# TSBB15 Computer Vision

Examiner:

Per-Erik Forssén

# 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.

# Course Information

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


Klas Nordberg


Michael Felsberg


Per-Erik Forssén

+ two special guests

# Course Information

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


Klas Nordberg


Michael Felsberg


Per-Erik Forssén

+ two special guests

- **Computer Exercises and Project Guides**


Kristoffer Öfjäll
CE + PG


Martin Danelljan
CE + PG


Tommaso Piccini
CE


Marcus Wallenberg
CE + PG


Hannes Ovrén
CE + PG

# What is Computer Vision?

- Course aims and structure
  methods, theory, exercises, projects
- Applications
- Research
  Examples from EU projects: MATRIS, DIPLECS
- Examples from CDIO student projects
- Methodology

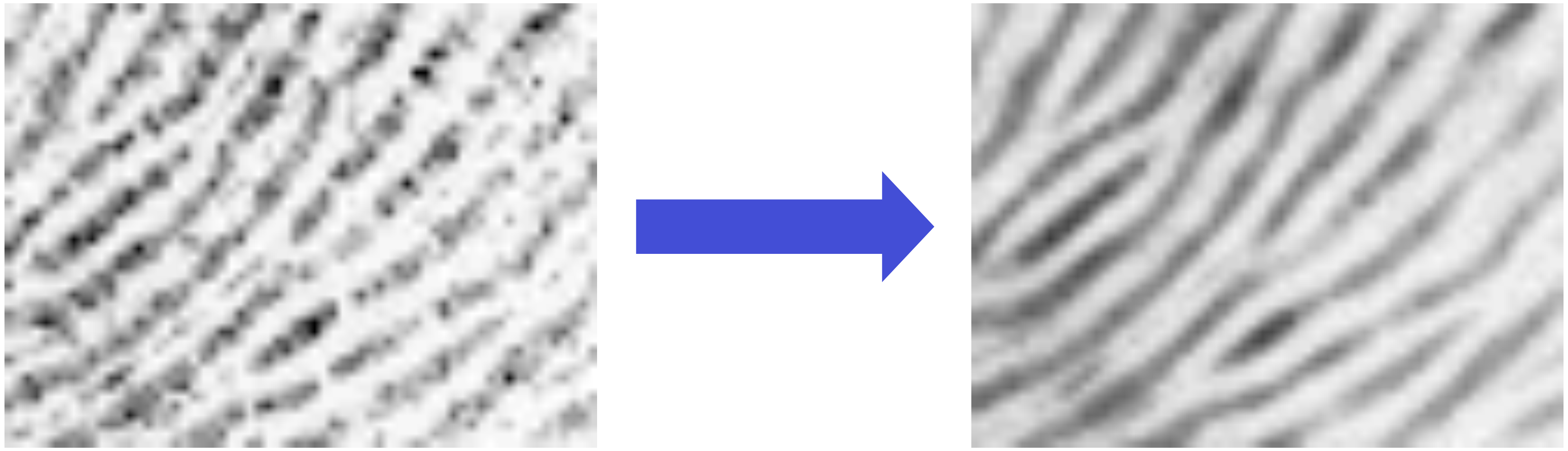
# Course Aims

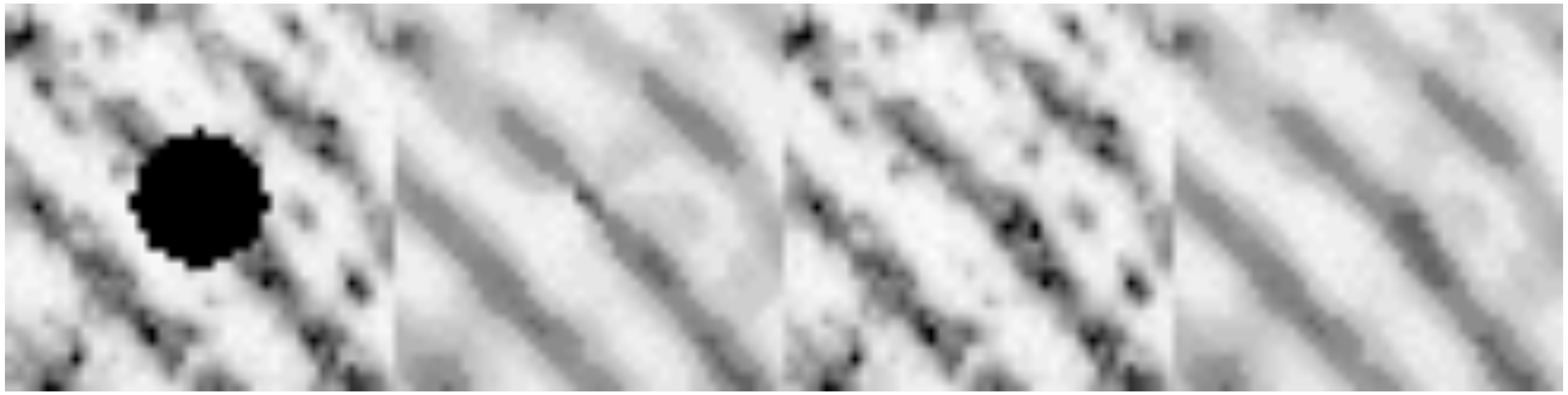
- From the LiTH Study Guide:

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

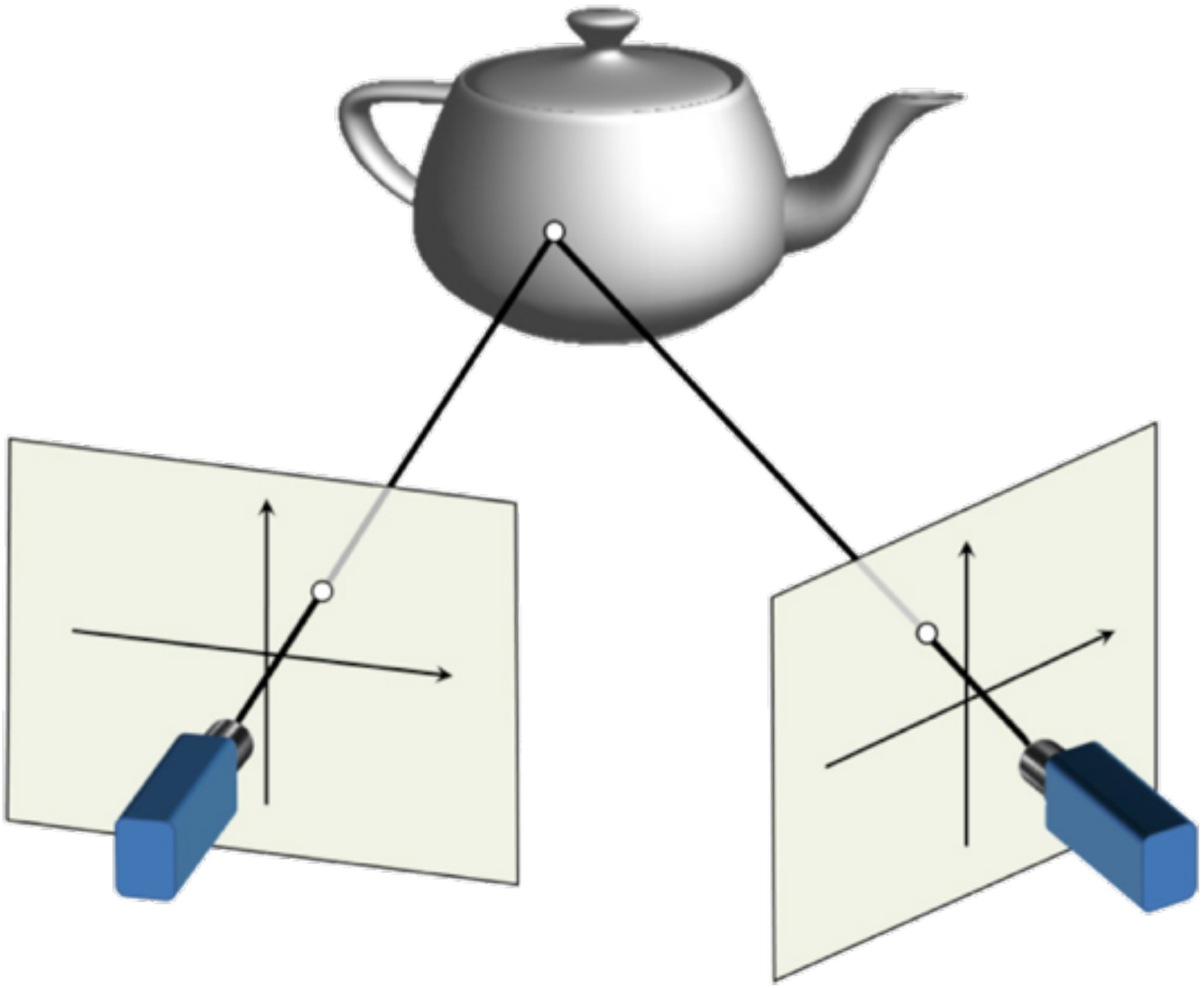
- tracking of image regions, triangulation from stereo images, estimation of optical flow, detection of several image features, matching of image features, graph and tree structures and other image representations, generative image models, segmentation of image regions, enhancement of images, debugging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

# Image Enhancement





# Triangulation

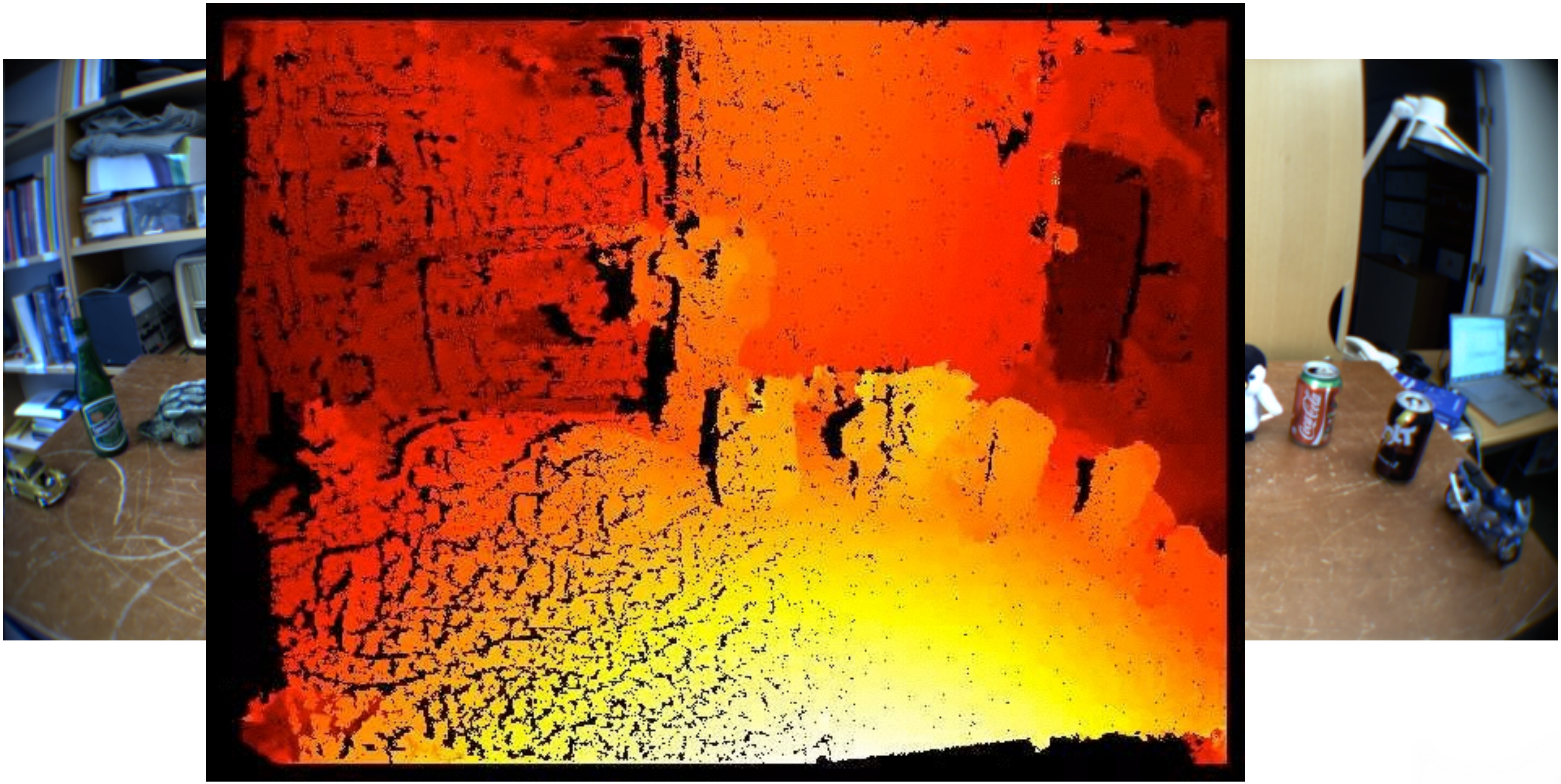


T. Moons et al. FTCGV 2008

# Triangulation

Left view Right view

# Triangulation



Disparity

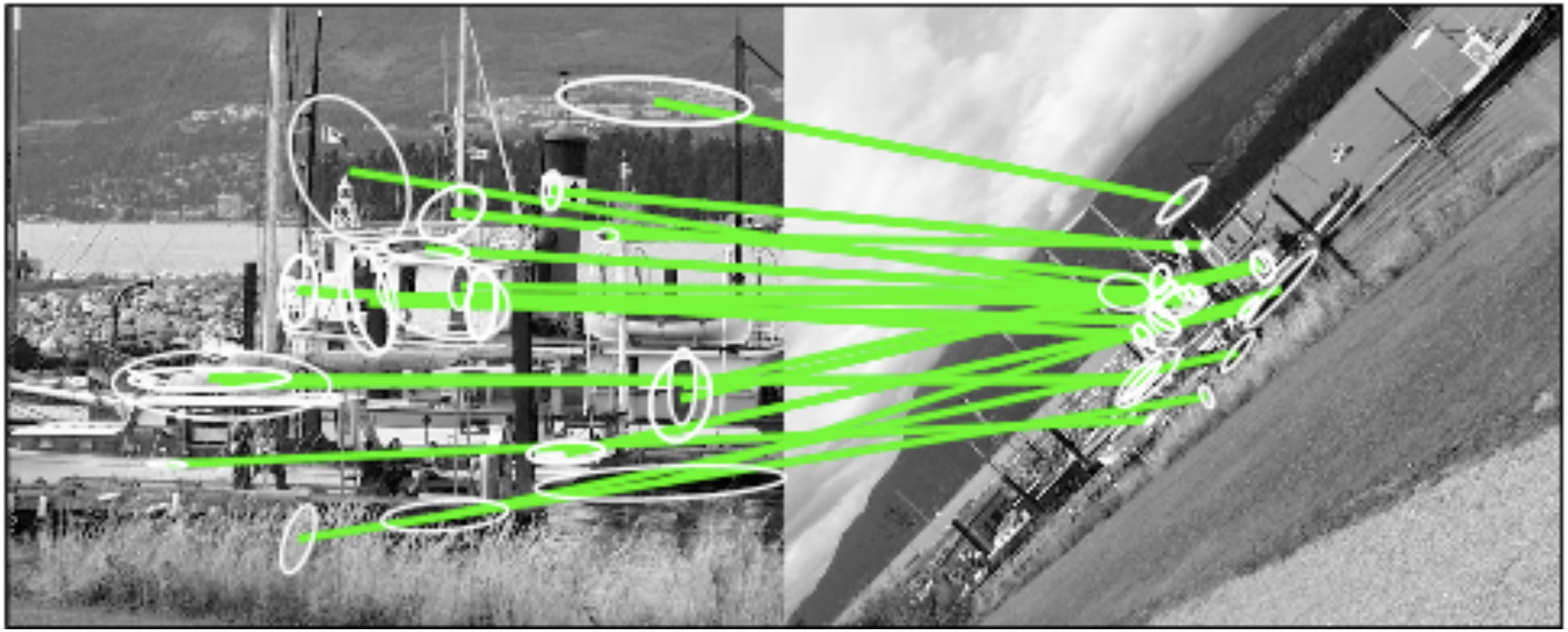
# Tracking

# Motion Estimation

# Feature Matching

Find out which image regions are corresponding

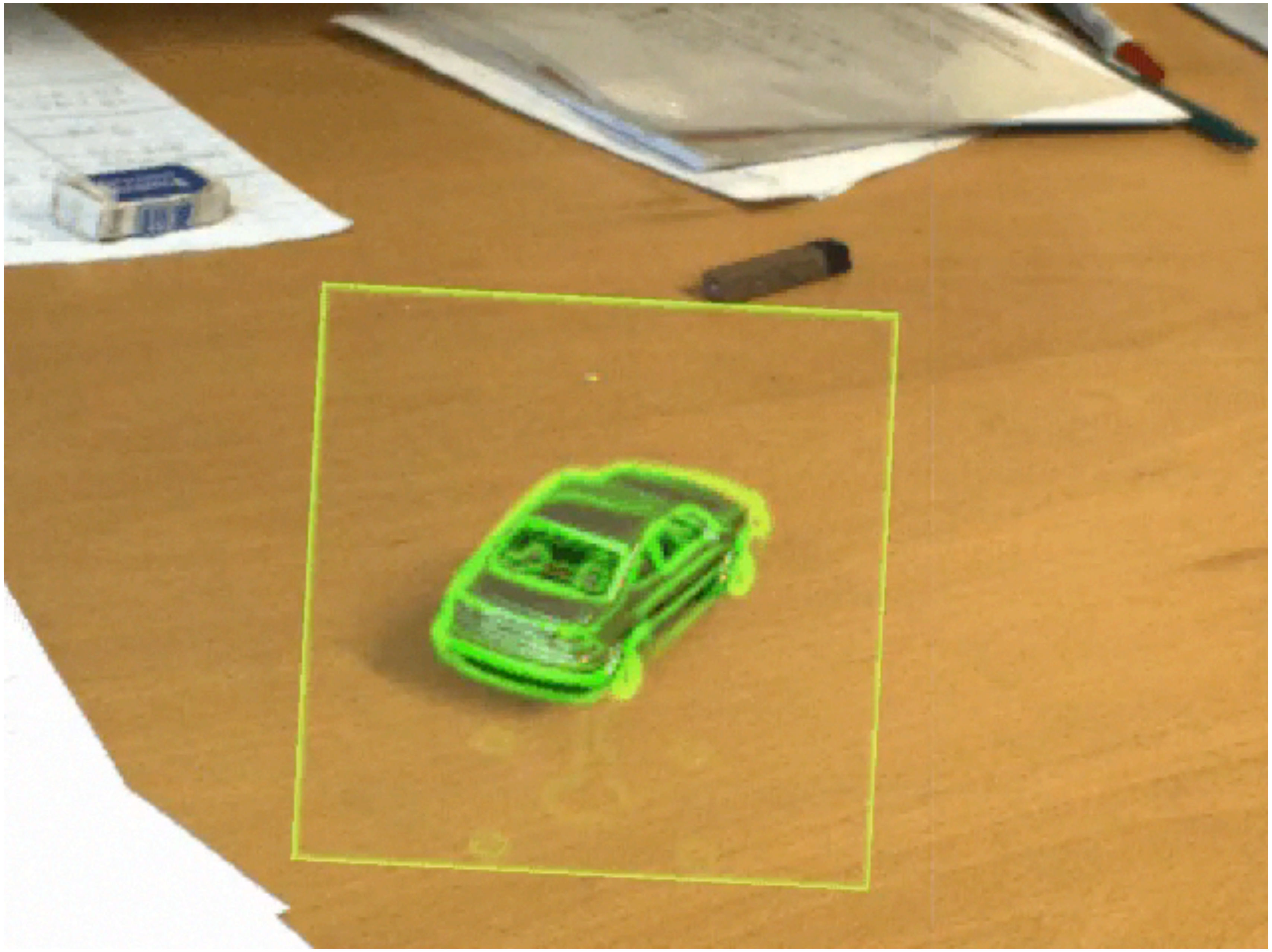
# Feature Matching



Find out which image regions are corresponding

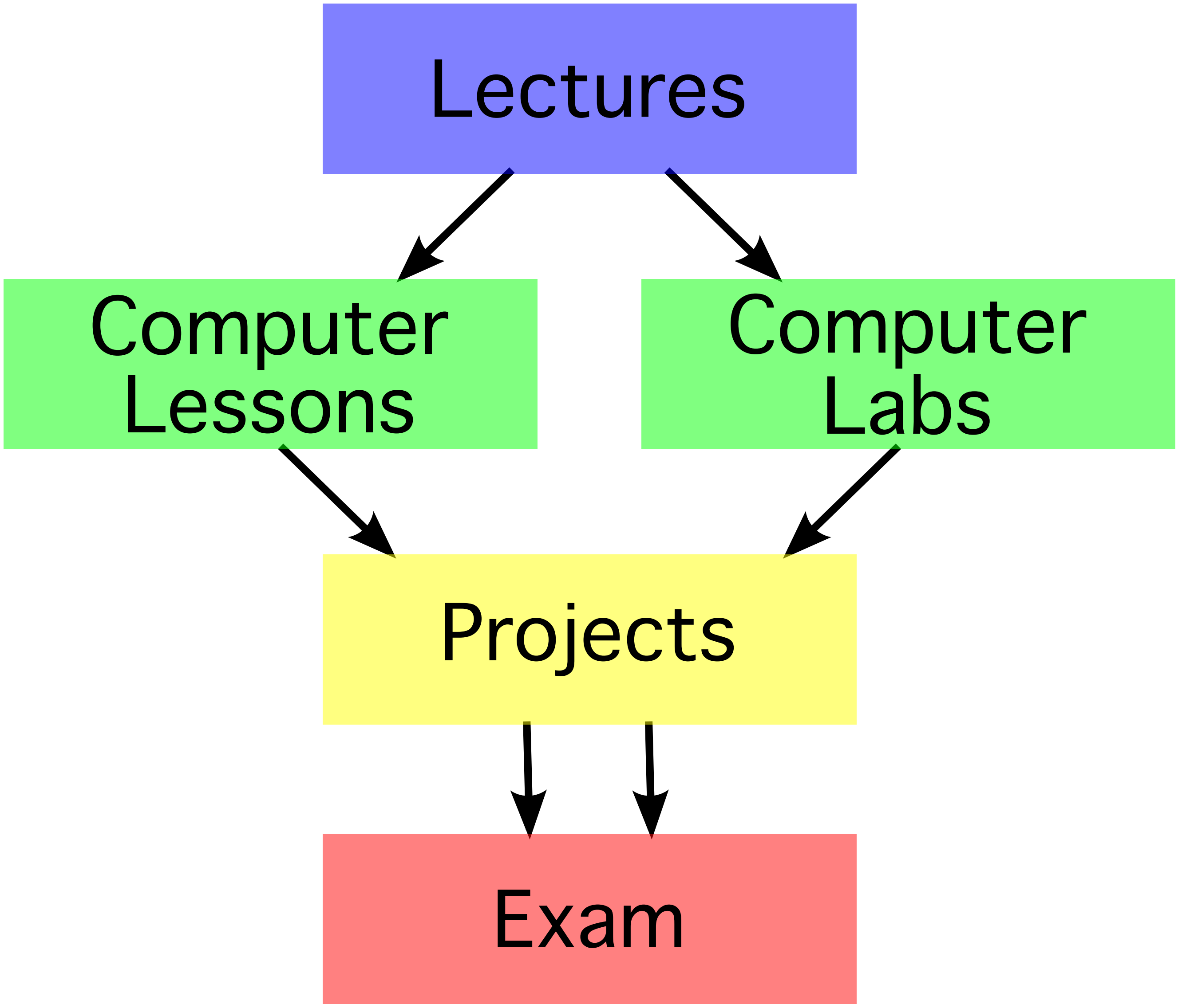
# Object Recognition

# Object Pose estimation



# Object Recognition

# 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 Matlab for computer vision
- **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)

# How do I pass the course?

- **Pass the four computer exercises**
  come prepared (read lab document and solve preparatory questions)
- **Pass the two group projects**
  code in svn, report, oral presentation
- **Pass the written exam**

# What grade will I get?

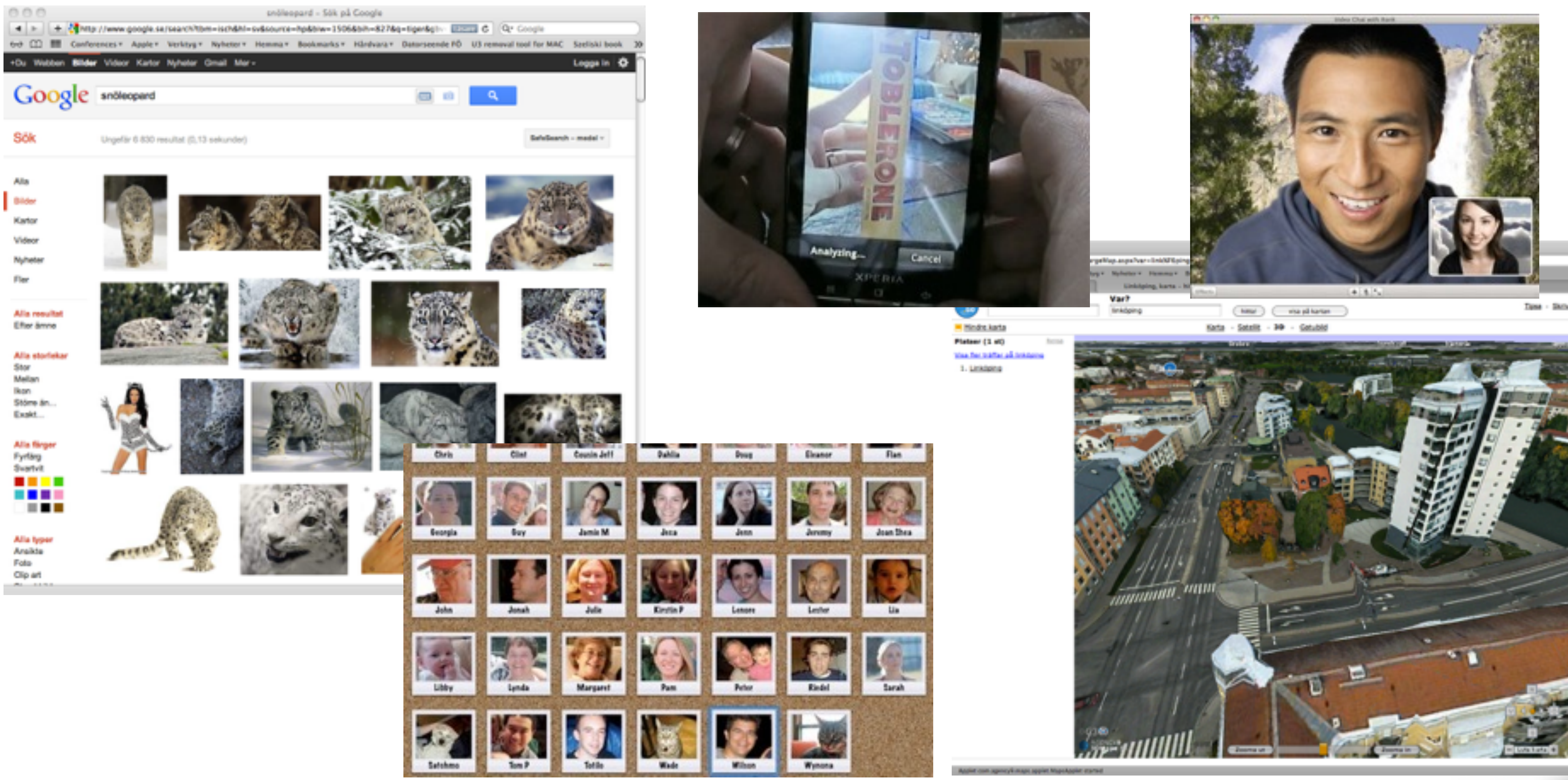
- If you have passed the course, the grade is a weighted average of the project grades and the exam grade:

```
course_grade = 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

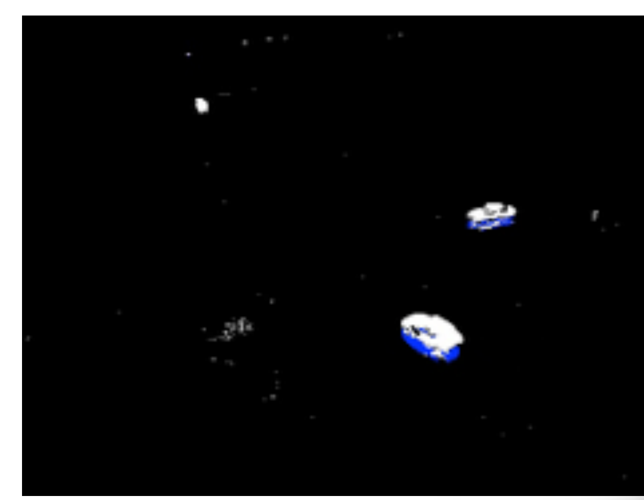
# Applications

- Google, Apple (including Polar rose and C3)

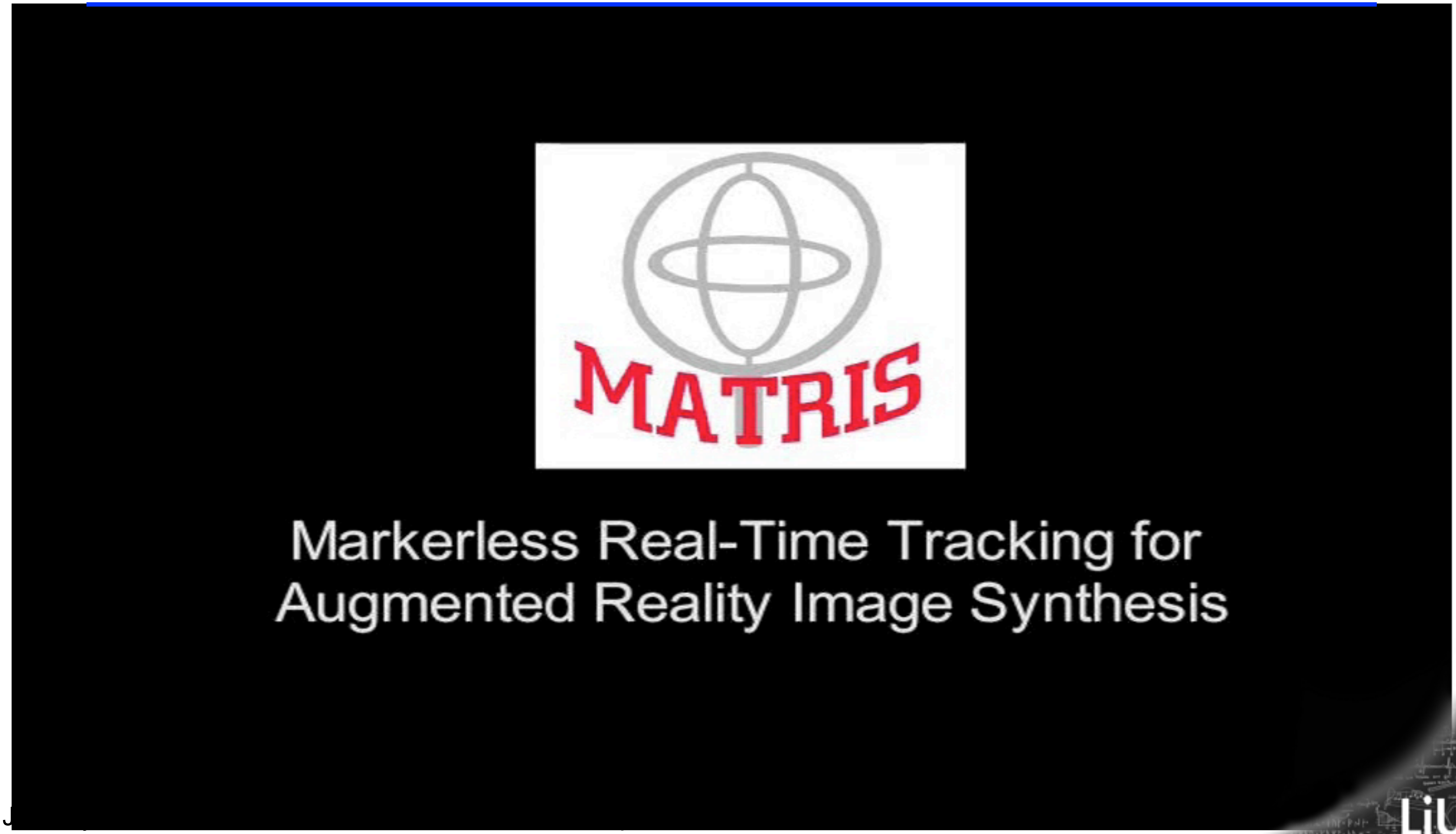


# Applications

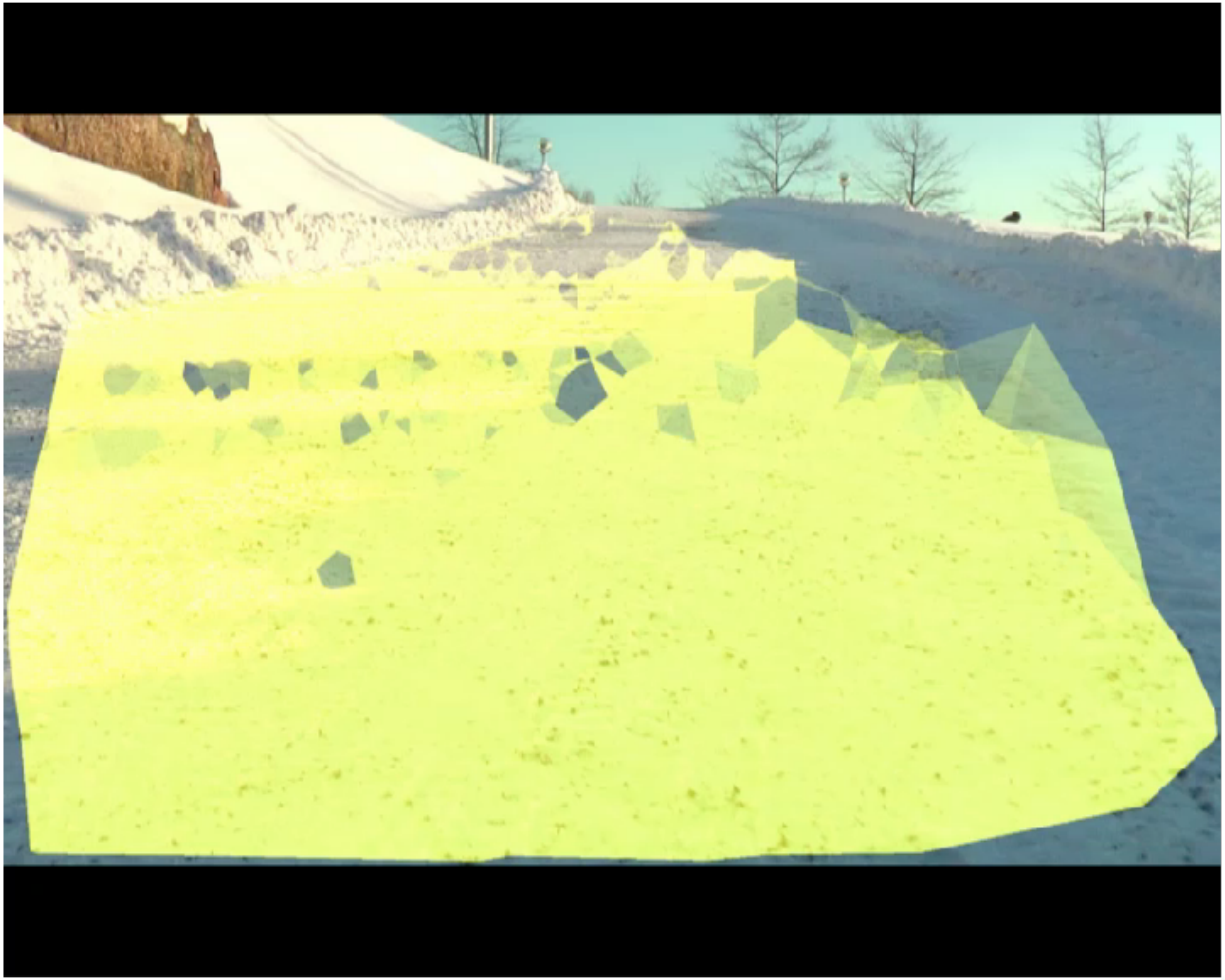
- Hollywood, Autoliv, Volvo, IVSS (e.g. Vägverket)

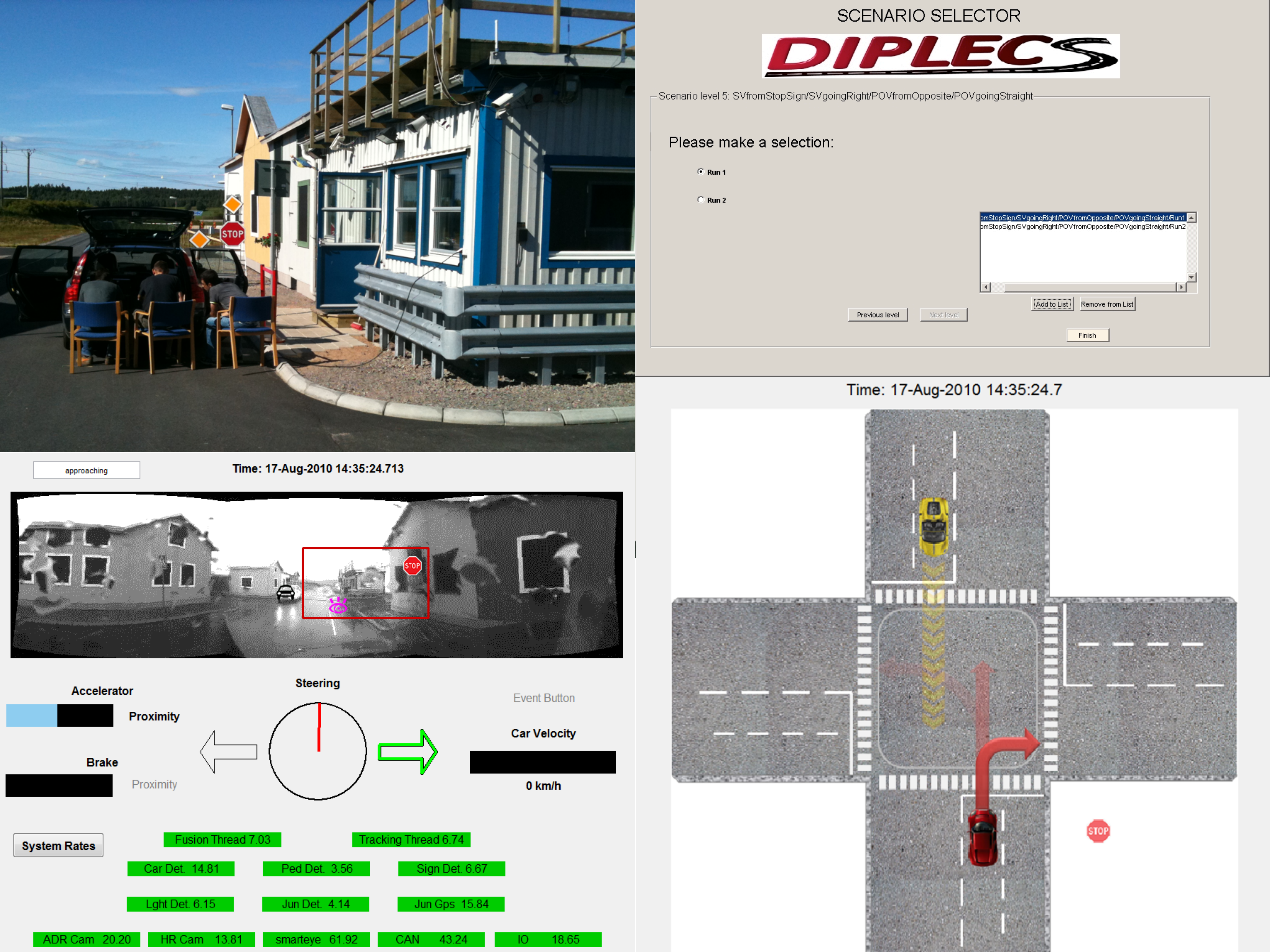



# MATRIS (2004-2007)



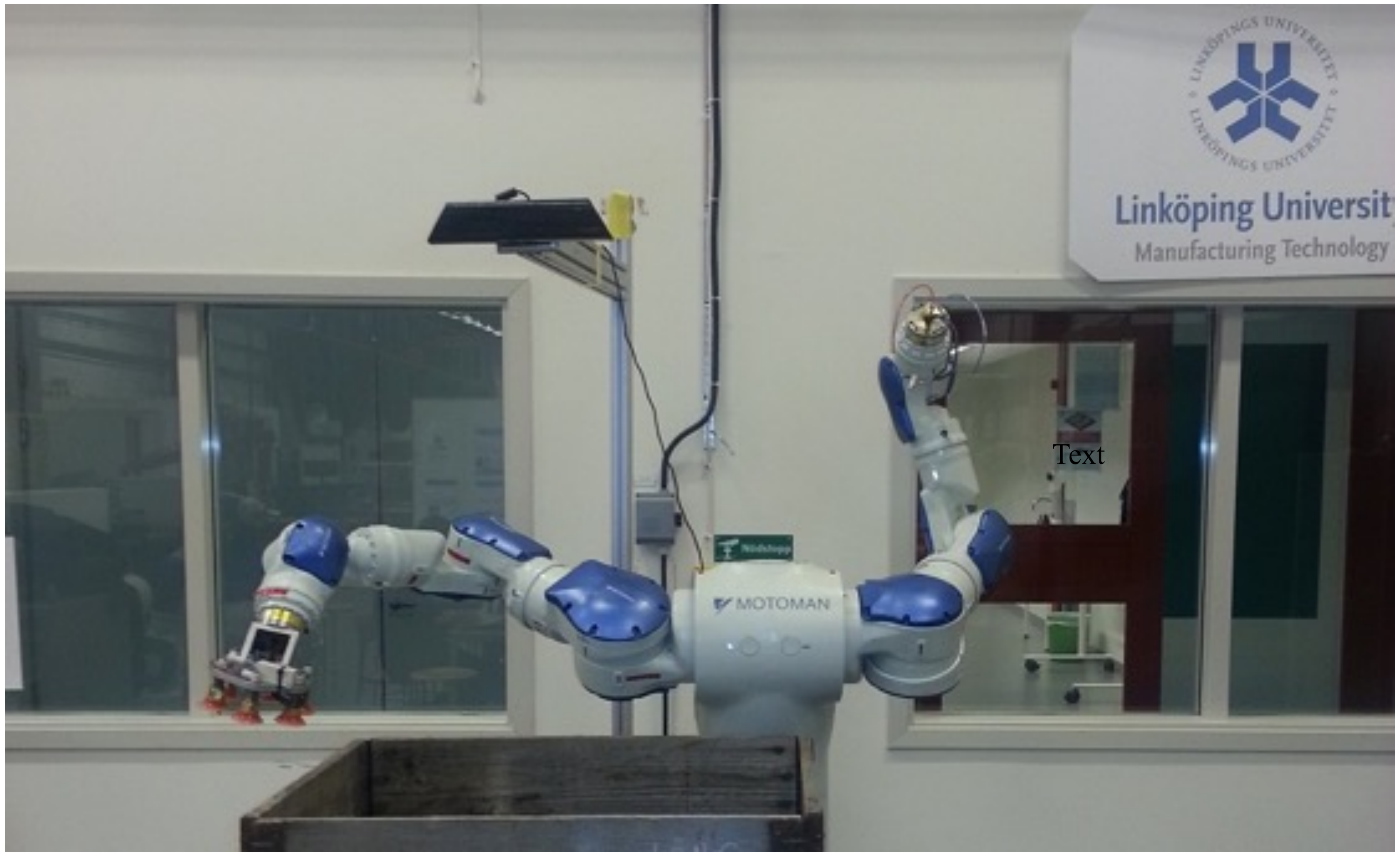
# DIPLECS (2007-2010)





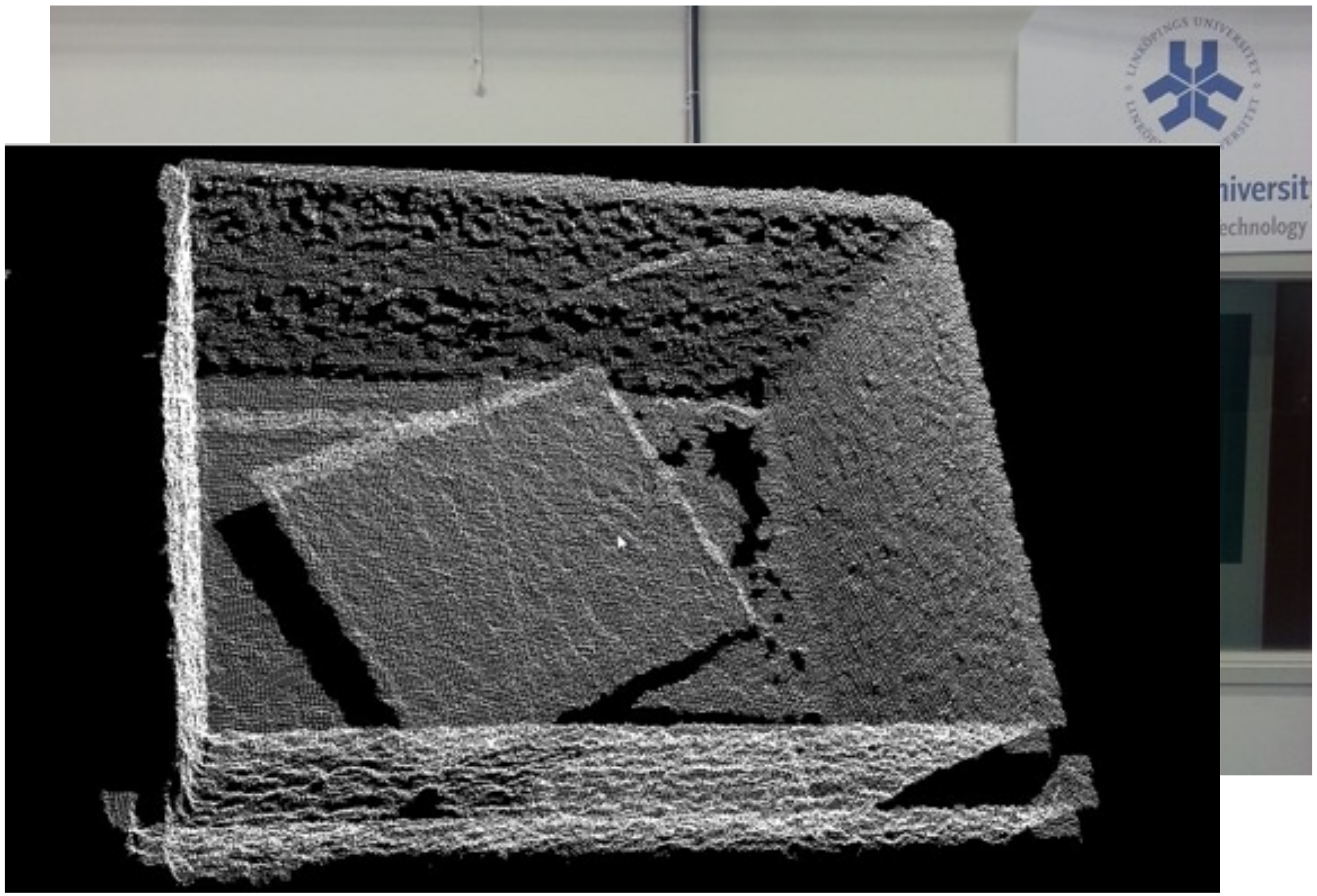
STOP
SCENARIO SELECTOR
DIPLECS
Scenario level 5: SVfromStopSign/SVgoingRight/POVfromOpposite/POVgoingStraight
Please make a selection:
Run 1
Run 2
omStopSign/SVgoingRight/POVfromOpposite/POVgoingStraight/Run1
omStopSign/SVgoingRight/POVfromOpposite/POVgoingStraight/Run2
Previous level
Next level
Add to List
Remove from List
Finish
approaching
Time: 17-Aug-2010 14:35:24.713
STOP
Accelerator
Proximity
Brake
Proximity
Steering
Event Button
Car Velocity
0 km/h
System Rates
Fusion Thread 7.03
Tracking Thread 6.74
Car Det. 14.81
Ped Det. 3.56
Sign Det. 6.67
Lght Det. 6.15
Jun Det. 4.14
Jun Gps 15.84
ADR Cam 20.20
HR Cam 13.81
smarteye 61.92
CAN 43.24
IO 18.65
Time: 17-Aug-2010 14:35:24.7
STOP

# CDIO Project Example (2012)



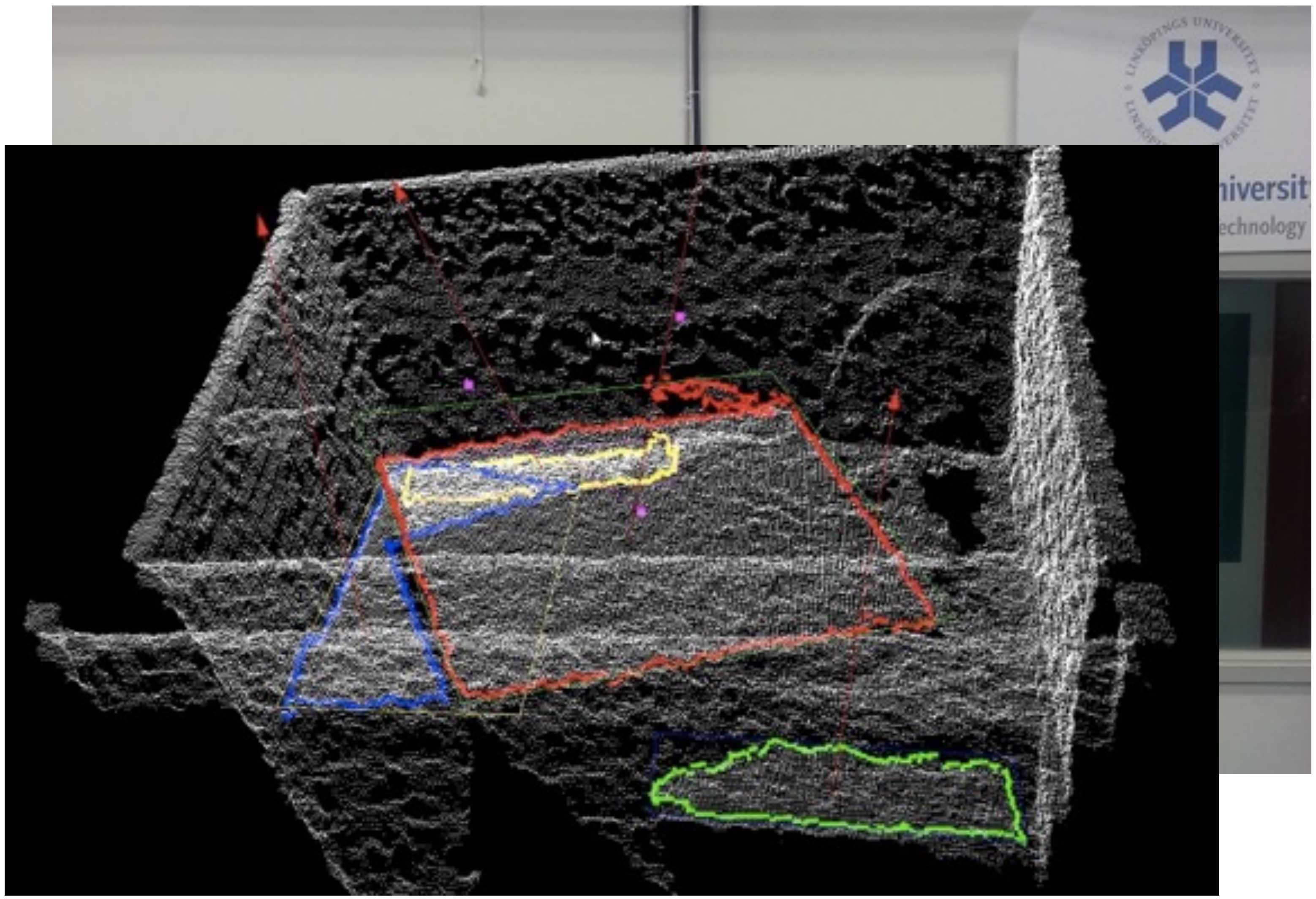

- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.

# CDIO Project Example (2012)



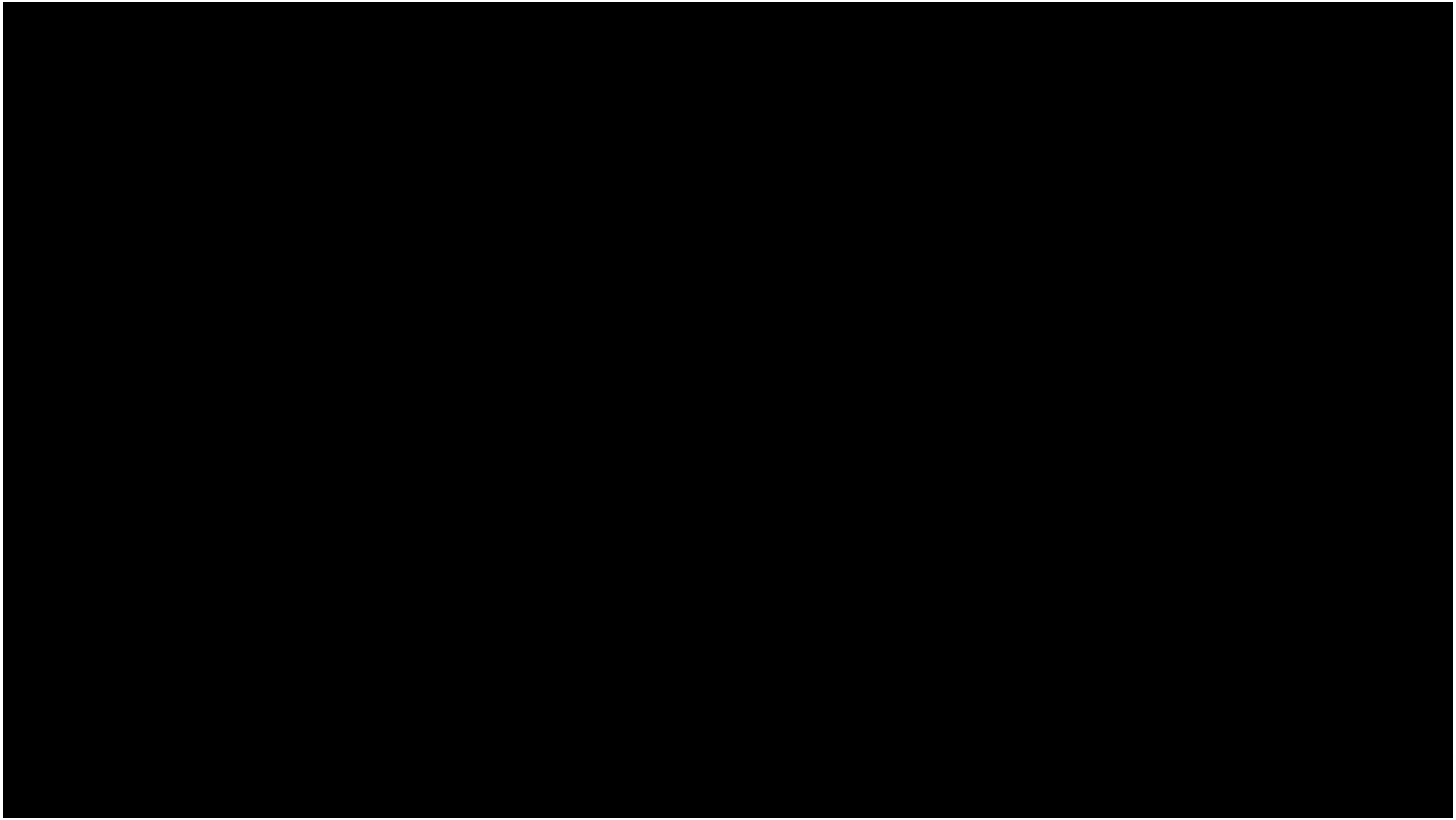
- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.

# CDIO Project Example (2012)

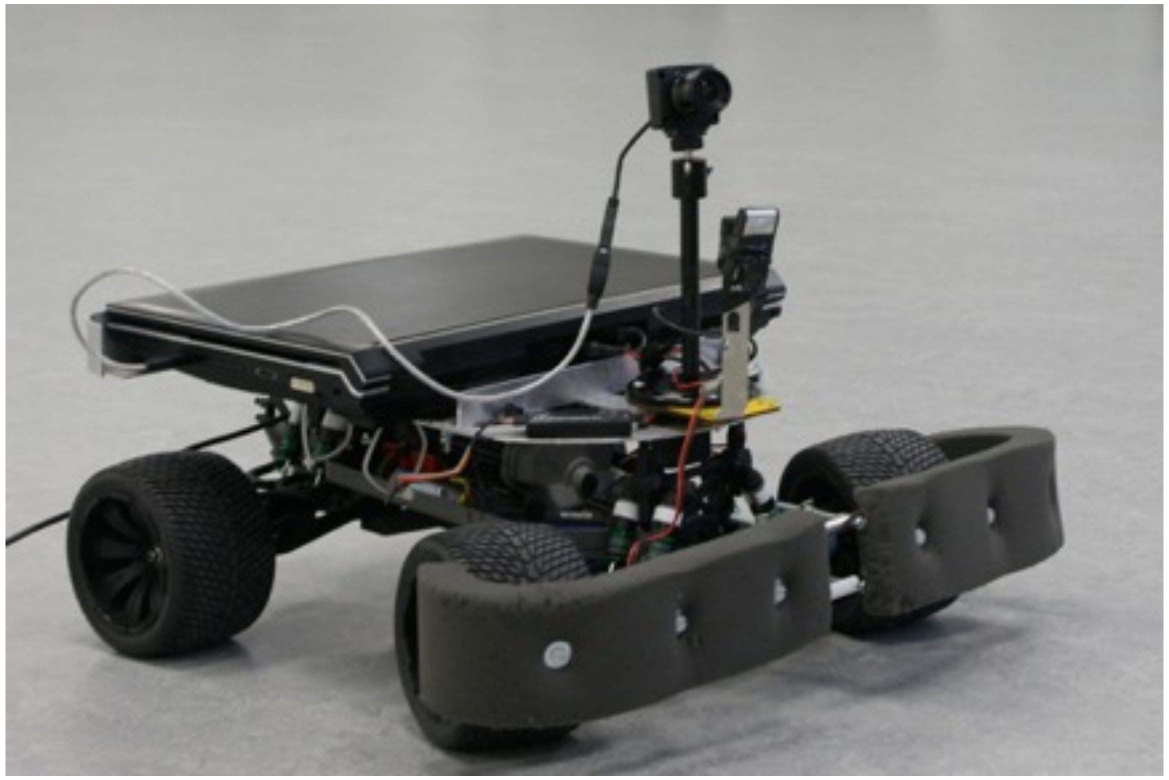


- Robot should pick up used computer monitors from a crate.
- Perception using a Microsoft Kinect depth sensor.

# CDIO Project Example (2012)



# CDIO Project Example (2012)



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

# CDIO Project Example (2012)

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

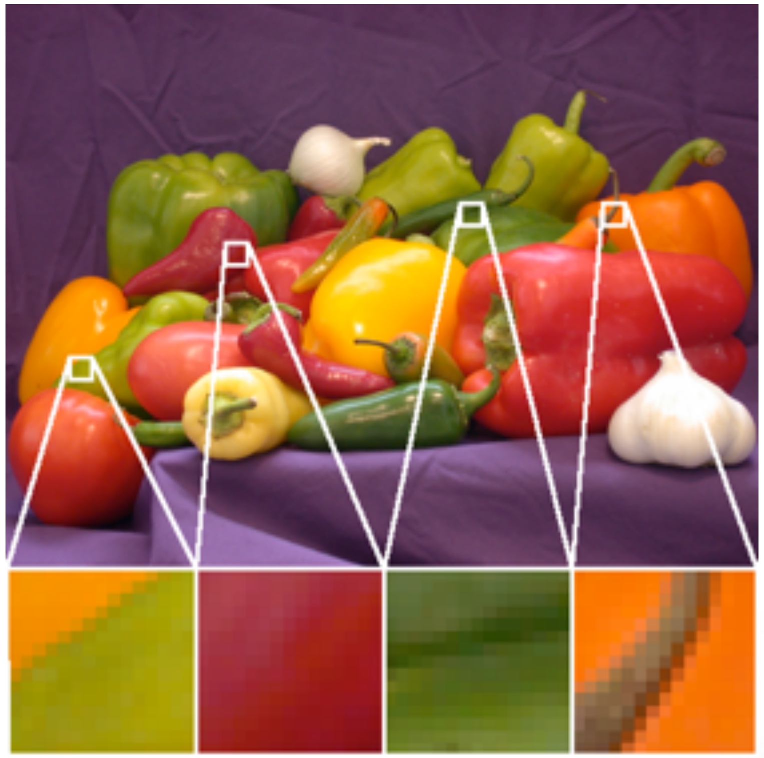
  stack all pixels in a vector:
  $\mathbf{v} = [\mathbf{I}(\mathbf{x}_1) \ldots \mathbf{I}(\mathbf{x}_k) \ldots \mathbf{I}(\mathbf{x}_K)] \quad \mathbf{x}_k \in \Omega$

# Local models

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

  stack all pixels in a vector:
  $\mathbf{v} = [\mathbf{I}(\mathbf{x}_1) \ldots \mathbf{I}(\mathbf{x}_k) \ldots \mathbf{I}(\mathbf{x}_K)] \quad \mathbf{x}_k \in \Omega$

- 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.

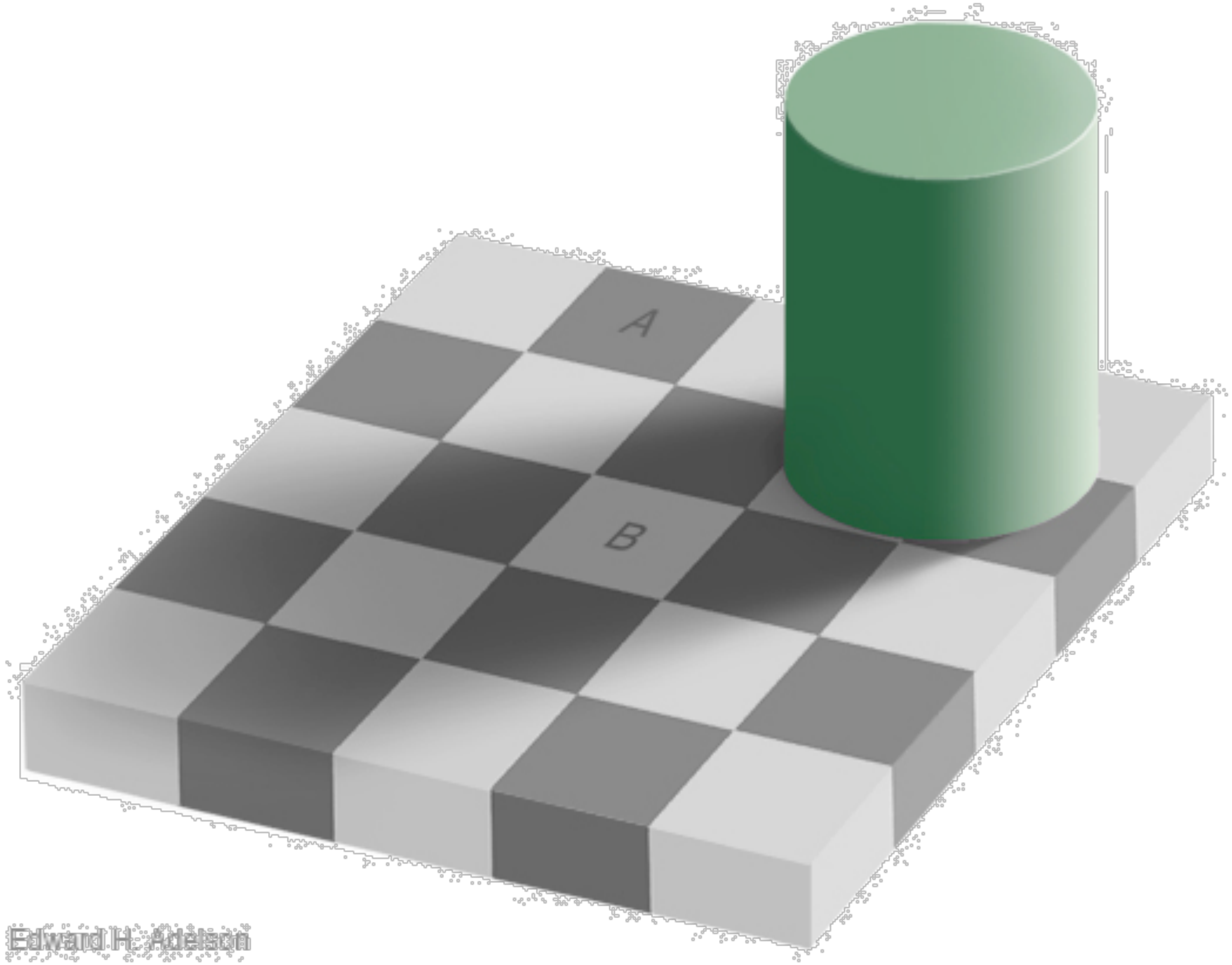


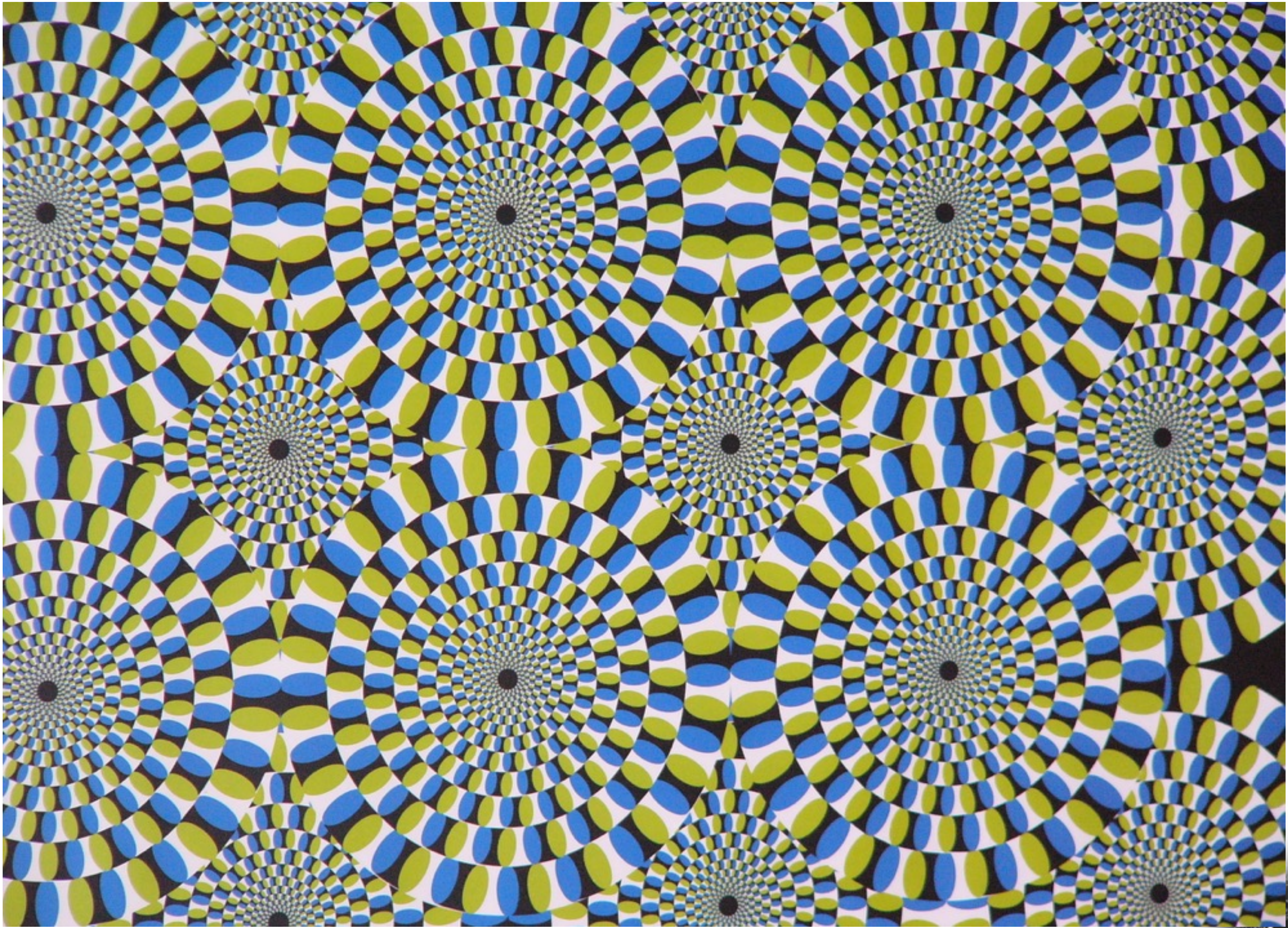
# Locality

- **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

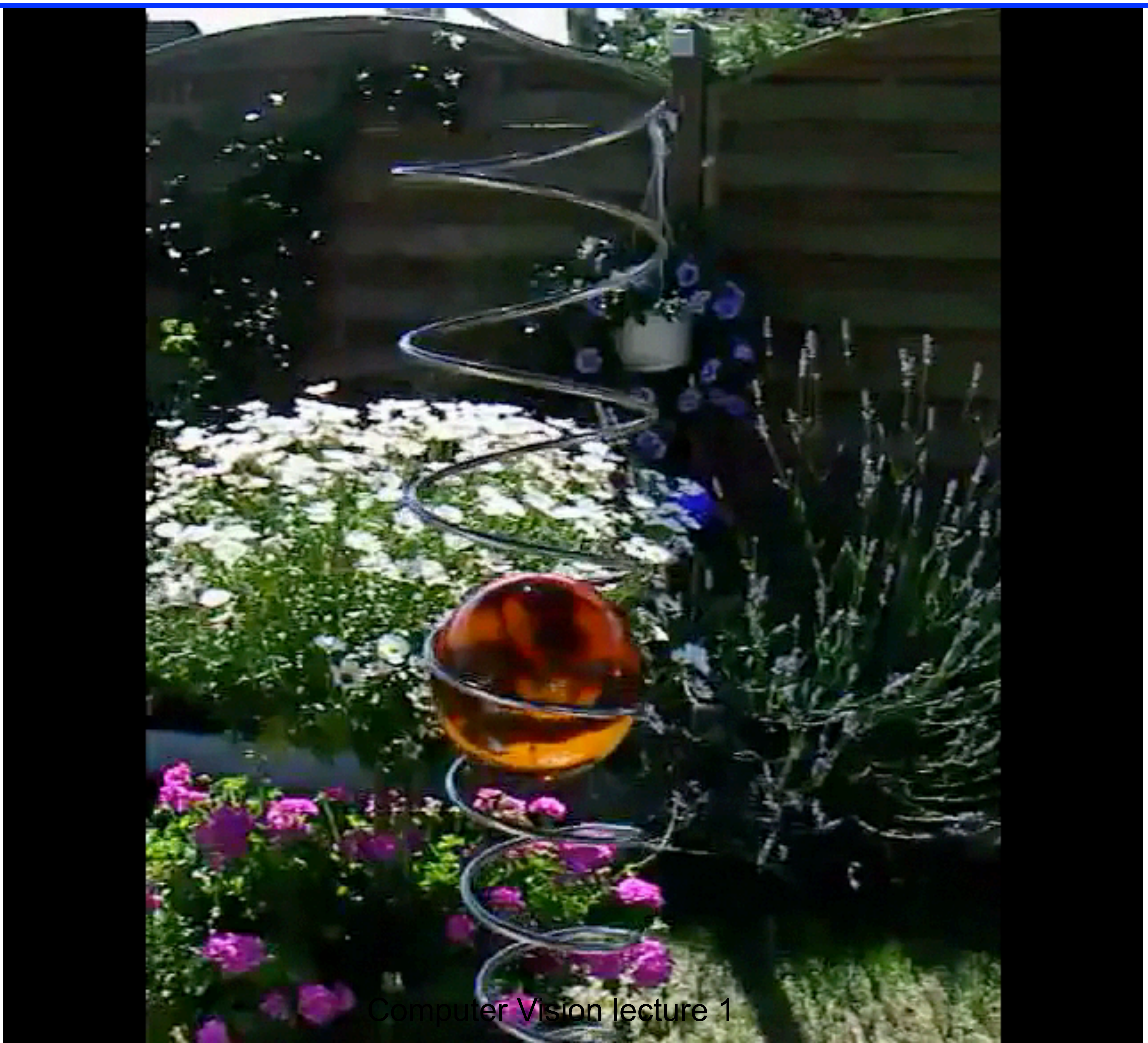
- This is also how the brain does it (Lecture 9).

# Locality & Intensity (illusion)





# Locality & Motion – Aperture



# Methodology

- Local models
  - –Mathematical model assumes existence of certain features
  - –Signal: intrinsically 1D, band-limited, ergodic
  - –Alternative: learn features from data
- Camera models
  - –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
  - **n** is a normal vector
- *f* is called *simple* signal or **intrinsically one-dimensional signal** (i1D)

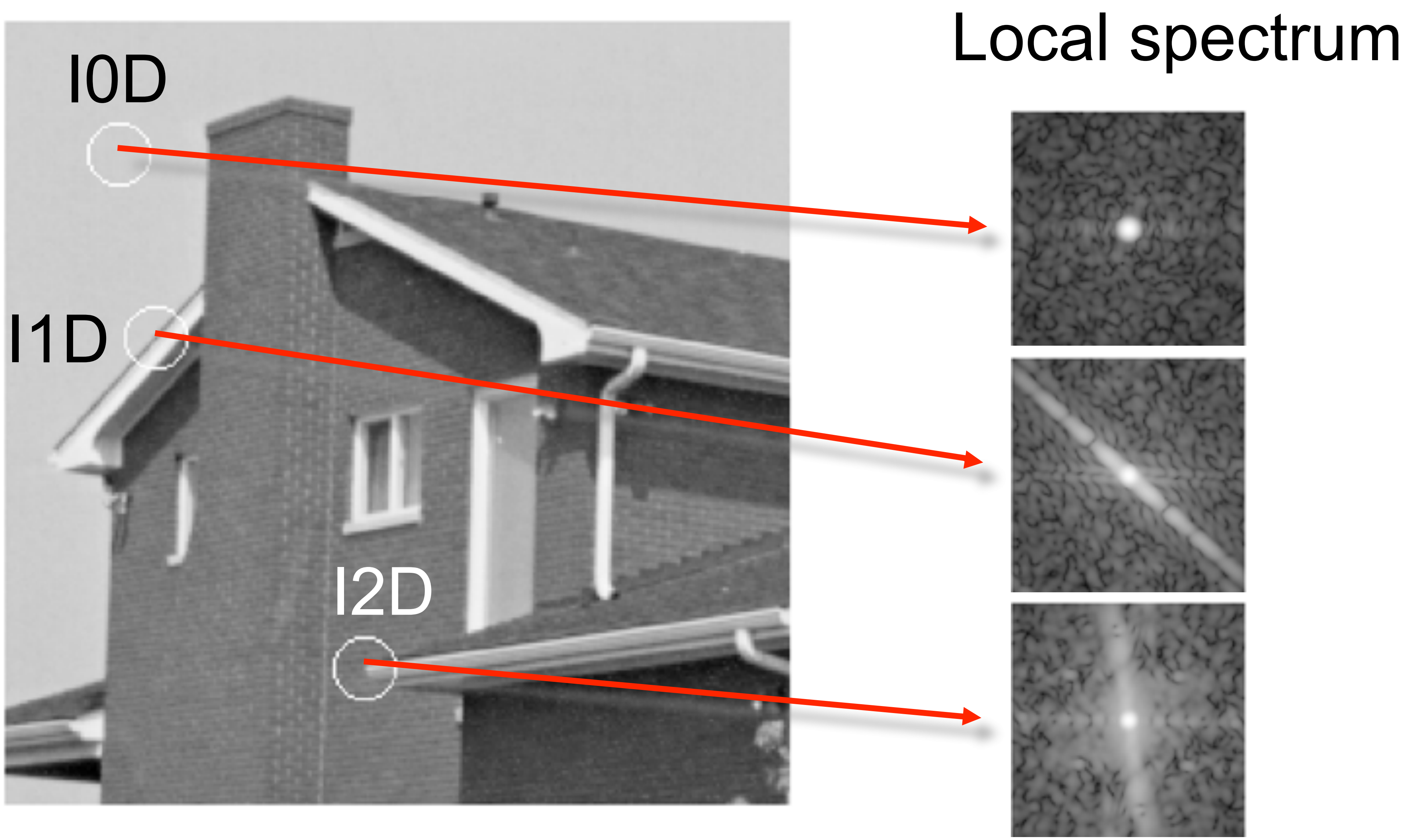
# Example: i1D-signals

- 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 **n** and which varies as *G*, the Fourier transform of *g* (why?)
  – *n* is the dimension of the signal

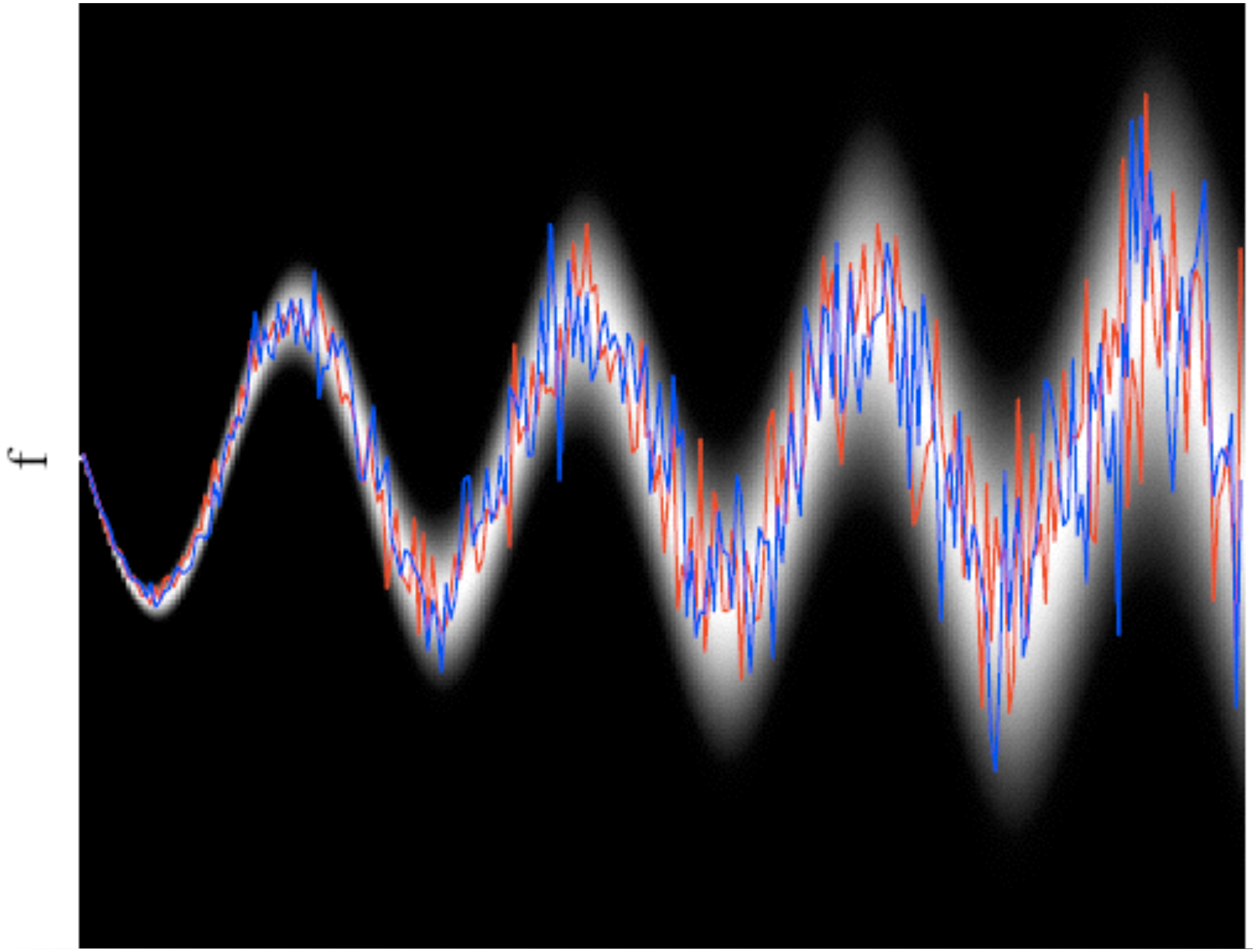
# The Intrinsic Dimension



# The Intrinsic Dimension

i0d i1d i2d

# Ergodic Signal



# Methodology

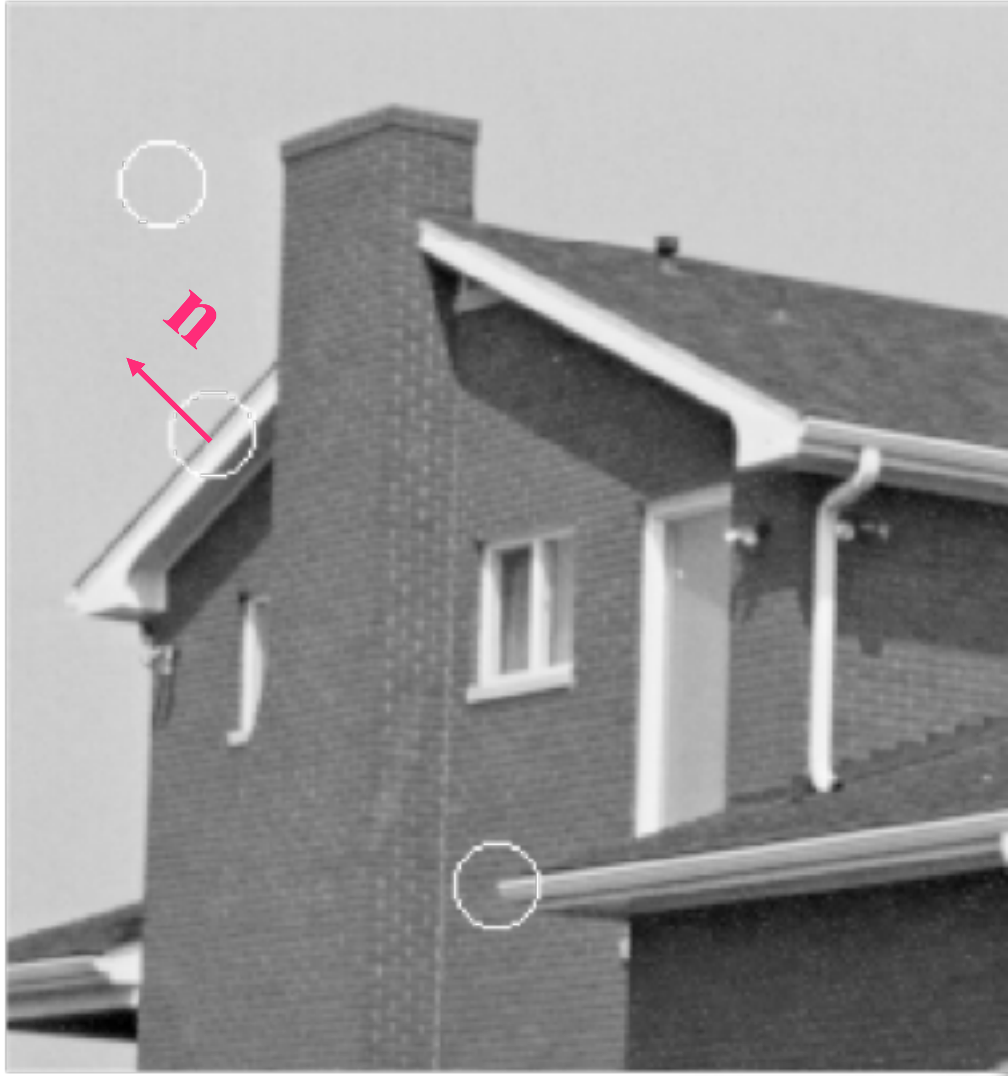
- Examples of local features:
  - Local orientation, motion, curvature, color, symmetry, phase...

# 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

# Example: Local Orientation

- A local signal which is i1D has a natural notion of orientation
- The signal is constant in all directions which are orthogonal to the vector **n**



# Methodology

- Representations
  - –Descriptors (uniquely defined, complete)
  - –Statistical properties (mean value, variance, orthogonality)
  - –Geometrical objects (e.g. epipolar line)

# 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**

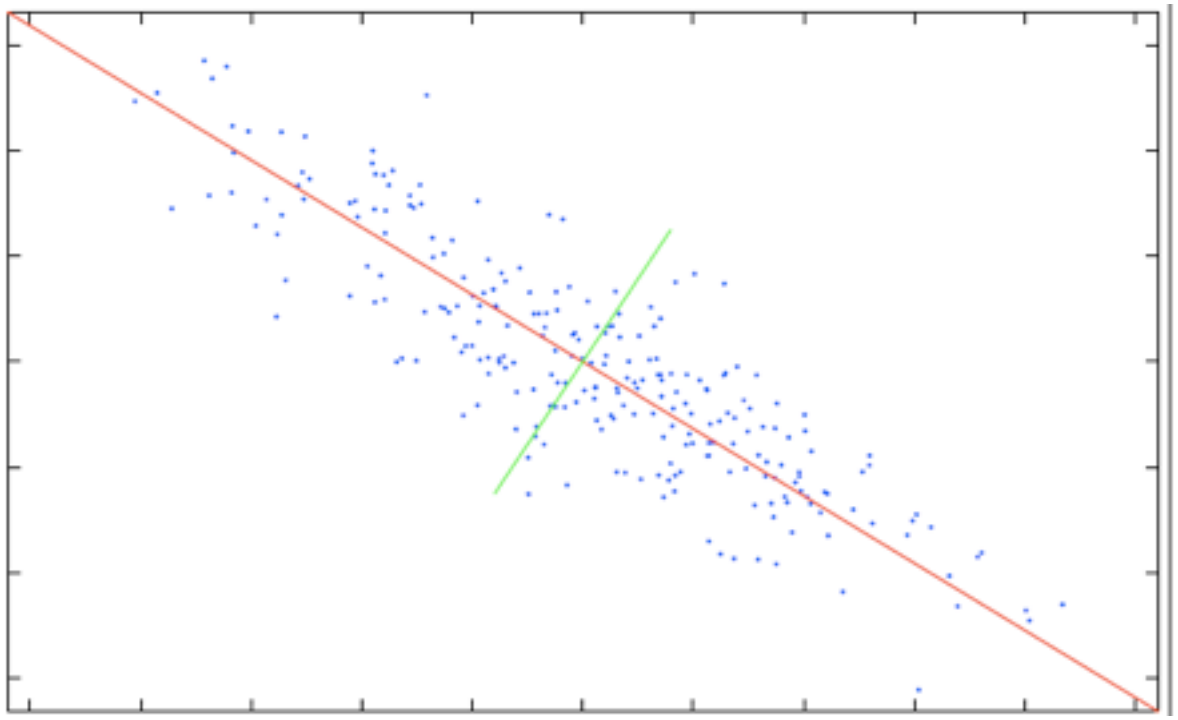
# Example: Local Orientation

- For the feature local orientation, the vector **n** is a possible representation.
- BUT: for an i1D-signal, **n** is not unique, also –**n** can be used. Why?
- *Unique* representations are normally preferred

# Example: Local Orientation

- In the 2D case, an angle in the interval 0°-180° between the vector **n** and the horizontal axis would be unique.
- BUT: if **n** is approximately horizontal, very small changes of orientation result in the angle jumping between 0° and 180°.

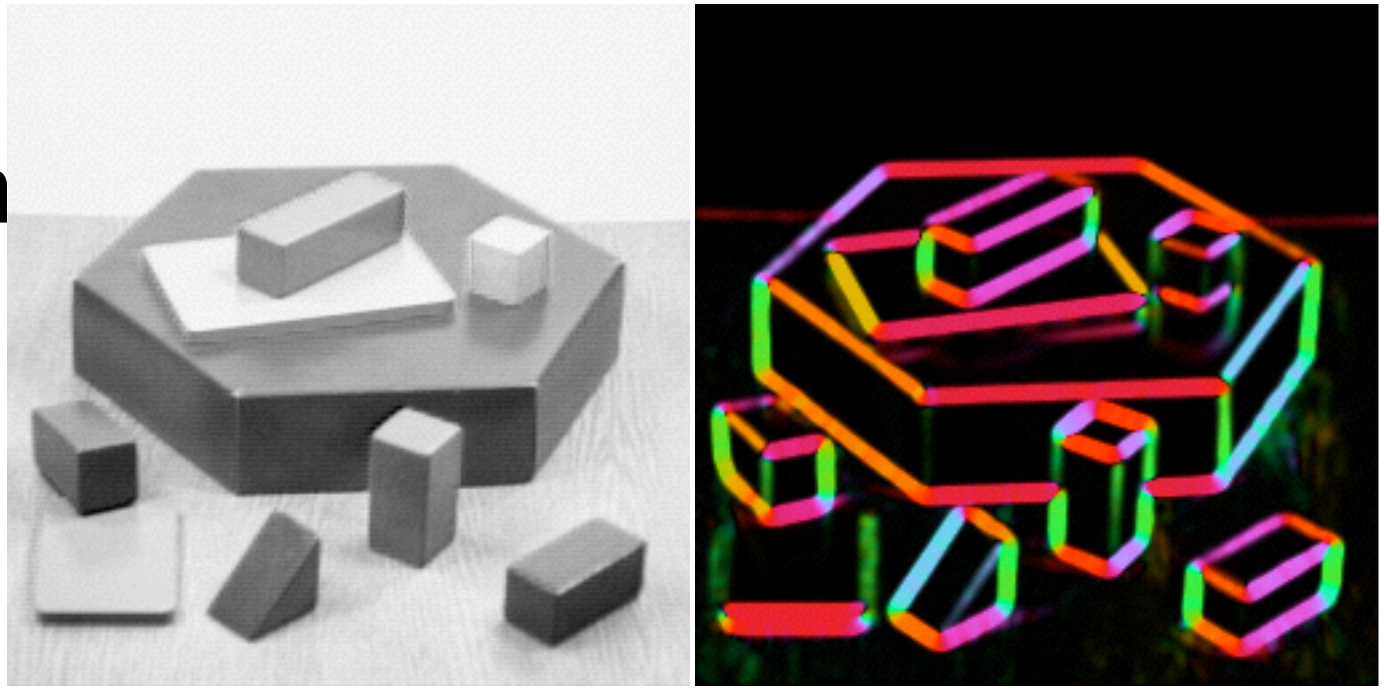
# Representations



- 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 can be used for weighted averaging (maximum likelihood estimation)

# Example: Local Orientation

- For representation of orientation (i.e. the vector **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



# 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**

# Methodology

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

# Estimation

- Which algorithm to choose depends on
  - Available computational power (how many operations can be computed for 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.