Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

Lecture 1

Course Overview
Human and Computer Vision
OpenCV & Software Development Tools

Goals of the Course

- Key elements of computer vision
- Practical application of the underlying theory
- Application areas
  - Robot vision
  - Industrial inspection
  - Video surveillance
- Effective implementation of solutions
  - Practical computer vision problems in a variety of environments
  - Bespoke software & standard computer vision libraries

Learning Outcomes

After completing this course, you should be able to

- Apply your knowledge of image acquisition, image processing, and image analysis to extract useful information from visual images
- Design, implement, and document appropriate, effective, and efficient software solutions for a variety of real-world computer vision problems
- Exploit standard computer vision software libraries in the development of these solutions

David Vernon

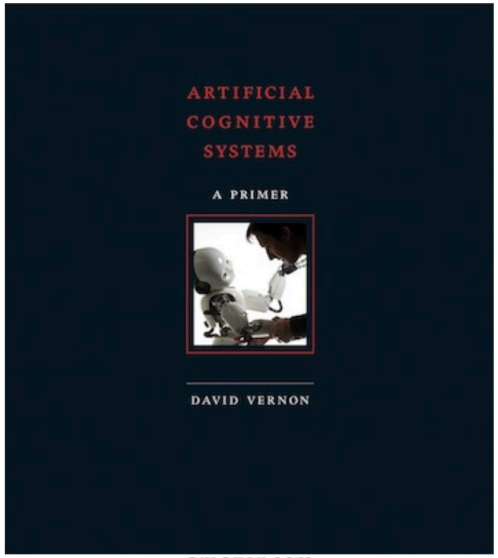
Home | Biography | Research | The iCub | euCognition | Publications | Courses | Cycling | Wiki

Welcome to my personal website. I am a professor at Carnegie Mellon University Africa in Rwanda where I teach cognitive robotics, artificial cognitive systems, applied computer vision, and data structures and algorithms for engineers. Prior to this I worked at the University of Skövde, Sweden. Over the past 40 years I have held positions in several information technology sectors: as a software engineer in industry, as a university professor (in Ireland, UAE, and Sweden), as a consultant, and as a project officer in the European Commission and Science Foundation Ireland.

My main research focus is on cognitive robotics and I have the privilege of serving as one of the co-chairs of the new IEEE Robotics and Automation Society (RAS) Technical Committee on Cognitive Robotics. I am proud to say that the IEEE Robotics and Automation Society has selected us as the recipient of the 2017 RAS Most Active Technical Committee Award.

As part of SPARC - The Partnership for Robotics in Europe - and euRobotics aisbl, I also contribute to the European Union ROCKeu2 coordination and support action to promote the deployment of cognitive systems in industry.

My most recent book, Artificial Cognitive Systems - A Primer, was published by MIT Press in 2014. A Chinese edition will be published in 2020.



amazon.com

"This volume fills a serious void in the literature on artificial cognition, spanning a wide range of past work and helping the reader assess the rich array of approaches that have been taken to constructing cognitive systems. It will be an essential text for anyone who seeks to advance work in this field."

ACM Computing Reviews
Review #: CR143563 (1509-0767)

Email: david@vernon.eu.

The Wonder of Rwanda

I live and work in Rwanda, a small country with big ambitions to become a leading knowledge economy in Africa. Here is a great video - The Wonder of Rwanda - by Made for minds, a German news and current affairs website, about how Rwanda is realizing these ambitions.

African GLO Tour 2018
Fundraising cycling challenge

Last year I took part in a seven-day 780 km cycle around Rwanda in aid of Burundi, one of the poorest countries in the world, struggling to recover from civil war and genocide. You can read all about it on my blog.

Any donation - even $10 / €10 / £10 - will make a difference so please consider giving something.

A heartfelt **Thank You** to the 138 people who have donated so far.

JustGiving Sponsor me

Supporting Texts

Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

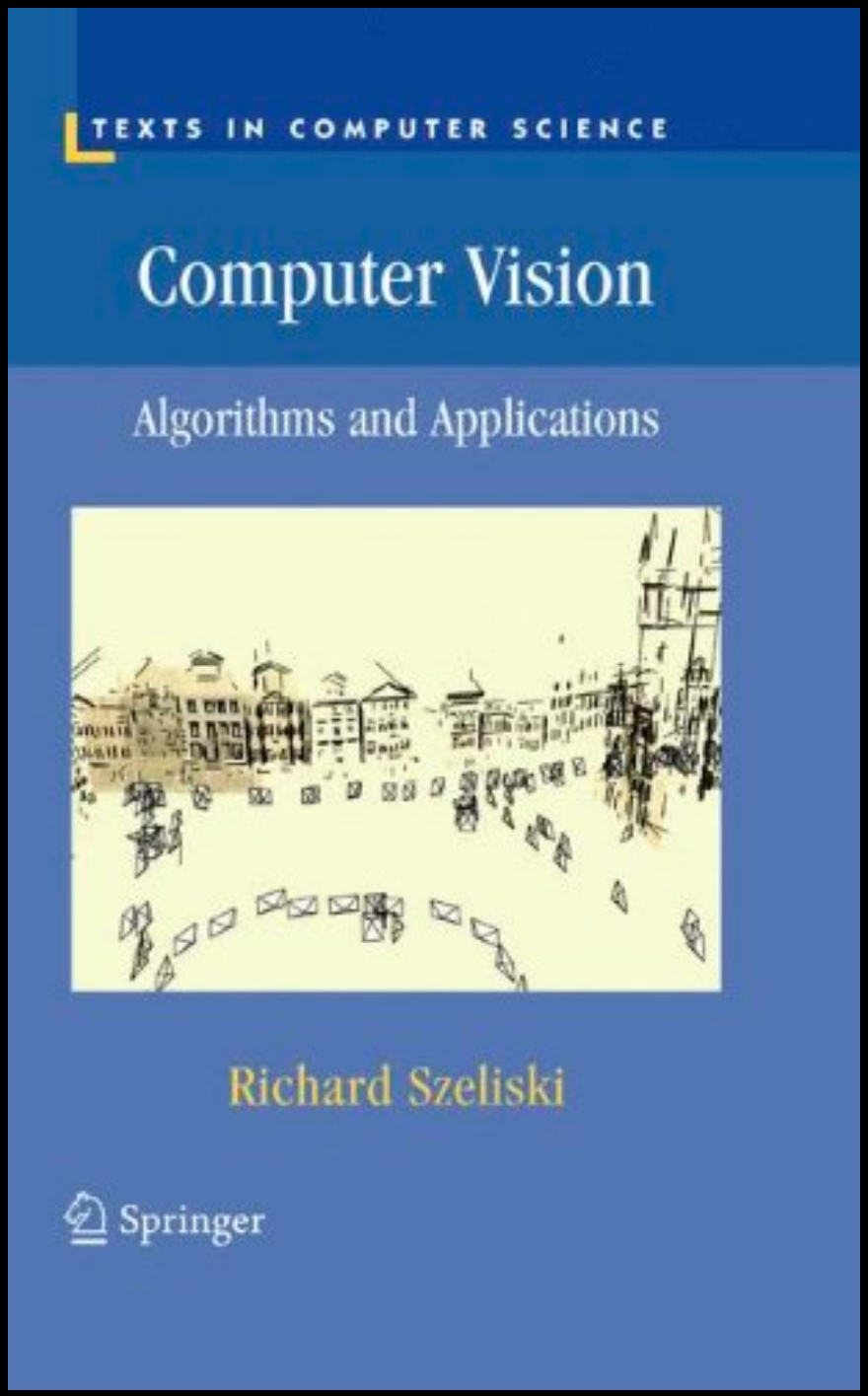
http://szeliski.org/Book/drafts/SzeliskiBook_20100903_draft.pdf

David Vernon, *Machine Vision: Automated Visual Inspection and Robot Vision*, Prentice-Hall, 1991.

http://vernon.eu/publications/91_Vernon_Machine_Vision.pdf

Kenneth Dawson-Howe, *A Practical Introduction to Computer Vision with OpenCV*, Wiley, 2014.

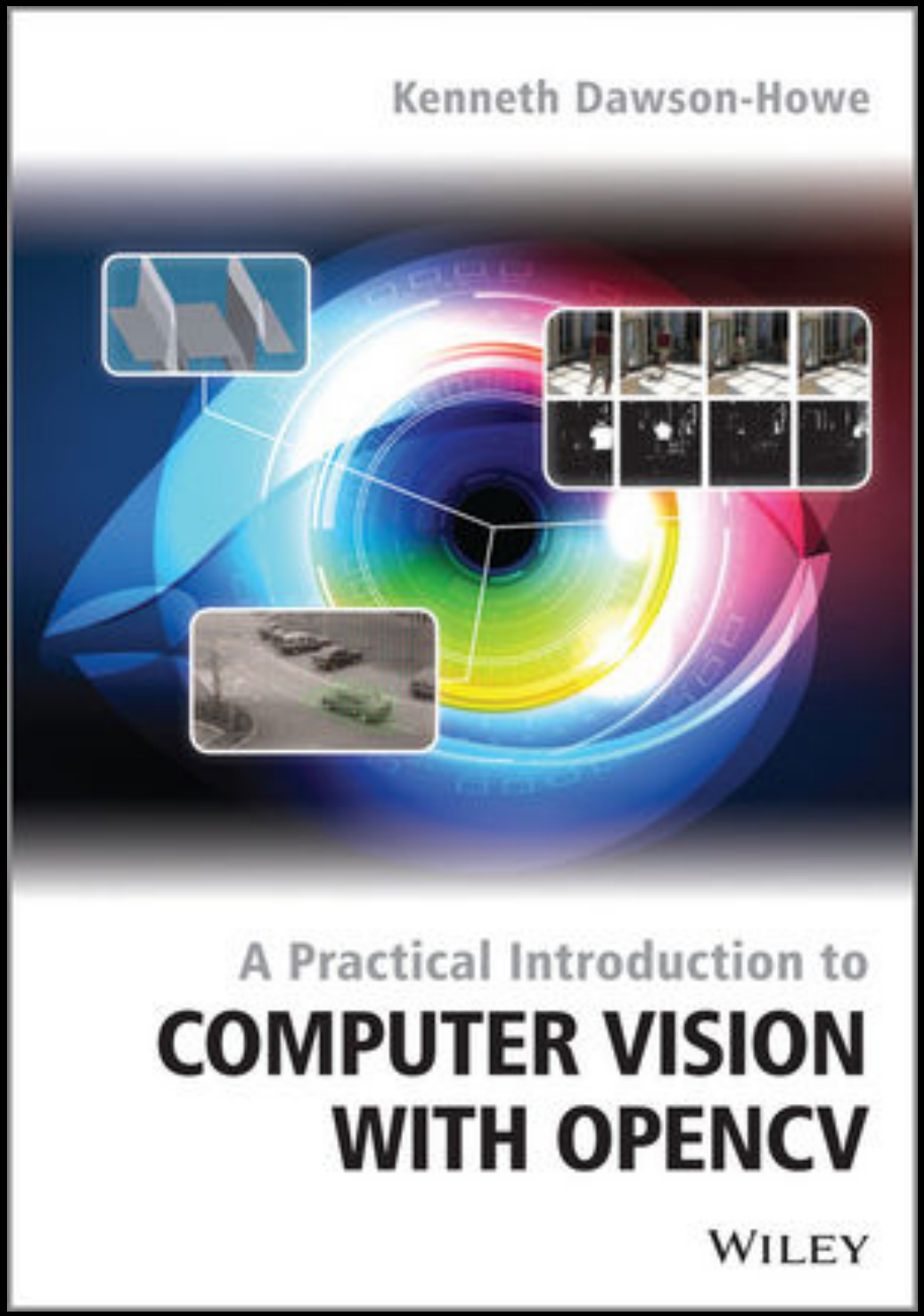
https://www.scss.tcd.ie/publications/book-supplements/A-Practical-Introduction-to-Computer-Vision-with-OpenCV/


TEXTS IN COMPUTER SCIENCE
Computer Vision
Algorithms and Applications
Richard Szeliski
Springer

David Vernon

MACHINE VISION

Automated Visual Inspection and Robot Vision


Kenneth Dawson-Howe
A Practical Introduction to
COMPUTER VISION
WITH OPENCV
WILEY

Course Operation

- Lectures will be posted in advance
  - Read them before coming to class
  - Read them again after class
- Readings
  - Read them after class
- Exercises
  - Learning exercises … not for credit but essential to do well and enjoy the course
- Assignments & Assessment
  - Best 3 of 4 individual programming assignments (20% each)
    - Marking schemes will be distributed with the assignments
    - Strict deadlines: NO EXTENSIONS except on truly exceptional compassionate grounds
  - Midterm exam (10%)
  - Final exam (30%)

Course Operation

- Do
  - Participate in class
  - Ask questions (you will be doing others a favour)
  - Discuss course material, readings, assignments with other students
  - Share thoughts but not written material (e.g. code, documentation)
  - Cite any work you use in assignments
- Don't
  - Cheat or plagiarize
    - Uncited use of any material from anywhere
    - Share / steal any material with/from former or current students
- Sanctions for cheating and plagiarism
  - Zero marks for first sharing infringement (both parties)
  - Fail the course (grade R) for second sharing infringement (both parties)
  - Fail the course (grade R) for first stealing infringement

Overview of Human and Computer Vision

Computer Vision

Computer Vision is concerned with the

- content
- organization, and
- behaviour

of a 3D world
by the automatic analysis of images of that world

Computer Vision

IMAGE → KNOWLEDGE

Extract descriptions of the world from images

Descriptions of what kind? qualitative vs. quantitative

Geometric: shape and position of object or distances in 3D world

Semantic: what objects do I see?

Dynamic: scene changes, object velocities, human actions, ...

Credit: Francesca Odone, University of Genova

Computer Vision

How tall is the dog?

How fast it it walking?

What's it looking for?

What's the diameter of the flower-bed?

Computer Vision

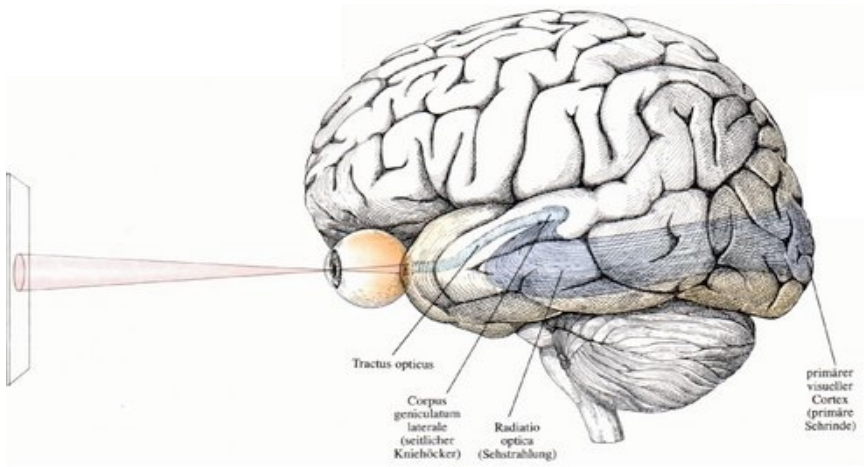
Recognizing objects from images . . .

. . . may be difficult!

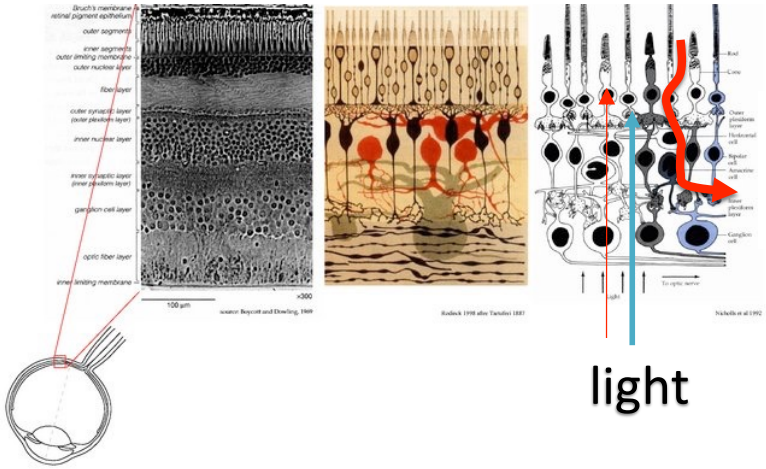
151	138	153	138	137	119	111	111	110	74	74	62	69	74	62	78
137	144	145	137	128	137	119	111	129	86	69	69	74	62	69	69
134	137	133	129	128	128	126	119	119	74	69	62	64	78	62	64
79	79	74	69	74	79	69	74	62	69	69	74	74	72	64	64
62	62	64	64	57	57	55	57	62	69	69	72	69	74	62	69
57	64	74	57	64	74	64	74	74	74	69	74	69	78	62	64
86	84	91	84	86	86	84	86	86	84	69	74	69	69	74	69
94	86	94	94	86	94	91	86	91	79	78	64	69	78	62	69
94	99	102	94	94	103	94	99	103	86	69	64	69	74	69	69
128	138	133	144	138	138	145	144	142	100	78	74	69	72	69	64
176	176	183	170	176	170	183	170	168	126	78	74	78	78	69	69
174	176	176	189	170	176	170	189	170	135	74	69	74	72	69	69
183	181	183	174	181	181	189	181	170	135	78	69	69	69	69	64
181	181	183	174	181	189	170	181	172	146	86	69	74	72	62	69
176	174	181	181	183	174	183	181	174	148	84	69	69	74	74	69
181	176	183	183	174	181	181	181	176	141	99	69	69	72	69	69
183	183	183	181	181	189	183	174	181	151	91	69	69	69	64	69
181	181	189	174	181	181	174	189	183	161	91	69	69	78	62	69
176	181	183	181	174	189	176	181	174	165	91	78	74	72	69	69
181	176	189	181	176	176	183	174	181	163	91	78	69	78	62	69
176	174	183	189	176	181	174	170	176	168	91	74	69	74	69	69

Credit: Alessandro Saffiotti, University of Örebro

The Human Visual System



Hubel 1985



neural signal

light

Credit: Francesca Odone, University of Genova

The Human Visual System

Cones

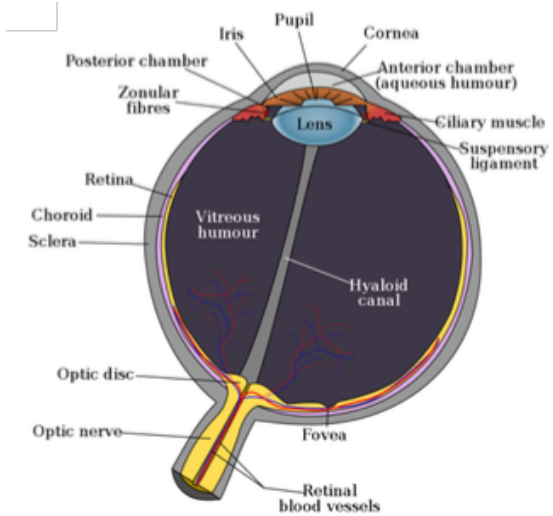
- color vision
- dense in the center
- 5-6 million

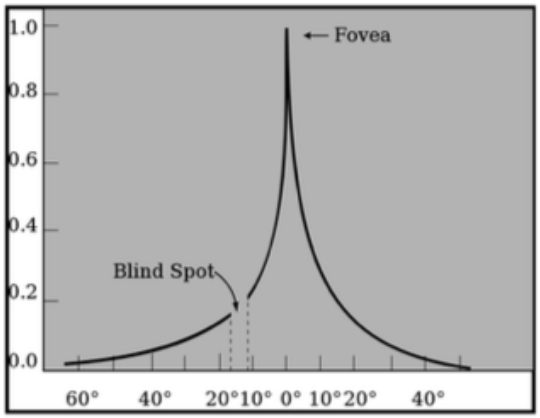
Fovea

- central zone of retina
- only cones, 27 times the density
- responsible for central vision

Rods

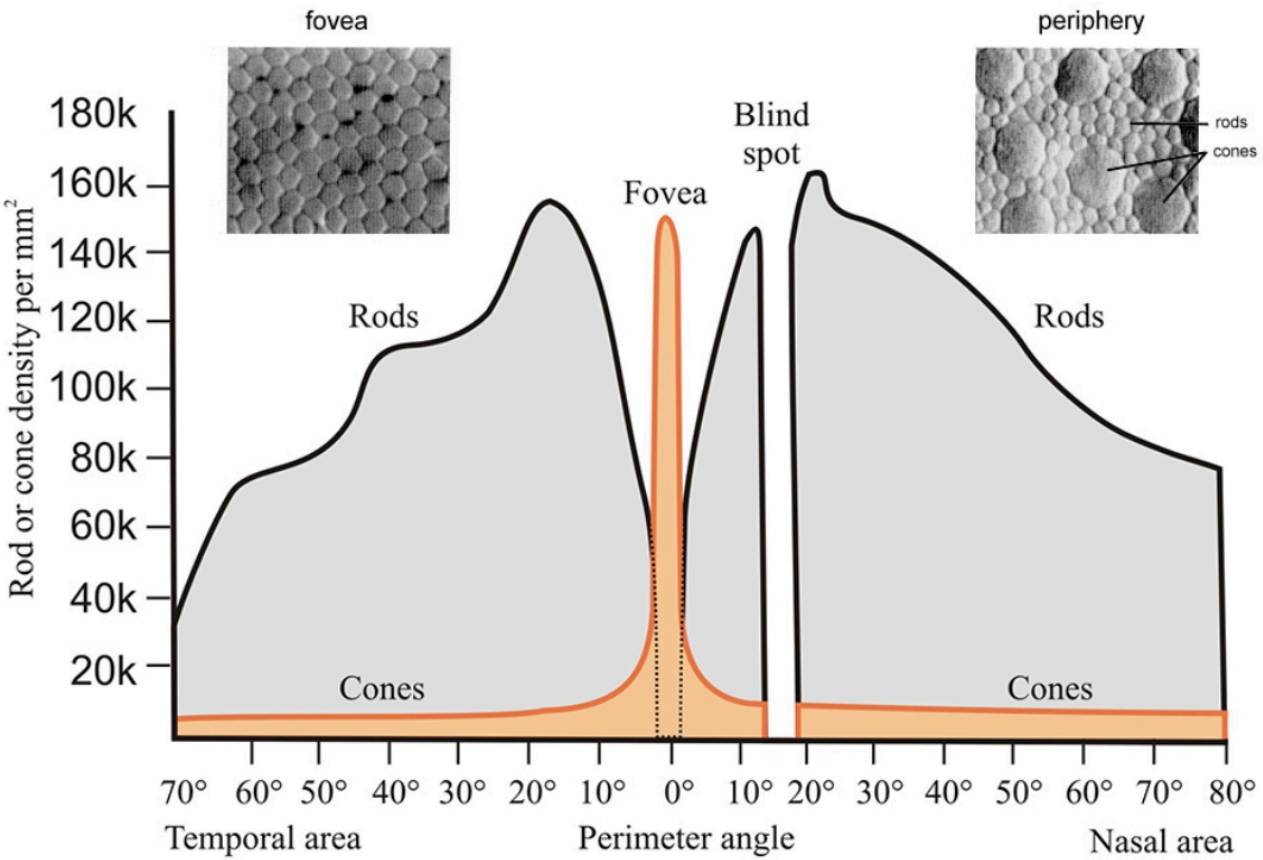
- black & white vision
- peripheral vision
- 120 million





Credit: Francesca Odone, University of Genova

The Human Visual System

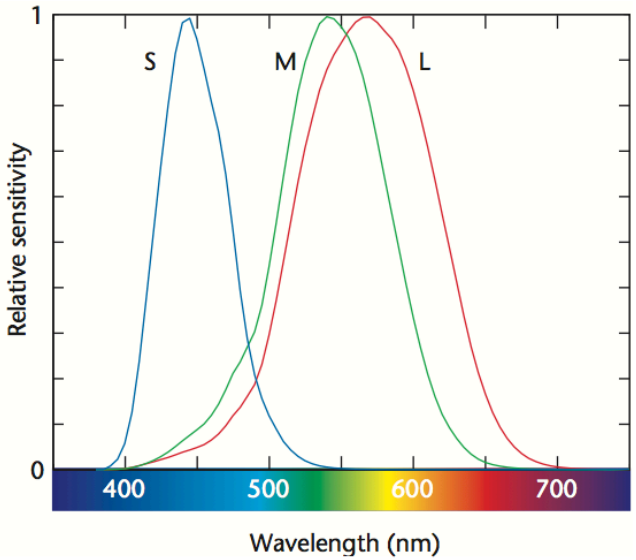


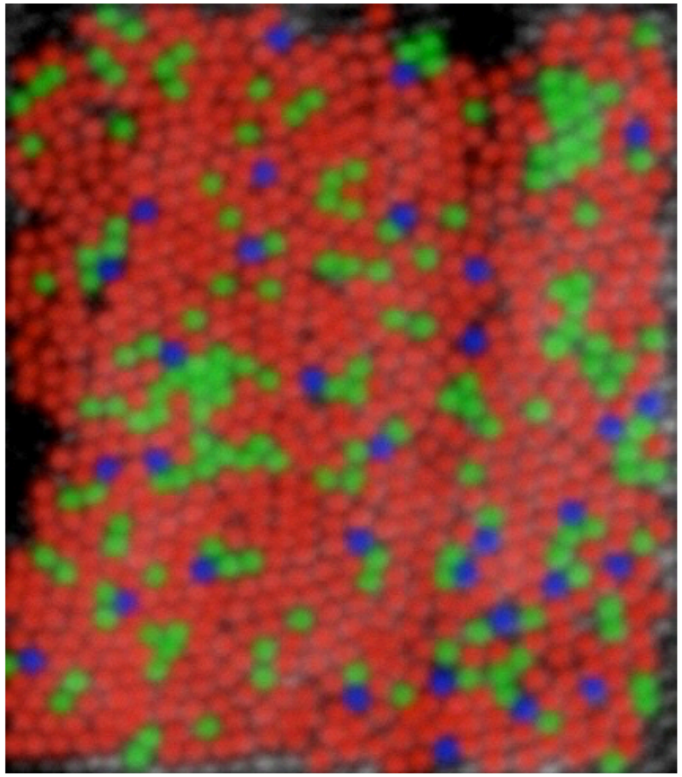
Credit: Francesca Odone, University of Genova

The Human Visual System

Three different types of cones:

- present with different densities
- each type sensitive to a different range of frequencies in the visible spectrum
- called L, M, S from the zones of spectrum at which they have highest response



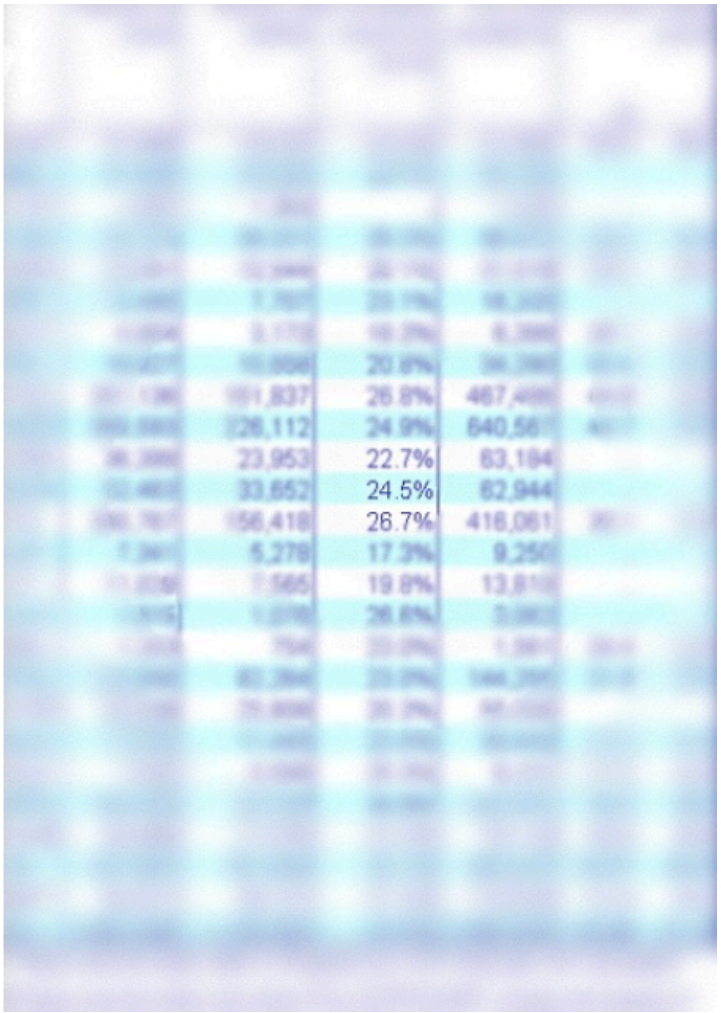


R 63% - G 31% - B 6%

Credit: Francesca Odone, University of Genova

The Human Visual System

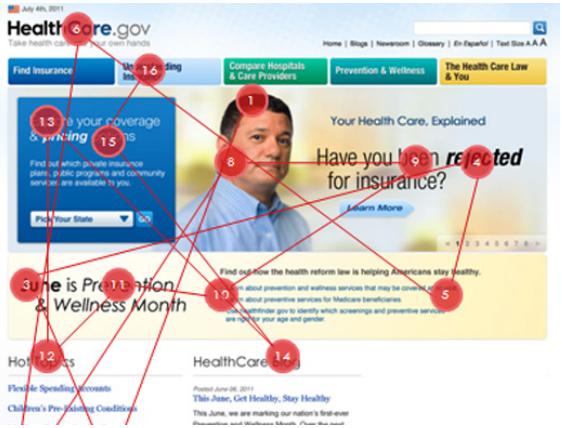
- The eye is not like a camera
- At each instant in time, our vision is sharp only in a very small area around the fovea
- All the rest is badly blurred and vaguely coloured
- Areas outside the fovea provide just contextual information



Credit: Francesca Odone, University of Genova

The Human Visual System

- Vision works as a sequence of fixations and saccades
  - Fixation: maintaining gaze on a single location (200-600 ms)
  - Saccade: moving between different locations (20-100 ms)



- Our mental images are a form of integration of several snapshots focused at different spots, as obtained with fixation

- Where do we fixate our gaze?
  - where there's something attractive
  - where we know there's something we want to look at

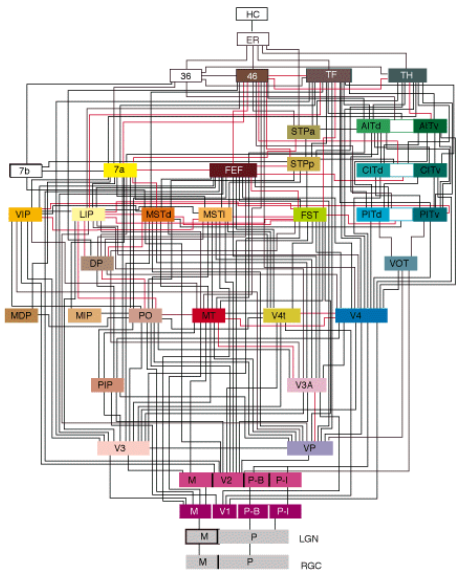
http://www.cs.ubc.ca/~mikewu/cs547/

Credit: Francesca Odone, University of Genova

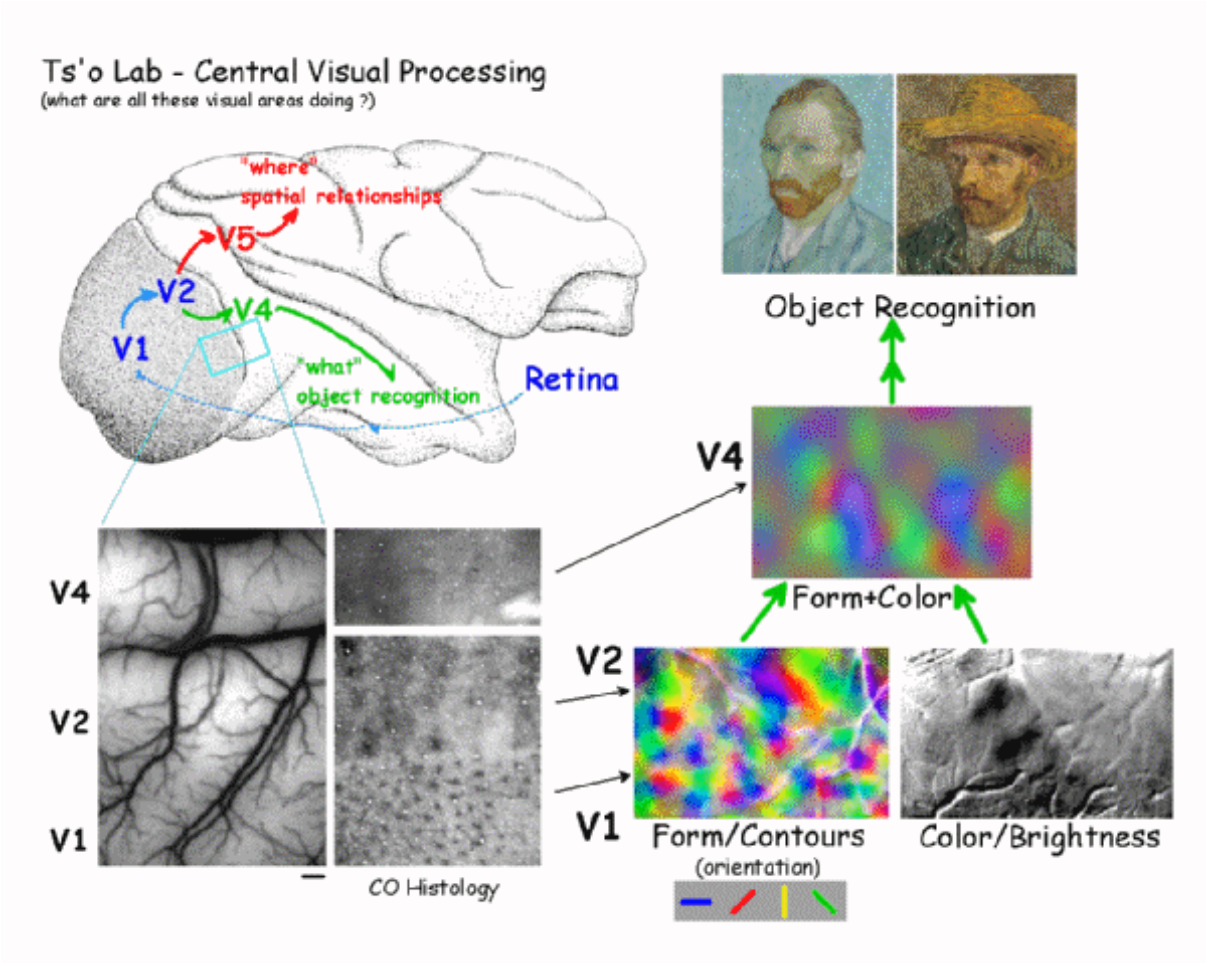
The Human Visual System

- 55% of total neo-cortex is concerned with “seeing”

- Partially understood



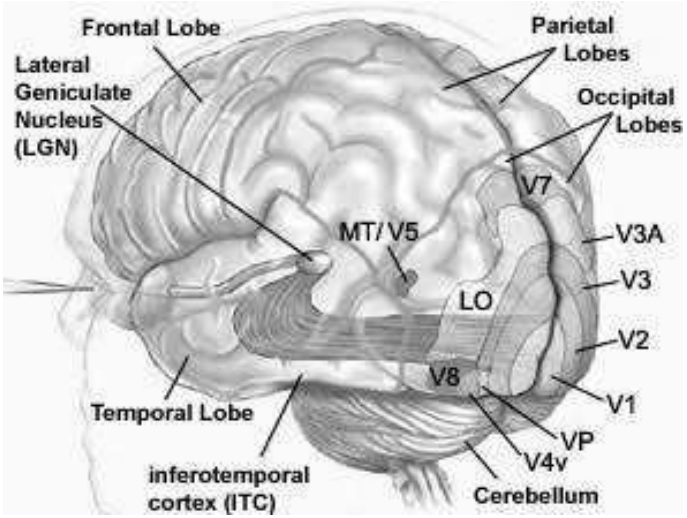
Hierarchy of visual areas
[Felleman, van Essen, 1991]



Credit: Markus Vincze, Technische Universität Wien

The Human Visual System

About 30 regions participate to seeing in primate cortex, e.g.,

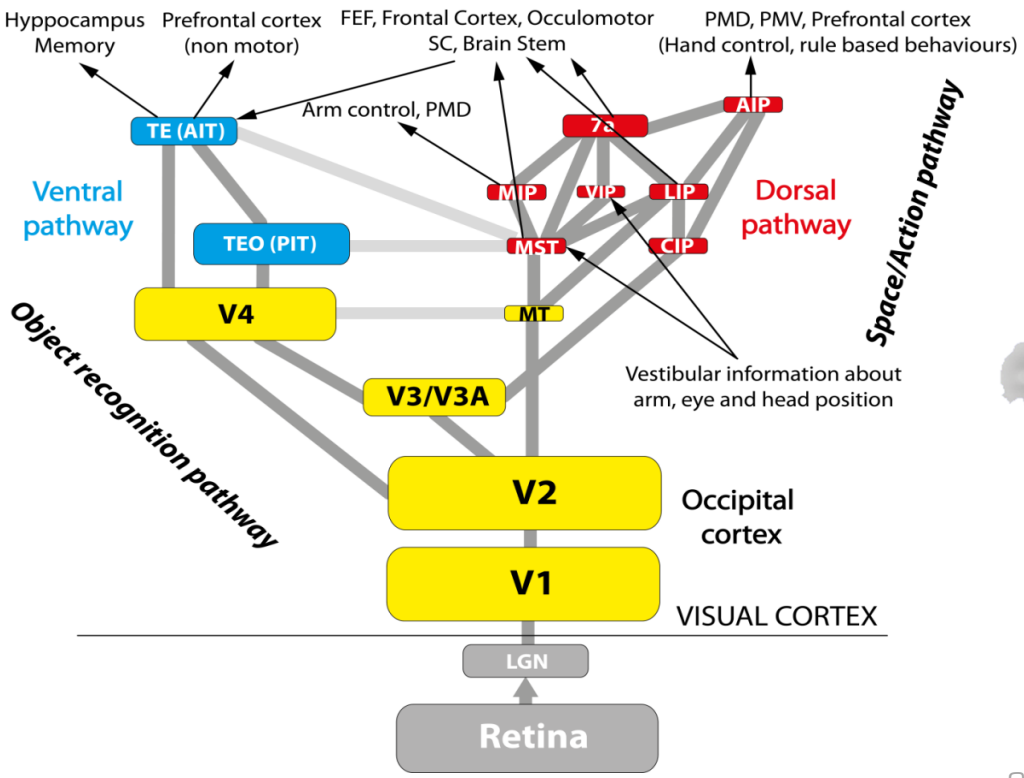


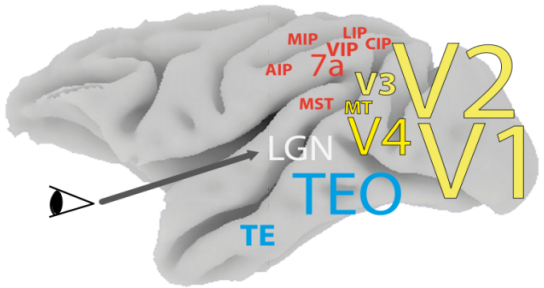
- MT reacts to motion in a certain view direction (0.1 sec after motion has happened)
- V1: cells react to edges of particular orientation;
  only 15% of axons come from the eyes,
  about 50% come from neighboring nerons (inhibition),
  remainder comes from higher level cortex (e.g., V2)
- Hippocampus: place and direction cells (cognitive map, [Tolman 1948])

Credit: Markus Vincze, Technische Universität Wien

The Human Visual System

- 55% of neo-cortex of primates devoted to vision
- 70% of this is in V1 and V2: general vision functions



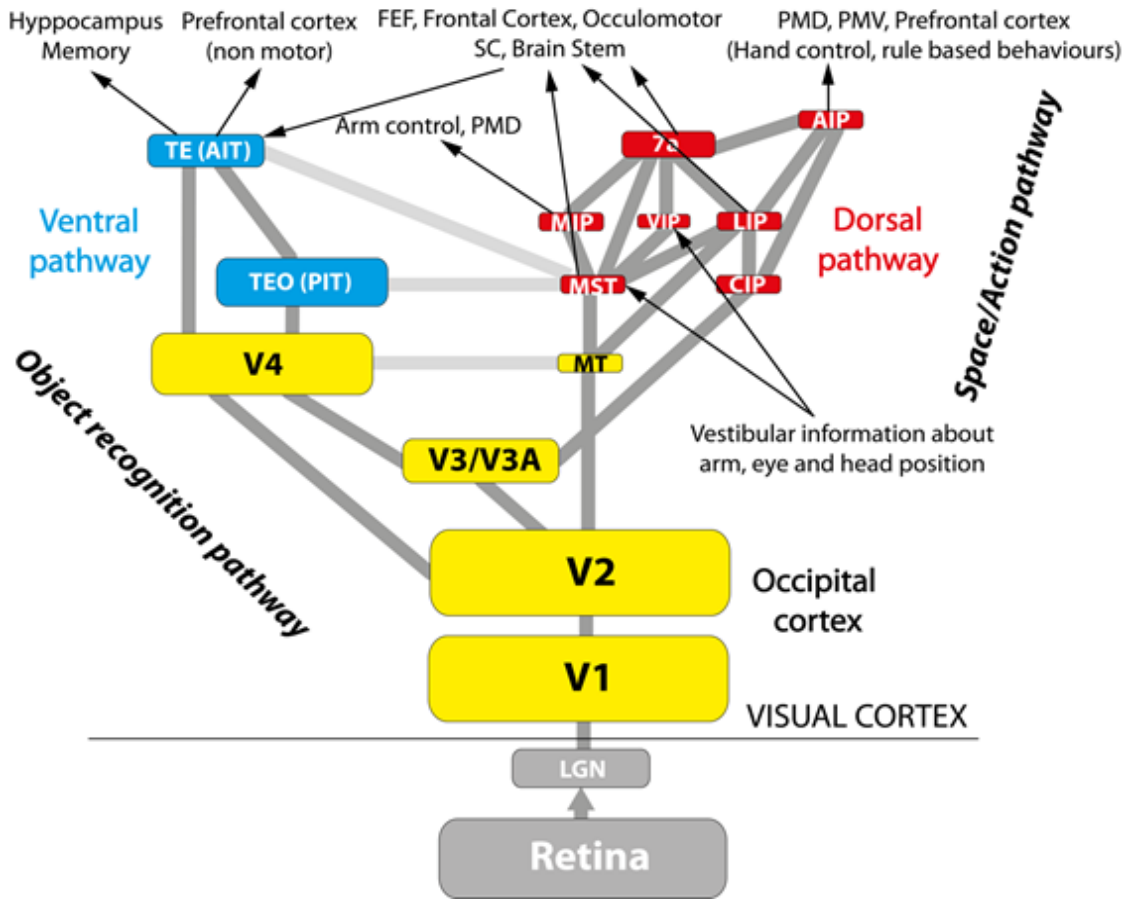


Credit: Norbert Krüger, University of Southern Denmark

The Human Visual System

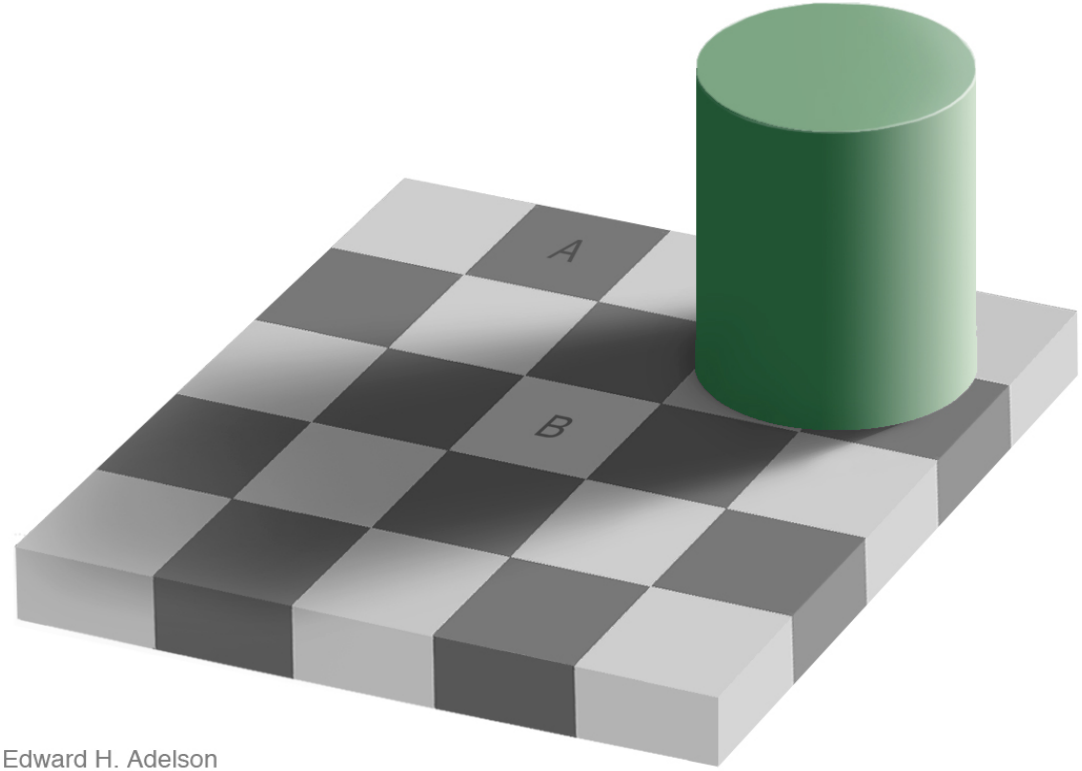
Areas

- Occipital Cortex
- Dorsal Pathway
- Ventral Pathway



Credit: Norbert Krüger, University of Southern Denmark

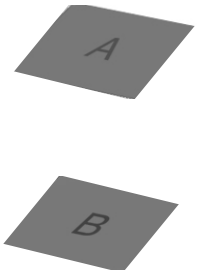
Cameras Measure Light
Humans Measure Scene Properties



Edward H. Adelson

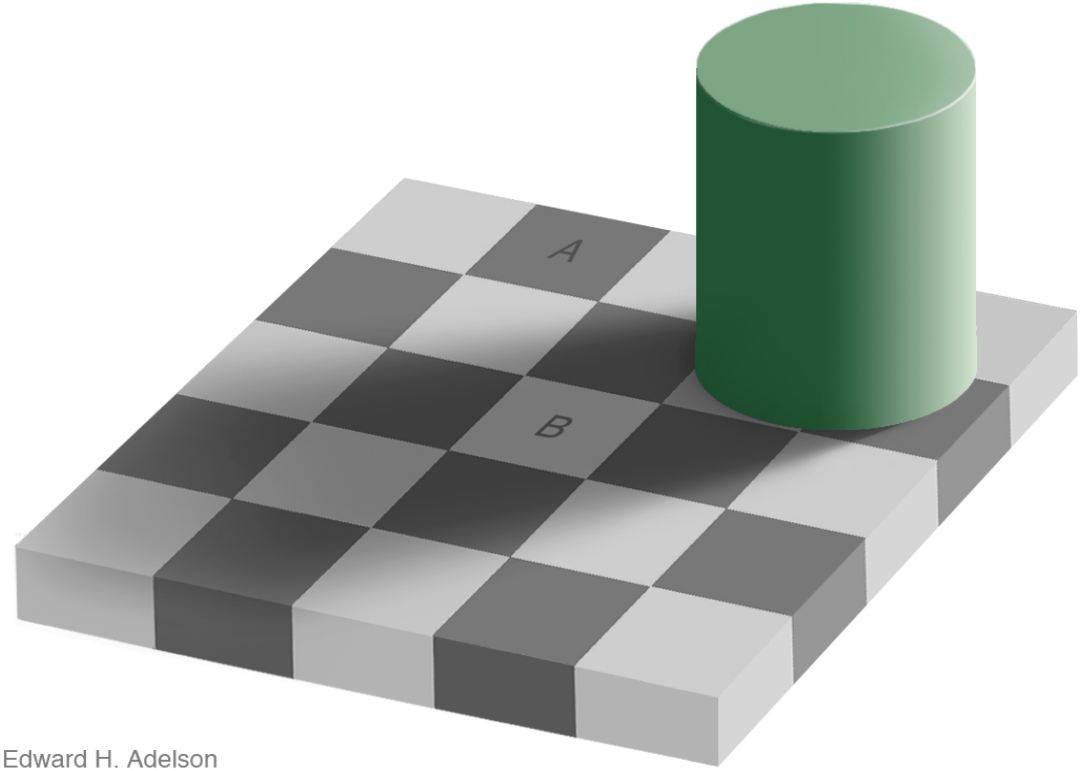
Credit: Markus Vincze, Technische Universität Wien

Cameras Measure Light
Humans Measure Scene Properties



Credit: Markus Vincze, Technische Universität Wien

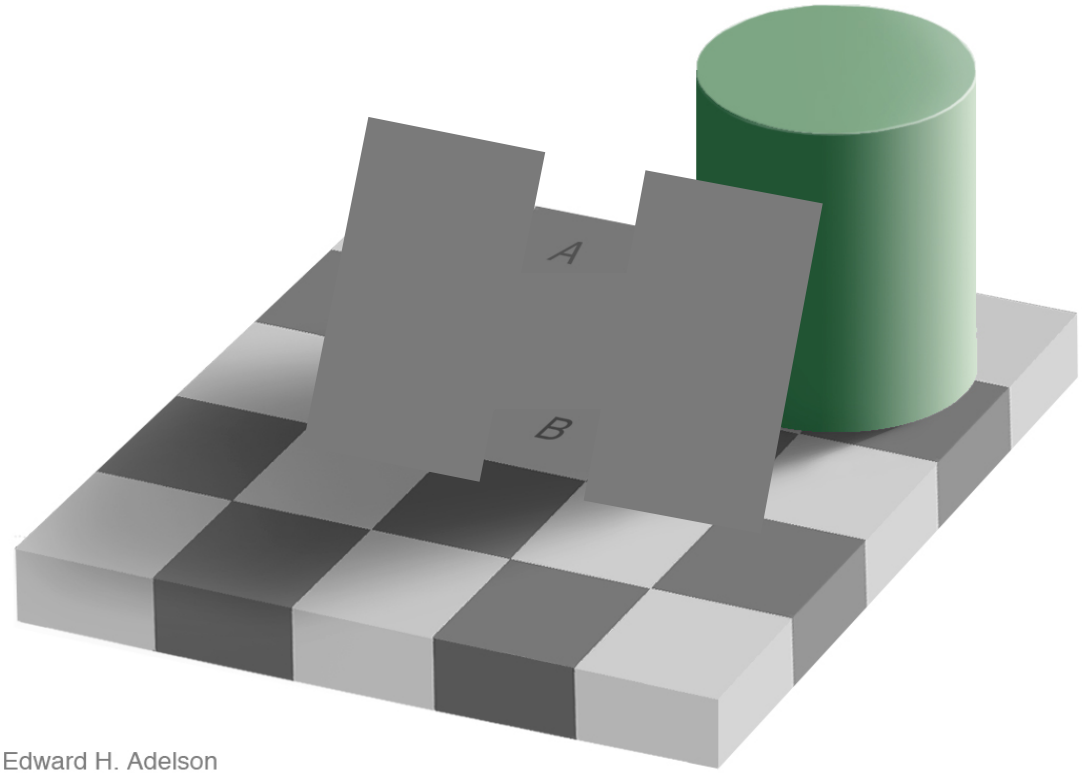
Cameras Measure Light
Humans Measure Scene Properties



Edward H. Adelson

Credit: Markus Vincze, Technische Universität Wien

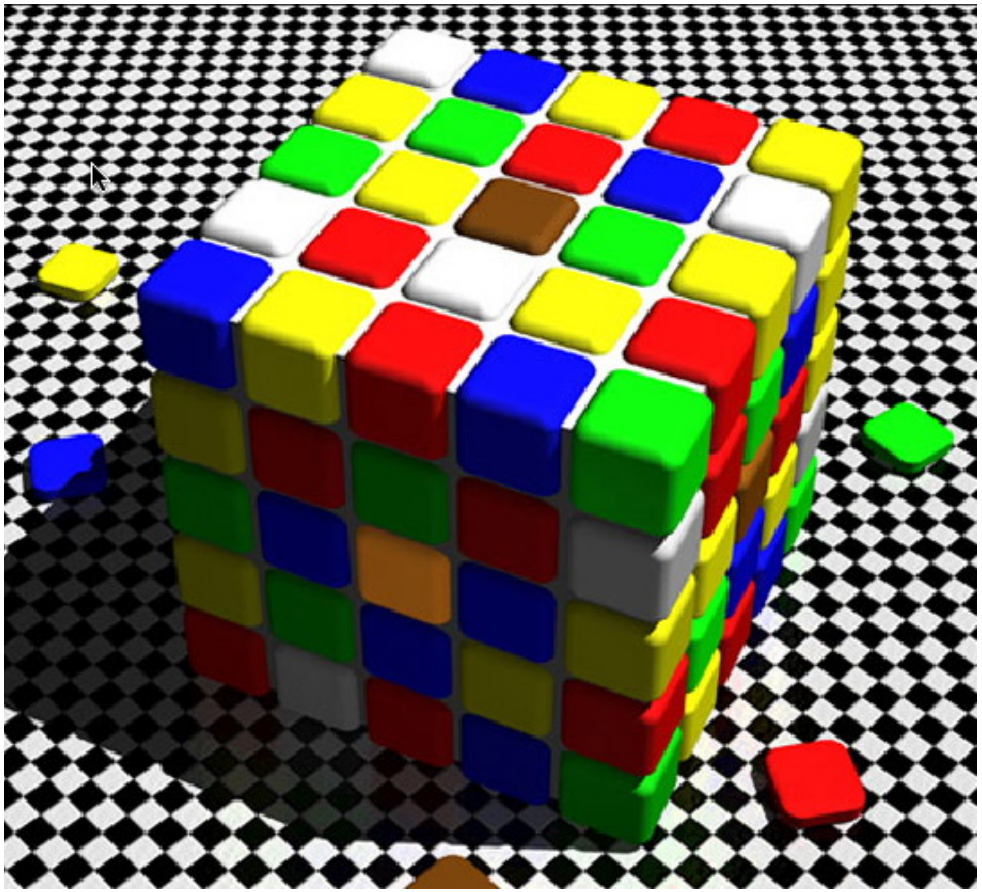
Cameras Measure Light
Humans Measure Scene Properties

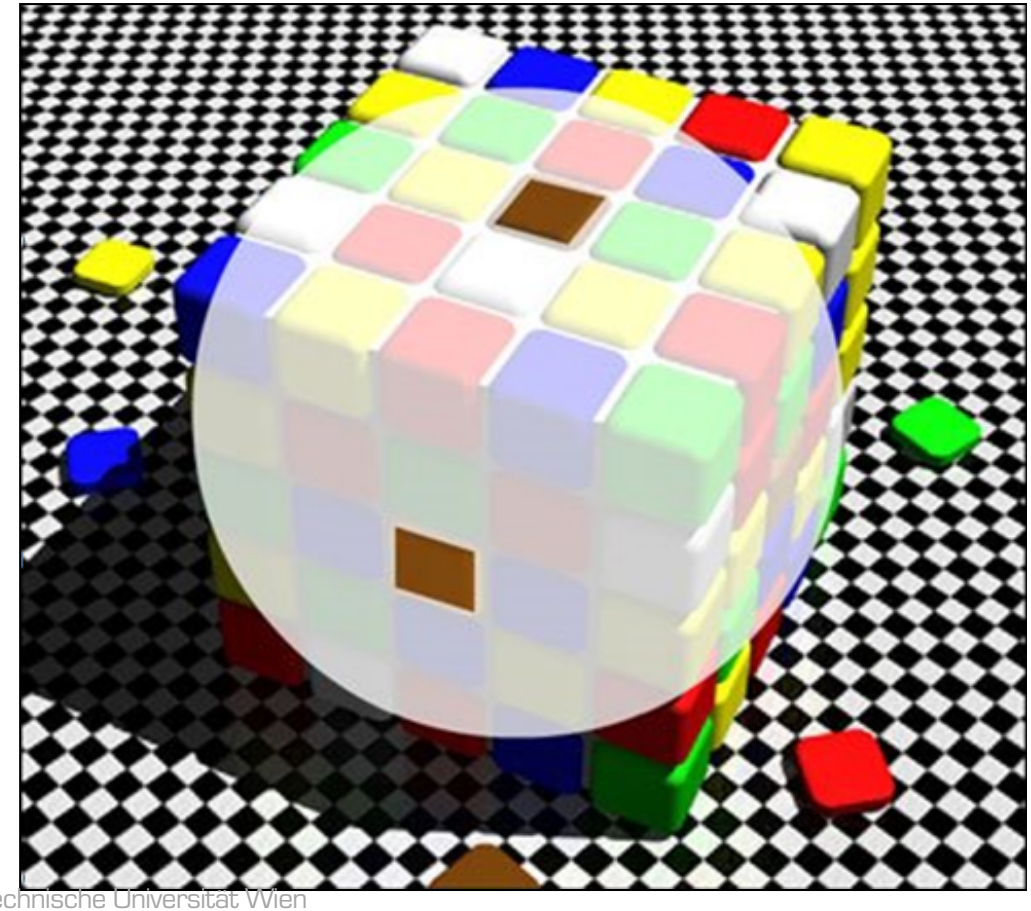


Edward H. Adelson

Credit: Markus Vincze, Technische Universität Wien

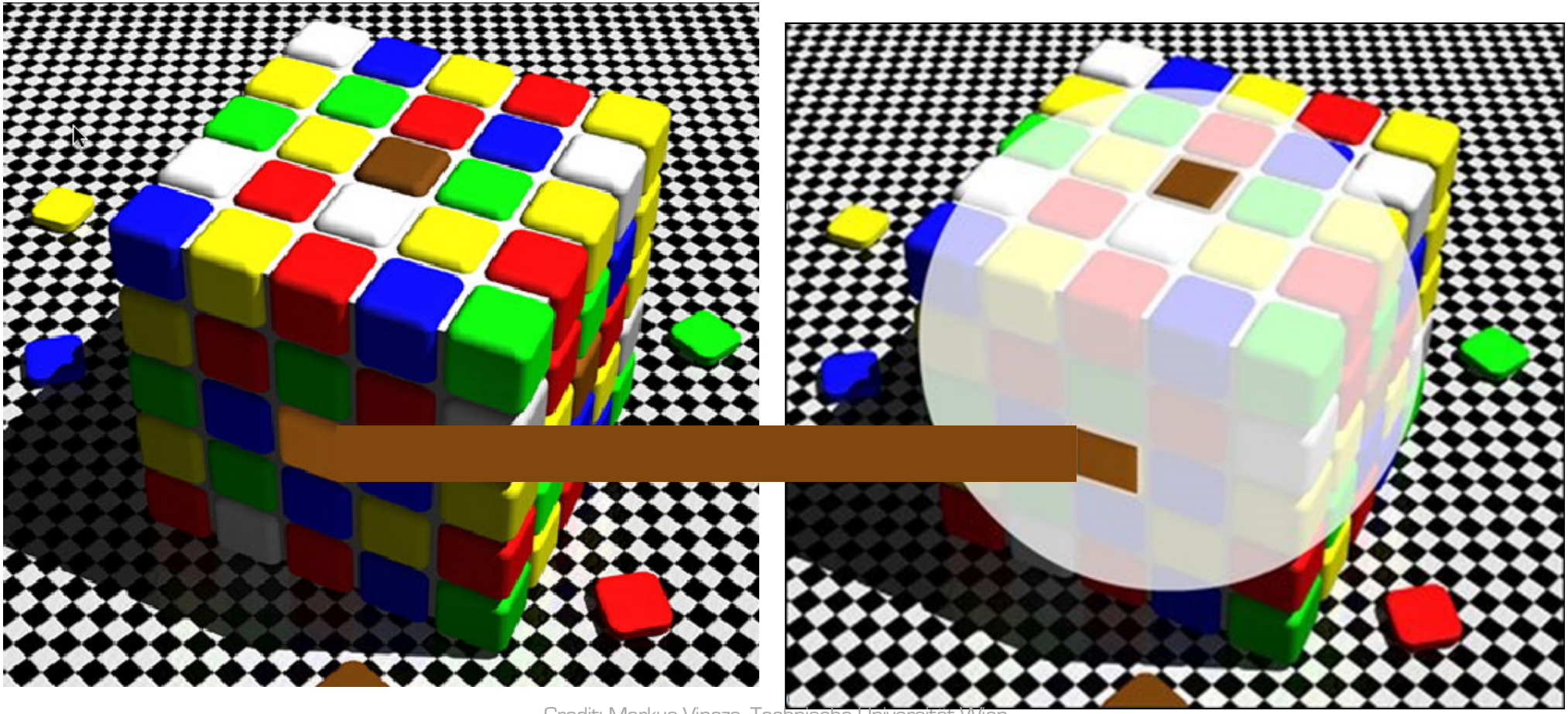
Colour Cameras Measure Light
Humans Measure Scene Properties





Credit: Markus Vincze, Technische Universität Wien

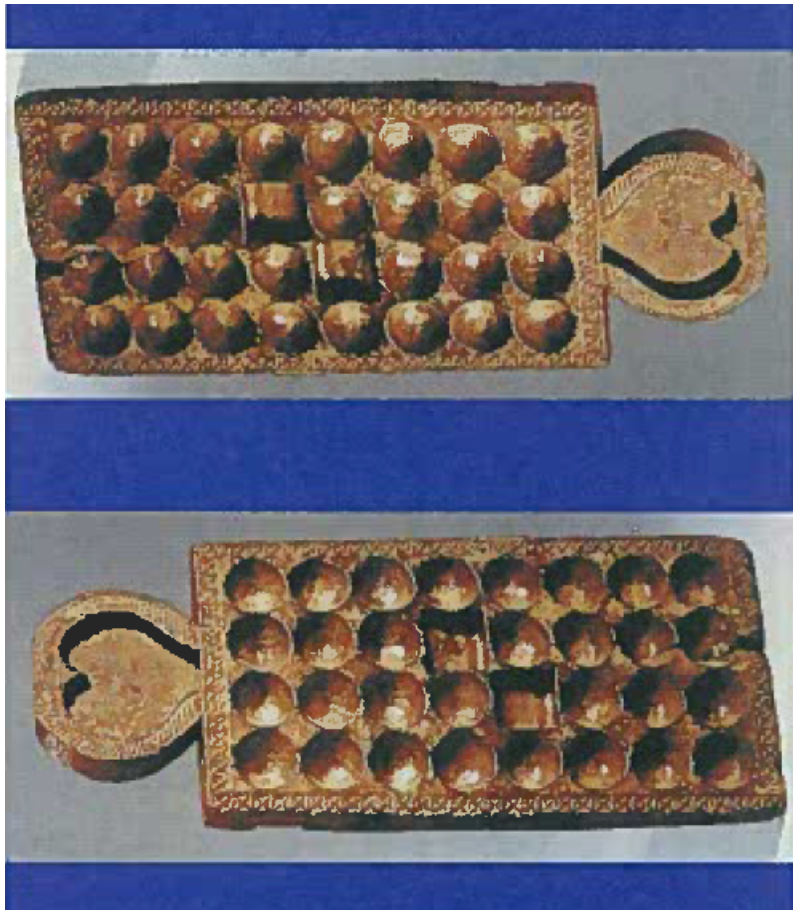
Colour Cameras Measure Light
Humans Measure Scene Properties



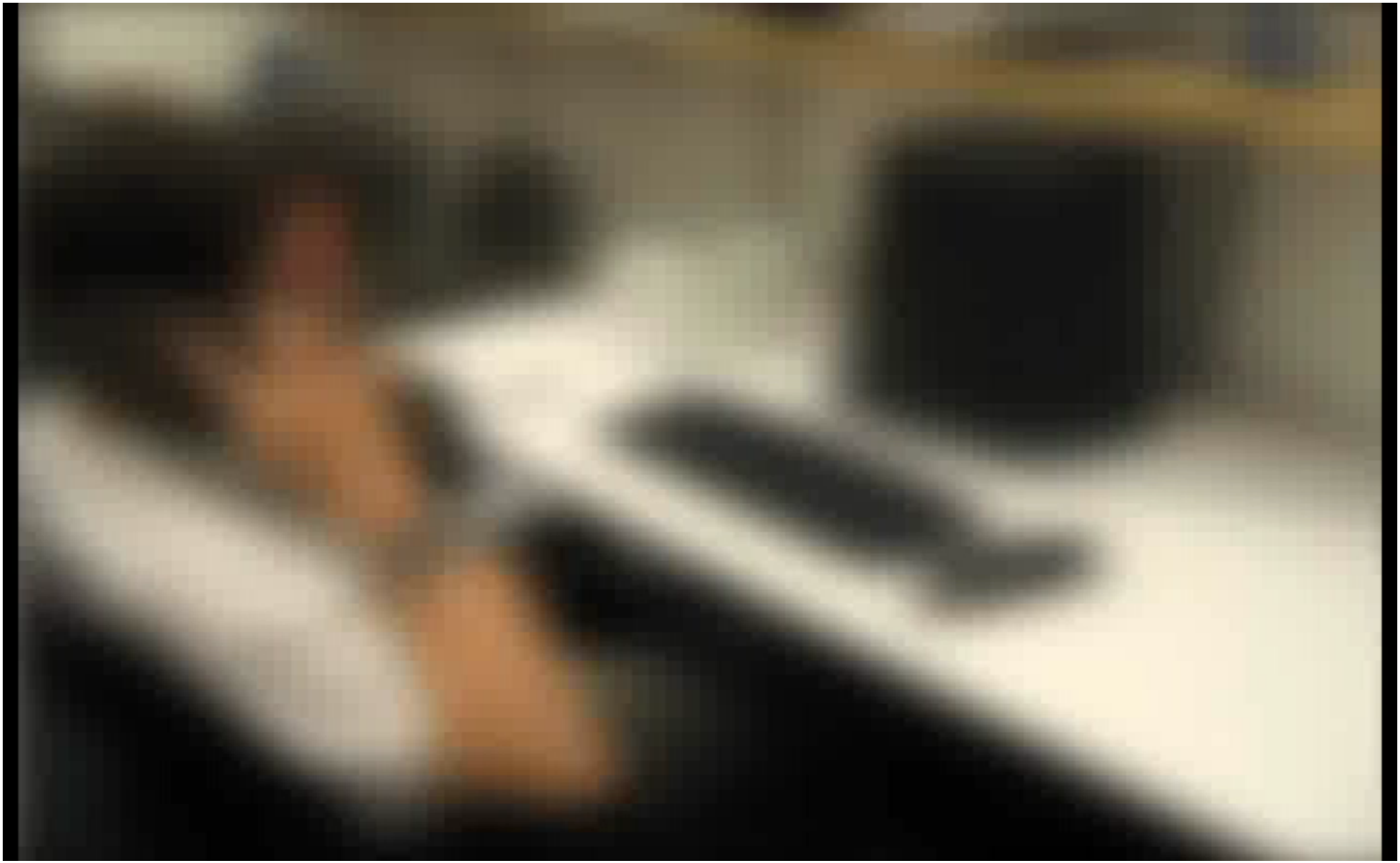
Credit: Markus Vincze, Technische Universität Wien

Measuring Light vs. Measuring Scene Properties

Assumption: light comes from above



Credit: Markus Vincze, Technische Universität Wien



Credit: Markus Vincze, Technische Universität Wien

Context is important but might be misleading

Credit: Markus Vincze, Technische Universität Wien

Vision has to solve an ill-posed Problem

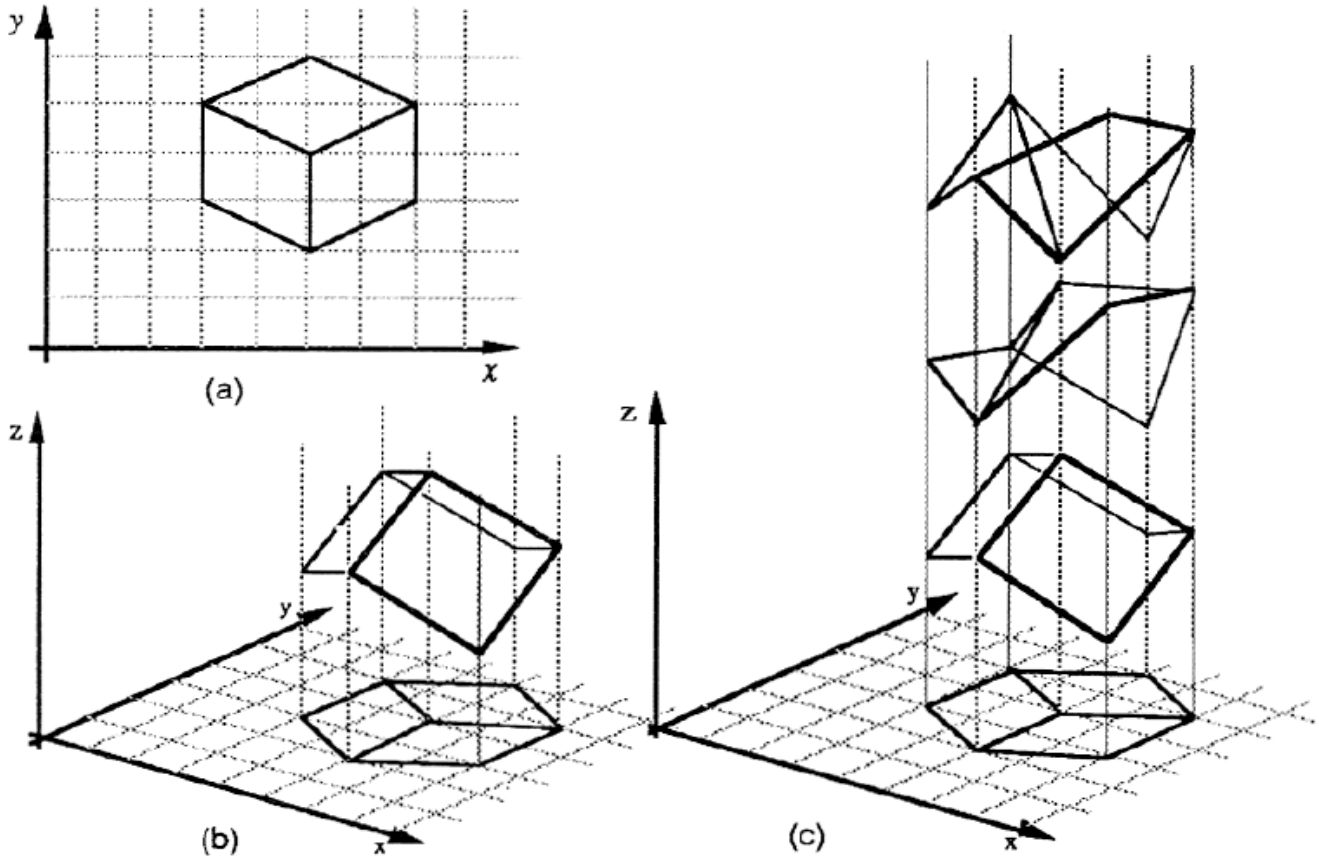
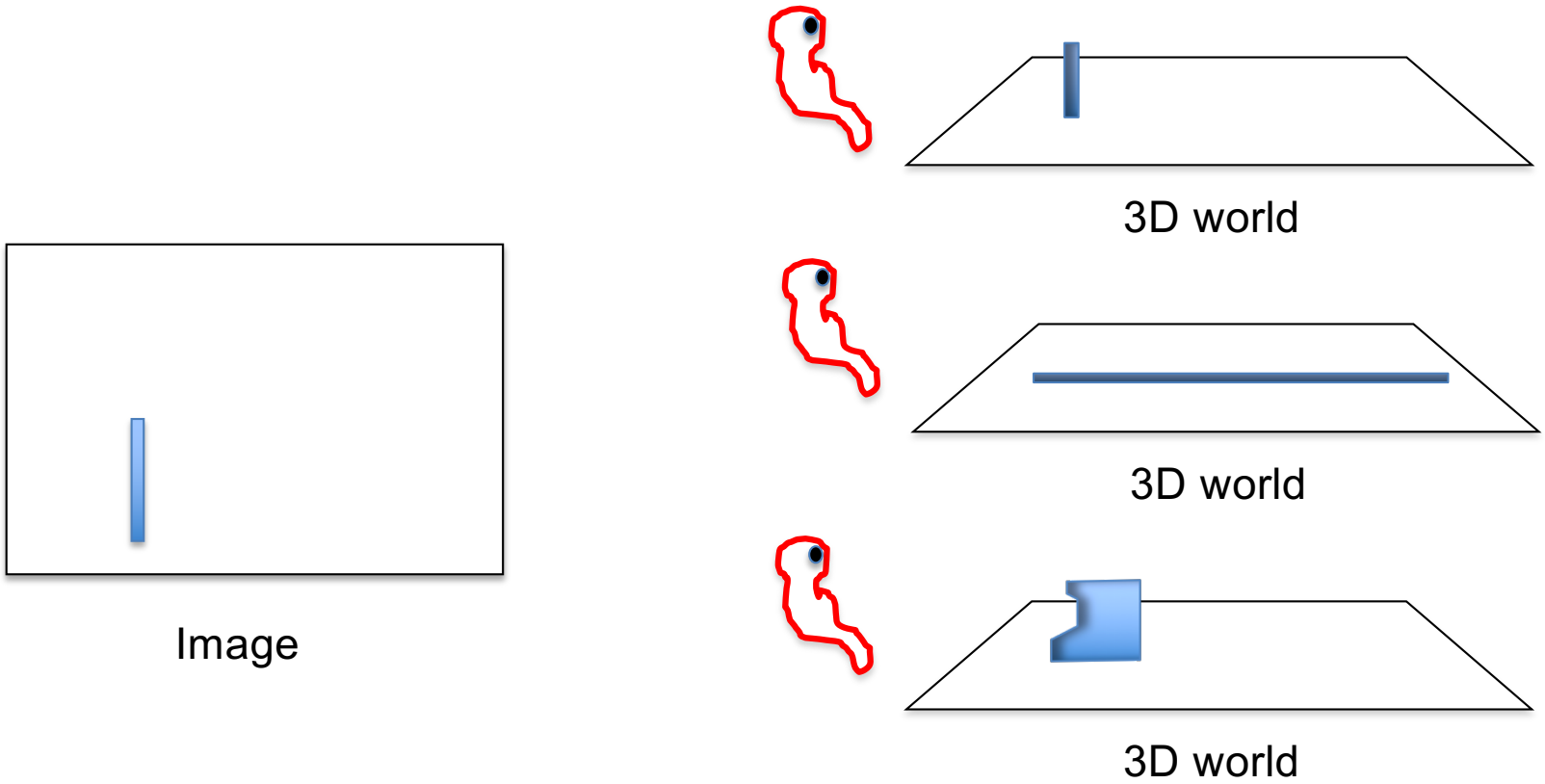


Figure 1. (a) A line drawing provides information only about the x, y coordinates of points lying along the object contours. (b) The human visual system is usually able to reconstruct an object in three dimensions given only a single 2D projection (c) Any planar line-drawing is geometrically consistent with infinitely many 3D structures.

[Sinha & Adelson 93]

Credit: Markus Vincze, Technische Universität Wien

Generic View Assumption

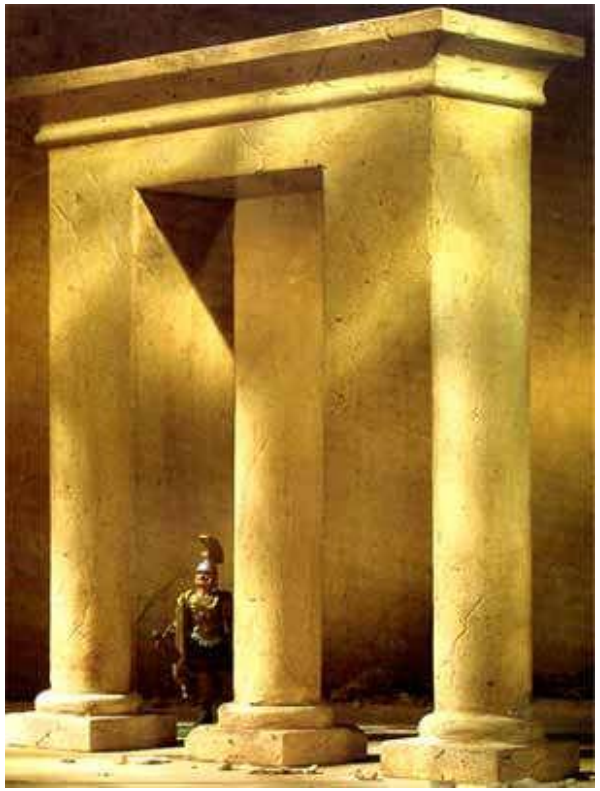


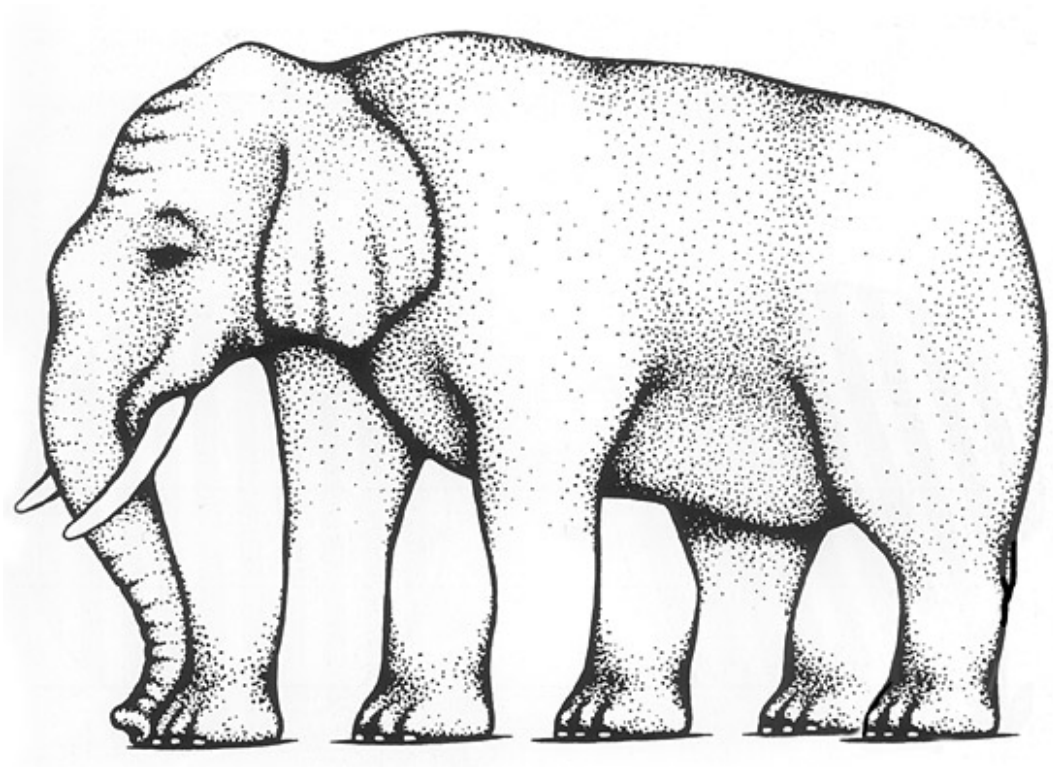
- Do not assume a special position in the world
- Most generic interpretation is to see a vertical line as a vertical line in 3D

Credit: Markus Vincze, Technische Universität Wien

[Freeman, 93]

Gestalts and Continuity





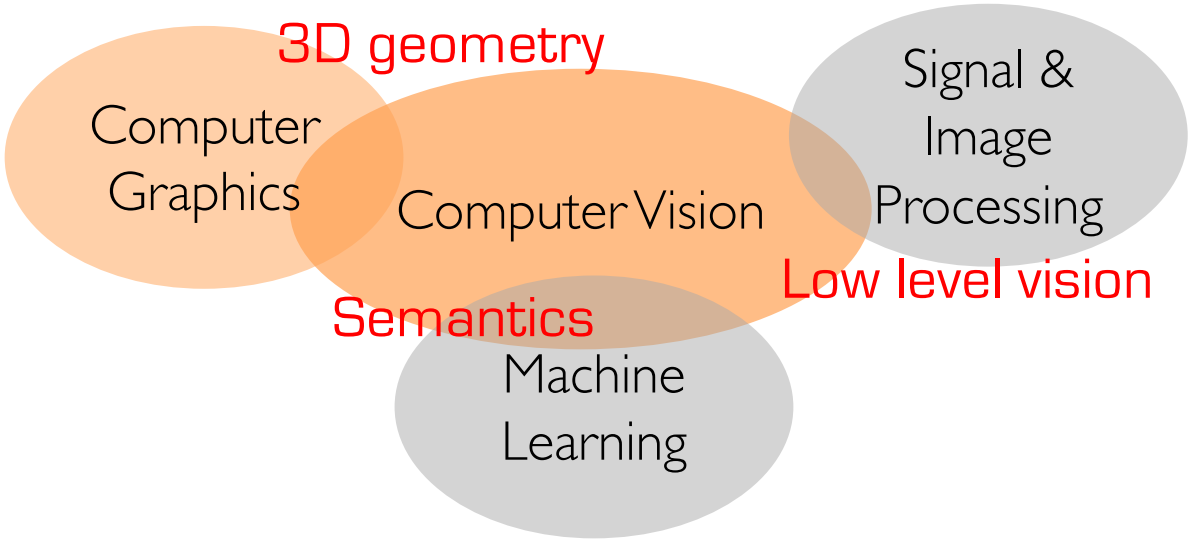
Credit: Markus Vincze, Technische Universität Wien

Computer Vision

Multi-disciplinary subject

- Optics
- Electronics
- Mechanical Engineering
- Computer Science
- Artificial Intelligence

Computer Vision



An incomplete view of related disciplines

Credit: Francesca Odone, University of Genova

Computer Vision

Includes many techniques

- Image processing
- Projective geometry
- Pattern recognition
- Graph theory
- Geometric modelling
- Statistical learning
- Cognitive processing
- ...

Computer Vision – A Brief History

1960s

started as a student summer project at MIT

the goal of the project was to link a camera to a computer, getting the computer to describe what it saw

1970s

mainly seen as a part of artificial intelligence (emulating human visual perception)

simplification of the world ("lego-like")

pioneering work of David Marr (mainly on) low level vision

1980s

more sophisticated mathematics

Credit: Francesca Odone, University of Genova

Computer Vision – A Brief History

1990s computer vision reaches maturity

- geometric approaches
- optical flow

2000s machine learning paves the way to scene understanding

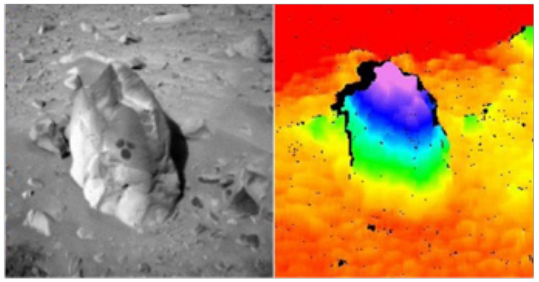
Today we deal with very large data and very large datasets
(deep architectures)

Credit: Francesca Odone, University of Genova

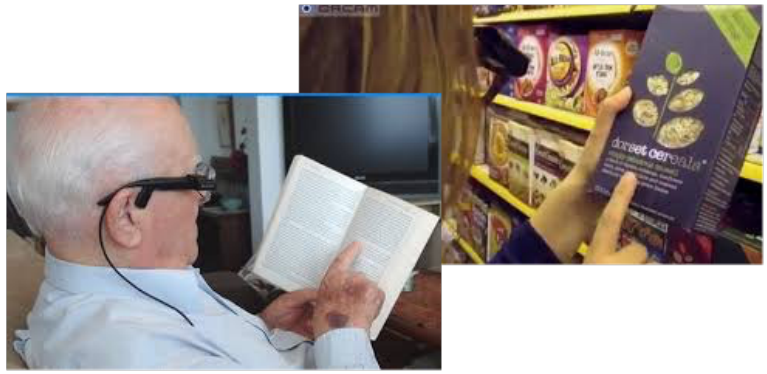
Computer Vision – A Brief History

60s	Blocks World / Micro Worlds
70s	Knowledge-based and Model-based vision
80s	Hierarchical modular information processing Mathematically-sound robust early vision
90s	Computational projective geometry Appearance-based vision
00s	Probabilistic techniques, machine learning, & video interpretation
10s	Deep learning and very large data-sets

Computer Vision



Where are we today?



Credit: Francesca Odone, University of Genova

Face detection

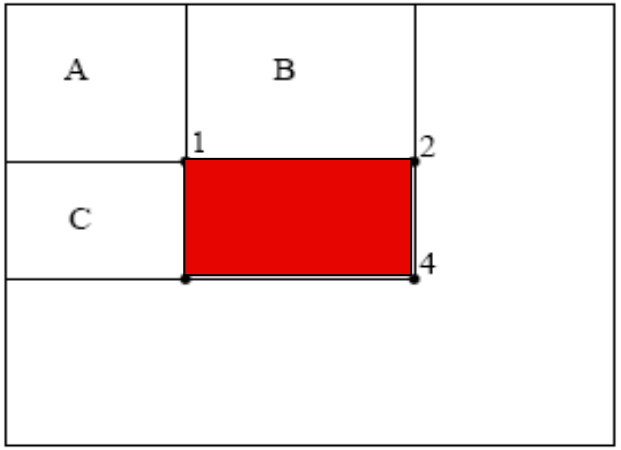
[Face priority AE] When a bright part of the face is too bright

Credit: Markus Vincze, Technische Universität Wien

Face detection

Haar filters and integral image

[Viola and Jones, ICCV 2001]

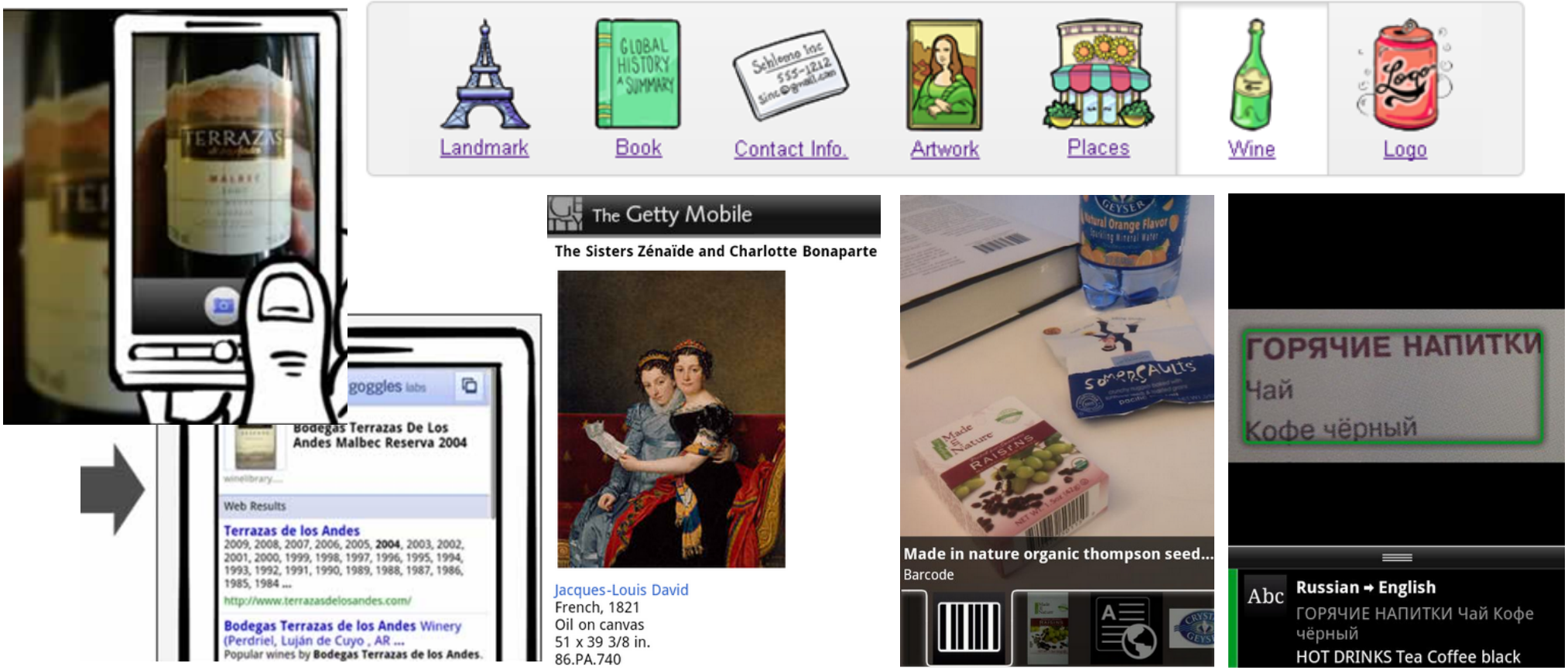


The average intensity in the block is computed with four sums independently of the block size

Credit: Markus Vincze, Technische Universität Wien

Google Goggles

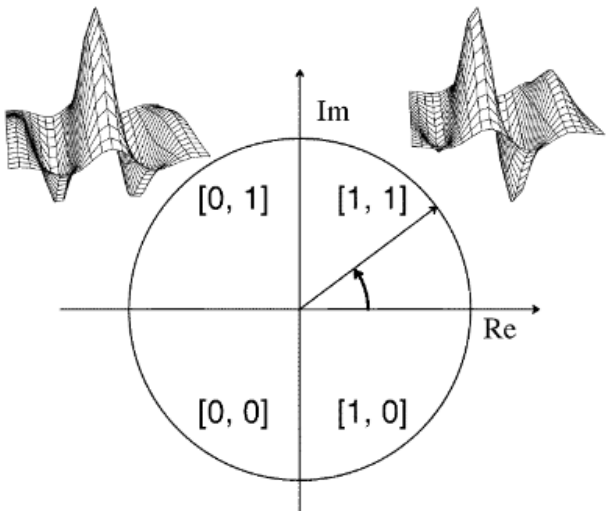
- Search the web with pictures
- Obtain name, company, similar pictures

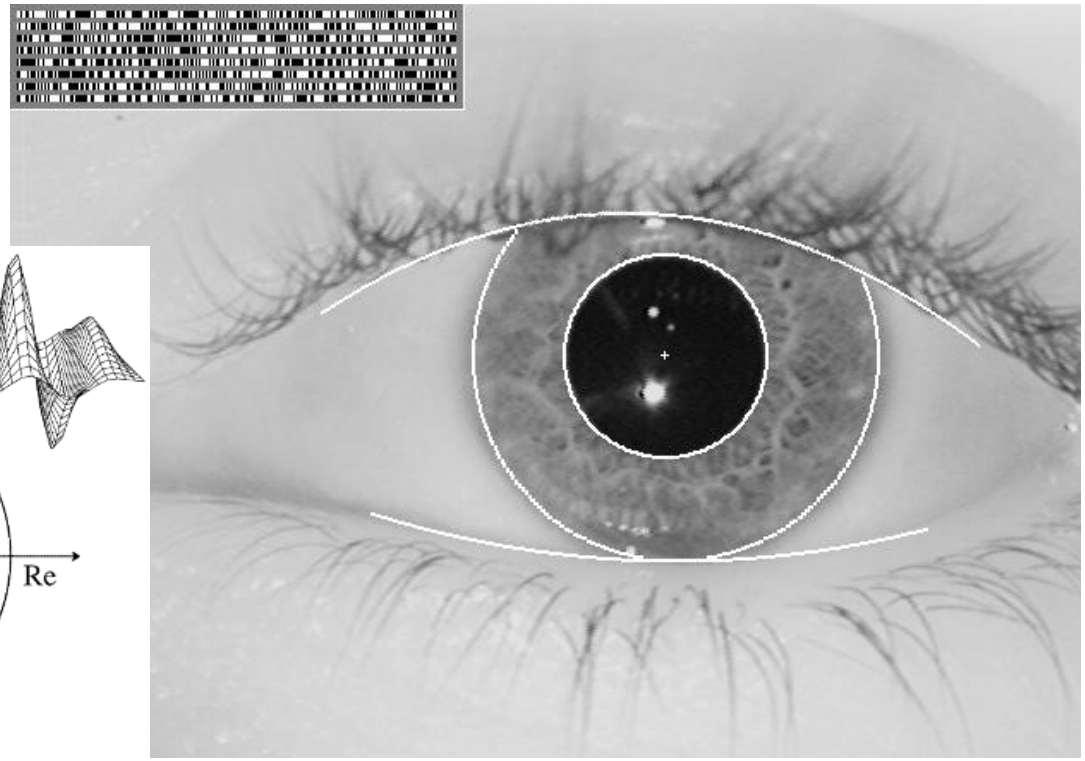


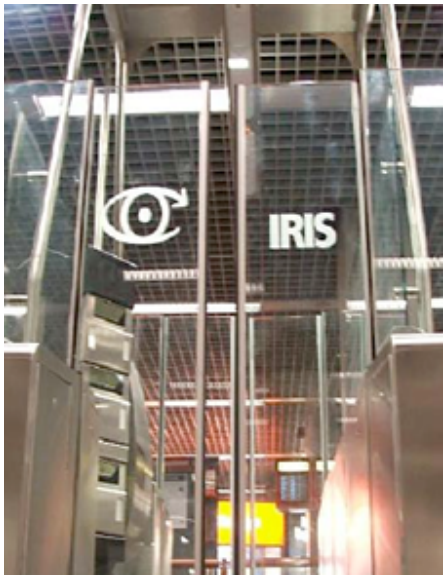
Credit: Markus Vincze, Technische Universität Wien

Iris Recognition

2D wavelet decoding





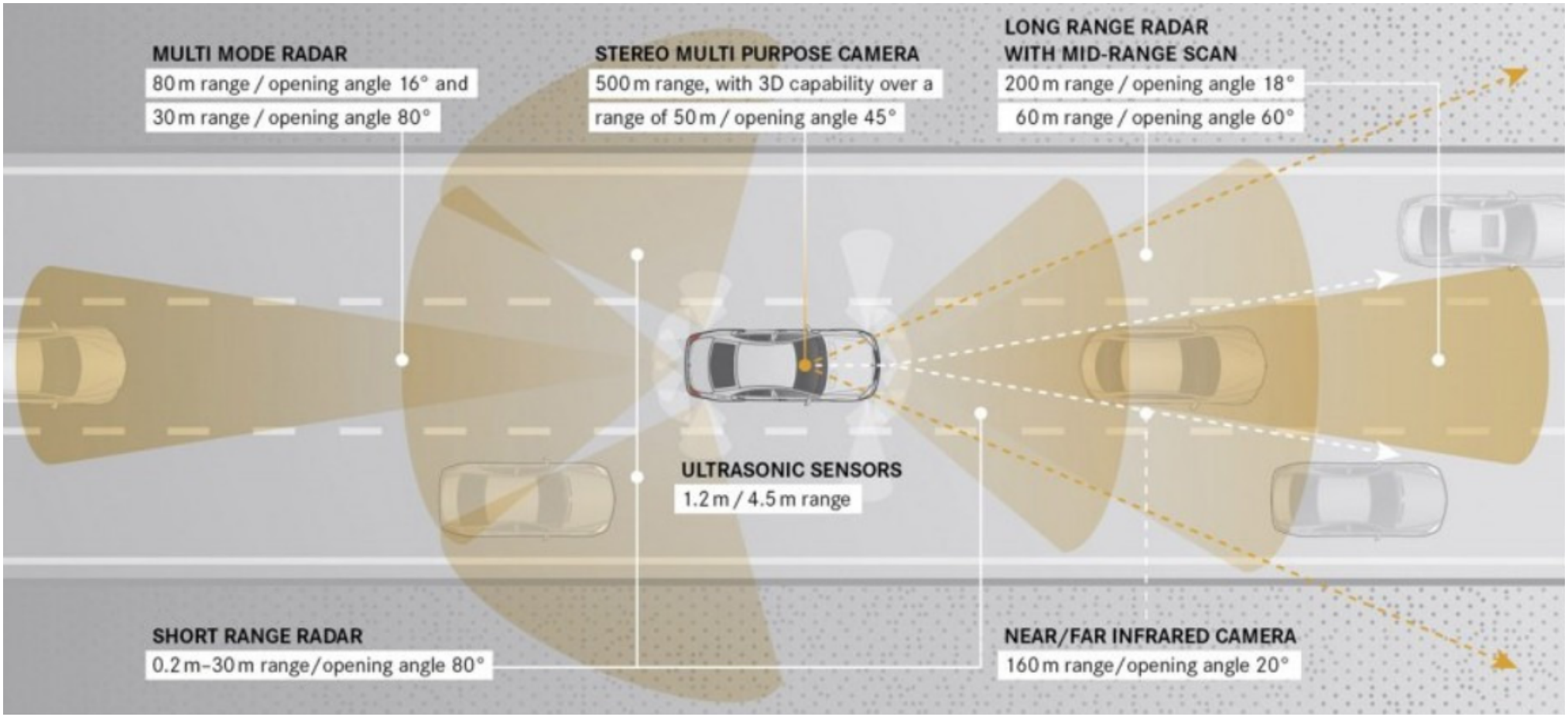


Access control, e.g., border

[John Daugman: How Iris Recognition Works, 2004]
Genetically identical eyes also have different iris.

Credit: Markus Vincze, Technische Universität Wien

Assisted Driving

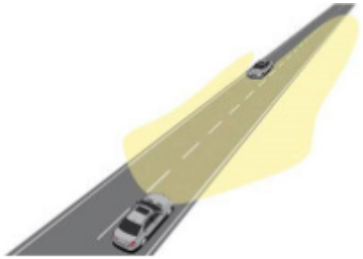


[Dariu M. Gavrila, Mercedes E- und S-Class; ICVS 2013]

Credit: Markus Vincze, Technische Universität Wien

Assisted Driving

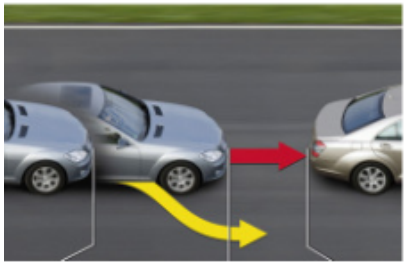
Traffic Signs



Adaptive High Beam

Nightview

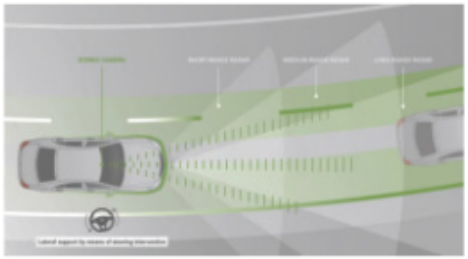
Attention



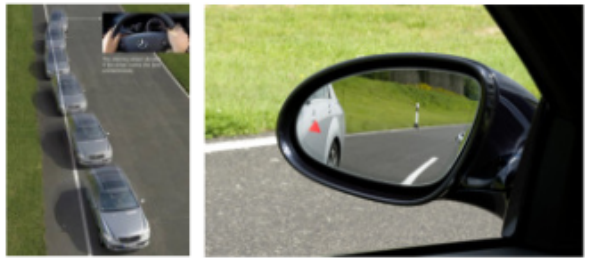
S-Class only

Pre-Crash Braking (longitudinal & lateral traffic) with Pedestrian Recognition (Active) Body Control

Parking



Adaptive Cruise Control
with Steering Assist



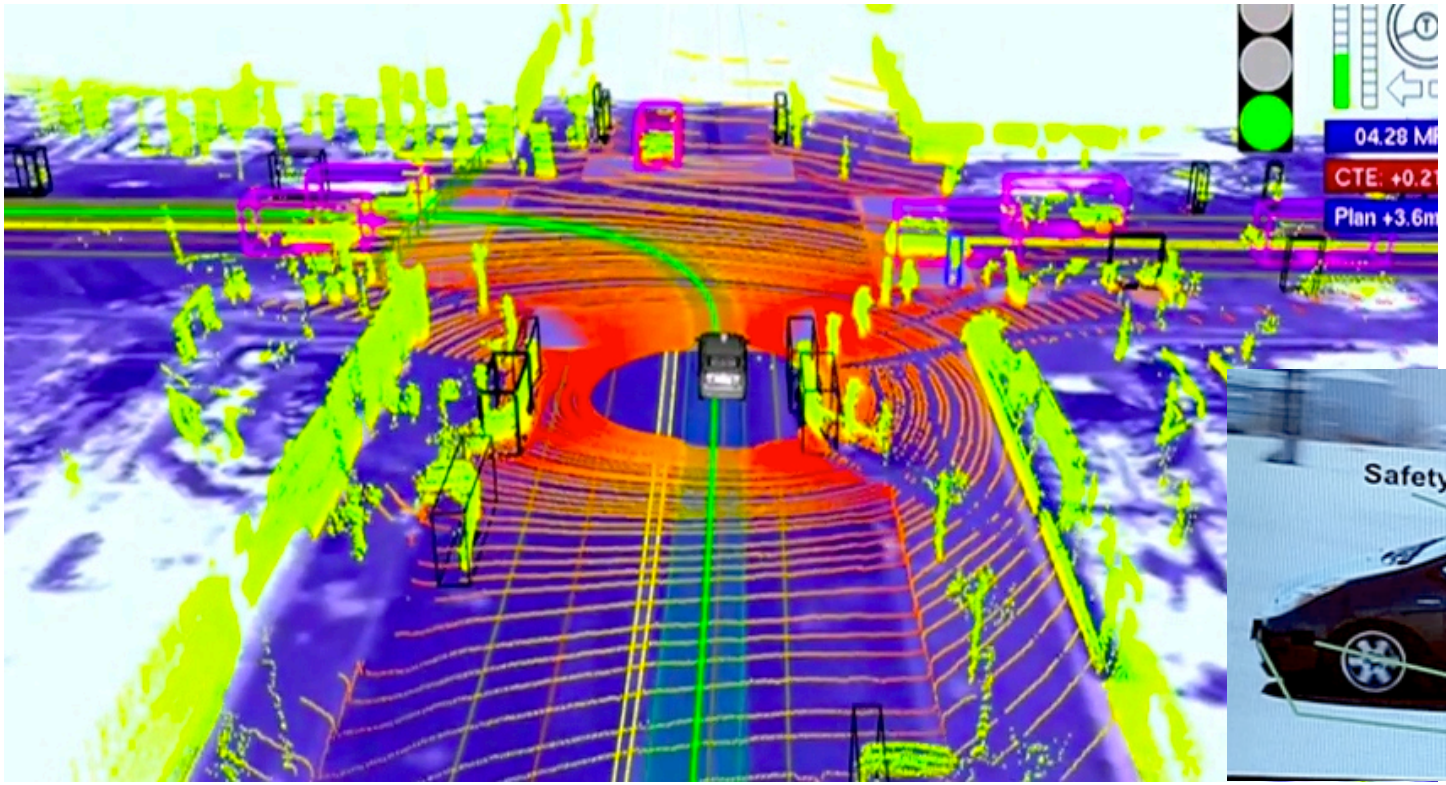
(Active)
Lane Keeping

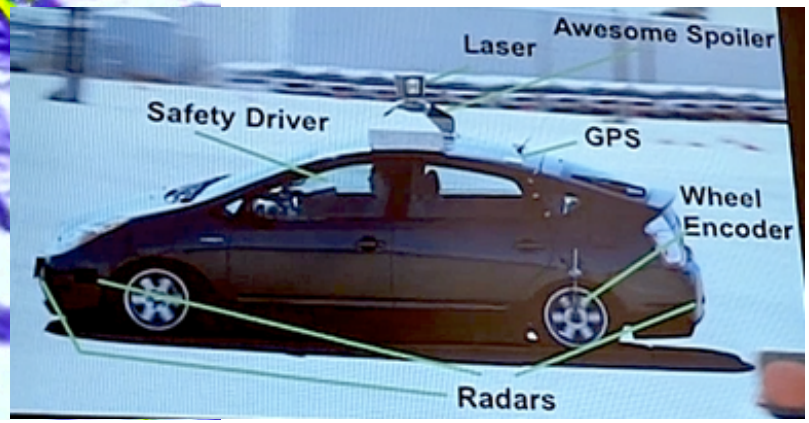
27

Credit: Markus Vincze, Technische Universität Wien

Google Driverless Car

Based on accurate maps, GPS, odometry, laser and radar (no cameras)





[E. Guizzo: How Google's Self-Driving Car Works, 2011]

Credit: Markus Vincze, Technische Universität Wien

Machine Vision in Robotic Inspection

Part position

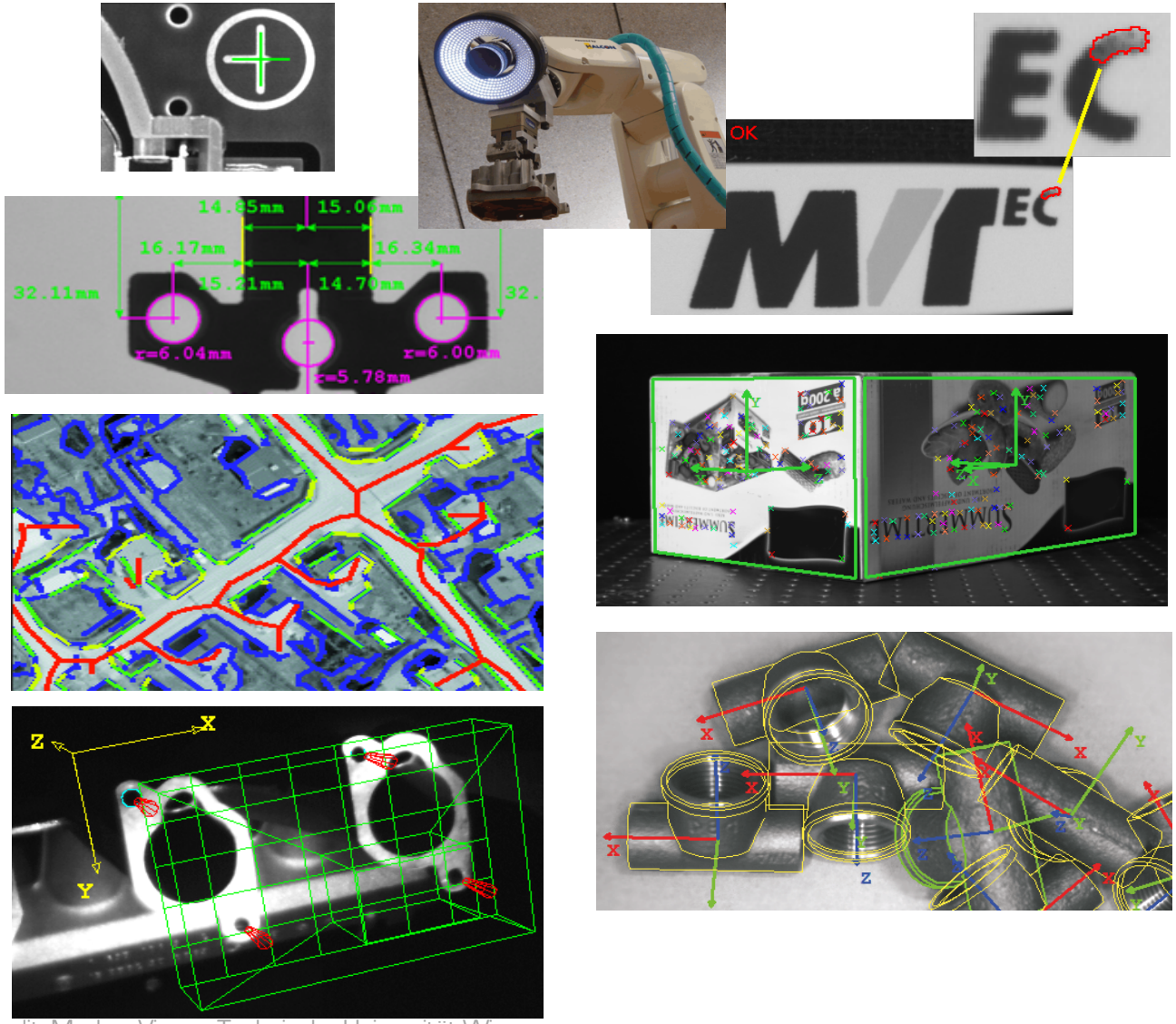
Print quality

Part measuring

Pose estimation

Plan generation

Pick from box
(simple parts)

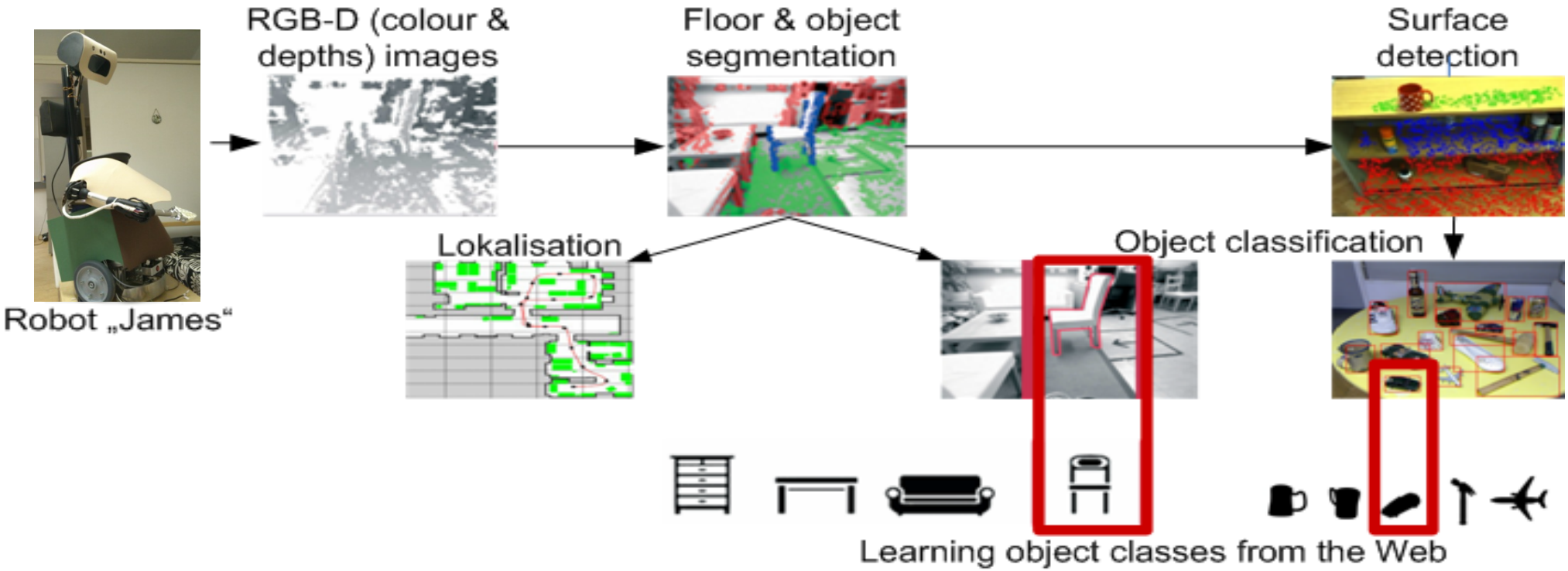


Credit: Markus Vincze, Technische Universität Wien

Situated Object Classification and Manipulation

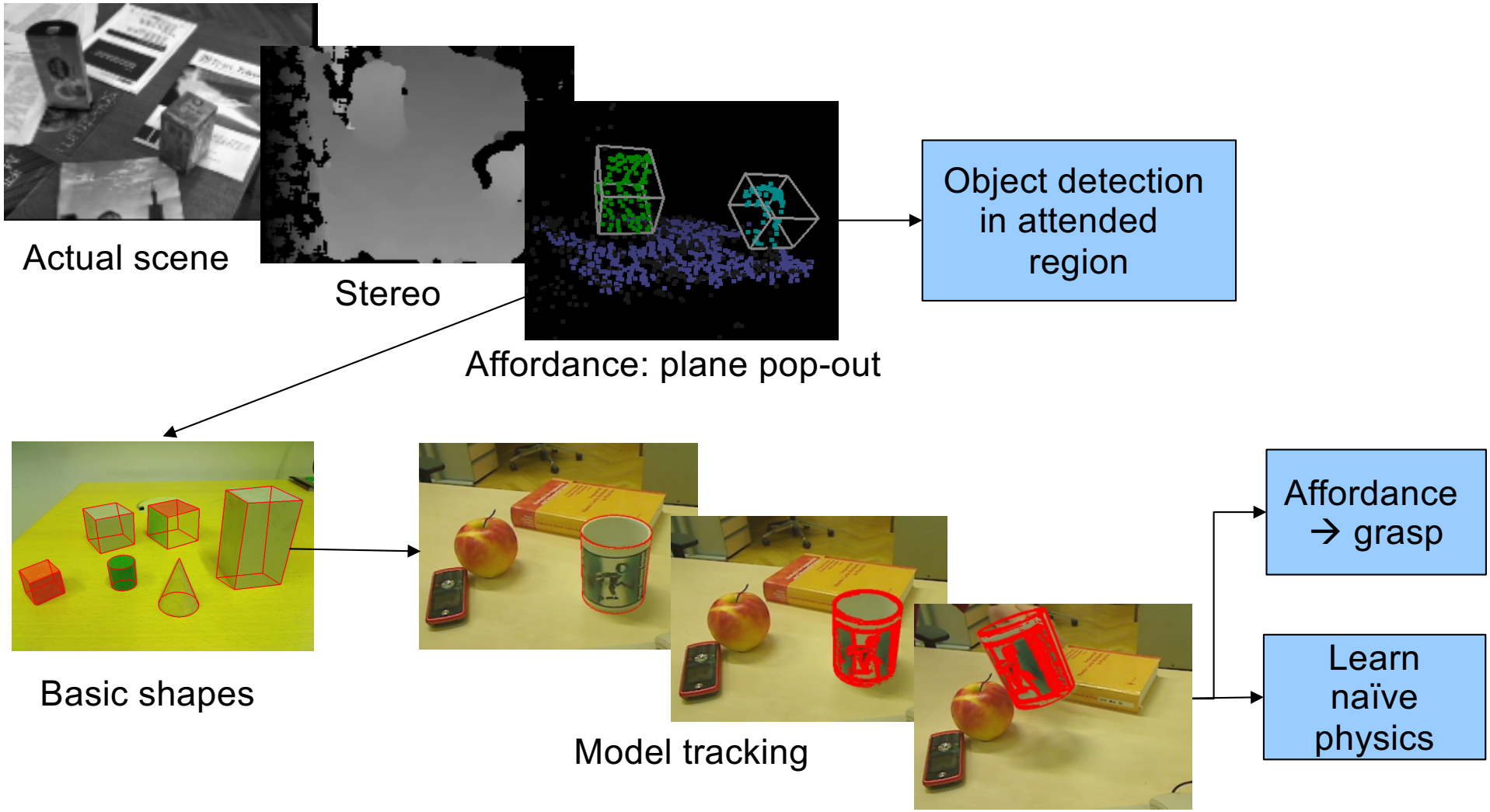
- Interaction of robot structures the environment

- Accumulation of object and scene knowledge



Credit: Markus Vincze, Technische Universität Wien

Table Plane Pop-out

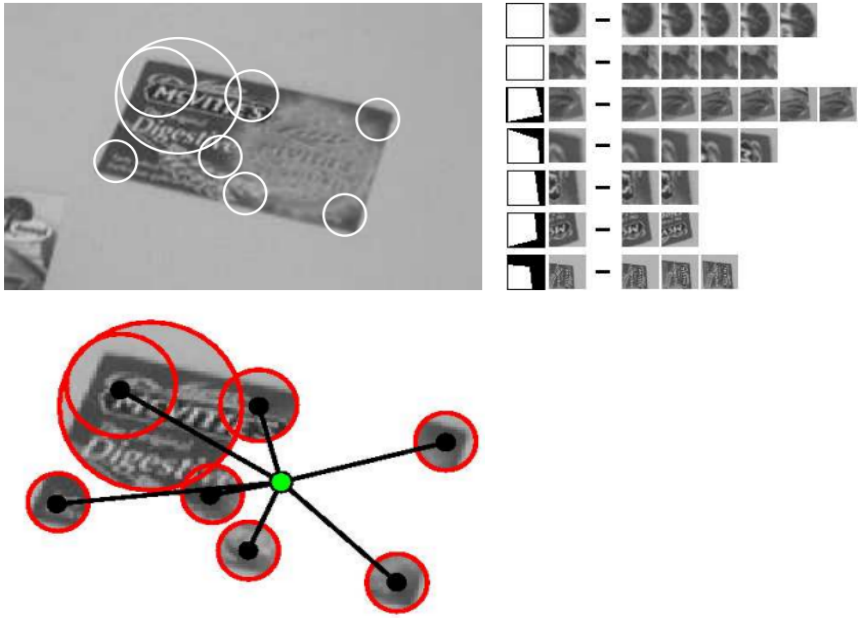


Credit: Markus Vincze, Technische Universität Wien

Object Recognition and Tracking

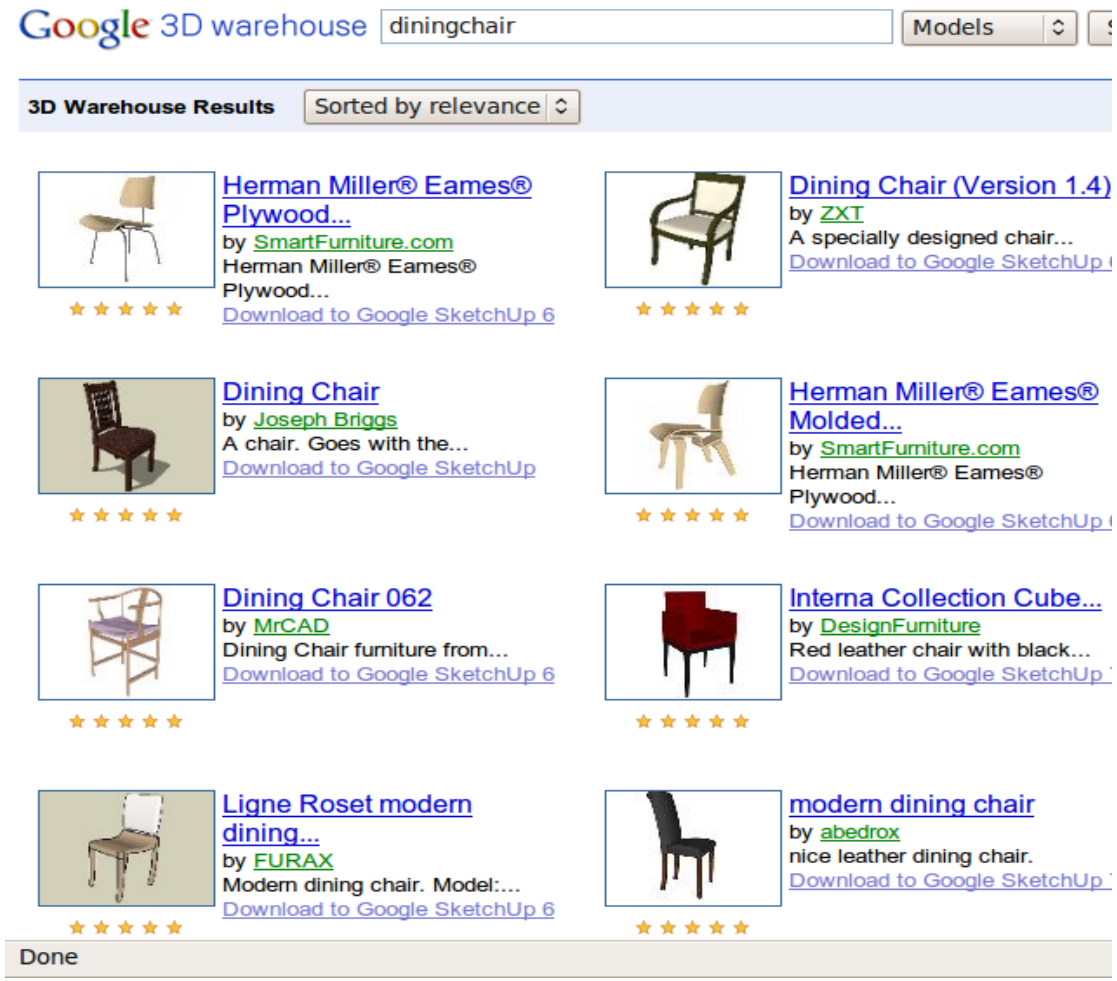
Interest points (corners) & local description of object

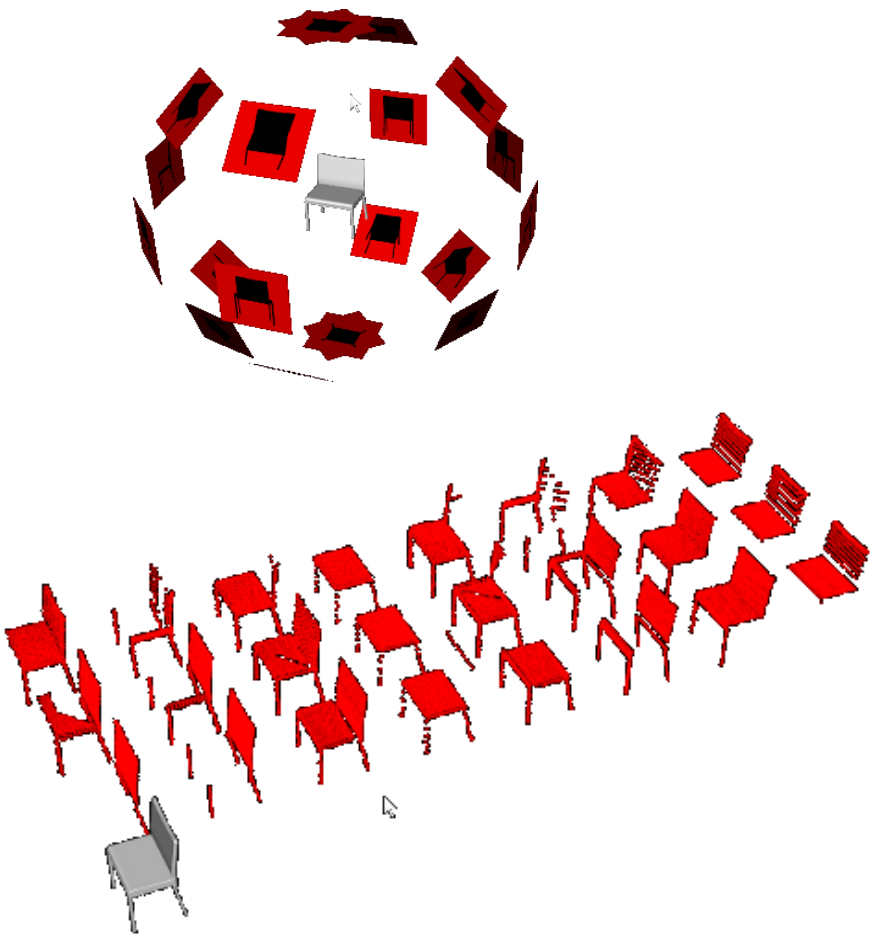
- "code book" for compact storage of many objects
- Recognition: relation of interest points to center



Credit: Markus Vincze, Technische Universität Wien

More Efficient: Learn from the Web





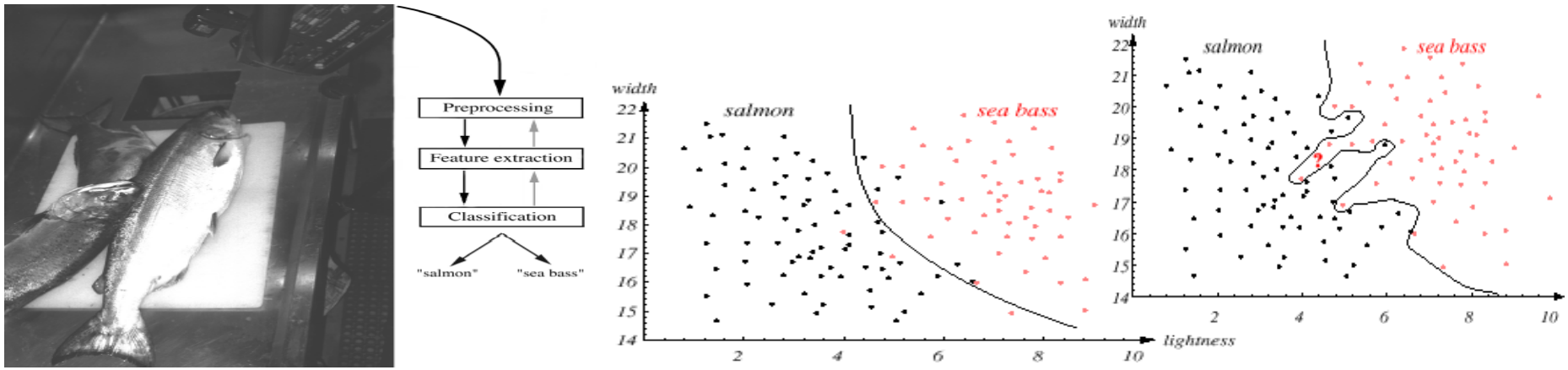
[Wohlkinger, IROS 2012]

Credit: Markus Vincze, Technische Universität Wien

Pattern Recognition

- "the act of taking in raw data and taking an action based on the category of the data" [1]

- *Methods:* statistics, machine learning, ...
-
- *Problem:* how to determine "category of data"



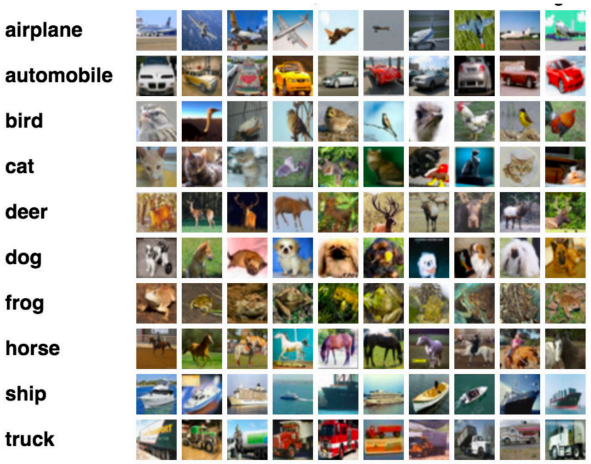
[1] Richard O. Duda, Peter E. Hart, David G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York.

Credit: Markus Vincze, Technische Universität Wien

Deep Learning

Facial Expression Classification with VGGnet CNN

Object Classification with Imagenet CNN

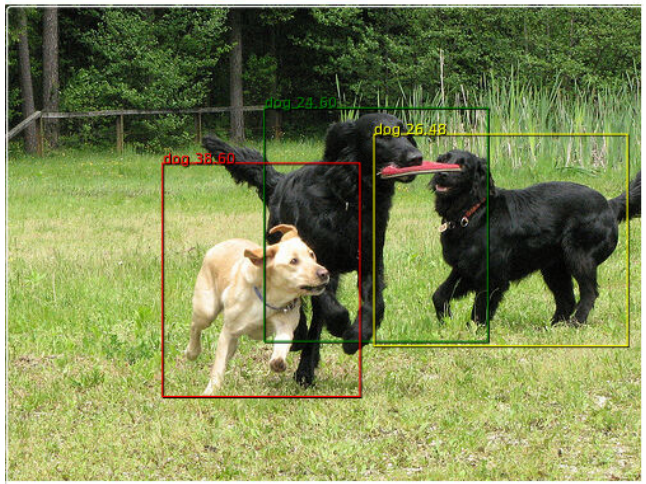


CIFAR-10 Object Classification with MiniVGGnet CNN

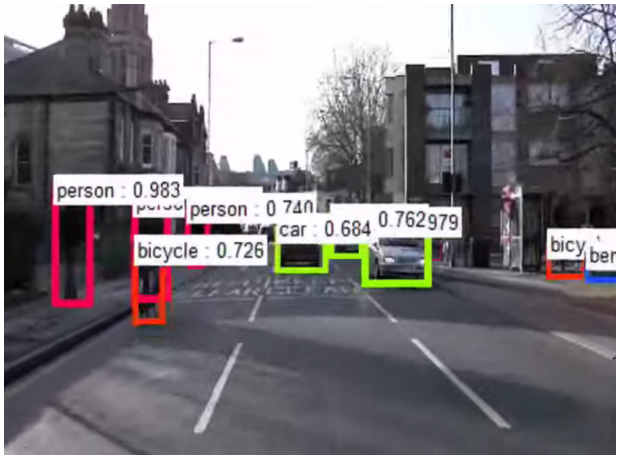
Credit A. Rosebrock, Deep Learning for Computer Vision 2017

Deep Learning

DeepPose Regression
Credit: Toshev and Szegedy CVPR 2014

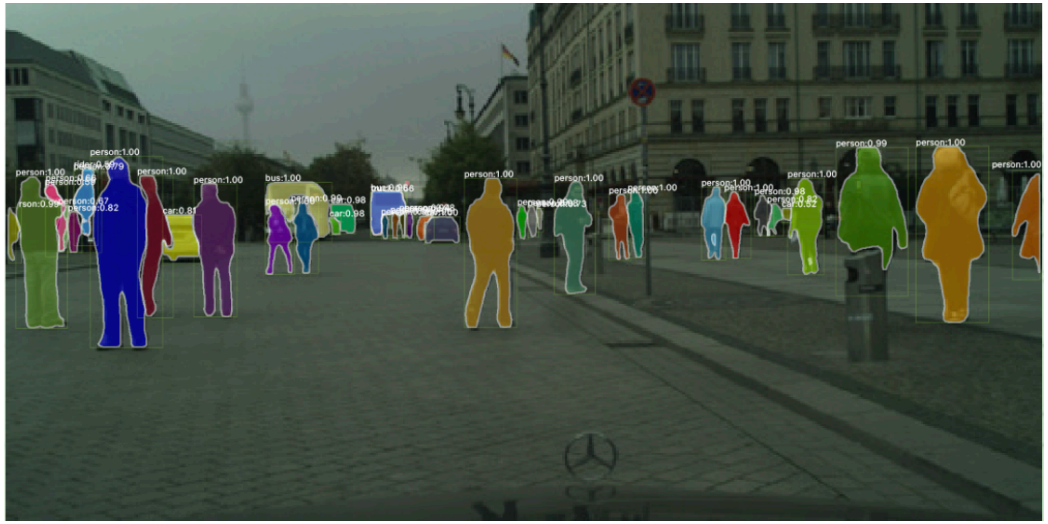


Object detection.
Credit: Toby Breckon, Durham University



Object detection with R-CNN
. Credit: Girshick et al. CVPR 2014

Deep Learning



Instance Segmentation

Credit J. C. Redmon, The Ancient Sectrets of Computer Vision 2017

Deep Learning

a cat is sitting on a toilet seat
logprob: -7.79

a display case filled with lots of different types of donuts
logprob: -7.78

a group of people sitting at a table with wine glasses
logprob: -6.71

Image captioning

Credit Karpathy and Li https://cs.stanford.edu/people/karpathy/sfmltalk.pdf

Computer Vision - Terminology

- The image is two-dimensional
- We lose information in the projection process,
  i.e., in passing from a 3D world to a 2D image
- The images are digital images
  - they are a discrete representation (i.e. they have distinct values at regularly sampled points)
  - they are a quantised representation (i.e. each value is an integer value)

Computer Vision - Terminology

Image formation system

- part illumination
- sensing element
- associated optics

is critical to the successful deployment of industrial systems

Computer Vision - Terminology

The task of the image acquisition and processing sub-system is

- to convert this signal into a digital image
- to manipulate the resultant image to facilitate the subsequent extraction of information

Computer Vision - Terminology

The image analysis phase is concerned with the extraction of explicit information regarding the contents of the image (*e.g.* object category, identity, position, size, orientation)

There is a fundamental difference between image processing and image analysis

- Image Processing facilitates transformations of images to (hopefully, more useful) images
- Image Analysis facilitates the transformation from an image to explicit (symbolic) information

Computer Vision - Terminology

Deep learning does end-to-end image processing and analysis

by learning the features, the classification criteria, the loss function ...

OpenCV
Software Development Tools for Exercises and Assignments

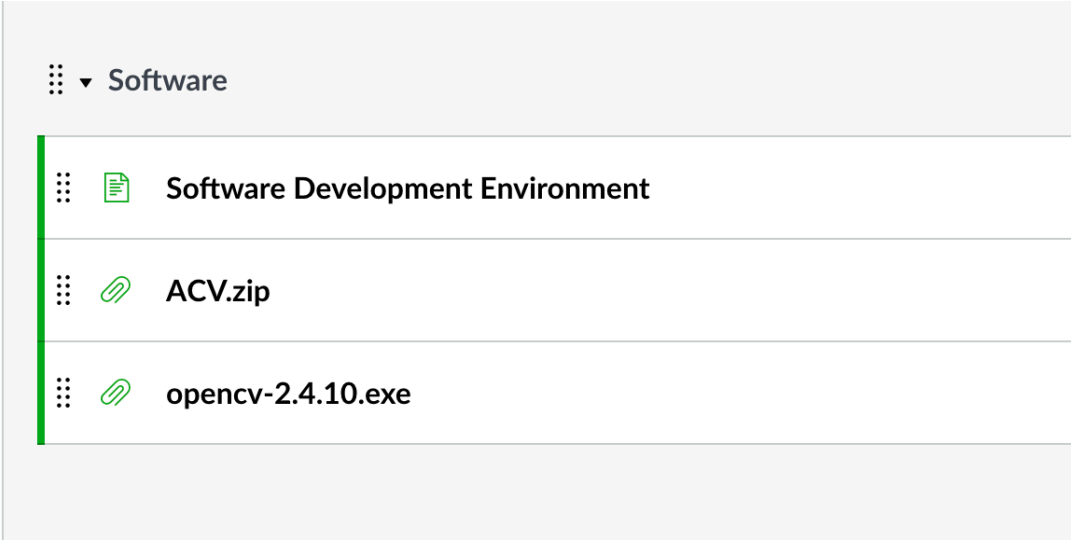
Software Development Tools for Exercises and Assignments

Installation of software development environment

- Windows 10 OS
- Microsoft Visual C++ Express compiler, version 10.0 (also known as Visual C++ 2010 or MSVC++ 2010)
- Cmake
- OpenCV (Version 2.4.10)
- Other libraries as the course progresses
- ACV Repository

Software Development Tools for Exercises and Assignments

Install software development tools



Warning: the instructions are very detailed

- Every line matters
- Do not skip over anything
- If you follow the instructions faithfully, everything will work
- If you don't … expect trouble and frustration

Software Development Tools for Exercises and Assignments

Carnegie Mellon University

Account
Dashboard
Courses
Calendar
Inbox
Help

18799-RW > Pages > Software Development Environment

Fall 2018

Home
Syllabus
Modules
Grades

View All Pages

Software Development Environment

This page provides a step-by-step guide to downloading, installing, and using the software required to run demonstration software and complete the assignments for the course. This environment is required to ensure that software developed by students can be compiled directly by the course instructor after it has been submitted.

Follow these instructions exactly as they are stated. Do not be tempted to skip through them, assuming that you will be able to figure it out yourself. You might, but it is more likely you will miss something small but important and it won't work as required. Take your time, follow these instructions carefully, and everything will work.

There are five steps in setting up the software development environment:

1. Operating System.
2. C/C++ Compiler.
3. Cmake.
4. OpenCV.
4. ACV repository.

Operating System

I assume you are running Windows 10.

If you are using Windows 7, follow the instructions here for the Operating System and C/C++ Compiler; follow the instructions below for Cmake and the ACV repository.

I strongly recommend you turn on the option in File Explorer to show file extensions. To do this, open File Explorer, click the *View* tab and click on the file name extensions checkbox so that it shows a tick. If it is not ticked, you will have problems identifying certain files later on (e.g. invoking the example.sln solution file to launch the C++ compiler).

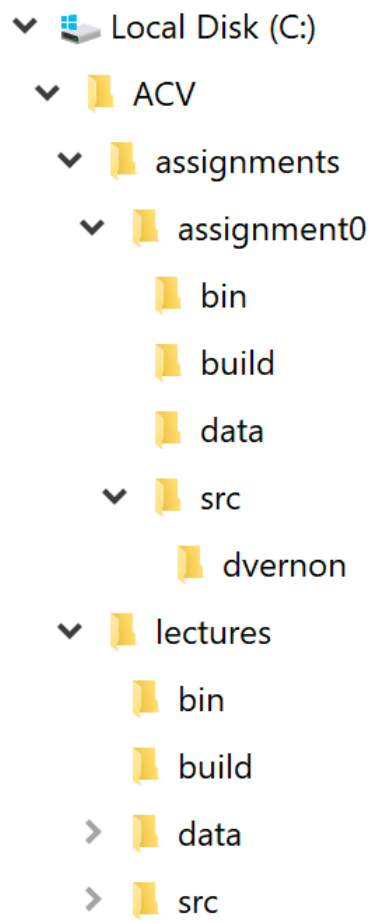
C/C++ Compiler

Download the Microsoft Visual C++ Express compiler, version 10.0 (also known as Visual C++ 2010 or MSVC++ 2010). It is important to use this version; earlier versions are no longer supported by Microsoft and later versions cause problems with some libraries.

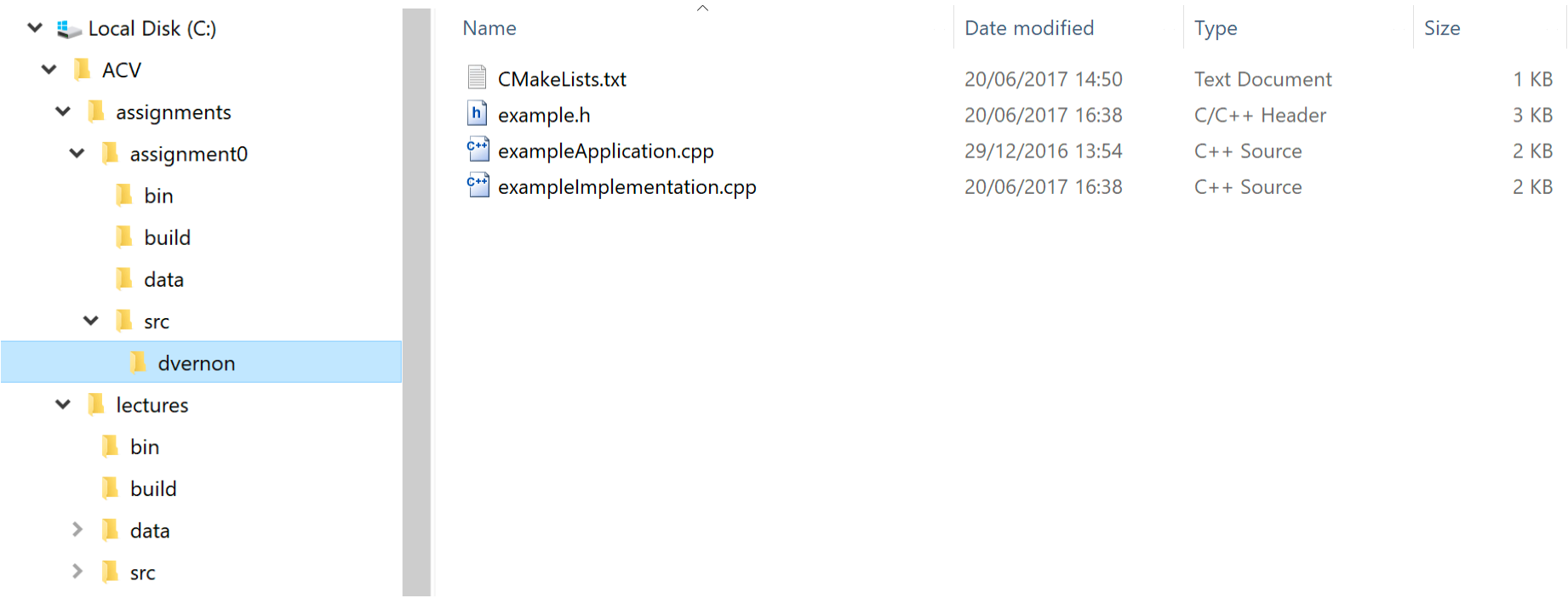
Software Development Tools for Exercises and Assignments

- Installation of software development environment
  - "`C:\ACV`" Applied Computer Vision Repository
  - Fixed file organization
- Let's walk through the process to compile and run the program in

  `C:\ACV\assignments\assignment0\dvernon`



Software Development Tools for Exercises and Assignments



Software Development Tools for Exercises and Assignments

- Preferred practice for software that supports encapsulation and data hiding (e.g. ADT & OO classes)

- 3 files: Interface, Implementation, and Application Files
  - Interface
    - between implementation and application
    - Header File that declares the class type
    - Functions, classes, are declared, not defined (except inline functions)

  - Implementation
    - `#includes` the interface file
    - contains the function definitions

  - Application
    - `#includes` the interface file
    - contains other (application) functions, including the `main` function

Software Development Tools for Exercises and Assignments

When writing an application, we are ADT/class users

- Should not know about the implementation of the ADT/class
- Thus, the interface must furnish all the necessary information to use the ADT/class
  - It also needs to be very well documented (internally)
- Also, the implementation should be quite general (cf. reusability)

Exercises

- Install software development tools and the ACV repository …
- Follow the instructions in the Software Development Environment document
- Build `C:\ACV\assignments\assignment0`
  - Run Cmake against

    `C:\ACV\assignments\assignment0 C:\ACV\assignments\assignment0\build`
  - Launch `C:\ACV\assignments\assignment0\build\assignment0.sln`
  - Build the solution
  - Run `C:\ACV\assignments\assignment0\bin\dvernon.exe`

Exercises

- Create, compile and run a new program in

  `C:\ACV\assignments\assignment0\myandrewid`

  Replace with your Andrew Id

- Follow the instructions in the Software Development Environment document

- Re-build `C:\ACV\assignments\assignment0`

Exercises

- For your first assignment, you will simply copy the `assignment0` directory to `assignment1` and follow a similar compilation procedure, writing new assignment-specific code.

- Follow the instructions in the Software Development Environment document

- Build `C:\ACV\assignments\assignment1`

Exercises

- Before building, there is just one thing you need to do: edit the

  `C:ACV\assignments\assignment1\CMakeLists.txt`

  and change the project name from assignment0 to assignment1, viz:

```
##############################################
PROJECT(assignment0)
##############################################
```

becomes

```
##############################################
PROJECT(assignment1)
##############################################
```

This is a different file to the one here:
`C:ACV\assignments\assignment1\src\CMakeLists.txt`

Name	Date modified	Type	Size
CMakeLists.txt	20/06/2017 14:50	Text Document	1 KB
example.h	20/06/2017 16:38	C/C++ Header	3 KB
exampleApplication.cpp	29/12/2016 13:54	C++ Source	2 KB
exampleImplementation.cpp	20/06/2017 16:38	C++ Source	2 KB

Exercises

When submitting an assignment, all you have to do is submit a zipped version of your `myid` directory with (e.g. `dvernon`) containing:

- Three source code files
- CmakeList.txt file
- Test data files (copied from your data directory)
- External documentation (if required)

Do not include sub-directories (e.g. src, data, build, bin, ...)

Exercises

- Build `C:\ACV\lectures`
- Follow the instructions in the Software Development Environment document
  - Run Cmake against

    `C:\ACV\lectures  & C:\ACV\lectures\build`
  - Open `C:\ACV\lectures\build\lectures\lectures.sln`
  - Build the solution
  - Run `C:\ACV\lectures\bin\<demoApplication>.exe`

Exercises

backgroundModelGMM
backgroundModelMedian
backgroundModelRunningAverage
backgroundModelSelectiveUpdate
backgroundModelStatic
binaryThresholding
binaryThresholdingAdaptive
binaryThresholdingOtsu
cameraCalibration
cameraInvPerspectiveMonocular
cameraInvPerspectiveBinocular
cameraModel
cameraModelData
cannyEdgeDetection
chamferMatching
colourHistogramBackprojection
colourHistogramMatching
colourSegmentation
colourToGreyscale
colourToHIS
connectedComponents
contourExtraction
faceDetection
featureExtraction
fourierTransform
gaussianFiltering
geometricTransformation
grabCut
harrisCornerDetection
histogram
histogramEqualization
houghTransformCircles
houghTransformLines
imageAcquisition
imageSamplingAndQuantization
kMeansClustering
laplacianOfGaussian
localAveraging
logPolarTransform
medianFiltering
morphologicalErosion
morphologicalDilation
morphologicalOpening
morphologicalClosing
morphologicalColourClosing
morphologicalColourOpening
opticalFlowFarneback
opticalFlowLucasKanade
personDetection
siftFeatureDetection
siftFeatureMatching
snakes
sobelEdgeDetection
trackingKalmanFilter
trackingMeanShift
templateMatching

Exercises - Summary

1. Install software development tools (should have been done already)

2. Download the current copy of the ACV repository

3. Unzip and merge with your existing copy (if you already have one)

4. Run Cmake against

   ```
   C:\ACV\assignments\assignment0 &
   C:\ACV\assignments\assignment0\build
   ```

5. Open
   `C:\ACV\assignments\assignment0\build\assignment0.sln`

Exercises - Summary

6. Build (i.e. compile and install) the project

7. Run `C:\ACV\assignments\assignment0\bin\dvernon.exe`

8. Create, compile and run a new program in

   `C:\ACV\assignments\assignment0\myid` Replace with your Id

Exercises - Summary

9. Run Cmake against

   `C:\ACV\lectures & C:\ACV\lectures\build`

10. Open

    `C:\ACV\lectures\build\lectures.sln`

11. Build (i.e. compile and install) the project

12. Run some of the 56 examples in `C:\ACV\lectures\bin`

Reading

R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 2010.

Section 1.1 What is computer vision?

Section 1.2 Brief history of computer vision

Kragic, D. and Vincze, M. (2010). "Vision for Robotics", Foundation and Trends in Robotics, Vol 1, No 1, pp 1–78.

Also take a look at David Lowe's website of industrial vision applications:

http://www.cs.ubc.ca/~lowe/vision.html