Automatic Image Analysis

Berlin University of Technology (TUB),
Computer Vision and Remote Sensing Group
Berlin, Germany

Ronny Hänsch

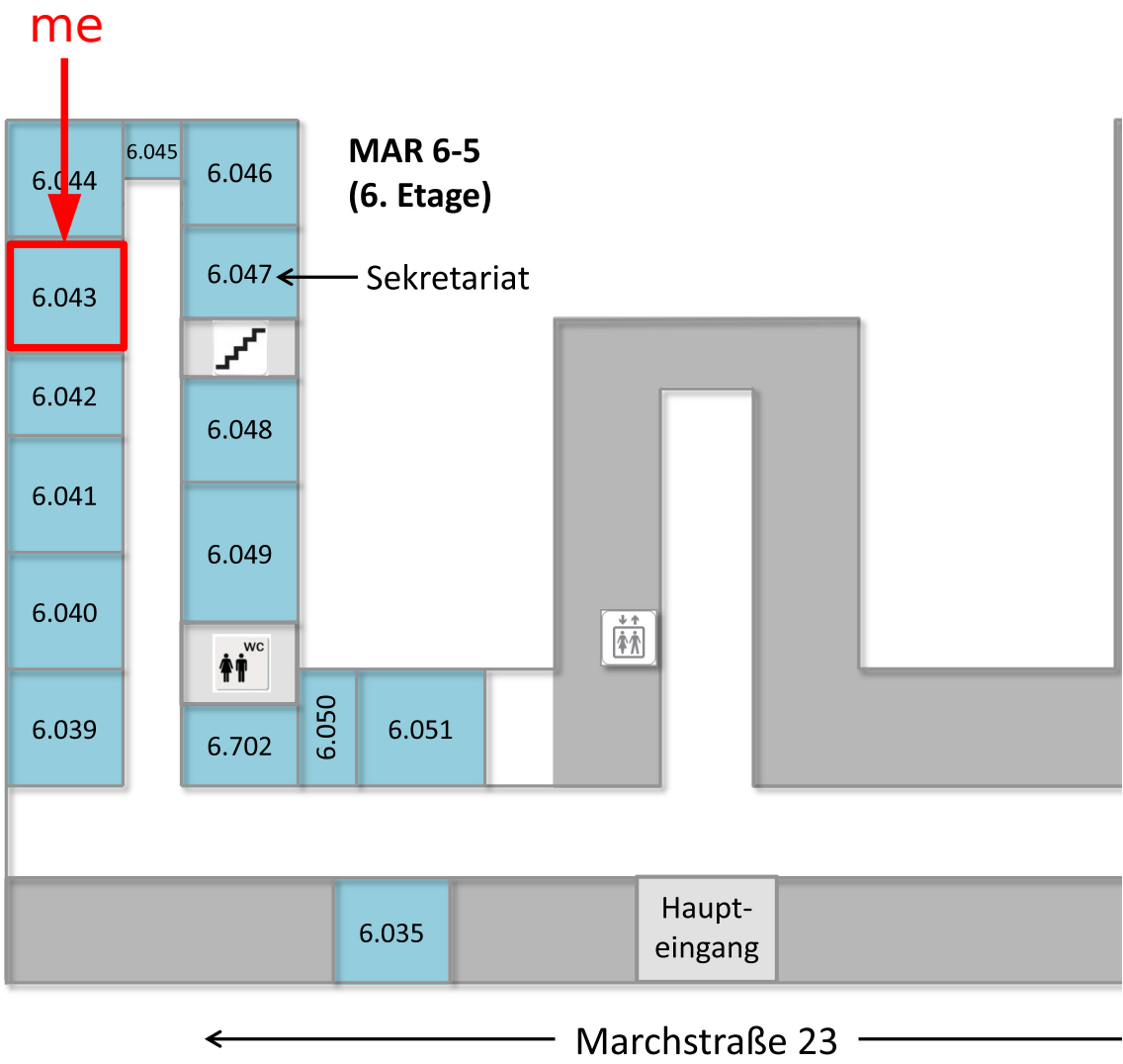
E-Mail: r.haensch@tu-berlin.de

Office

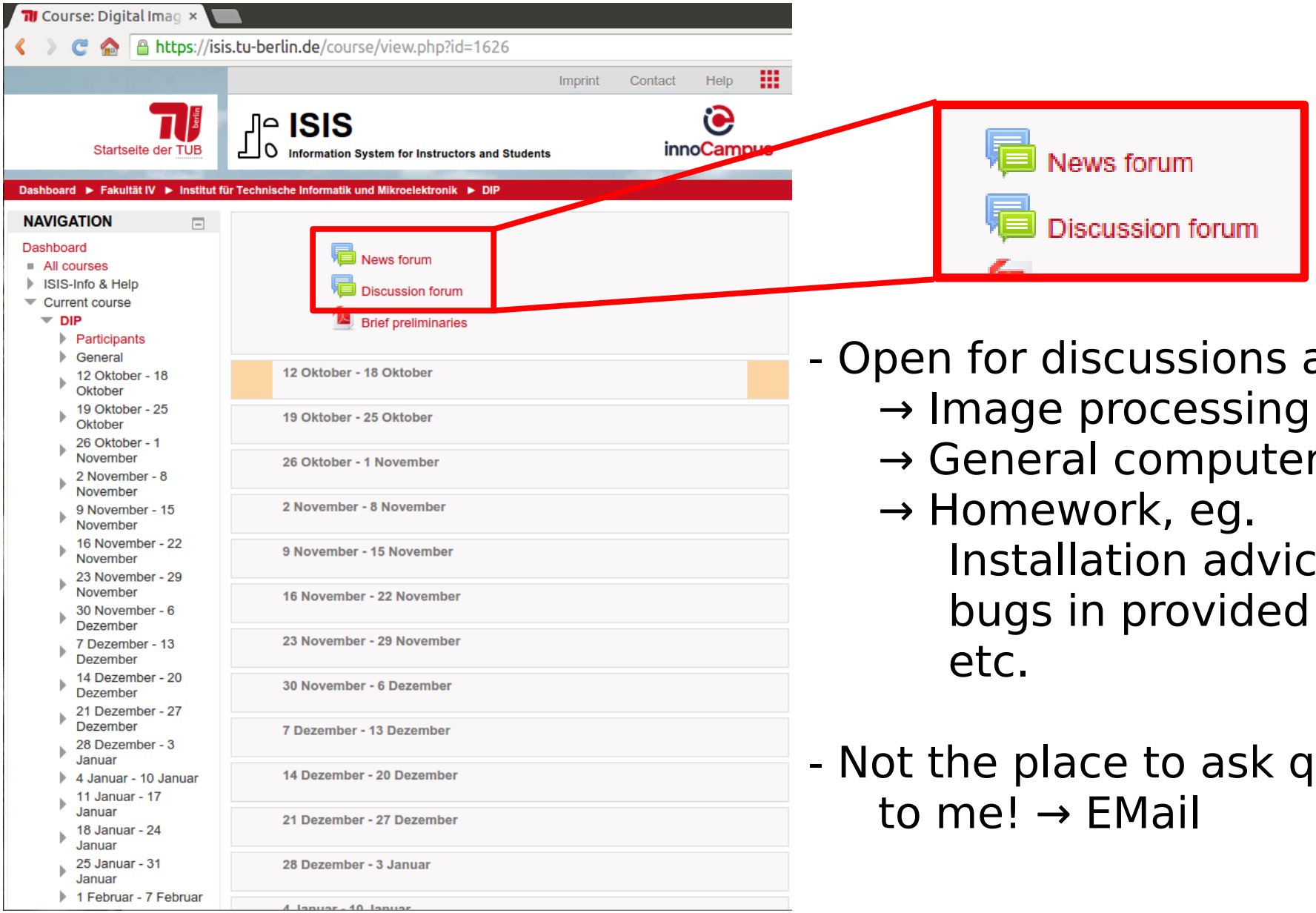
- MAR6.043,
  March Building, 6th Floor

Consultation Time

- Wednesday,
  16:00-18:00 o'clock
- (Or by arrangement)



Information



- Open for discussions about
  - → Image processing topics
  - → General computer vision
  - → Homework, eg. Installation advices, bugs in provided code, etc.

- Not the place to ask questions to me! → EMail

Exams

- **Mid-term:**
  - Near the middle of the term: **09.06.2017**
  - Room: tba
  - Duration: ca. 45 min
  - In place of an exercise
  - No grade, but pass is necessary to take part at the final exam

- **Final:**
  - At the end of the term: **18.07.2017**
  - Time: 10 – 12:00
  - Room: H3010
  - Duration: 90 min

- Questions in English, answers in English or German

Homework

What to do?

I: Answering theoretic questions

II: Implementation of methods for processing digital images

How?

In groups of 3-4 students

Programming Language: C++ [and OpenCV **2.4** or **3.0**]

Completion of provided software packages

- Class descriptions (header files): given
- Includes: given
- Basic functionality: given
- Specific functions: Your task!

Goal?

Practising, Learning. No grades!

But pass is necessary to take part at the final exam

Homework

- Next meeting in two weeks

- **BEFORE Friday, 10am**:
  - Hand in your solution via ISIS

  - Solution includes (red denotes mandatory material):
    - → **All program files** (.h & .cpp) of the provided material
    - → **Input**, intermediate, and **output images**
    - → Pdf-file with
      - → Short discussion / presentation of your solution
      - → **Answers to theoretical questions**

- Algorithms more important than code (but try!)

Homework

“Grades”

+++	more than just a correct solution (efficient, clever, cool, …)
++	correct solution
+	some minor errors, but still acceptable
-	not acceptable → re-work *(within 1 week, parallel to new assignment)*
- -	failed: you are not allowed to write the exam!

1. Exercise – Part II : Practical

C++ and OpenCV

Given:

- Main function (main.cpp)
- Function declaration (aia1.h)
- Basic functionality (aia1.cpp)

Todo:

- **[Install C++-compiler]**
- **[Install OpenCV]**
- Aia1.cpp
  - Mat Aia1::doSomethingThatMyTutorIsGonnaLike(Mat&)
    → Do something (reasonable)

Deadline:

- Next meeting at **12.05.2014**, 10am

1. Exercise – Part I : Theory

1. Provide

   → **name**

   → **student ID**

   → and **course of study** (e.g. Computer Science, Master)

   of all group members!

1. Exercise – Part II : Given

FILE: main.cpp

```
int main(int argc, char** argv)
```

- Main function
  - Usage:
    - aia1 path_to_image
      - Calls Aia1::run(...)
      - Calls Aia1::test(...) for basic testing!

FILE: Aia1.cpp/h

```
void Aia1::run(string fname)
```

- Loads image
- Calls related functions

1. Exercise – Part II : To Do

`Mat doSomethingThatMyTutorIsGonnaLike(Mat& img)`

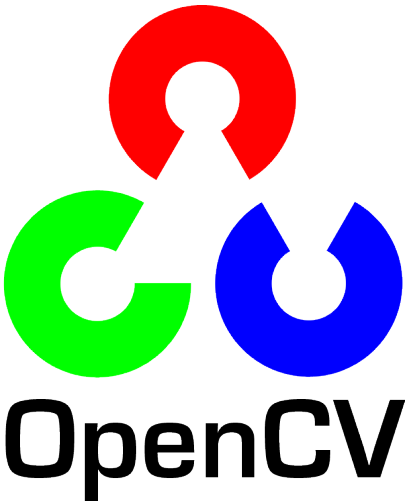
`img` : input image

`return` : output image

→ does something cool... (hopefully)

Brief introduction to OpenCV

- One of most common image processing libs
- Open source (BSD)
- C/C++, Python, Java interfaces
- Supported: Windows, Linux, Mac OS, iOS, Android
- Strong focus on real-time applications
- Multi-core processing, hardware acceleration
- 47 thousand people of user community
- 9 million downloads



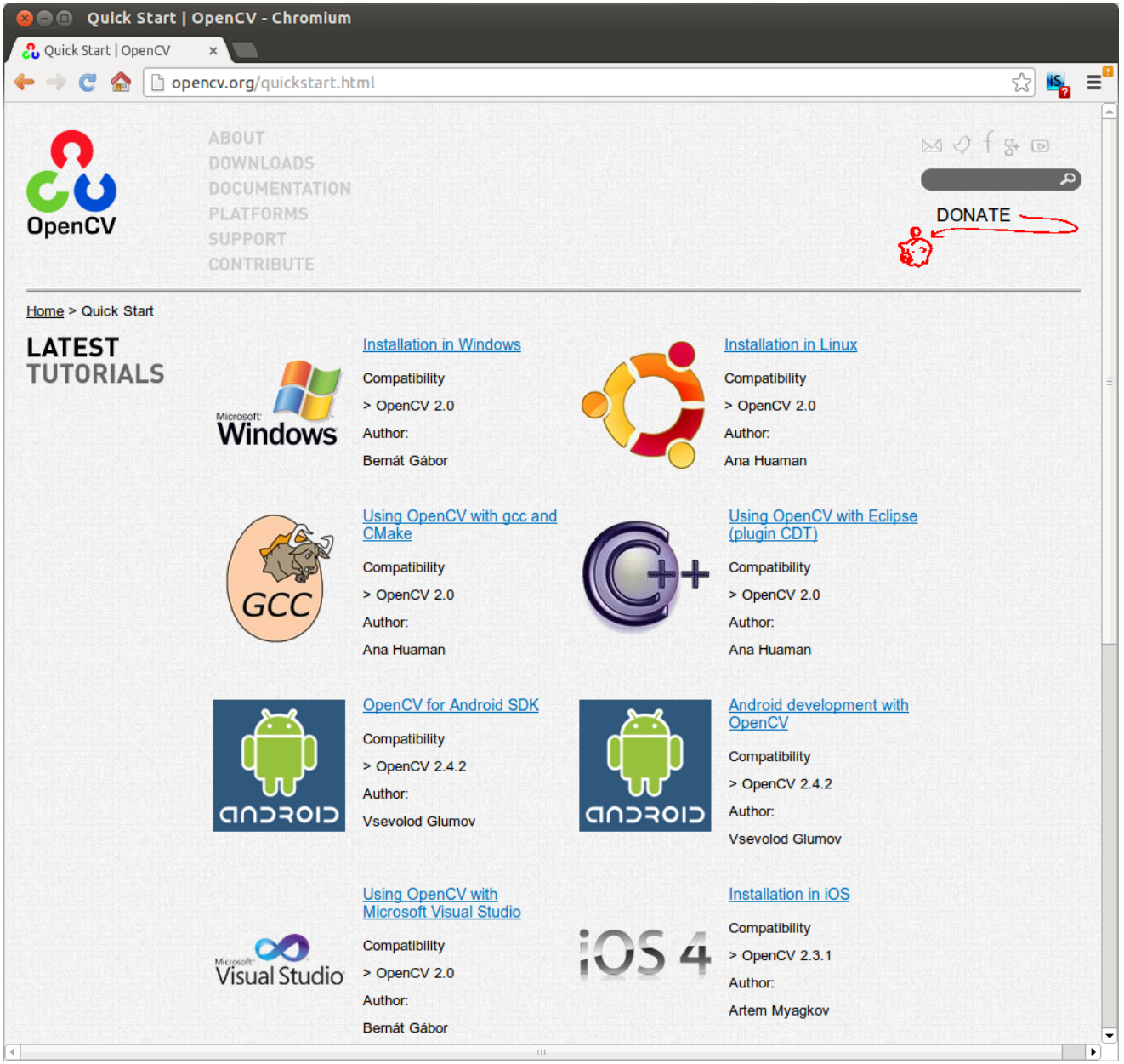
Brief introduction to OpenCV

OpenCV API Reference

- Introduction
  - API Concepts
- core. The Core Functionality
  - Basic Structures
  - Basic C Structures and Operations
  - Dynamic Structures
  - Operations on Arrays
  - Drawing Functions
  - XML/YAML Persistence
  - XML/YAML Persistence (C API)
  - Clustering
  - Utility and System Functions and Macros
  - OpenGL interoperability
- imgproc. Image Processing
  - Image Filtering
  - Geometric Image Transformations
  - Miscellaneous Image Transformations
  - Histograms
  - Structural Analysis and Shape Descriptors
  - Motion Analysis and Object Tracking
  - Feature Detection
  - Object Detection
- highgui. High-level GUI and Media I/O
  - User Interface
  - Reading and Writing Images and Video
  - Qt New Functions
- video. Video Analysis
  - Motion Analysis and Object Tracking
- calib3d. Camera Calibration and 3D Reconstruction
  - Camera Calibration and 3D Reconstruction
- features2d. 2D Features Framework
  - Feature Detection and Description
  - Common Interfaces of Feature Detectors
  - Common Interfaces of Descriptor Extractors
  - Common Interfaces of Descriptor Matchers
  - Common Interfaces of Generic Descriptor Matchers
  - Drawing Function of Keypoints and Matches
  - Object Categorization
- objdetect. Object Detection
  - Cascade Classification
  - Latent SVM
- ml. Machine Learning
  - Statistical Models
  - Normal Bayes Classifier
  - K-Nearest Neighbors
  - Support Vector Machines
  - Decision Trees
  - Boosting
  - Gradient Boosted Trees
  - Random Trees
  - Extremely randomized trees
  - Expectation Maximization
  - Neural Networks
  - MLData
- flann. Clustering and Search in Multi-Dimensional Spaces
  - Fast Approximate Nearest Neighbor Search
  - Clustering
- gpu. GPU-accelerated Computer Vision
  - GPU Module Introduction
  - Initalization and Information
  - Data Structures
  - Operations on Matrices
  - Per-element Operations
  - Image Processing
  - Matrix Reductions
  - Object Detection
  - Feature Detection and Description
  - Image Filtering
  - Camera Calibration and 3D Reconstruction
  - Video Analysis
- photo. Computational Photography
  - Inpainting
  - Denoising
- stitching. Images stitching
  - Stitching Pipeline
  - References
  - High Level Functionality
  - Camera
  - Features Finding and Images Matching
  - Rotation Estimation
  - Autocalibration
  - Images Warping
  - Seam Estimation
  - Exposure Compensation
  - Image Blenders
- nonfree. Non-free functionality
  - Feature Detection and Description
- contrib. Contributed/Experimental Stuff
  - Stereo Correspondence
  - FaceRecognizer Documentation
  - Retina Documentation
  - OpenFABMAP
- legacy. Deprecated stuff
  - Motion Analysis
  - Expectation Maximization
  - Histograms
  - Planar Subdivisions (C API)
  - Feature Detection and Description
  - Common Interfaces of Descriptor Extractors
  - Common Interfaces of Generic Descriptor Matchers
- ocl. OpenCL-accelerated Computer Vision
  - OpenCL Module Introduction
  - Data Structures and Utility Functions
  - Data Structures
  - Operations on Matrics
  - Matrix Reductions
  - Image Filtering
  - Image Processing
  - ml.Machine Learning
  - Object Detection
  - Feature Detection And Description
  - Video Analysis
  - Camera Calibration and 3D Reconstruction
- superres. Super Resolution
  - Super Resolution
- viz. 3D Visualizer
  - Viz
  - Widget

Brief introduction to OpenCV

How to install....



Brief introduction to OpenCV

```
#include <iostream>
#include <opencv2/opencv.hpp>
using namespace std;

int main(int argc, char** argv){

    cv::Mat img = imread( argv[1], 0 );            // load image as gray-scale

    // show image
    cv::namedWindow( "example");
    cv::imshow( "example", img);

    Mat newImg( img.rows, img.cols, CV_8UC3, cv::Scalar(0) );
    // do something fancy
    fancyFunction(img, newImg);

    cv::imwrite("coolResult.png", newImg);

}
```

Brief introduction to OpenCV

Matrix generation, some examples:

```
Mat M1 = Mat(2, 3, CV_32FC1);        // creates 2x3 matrix of floats (one channel)
Mat M2 = Mat(3, 2, CV_64FC2);        // creates 3x2 matrix of doubles (two channels)
Mat M3 = Mat(3, 3, CV_8UC3);         // creates 3x3 matrix of uint (three channels)

Mat M4 = Mat::zeros(3, 3, CV_32FC1);     // creates 3x3 matrix of floats, all set to 0
Mat M5 = Mat::ones(3, 3, CV_32FC1);      // creates 3x3 matrix of floats, all set to 1

Mat M6 = (Mat_<float>(3,3) << 1, 2, 3, 4, 5, 6, 7, 8, 9);
```

Accessing matrix data (the easy way)

```
M1.at<float>(row, column)      =    22.0 / 7.0;
M2.at<Vec2d>(row, column)   =    Vec2d(0,1);
int s = M3.at<Vec3b>(row, column)[0];
```

Brief introduction to OpenCV

Compilation (Linux etc.)

user@comp:~/path$ g++ -o aia aia1.cpp main.cpp -lopencv_core -lopencv_imgproc -lopencv_highgui
user@comp:~/path$ g++ -o aia1.cpp main.cpp `pkg-config opencv --cflags –libs`

Or using *cmake* and *make*

Further information:

- http://opencv.org/
  - Install guides
  - Documentation
  - FAQ
- OpenCV 2 Computer Vision – Application Programming Cookbook
- “AIA 0 – Preliminary”-slides on ISIS2