Perceiving Things

Friedrich Fraundorfer
Institute of Computer Graphics and Vision
Graz University of Technology (TUG)
Austria

(Slides from a Colloquium talk at Graz University of Technology from 19.1.2018, slightly adapted for publication)

A

Perceiving Thing

?

Transcript of the introduction

Today I will be talking about perceiving things.

This topics is important as you will recognize that perceiving things are already around of us all and that the number of such things will grow rapidly beyond any limits.

In this presentation I will first introduce the term „perceiving things“. Then I will motivate the research challenges related to perceiving things. And I will show examples and my own contributions in this area.

But first, what do I mean with the term “a perceiving thing”.

I mean a technical gadget, a household appliance, a tool that can perceive the environment and the user who is interacting with it.

I brought such a perceiving thing for you. My smart phone. It has a camera and if I take a picture of an object it can tell me what I see (e.g. using the google goggles app)

Of course a more classical example of a perceiving thing would be a mobile robot like the DLR Justin robot. It uses cameras in the head as main navigation sensor. It can catch basket balls.

But more interestingly, we will see very soon that simple tools and appliances will get the ability to perceive the environment and the user that interacts with them. One example would be a stove.

It will recognize what you put on the heating plates. If it should be heated or not.

One last example from my side.

A perceiving thing can also be a shoe. Yes, I said a shoe.

This is an example of one of my own projects.

This shoe is an assistive device for visually impaired persons. It is a shoe equipped with a camera that senses the environment and guides a blind person through complicated situations.

Now these have been enough examples. I am sure that you can think of many more.

My scientific quest

How can we create a **computer vision system** that is comparable in performance to the human visual system?

My scientific quest

How can we create a **computer vision system** that is comparable in performance to the human visual system?

Is it possible to create a computer vision system that surpasses human capabilities (more precise, faster)?

Capabilities of such a computer vision system

- Localize and name objects (person, cat, flower, car)
- Name actions
- Estimate sizes of objects
- Estimate distances to objects
- Estimate motion of objects (speed, direction)
- Measure ego-motion
- Detect unusual appearance (e.g. scratches in a smooth surface, rust)
- Name relations between objects (maybe beyond a pure visual system)

Capabilities of such a computer vision system

- Localize and name objects (person, cat, flower, car)
- Name actions
- **Estimate sizes of objects**
- **Estimate distances to objects**
- **Estimate motion of objects (speed, direction)**
- **Measure ego-motion**
- Detect unusual appearance (e.g. scratches in a smooth surface, rust)
- Name relations between objects (maybe beyond a pure visual system)

Which technology is needed for a perceiving thing

digital camera + **computer vision algorithms**

Team


Michael Maurer


Fabian Schenk


Thomas Holzmann


Tobias Koch (TUM)


Xiangyu Zhu (DLR)


Christian Mostegel


Jesús Pestana-Puerta


Christian Ertler


Tahmineh Partovi (DLR)

Former Phd collaborators:
Chris Engels, Changchang Wu, Guan Banglei, Gellert Mattyus, Olivier Saurer, Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Dominik Honegger, Christian Häne, Johannes Schönberger, Florian Burkert, Markus Rumpler

Camera motion estimation

Camera motion estimation

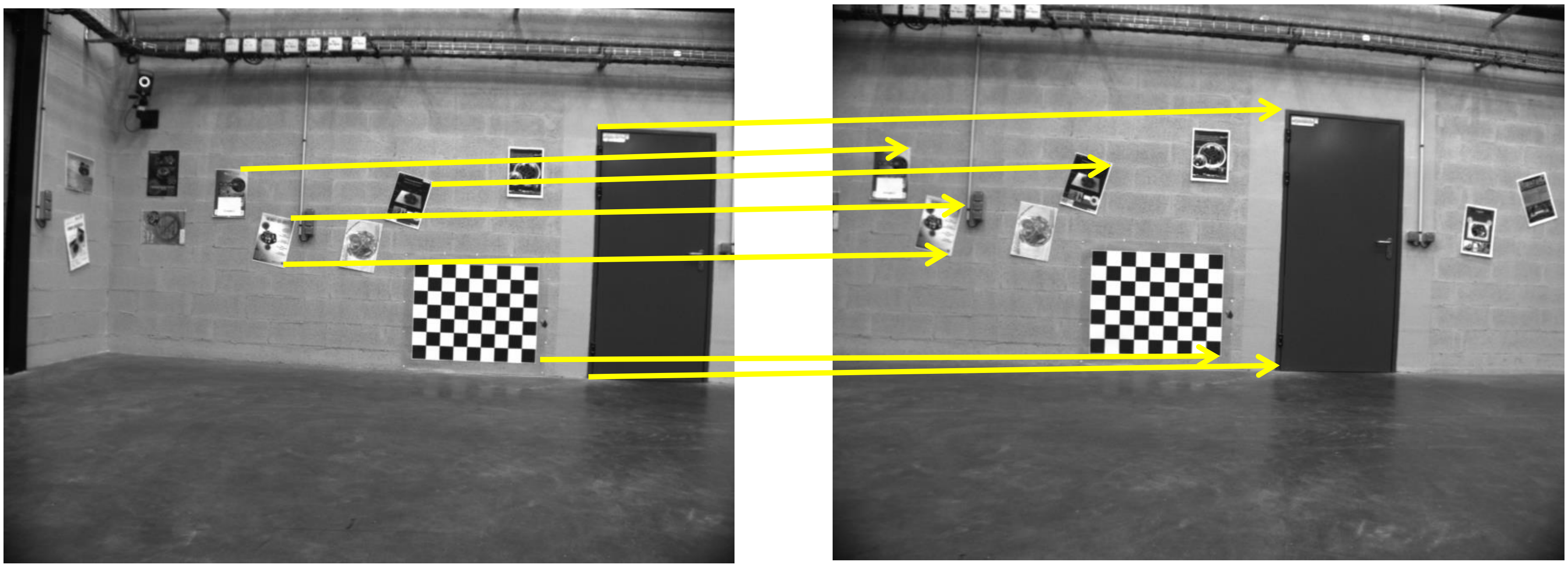
$$x = \mathbf{P}X$$

$$x'^{T}\boldsymbol{E}x = 0$$

Mathematical models for image projection and two-view geometry

Camera motion estimation

$$x'^T \boldsymbol{E} x = 0$$



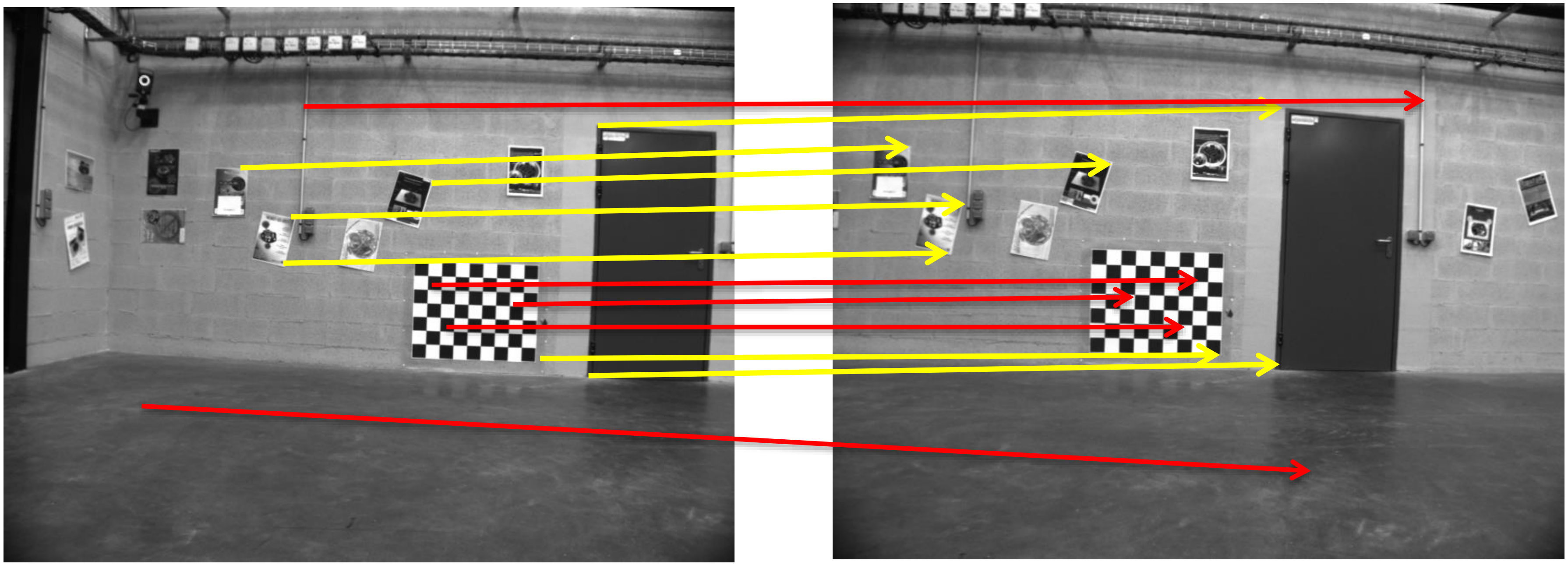
x

image matches

x'

Camera motion estimation

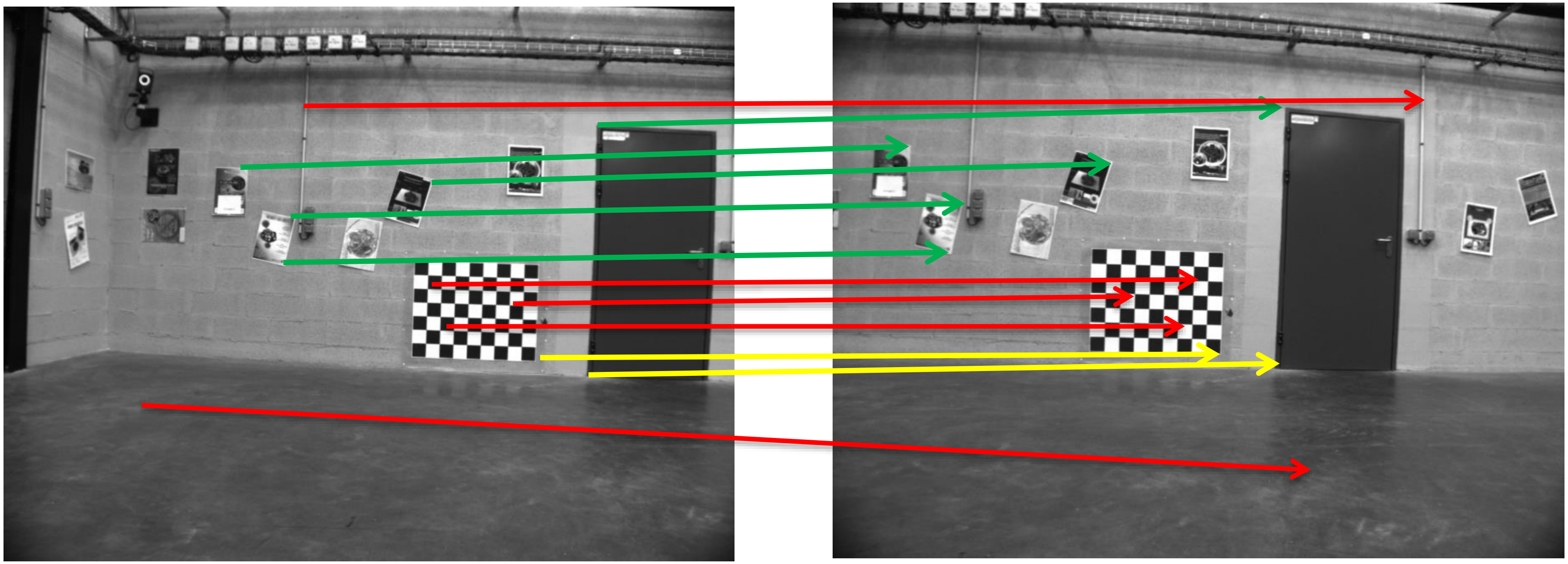
$$x'^T \boldsymbol{E} x = 0$$



x image matches with mis-matches x'

A minimum sample

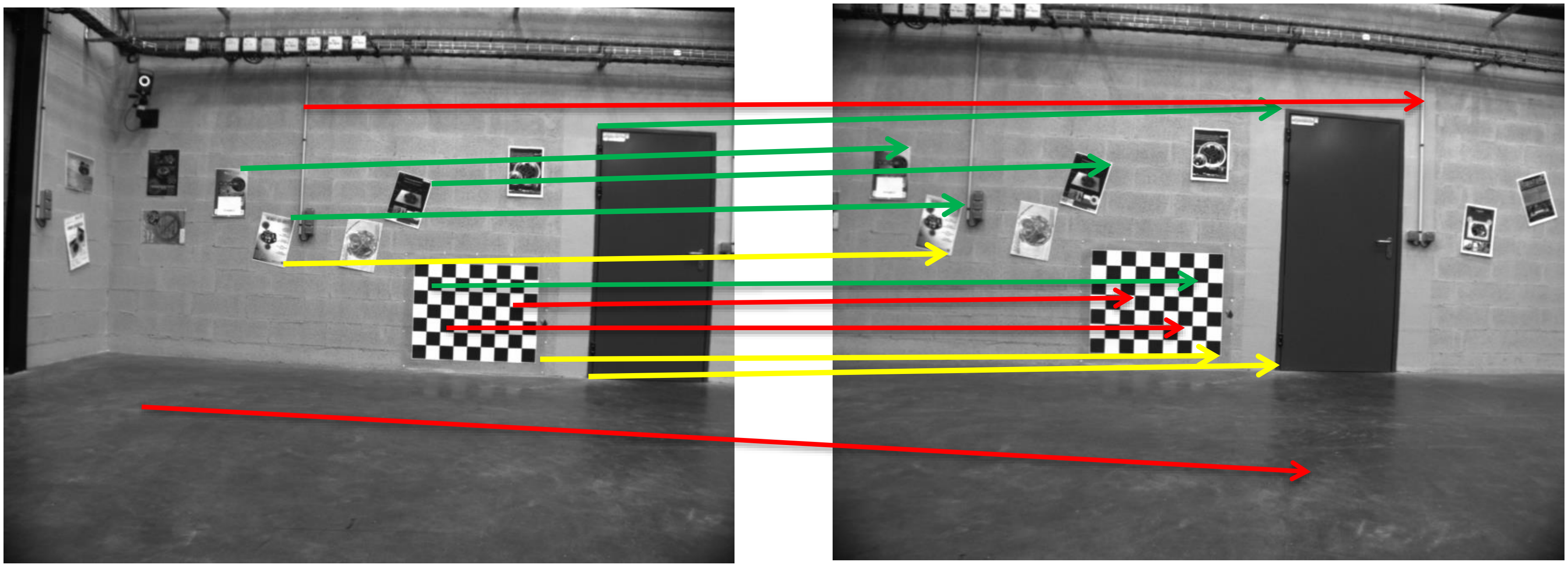
$$x'^T \boldsymbol{E} x = 0$$



x image matches with mis-matches x'

A minimum sample

$$x'^T \boldsymbol{E} x = 0$$



x image matches with mis-matches x'

How many samples are necessery

…to guarantee that at least one correct sample could be selected

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)}$$

s is the number of data points from which a model can be minimally computed

ε is the percentage of outliers in the data (can only be guessed)

p is the requested probability of success

[Fischler&Bolles 1981, Random sample consensus-Ransac]

How many samples are necessery

Example: p = 0.99, s = 5, ε = 0.5 -> N = 145

Number of points (s) p=0.99, e=0.5	8	7	6	5	4	2	1
Number of samples (N)	1177	587	292	145	71	16	7

Simplify the model to reduce the need for data points

- Motion constraints
- Environment constraints
- Measurements from other sensors

Ego-motion estimation with known gravity direction

$$x'^T \boldsymbol{E} x = 0$$

$$E = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \rightarrow \begin{bmatrix} a & b & c \\ -b & a & d \\ e & f & 0 \end{bmatrix}$$

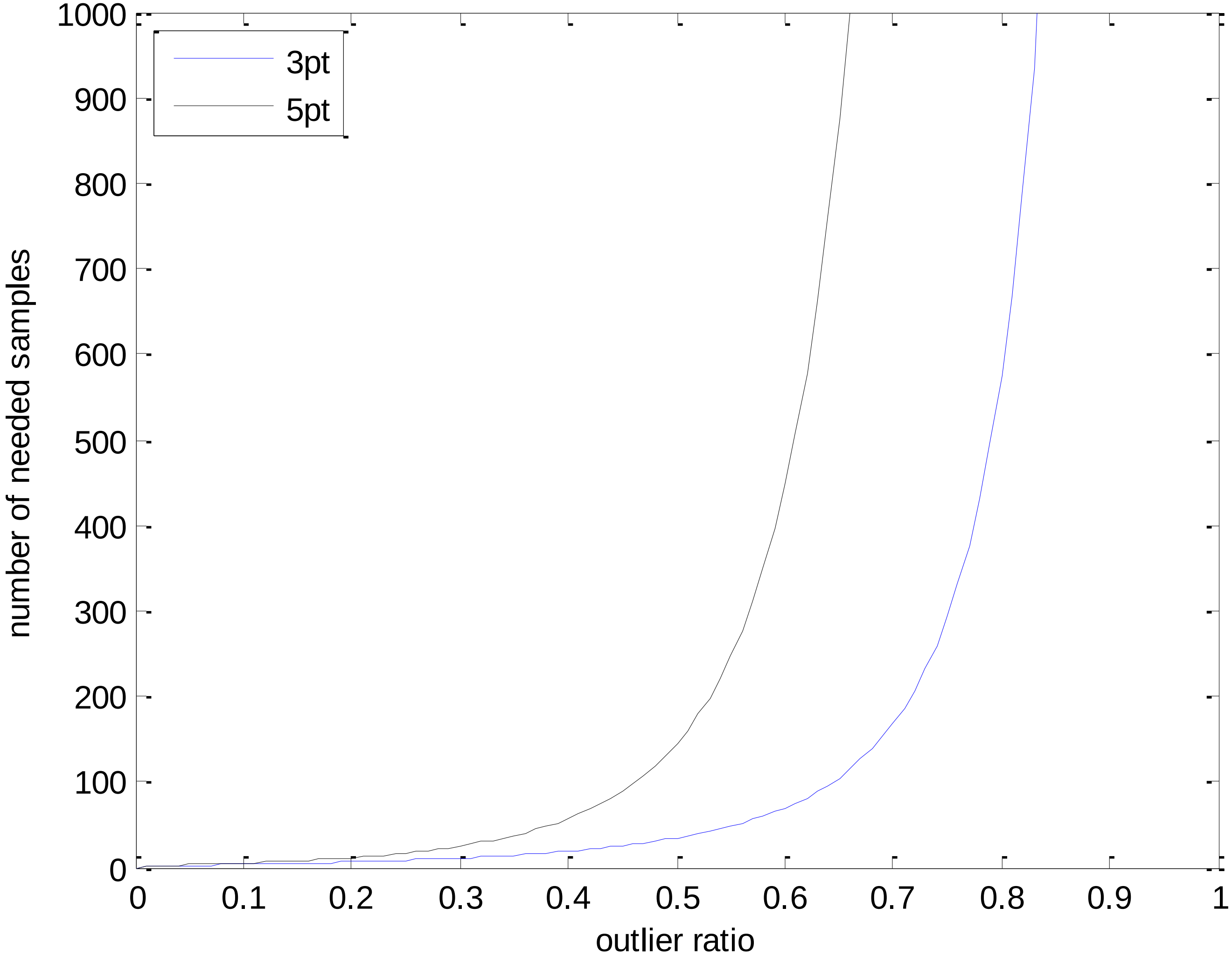
6 instead of 9 unknowns, 1 element equal to 0

Polynomial equation system, where only 3 instead of 5 data points are needed.

$$v_1 ys + v_2 zc + v_3 xs - v_4 yc - v_5 z + v_6 x + v_7 zs - v_8 xc = 0$$
$$u_1 ys + u_2 zc + u_3 xs - u_4 yc - u_5 z + u_6 x + u_7 zs - u_8 xc = 0$$
$$w_1 ys + w_2 zc + w_3 xs - w_4 yc - w_5 z + w_6 x + w_7 zs - w_8 xc = 0$$
$$s^2 + c^2 - 1 = 0$$
$$x^2 + y^2 + z^2 - 1 = 0$$

Fraundorfer et al. (2010)

3pt vs. 5pt

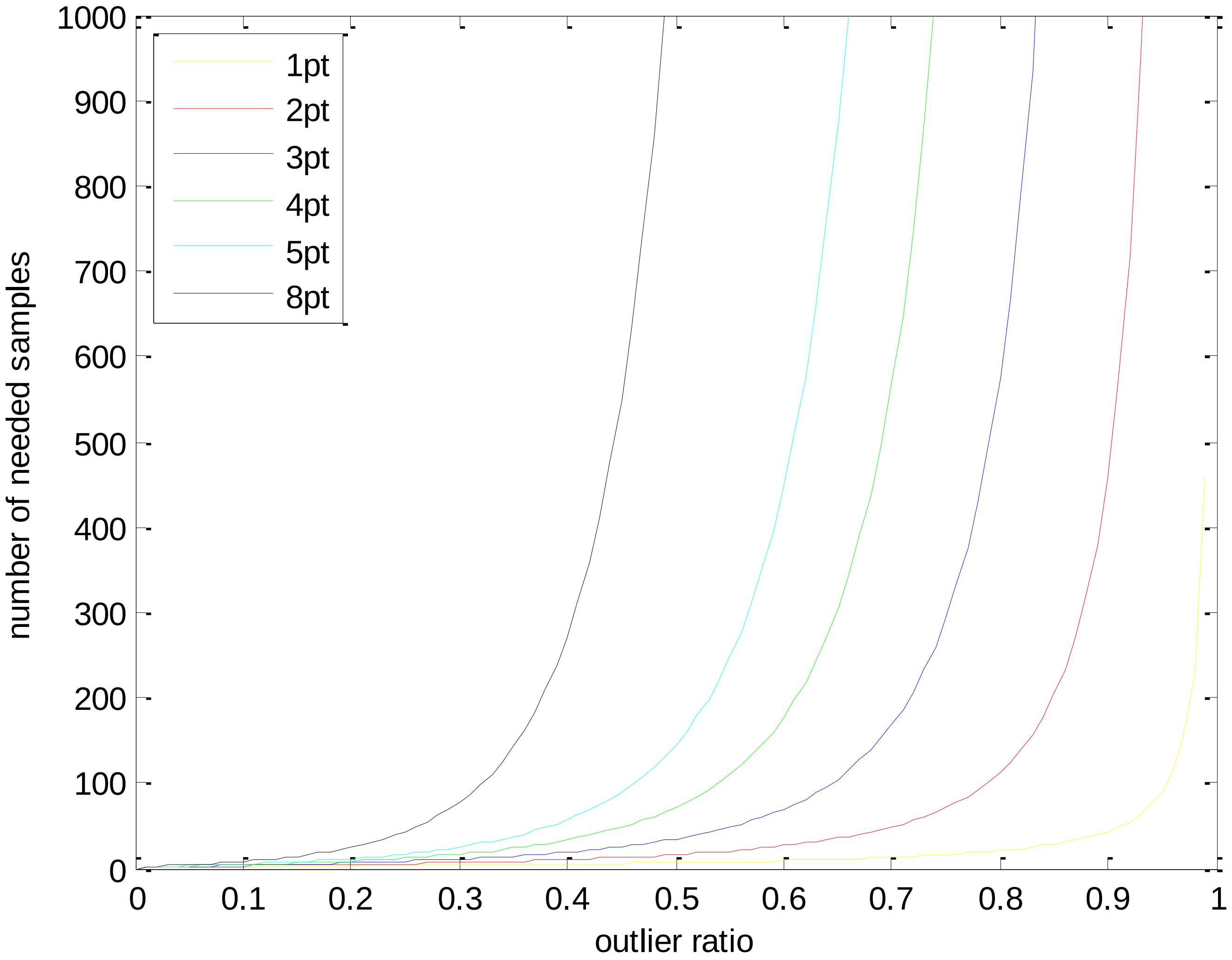


1000
900
800
700
600
500
400
300
200
100
0
number of needed samples
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
outlier ratio
3pt
5pt

Contributions in this field

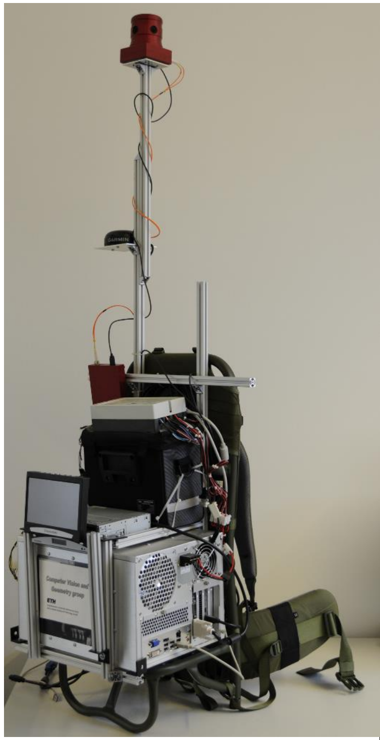
- Banglei et al. (2018) – 1pt relative motion
- Saurer et al. (2017) – 2pt relative motion
- Koch et al. (2016) – 2pt image registration
- Lee at al. (2015) – 3pt multi-camera pose
- Saurer et al. (2015) – 3pt relative pose
- Lee et al. (2014) – 4pt relative motion
- Lee et al. (2013) – 3pt multi-camera pose
- Lee et al. (2013) – 2pt relative motion
- Saurer et al. (2012) – 2pt relative motion
- Fraundorfer et al. (2010) – 3pt relative motion
- Scaramuzza et al. (2009) – 1pt relative motion

All variants



Mathematically involved

```
c(4) =
3*r33^5*r36^2*u31^4*u32*a*r35^3*u21*U12*r12*r31^2*r25*r13*u12*r21+2*r33^3*r36^5*u31^4*u32*a*d*r34*u21*r16*r26*r24*r31^2*r13*U33*r21+3*r33^5*r36*u31^4*u32^2*b*r35^3*r32*r
12*r31^2*u22*r25*r13*u12*U13*r23-
2*r33^3*r36^2*u31^3*u32^2*b*d*r35^3*r32*r34*U12*r16*r24*r31*r22*u22*r25*r13+2*r33^2*r36*u31^2*u32^4*a*d*r35^3*r32*r34^3*u21*r16*r26*r24*r31*r13*U13*r21+r33^3*r36^5*u31^5
*b*r34^2*r24*r31^2*r11*r22*u22*r25*r15*U13+6*r33^5*r36^2*u31^4*u32*a*r35^3*r32*r16*r14*r12*r26*r31*r22*u22*U32+r33^4*r36^2*u31^2*u32^3*c*r35^3*r32*r34*r16*r31^2*U11*u22*
r25*r13*r23-2*r33^2*r36*u31^2*u32^4*a*d*r35^3*r32*r34^3*u21*r16*r26*r24*r31*r13*U33*r21-
6*r33^5*r36*u31^3*u32^2*c*r35^4*r32*r14*r24*r31*u22*r25*U31*u12*r21+3*r33^3*r36^2*u31^5*u32*a*r35^2*r32^3*r34^2*u21*U12*r16*r14*r12*r26*r23-
6*r33^5*r36^4*u31^4*u32*a*d*r35*r32*r16*r26*r31*r22*u22*r13*U13+2*r33^3*r36^4*u31^5*b*d*r35*r32*r34*r14*r26*r31*u22*U32*u12*r21-
r33^4*r36*u31^2*u32^4*c*r35^3*r34*r14*r26*r31^2*U11*u22*u12*r21-
2*r33^3*r36*u31^2*u32^3*c*r35^4*r32*r34*r16*r14*r12*r31*u22*r25*U21*r23+4*r33^3*r36^4*u31^3*u32^2*b*r35*r32*r34*r24*r31*r11*r22*u22*r25*r15*U13+2*r33^3*r36^3*u31^3*u32^3
*a*d*r35*r34*r14*r31^3*u22*r25*u12*U23*r23-
r33^2*r36*u31^2*u32^4*a*d*r35^3*r34^3*u21*r14*r31^2*r25*u12*U13*r21+r33^4*r36^4*u31^2*u32^2*b*r35^2*u21*r31^2*r11*r25*r13*u12*U23*r21-
2*r33^3*r36*u31^2*u32^3*a*d*r35^4*r32*r34*u21*r16*r31*r25*r13*U13*r21+r33^2*r36^2*u31^3*u32^2*c*r35^3*r32^2*r34^2*r16*r31*U11*u22*r25*r13*r23+2*r33^4*r36*u31^3*u32*a*r35
^5*u21*U12*r12*r31^2*r22*r25*r13*u12-
r33^2*r36^2*u31^2*u32^4*c*r35^2*r32*r34^2*u21*r14*r31*U11*r25*u12*r21+r33^2*r36^3*u31^4*u32*a*r35^2*r32^2*r34^2*u21*U12*r12*r26*r31*r13*u12*r23+r33^2*r36^2*u31^6*a*d*r35
^2*r34^2*u21*r14*r31^2*r22*r25*u12*U13+8*r33^4*r36^3*u31^5*u32*a*d*r35*r32*r34^2*r16*r14*r12*r31*u22*r25*r23*U33+2*r33^5*r36^3*u31^5*a*r35^2*r32*u21*r16*r14*r12*r26*r24*r
31*U32*r21+6*r33^6*r36^2*u31^4*u32^2*a*d*r35^2*r32*r14*r26*r31*u22*u12*U13*r21-
r33^2*r36^2*u31^6*b*d*r35^2*r32*r34^2*u21*r12*r26*r31*r13*u12*r23*U22+8*r33^4*r36^3*u31^5*u32*a*d*r35*r32*r34^2*r16*r24*r31*r22*u22*r25*r13*U13+r33^2*r36^4*u31^6*b*d*r32
*r34^4*r16*r24*r31*r22*u22*r25*r13*U32+r33^2*r36^4*u31^6*b*d*r32*r34^4*r14*r24*r31*r22*u22*r25*U32*u12+r33^2*r36^4*u31^6*b*d*r32*r34^4*r12*r31*u22*r25*r13*U32*u12*r23-
8*r33^4*r36^3*u31^5*u32*a*d*r35*r32*r34^2*r16*r24*r31*r22*u22*r25*r13*U33+r33^2*r36^4*u31^6*b*d*r32*r34^4*u21*r16*r14*r12*r26*r31*U32*r23+r33^2*r36^4*u31^6*b*d*r32*r34^4*
u21*r12*r26*r31*r13*u12*r23*U22-
2*r33^2*r36^4*u31^5*u32*a*d*r32*r34^3*u21*r16*r31*r22*r25*r13*U33+4*r33^4*r36^3*u31^3*u32*b*r35^3*r32*u21*r26*r24*r31*r11*r22*r15*U23+r33^2*r36^2*u31^5*b*r35^3*r32^2*r34
^2*u21*r16*r14*r12*r26*r31*r23*U33-6*r33^2*r36^3*u31^4*u32^2*a*r35*r32*r34^3*u21*r26*r24*r31*r11*r15*U32*r21-
```




WOB VH 453


Panasonic

Camera drones

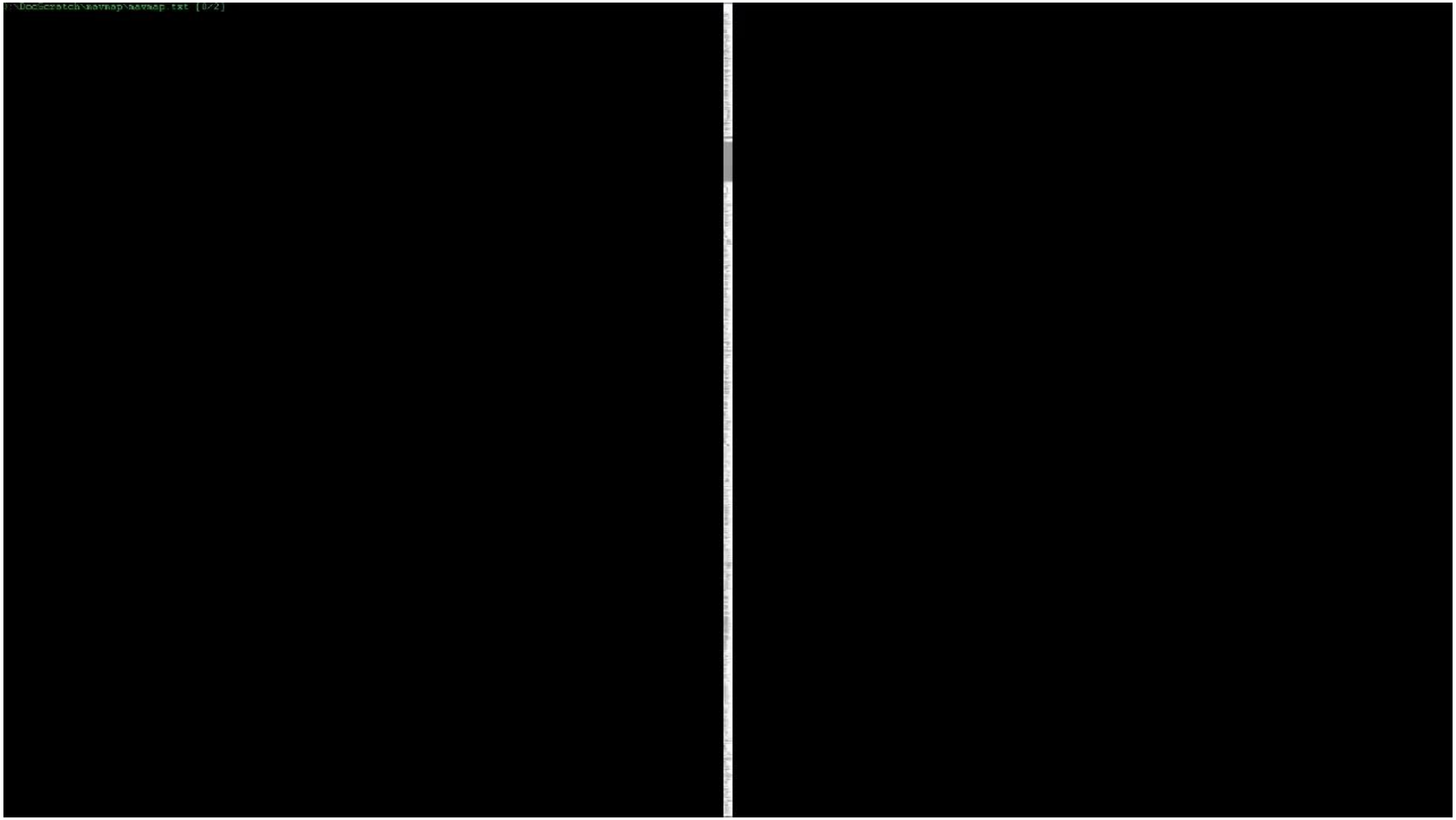
Project list:

- Autonomous Drones (SNF)
- Search&Rescue Drones Sfly (FP7)
- Autonomous Drones (SNF)
- Collaborative Drones (FWF-DFG-SNF)
- Semantic Mapping (Industry)
- Inventory drone (Industry)
- Delivery drone (Post)
- Amazon Gift
- Drone competition (DJI)
- Drones for mining SLIM (H2020)

EU Project SFly

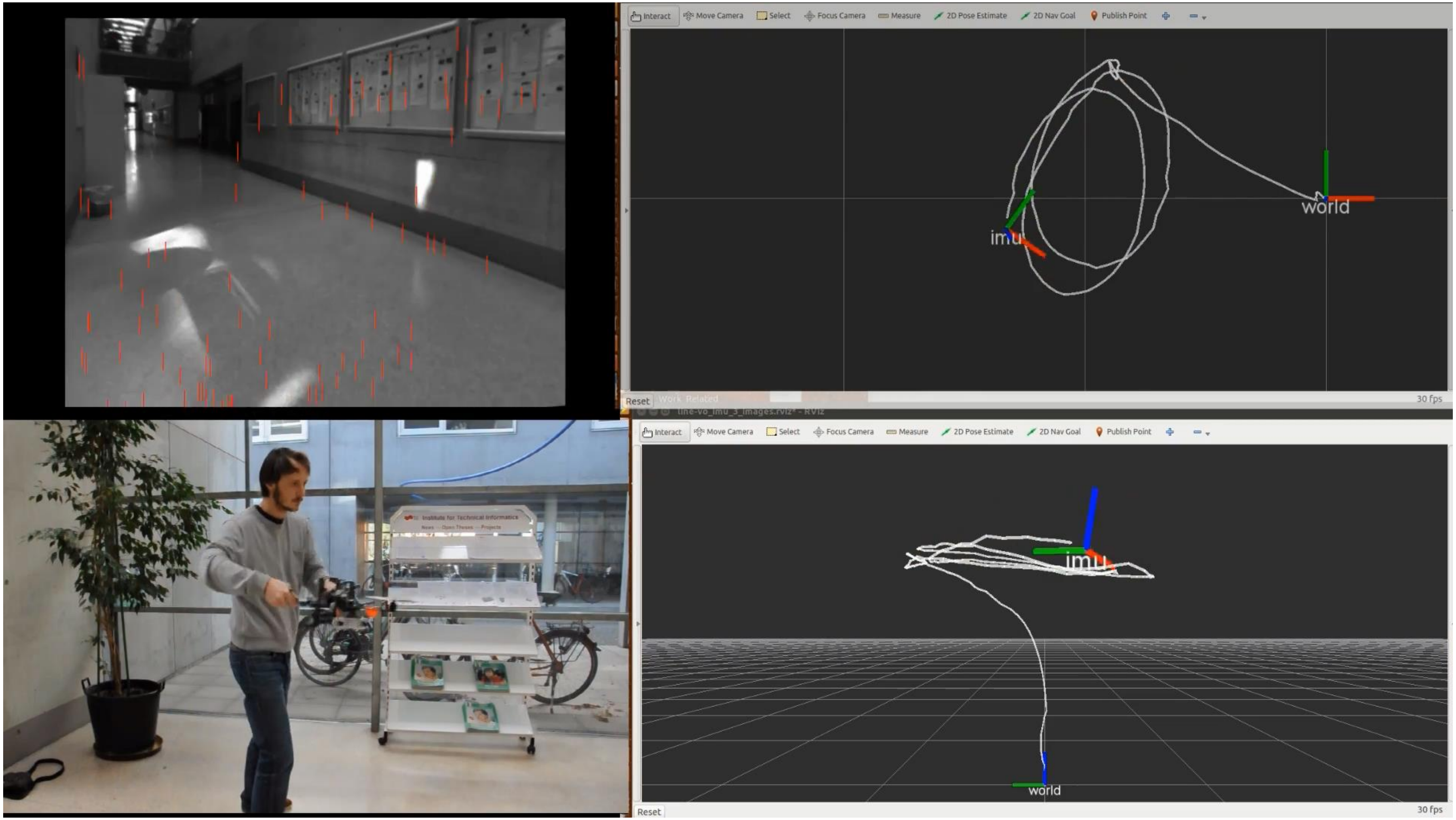
[Doitsitis et al. (2014)]

MAVMAP



https://github.com/mavmap/mavmap

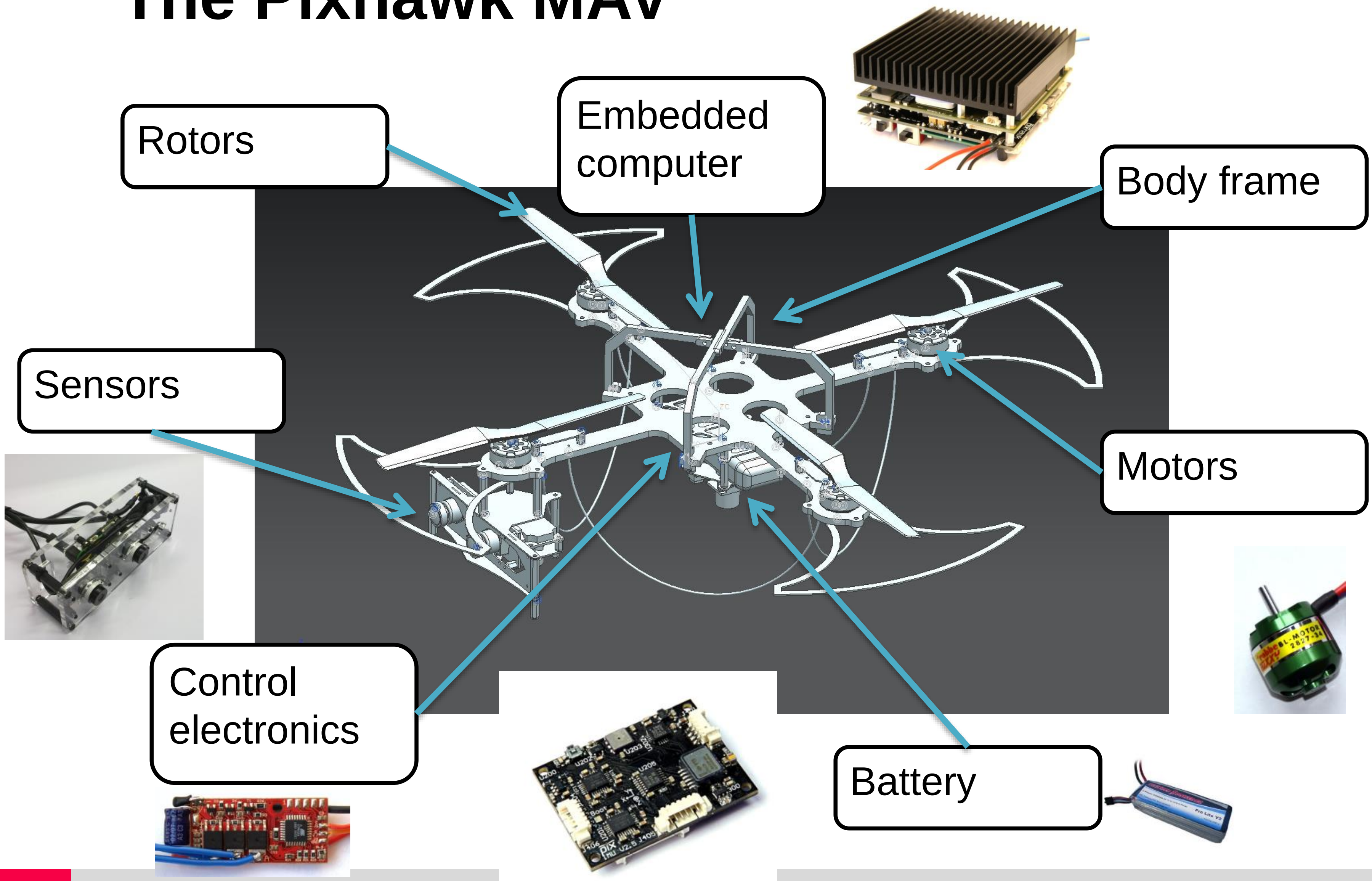
FWF-DFG-SNF Project VMAV



[Holzmann et al. (2016), Direct Stereo Visual Odometry Based on Lines, VISAPP2016]

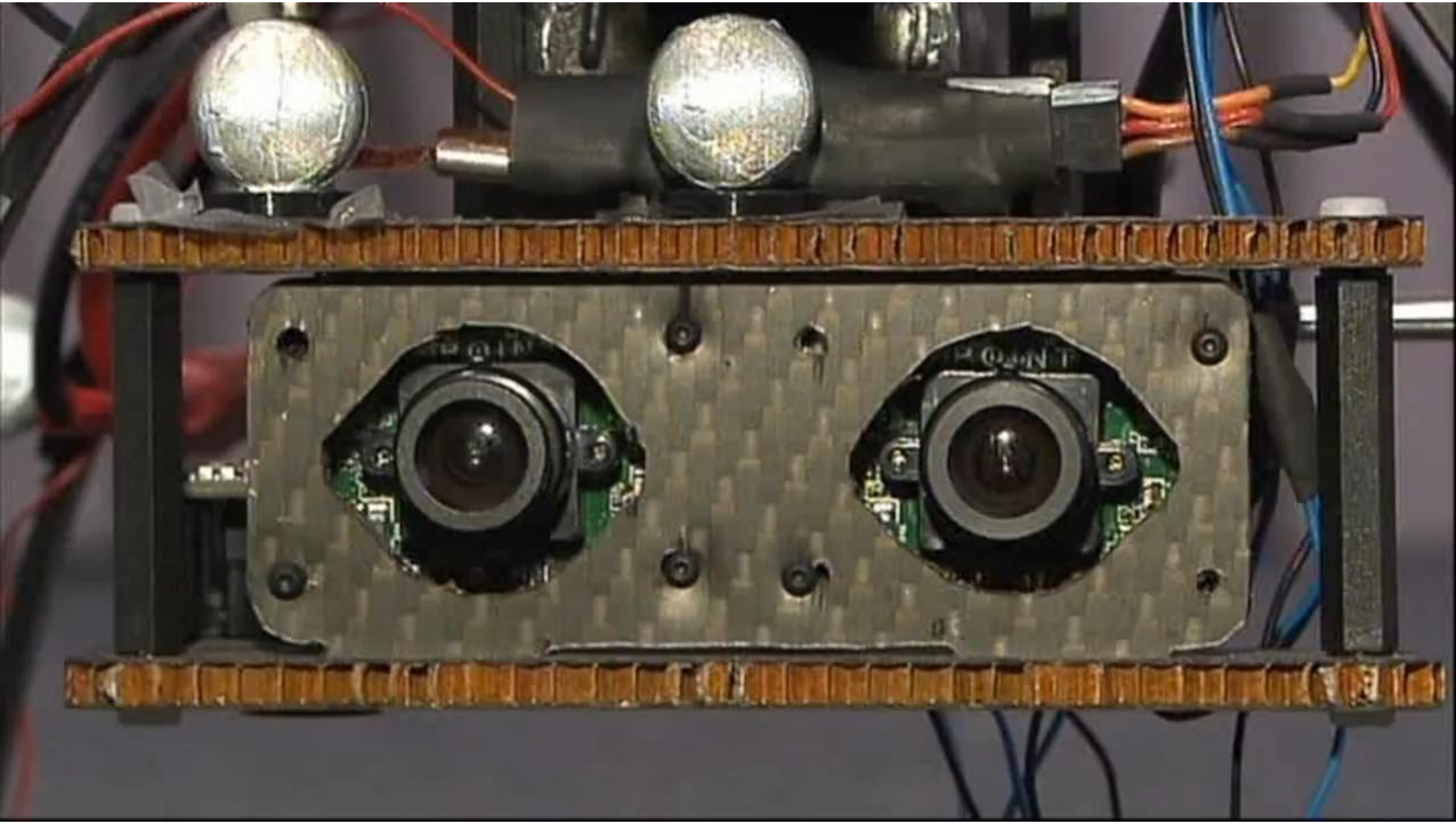
The Pixhawk student team

The Pixhawk MAV



[Meier et al. (2011)]

Onboard computer vision



Navigation

[Heng et al. (2011)]

Dronespace@ICG

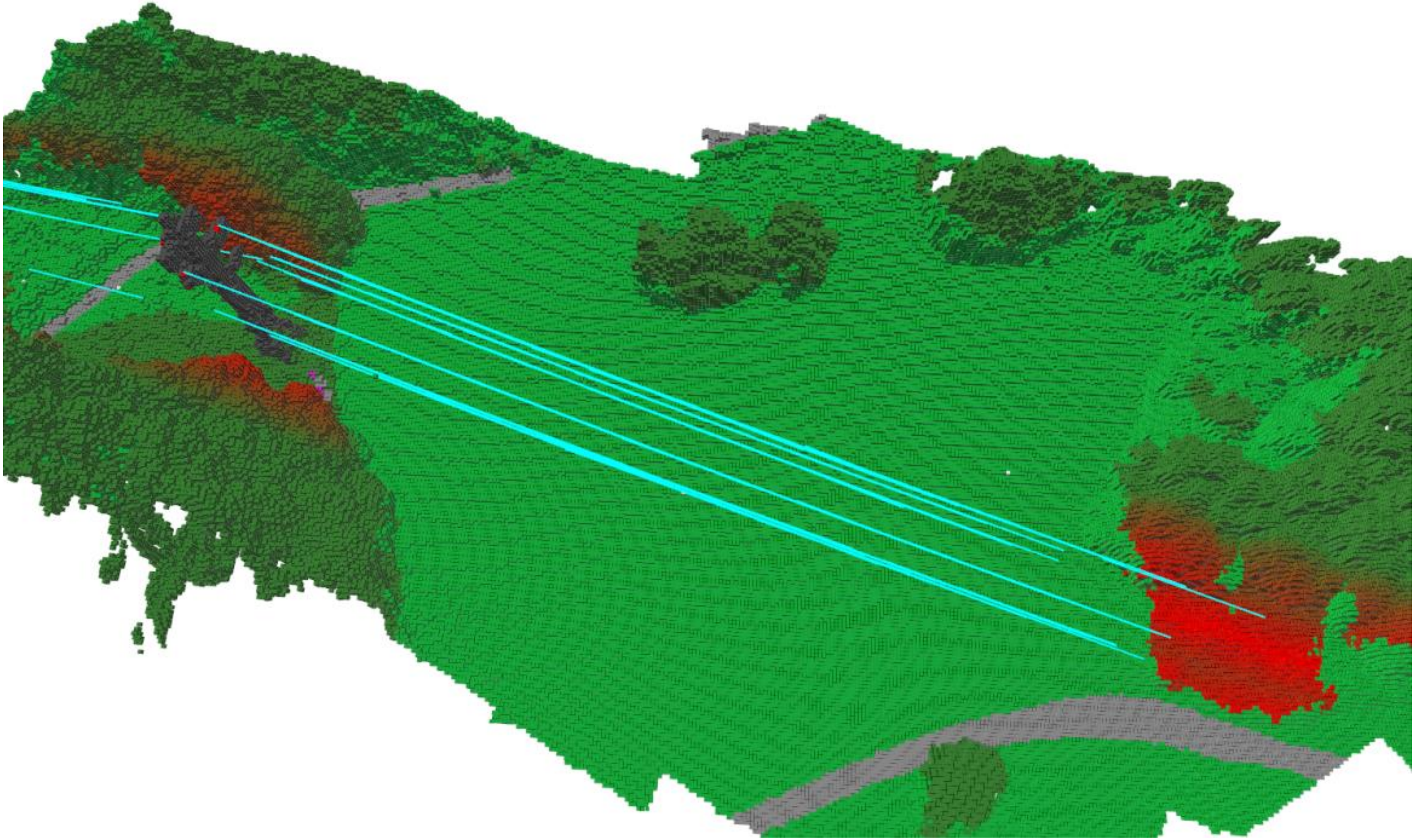
Cortex-A15 2.0Ghz quad core, 2GB RAM
Ubuntu Linux

Drone weighs 500g, is equipped with standard cameras or depth camera

Camera drones lecture

Future research directions

Semantic 3D



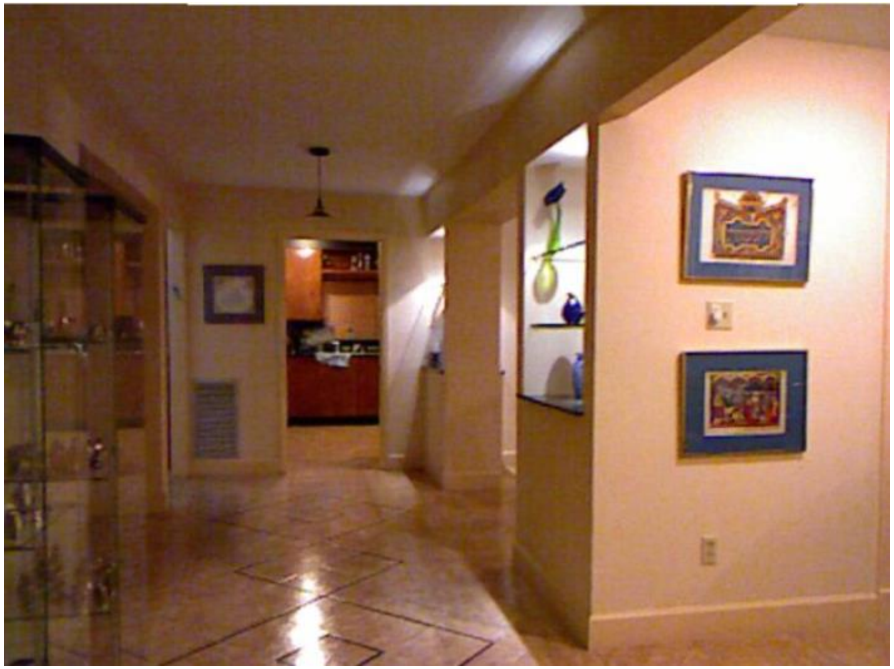
[Maurer et al. (2017), Automated inspection of power line corridors to measure vegetation undercut using UAV-based images, UAVG 2017]

Semantic Drone Data Set

URL: **dronedataset.icg.tugraz.at**

Convolutional neural networks for geometry

Single image depth estimation



Limits of current method

Network estimates depth for a picture on a flat wall

[Koch et al. (2018)]

Thank you