

Applied Computer Vision for Robotics

17.4.2013

Philipp Heise

Brian Jensen

Sebastian Klose

GitHub

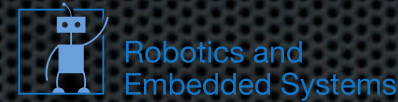


Robotics and
Embedded Systems



- ✦ Find together in groups of 3 people
- ✦ Everyone needs to create a GitHub account
<http://github.com>
- ✦ Write a mail to kloses@in.tum.de containing:
 - ✦ team name (be creative here)
 - ✦ real name, GitHub account name and e-mail address for each team member

Syllabus



- ✦ New sheet every 2 weeks
- ✦ **Mandatory** meeting every week
- ✦ Short presentation of the sheet results by the teams
- ✦ Everyone will be registered to a RVC mailing list - ask questions here
- ✦ You can also ask us questions or write mails

ROS

- ✦ We are going to use ROS - Robot Operating System (www.ros.org)
- ✦ We'll introduce it on the fly and give hints which packages you might need
- ✦ Check out the tutorials and wiki on ros.org for more details

Cameras

- ✦ If you don't have an own camera working in ROS, you can borrow one from us
- ✦ PSEye Camera - 640x480 @60Hz
- ✦ **Deposit: 10 €**

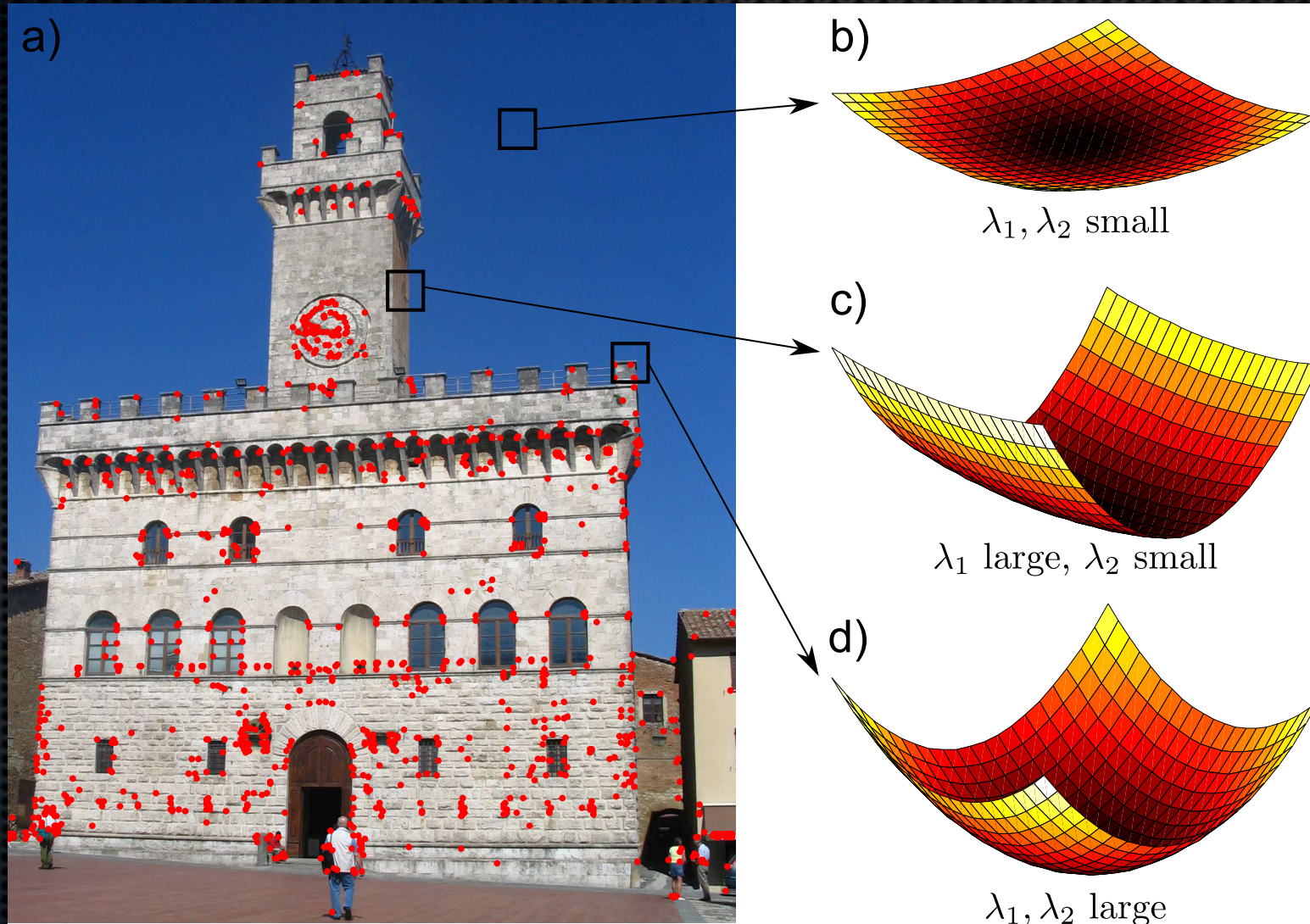
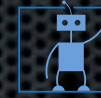
Harris

- ✦ Detect image corners
- ✦ Calculate structure tensor

$$M = \sum_{u,v} w(u,v) \begin{pmatrix} \nabla_x I & \nabla_x I \nabla_y I \\ \nabla_x I \nabla_y I & \nabla_y I \end{pmatrix}$$

- ✦ Be careful about the image data types
- ✦ Use the eigenvalues λ_1, λ_2 to calculate the “corneress”

Harris



Harris

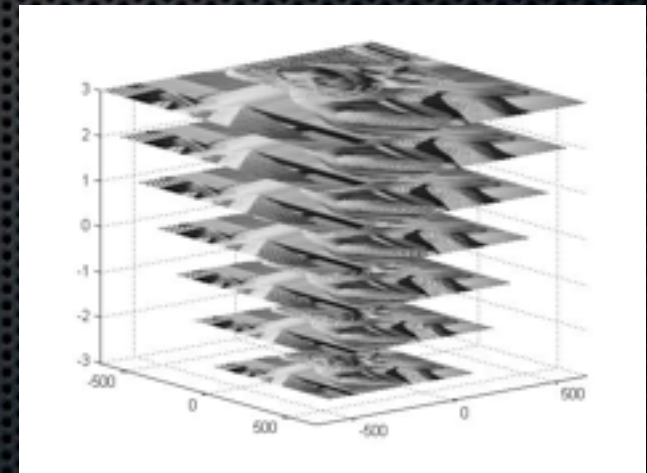
- Use the Harris-Stephens formula to avoid the exact computation of the eigenvalues

$$\begin{aligned}M_c &= \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 \\ &= \det(M) - \kappa \text{trace}^2(M)\end{aligned}$$

- Kappa is a parameter and can be tuned to get reasonable results
- Threshold to get corners and perform non-maximum suppression

Scale Space

- ✦ Perform Harris corner detection on different scales
 - ✦ Create an image pyramid with a scale factor (<1) and a fixed number of octaves
 - ✦ Detect the corners in each octave



Orientation

- ✦ Find the orientation of each corner
- ✦ Compute the angle using

$$\theta = \text{atan2} \left(\sum_{u,v} w(u,v) \begin{pmatrix} \nabla_x I \\ \nabla_y I \end{pmatrix} \right)$$

ANMS

- ✦ Very often the detected features are not well distributed over the image
- ✦ Instead of using a fixed threshold ANMS uses only non maximum suppression within a certain radius