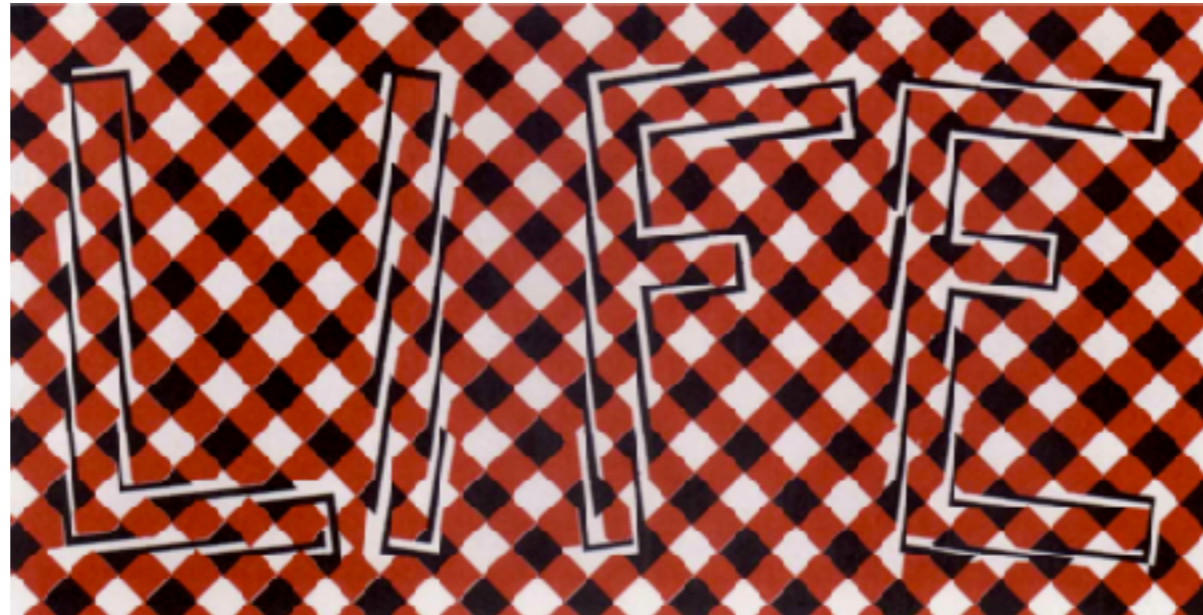


Image source: <http://www.clarifai.com/technology>



# Locality & Orientation (illusion)

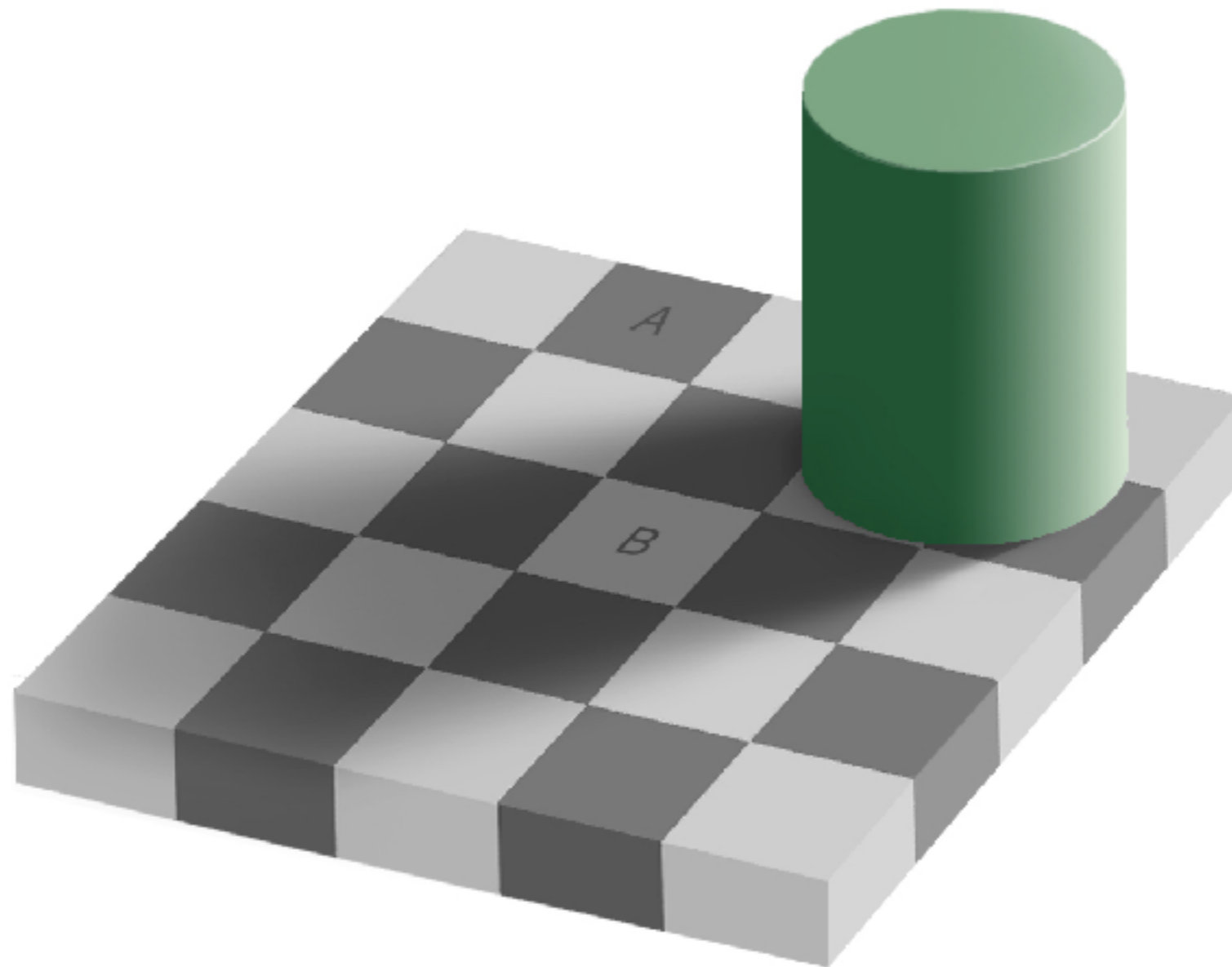
---





# Locality & Intensity (illusion)

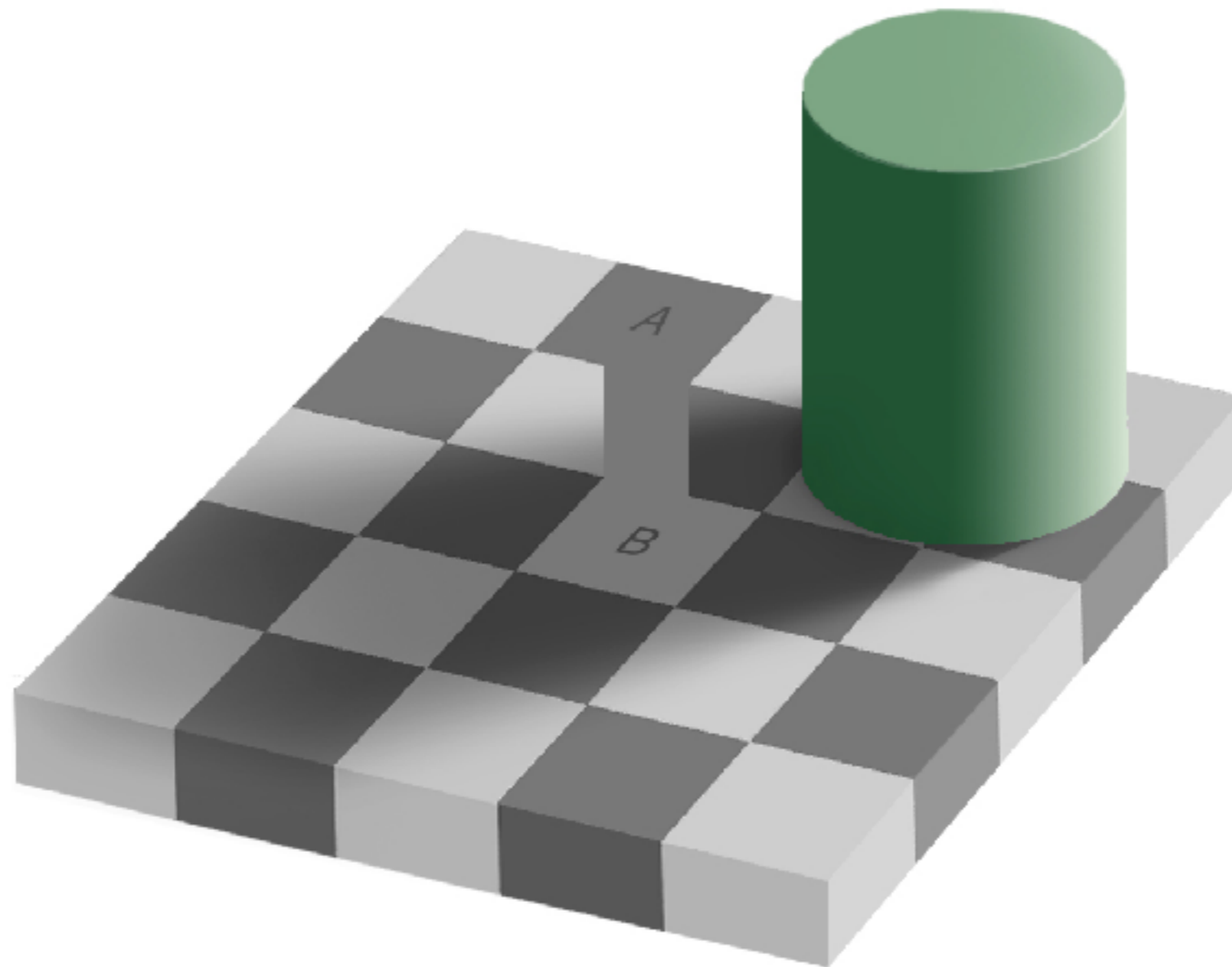
---





# Locality & Intensity (illusion)

---





# Locality & Motion – Aperture







# Methodology

---

- Local models
  - Signal model: intrinsically 1D, band-limited, ergodic
  - Alternative: local model learned from data
- Camera models (global)
  - pinhole model
  - multiple views will be used



# Example: i1D-signals

---

- Local neighbourhoods  $f$  which contain edges or lines can be described by

$$f(\mathbf{x}) = g(\mathbf{x}^T \hat{\mathbf{n}})$$

- where  $g$  is a function of a single variable
- $\mathbf{n}$  is a normal vector
- $f$  is called *simple* signal or **intrinsically one-dimensional signal** (i1D)



# Example: i1D-signals

---

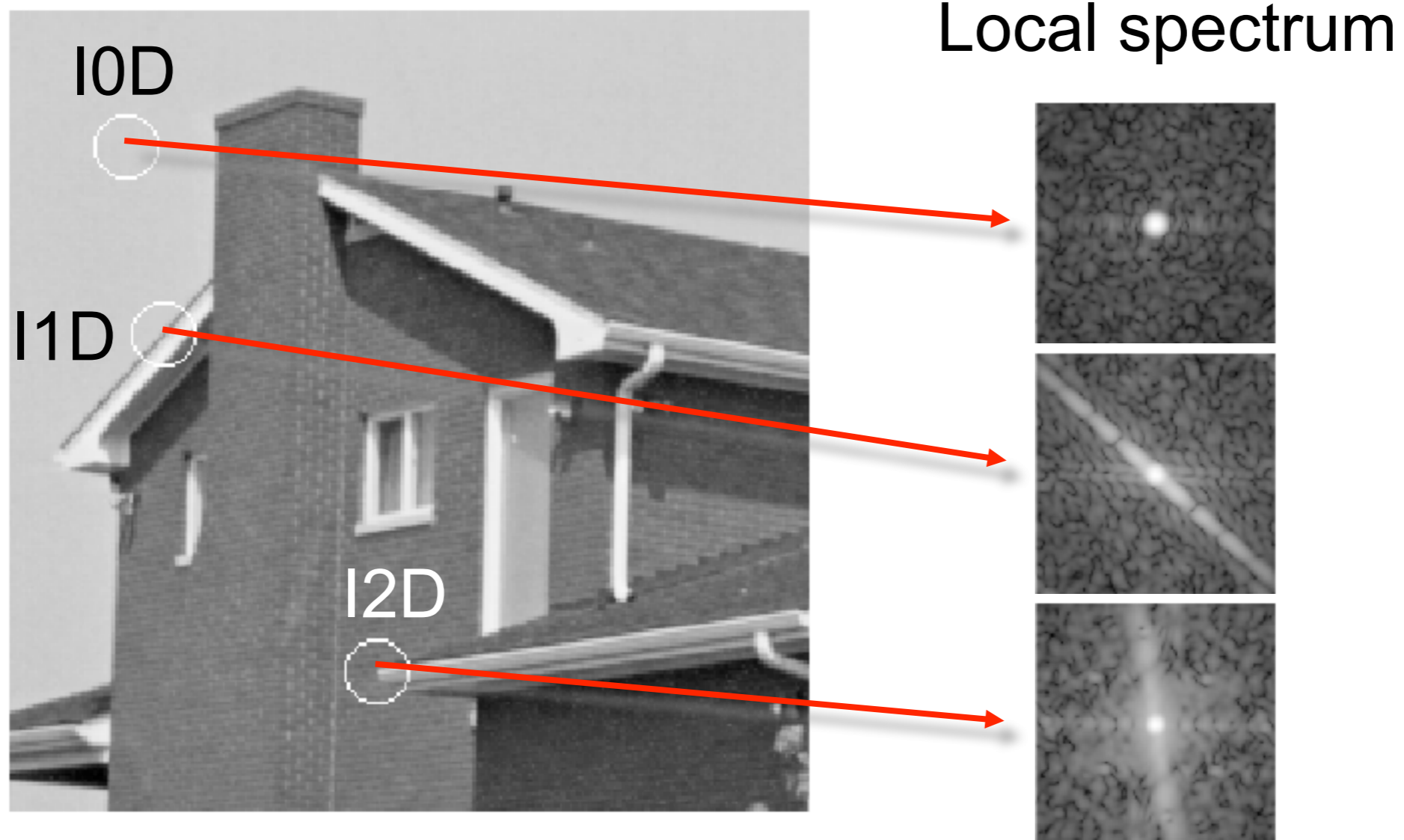
- For an i1D-signal we know that its *Fourier transform* reads

$$F(\mathbf{u}) = (2\pi)^{n-1} \delta_{\hat{\mathbf{n}}}^{line}(\mathbf{u}) G(\mathbf{u}^T \hat{\mathbf{n}})$$

- an impulse line which goes through the origin with orientation  $\hat{\mathbf{n}}$  and which varies as  $G$ , the Fourier transform of  $g$  (**see EDUPACK**)
- $n$  is the dimension of the signal



# The Intrinsic Dimension





# The Intrinsic Dimension

---



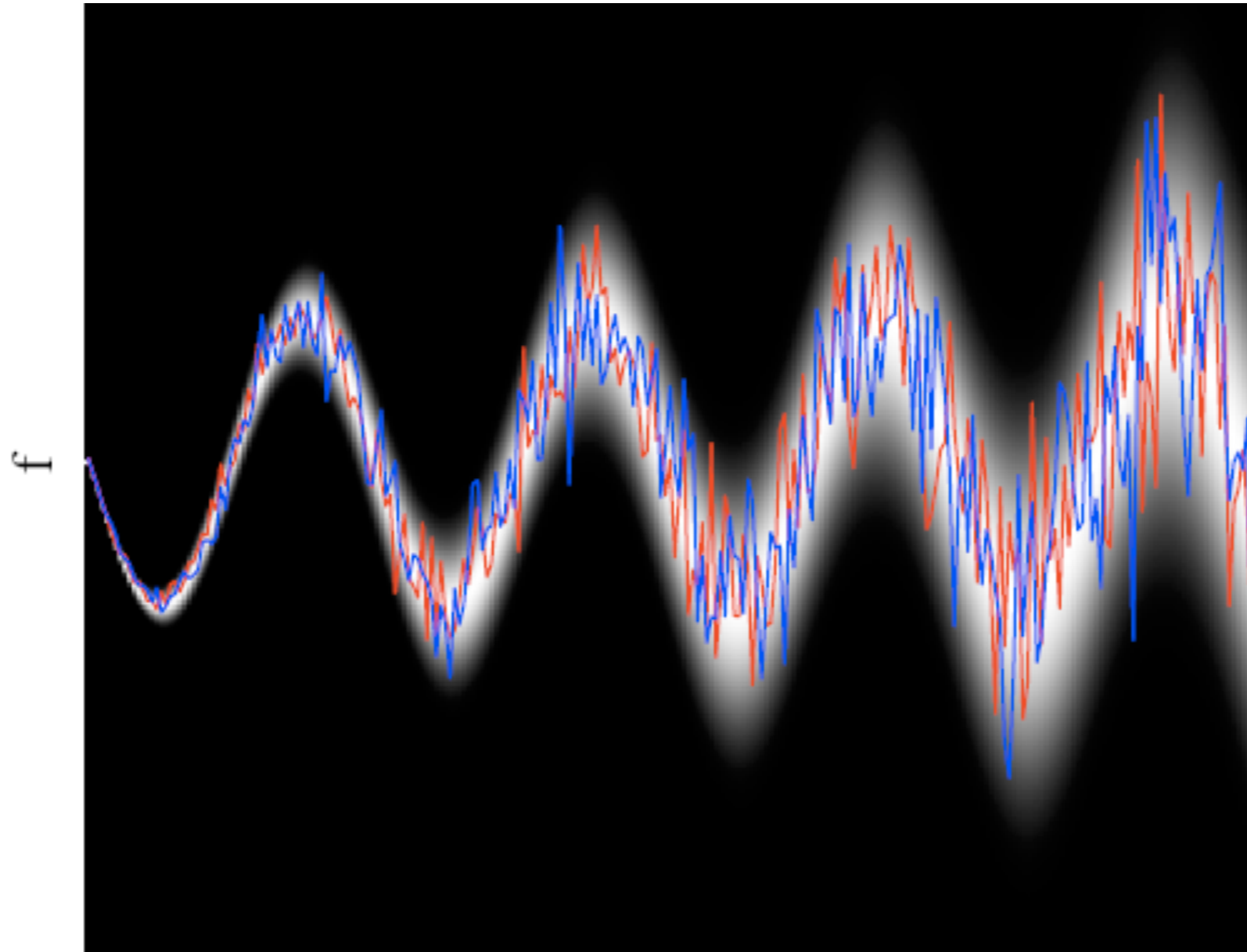
i0d

i1d

i2d



# Ergodic Signal





# Methodology

---

- Examples of local features:
  - Local orientation, motion, curvature, color, symmetry, phase...



# Methodology

---

- Examples of local features:
  - Local orientation, motion, curvature, color, symmetry, phase...
- How local is local?
  - What locality is best depends on the signal (E.g. brick wall at coarse scale and fine scale)
  - For better generality one needs to try several choices of locality in parallel (LE2)





# Example: Local Orientation

- A local signal which is 1D has a natural notion of orientation
- The signal is constant in all directions which are orthogonal to the vector  $\mathbf{n}$





# Representations

- A **representation** is a mathematical description of a feature
- A feature can have one or several representations, for instance
  - Real or complex numbers (scalars)
  - Vectors
  - Matrices
- A collection of one or more features for a particular image object or region is called a **descriptor**



# Example: Local Orientation

- For the feature local orientation, the vector  $\mathbf{n}$  is a possible representation.
- BUT: for an 1D-signal,  $\mathbf{n}$  is not unique, also  $-\mathbf{n}$  can be used!
- **Unique** representations are normally preferred





# Example: Local Orientation

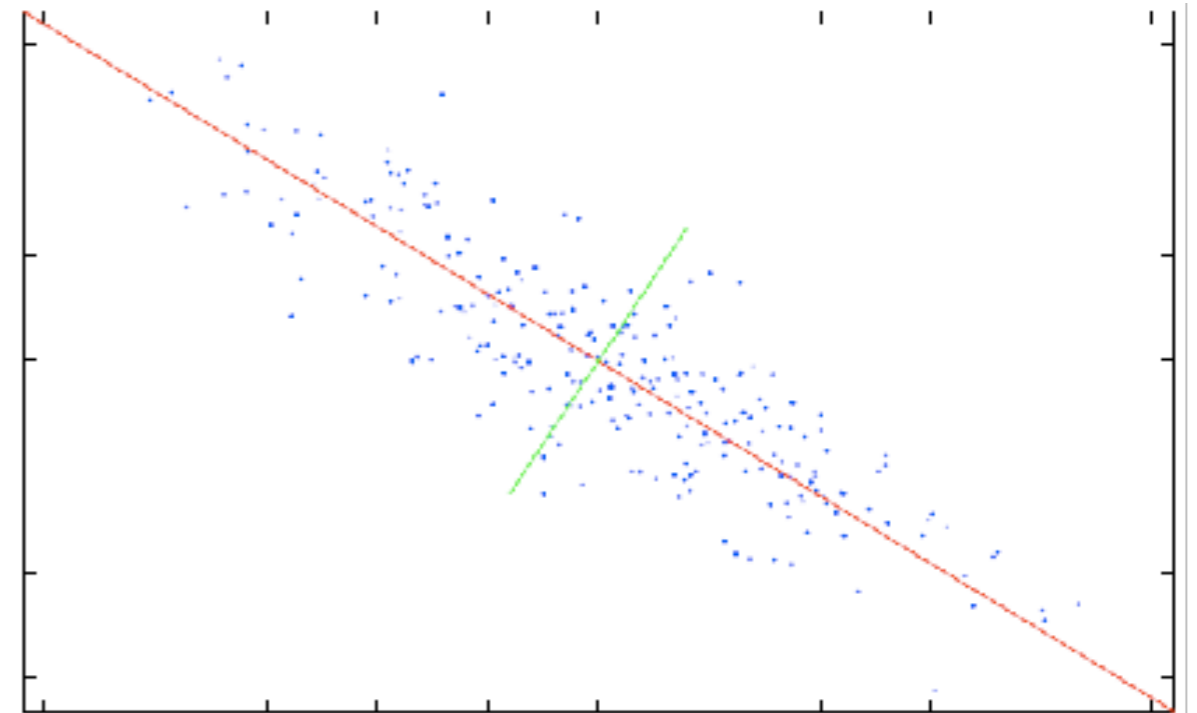
---

- In the 2D case, an angle in the interval  $0^\circ$ - $180^\circ$  between the vector  $\mathbf{n}$  and the horizontal axis would be unique.
- BUT: if  $\mathbf{n}$  is approximately horizontal, very small changes of orientation result in the angle jumping between  $0^\circ$  and  $180^\circ$ .
- Smoothly varying representations are also desirable.



# Representations

- The **confidence** of a signal is a measure of how well it fits the assumed model. It does not depend on the feature value
- E.g. for a fitted line, the inverse variance orthogonal to the line can be used
- The confidence value can be used for weighted averaging (maximum likelihood estimation)

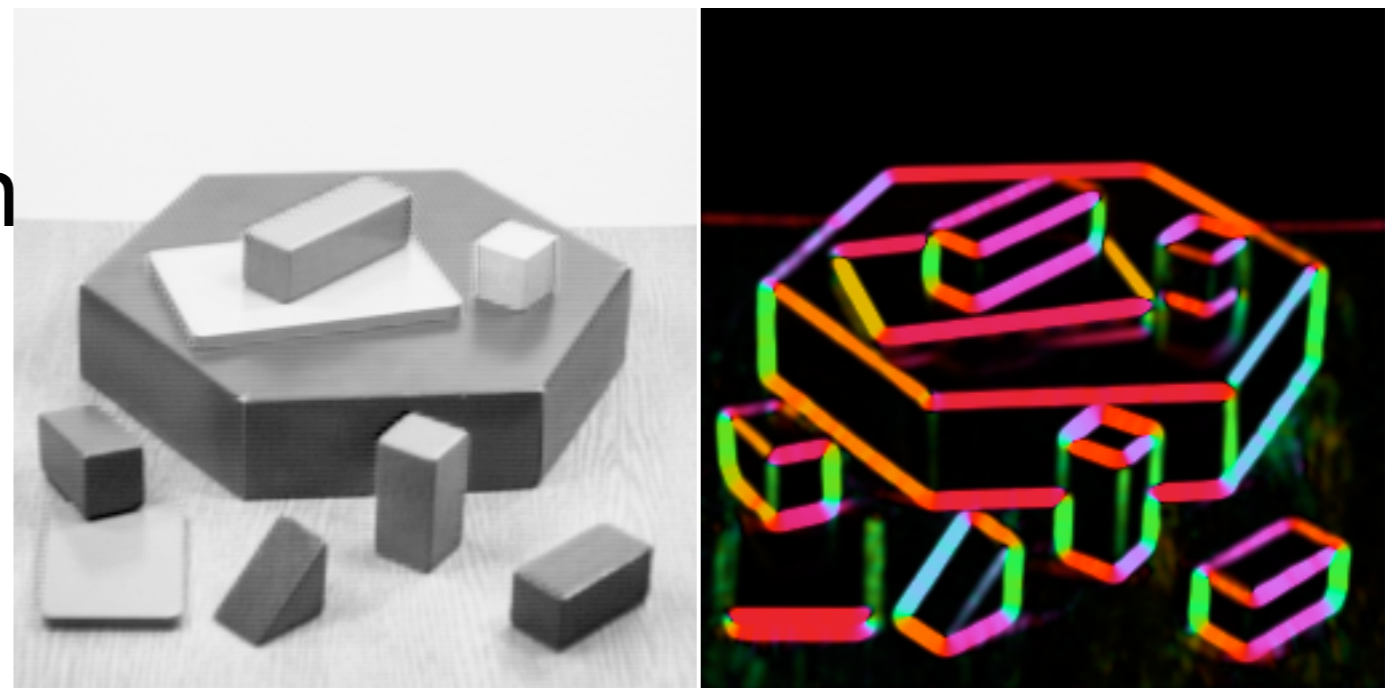


$$\hat{\mathbf{x}} = \frac{1}{K} \sum_k \mathbf{x}_k \quad \text{vs.} \quad \hat{\mathbf{x}} = \frac{1}{\sum_k w_k} \sum_k w_k \mathbf{x}_k$$



# Example: Local Orientation

- For representation of orientation (i.e. the vector  $\mathbf{n}$ ), we need a confidence measure that is related to the fitness of the corresponding local signal to the i1D model. Because some regions do not have a unique orientation.
- e.g. the local orientation variance can be used
- We will later introduce several such representations (LE3)





# Estimation

---

- For a given feature representation, we want to compute the numerical descriptor value for the local image region
- This step is called **estimation**
- The same representation can often be estimated in several ways, with different **algorithms**



# Estimation

- Estimation examples:
  - **linear filtering** convolution in the spatial domain and multiplication in the Fourier domain (e.g. gradient)
  - **histogramming** (e.g. colour distributions, and HoG - gradient orientation histograms)
  - solving a **system of equations** (e.g. fundamental matrix estimation)
  - **gradient descent** (e.g. tracking of image regions)





# Estimation

---

- Which algorithm to choose depends on
  - Available computational power (how much computation can be spent on each image)
  - Required accuracy (acceptable measurement error)
  - Required robustness (e.g. handling of outliers in the data)
  - Required dynamics of the feature space (estimation of very large and very small values)



# Example: Local Orientation

---

- Convolve the image with two (or more) filters
- In each image point:
  - Combine the filter responses in a non-linear way to obtain the orientation estimate
  - Combine the filter responses in a non-linear way to obtain a confidence value
- During the course, we will introduce several estimation methods of this type. e.g. in the computer lesson tomorrow.