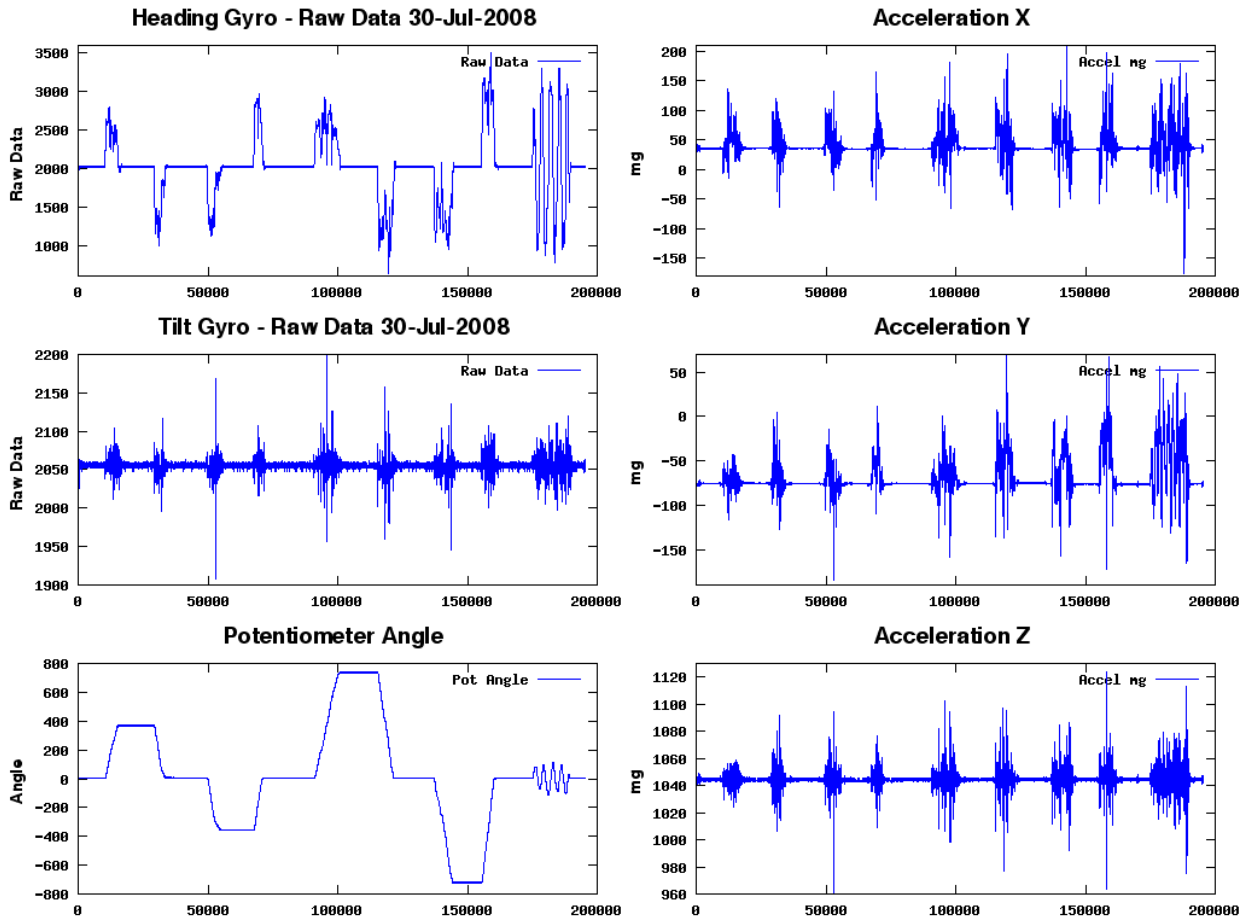
Introduction to Data Processing



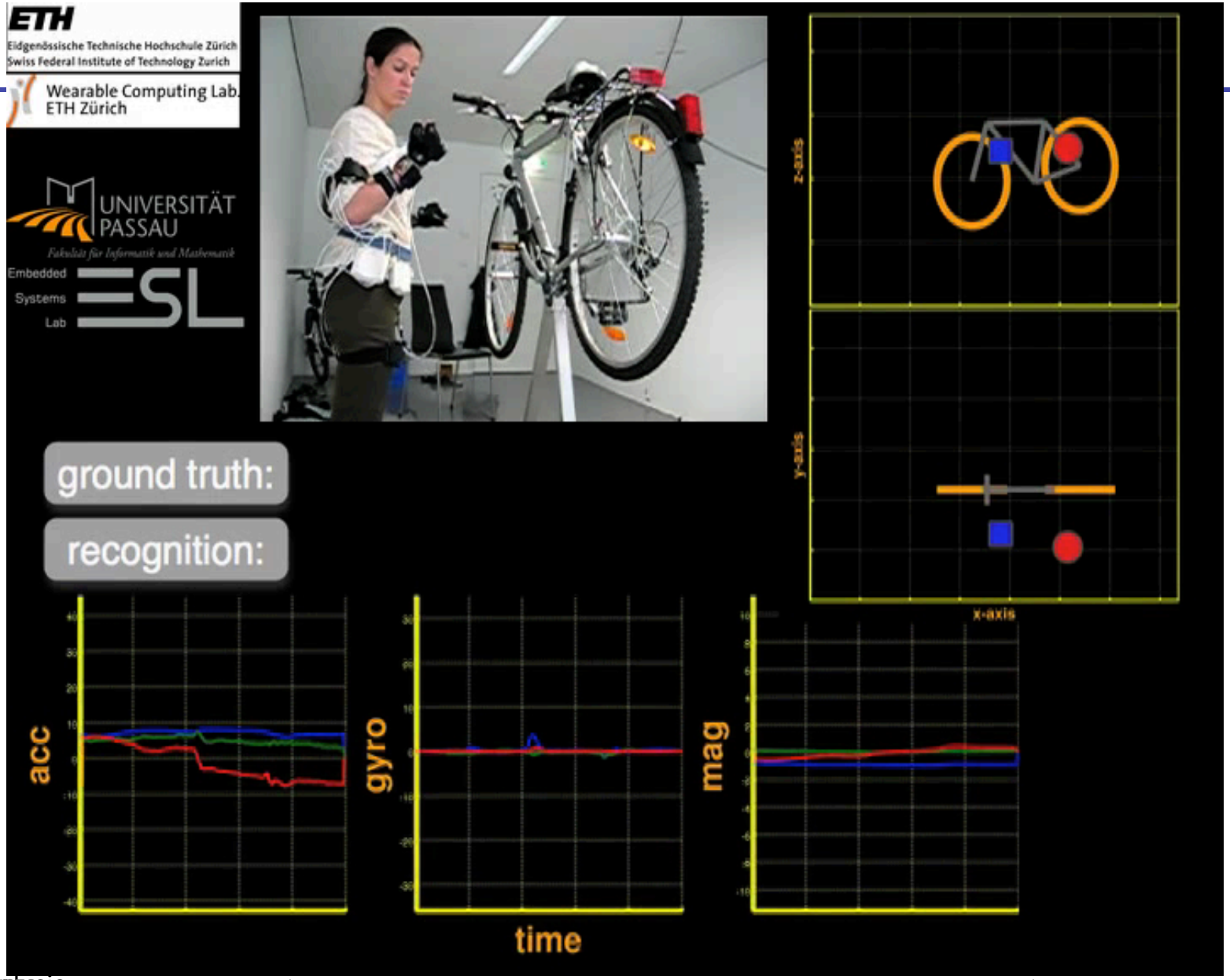
Overview

- Introduction
- General Tips
- Some of my tools
  - Set of commandline tools (zsh, git, mvim …)
  - matlab
  - ipython, R
- Case Study: Workshop Scenario

Disclaimer

- Data analysis/processing is a very broad term
  - A lot of different definitions
  - A lot of different tools
- In this tutorial I show you what works for me:
  - The methodology
  - Some useful tools

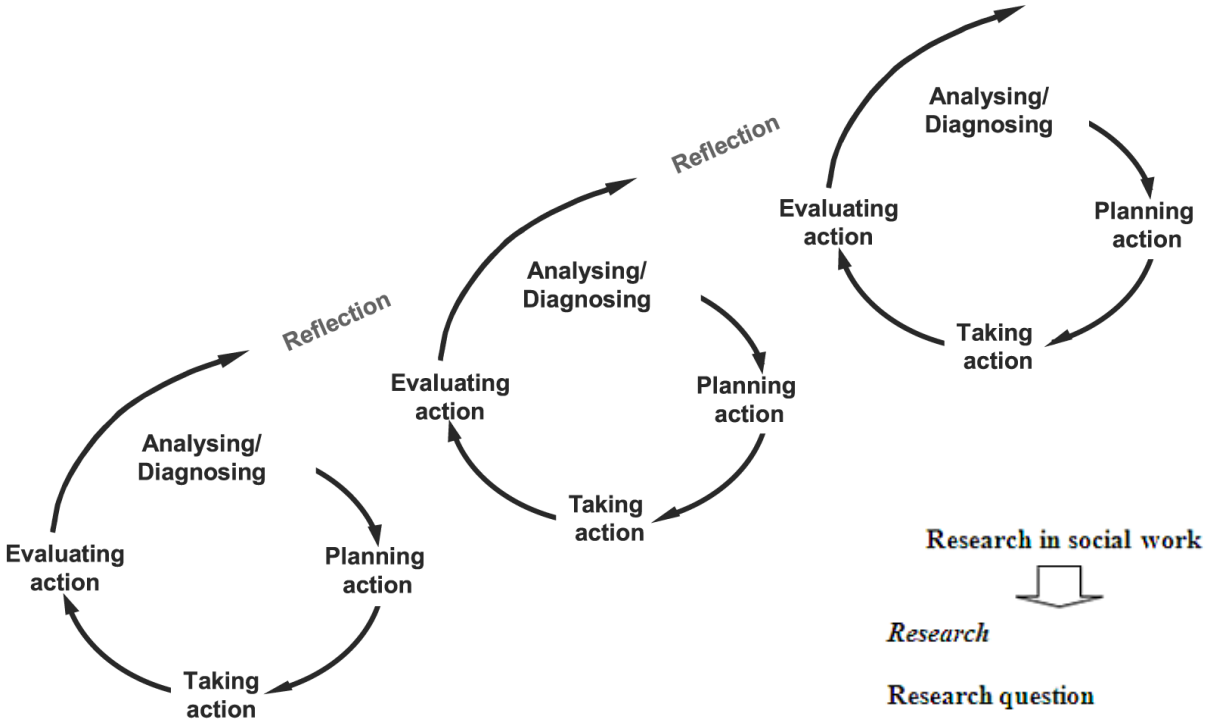
- Most important:
  - Have fun playing with data!

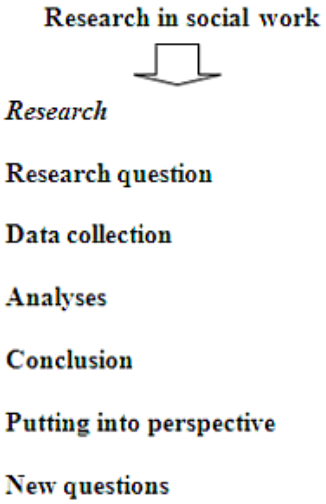

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Wearable Computing Lab.
ETH Zürich
UNIVERSITÄT
PASSAU
Fakultät für Informatik und Mathematik
Embedded
Systems
Lab
ESL
z-axis
y-axis
x-axis
ground truth:
recognition:
acc
gyro
mag
time

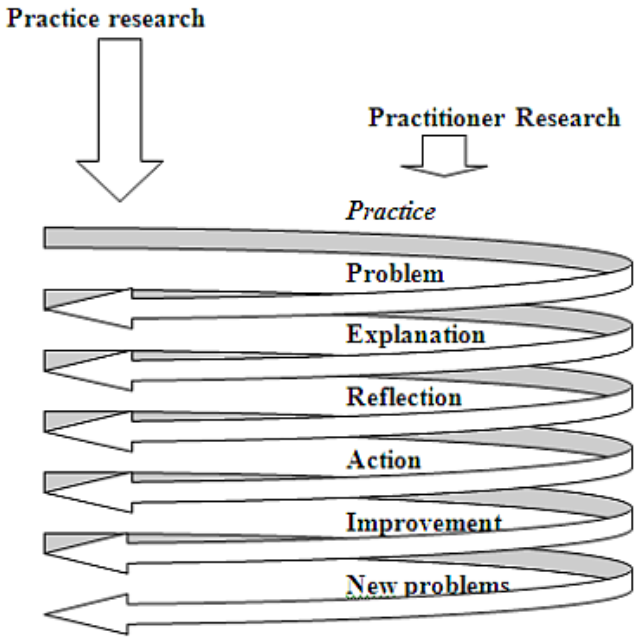
Research

- As researcher your product is not data or code … your product is **knowledge**
- Empirical science uses data and code to obtain knowledge
- good research is about reproducibility

Research “life” cycle







Getting data …

- Standard datasets
  - http://contextdb.org
- Own Experimental Design
  - Difficult !! Don't underestimate the design
  - Some good starting points (for UI design experiments, yet also valid for other designs)
  - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-831-user-interface-design-and-implementation-spring-2011/lecture-notes/MIT6_831S11_lec14.pdf
  - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-831-user-interface-design-and-implementation-spring-2011/lecture-notes/MIT6_831S11_lec15.pdf

How I handle data …

- Separate code from data
  - One data set, multiple types of analysis
- Have separate directories for data
  - Suggestion: input, working, output
  - input: never changes!
  - working: calculated features, processing steps
  - output: classification results etc.
- NEVER change the raw data directly
- NEVER do changes by hand to the data …

Sample Project directory

- My_project
  - data
    - Working
    - Input
    - Output
  - Code
    - Matlab
    - Python
    - C
  - README.md
- My_Data
  - EEG
    - Working
    - Input
    - Output
  - Face_rec
- My_EEG_project
  - papers
  - code
  - demo
  - test
  - README.md

Use Source Control (for code, docs …)

- Use a distributed version control system
  - My favorite: git
- I use it for everything … (except DATA)
  - papers
  - My website
  - Every text file

- Text files are your friends ☺
- Don’t use git for data (use .gitignore file to exclude it)

Save frequently and backup

- Save everything to disk frequently
  - Features you calculated
  - Data preprocessing steps
  - Your models, your results
- Disk-space is cheap, use it
- Use naming conventions:
  - For example one of my working directories:
    - Working/2012-12-02-features-accel-sw100.mat
- Dropbox is nice
  - if the data is not too large and sensitive (privacy!!)
  - works for code

On tests, timing …

- Test the data + code as early and often as possible
  - Work with input /output files
  - General tests used for:
    - Prototyping language (perl, python …)
    - Demo implementation in faster language (C, C++ …)
- Estimate the timing of your methods
  - Paper deadline driven

Tips

- Use folder structures (with docs, tests)
- Use Version Control
- Save everything frequently (intermediate steps)
- Easy to execute part of the analysis (modular setup)

Pick the right tool for the right purpose …

- Remember we want to produce knowledge
- There are a lot of data processing/ analysis software out there:
  - Matlab, mathematica, maple, S, Strata
  - Octave, Sage, R
  - Libraries in c, java, pyhton, ruby, javascript, perl

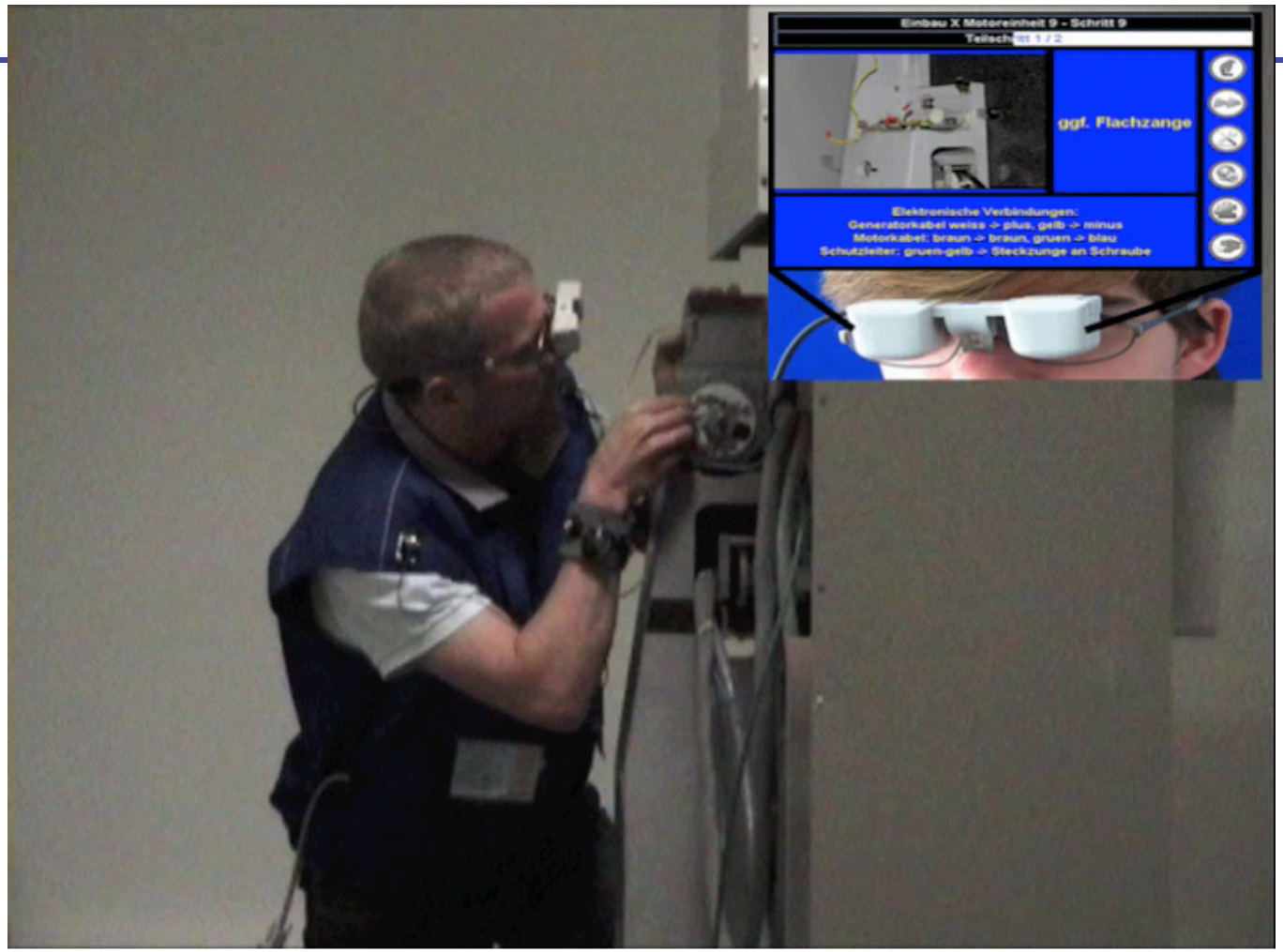
- Every software comes with advantages and disadvantages!

Tools I use …

- Commandline tools
  - screen, cat, grep, sed, head, tail, awk, find, xargs, sort, wc …
  - Zsh, vim
    - I use YADR (for mac):
      - http://skwp.github.com/dotfiles/
- For the initial data processing
  - Matlab, python (ipython, scipy …) http://www.enthought.com
- For most plots: R
- For demos, production code:
  - Depends, whatever does the job
  - A good knowledge of C is very helpful ☺

Case Study: Workshop Scenario

- Let's assume you need to build a prototype for a project
- Support of a worker during an assembly scenario
- Requirements:
  - You should recognize the following tasks:
    - hammering, screw driving, sand papering, sawing
  - The classifier should run on an embedded platform
    - Linux –arm, C implementation
- How do you start?



Kunze, K.,Wagner, F., Kartal, E., Morales Kluge, E., and Lukowicz, P. Does Context Matter ? - A Quantitative Evaluation in a Real World Maintenance Scenario. *In Proceedings of the 7th international Conference on Pervasive Computing Nara*, Japan, May 11 - 14, 2009.

Case Study: Workshop Scenario

Plotting in R

Excel is not nice to do plotting ☹

I don’t know how to program in R

- Really
- However, it does not matter ☺

Rapid prototyping for demos …

- Processing
- java
- Node.js /ruby on rails
- Javascript:
  - d3.js (for http://contextdb.org)
  - http://square.github.com/cubism/

Further References

Data analysis, machine learning, dsp

https://class.coursera.org/dataanalysis-001/

https://www.coursera.org/course/dsp

https://www.coursera.org/course/ml

Visualization

http://cs171.org

Help

http://stackoverflow.com

http://stats.stackexchange.com