

Introduction to TEK5030 – Computer Vision

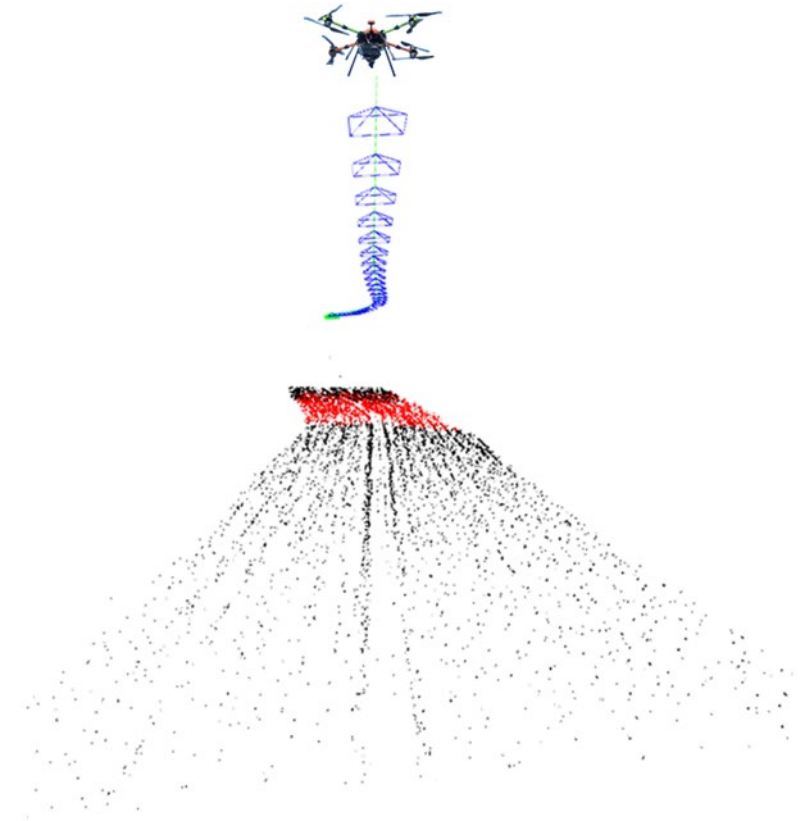
27.01.2023

Trym Vegard Haavardsholm (trym.haavardsholm@its.uio.no)

Idar Dyrdal (idar.dyrdal@its.uio.no)

Thomas Opsahl (thomasoo@its.uio.no)

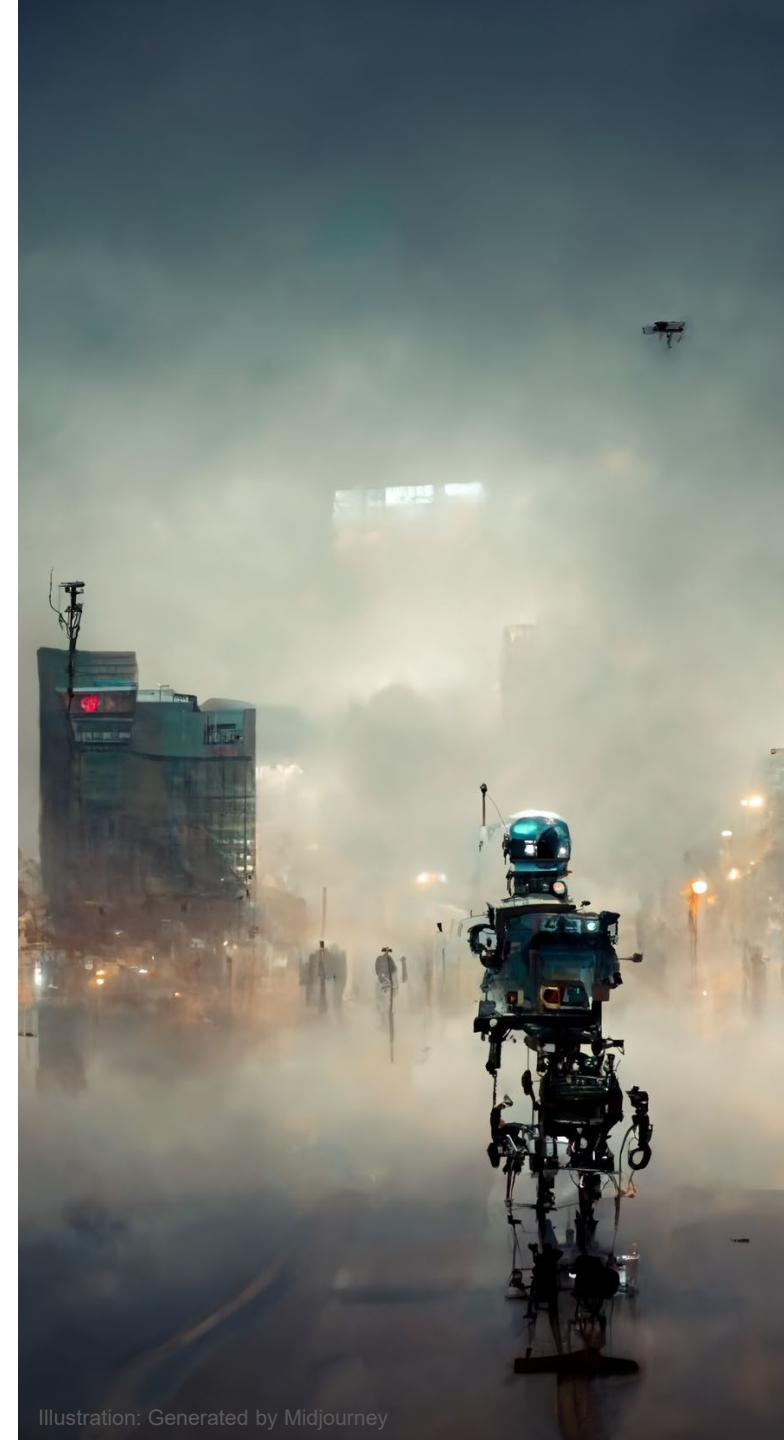
Ragnar Smestad (ragnarsm@its.uio.no)



Today

- A quick introduction to computer vision
- About the course

- Lab: An introduction to OpenCV
 - Processing live video streams with OpenCV!



«An image is worth more than a thousand words»



«An image is worth more than a thousand words»

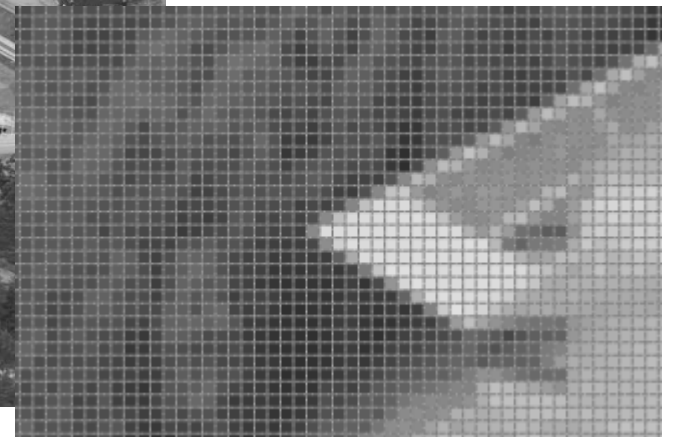
How can we extract
this information?



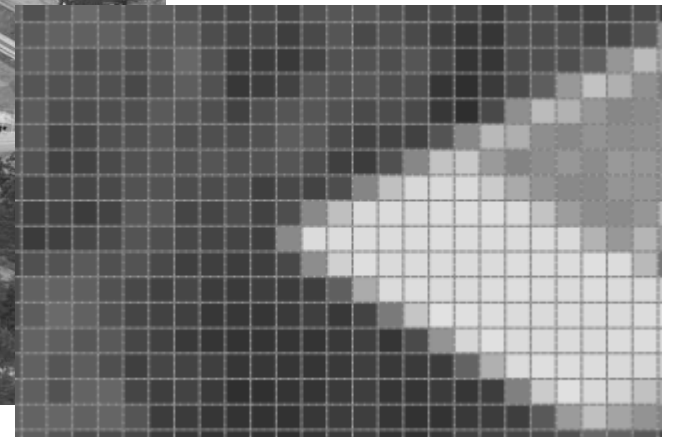
Images and pixels



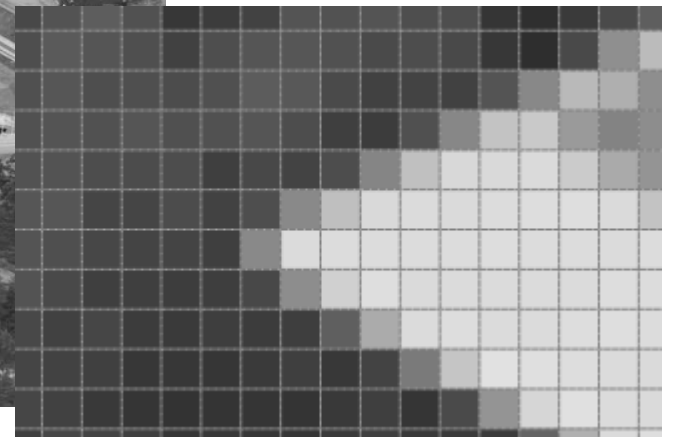
Images and pixels



Images and pixels



Images and pixels



Images and pixels



| | | | | | | | | |
|----|----|-----|-----|-----|-----|-----|-----|-----|
| 80 | 81 | 85 | 77 | 63 | 60 | 80 | 135 | 191 |
| 77 | 63 | 72 | 67 | 77 | 133 | 192 | 216 | 217 |
| 76 | 66 | 78 | 137 | 191 | 217 | 219 | 218 | 220 |
| 68 | 63 | 137 | 219 | 220 | 220 | 220 | 220 | 220 |
| 61 | 62 | 72 | 141 | 208 | 222 | 218 | 219 | 219 |
| 58 | 62 | 60 | 62 | 95 | 171 | 219 | 221 | 219 |

It is easy to calculate with images!



It is easy to calculate with images!



=



**How fast can a machine
crunch pixel values?**



Illustration: Generated by Midjourney

Floating point operations (FP32) per second (FLOPS)



Photo: Miiichiaieil Hieinizileir ([CC BY-SA 4.0](#))

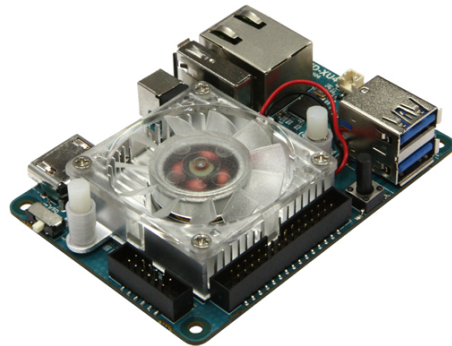


Photo: [Alexandre.willame \(CC BY-SA 4.0\)](#)

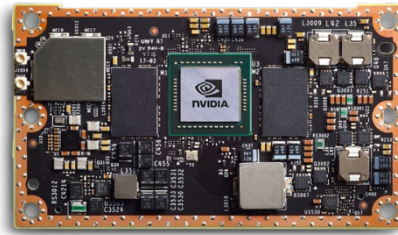


Photo: NVIDIA



Photo: HP



Photo: RIKEN

Raspberry PI 4B
48 GFLOPS

Odroid XU-4
214 GFLOPS

Jetson TX2
782 GFLOPS

My laptop
3 TFLOPS

Fugaku supercomputer
>1000 PFLOPS

(theoretical *peak*)

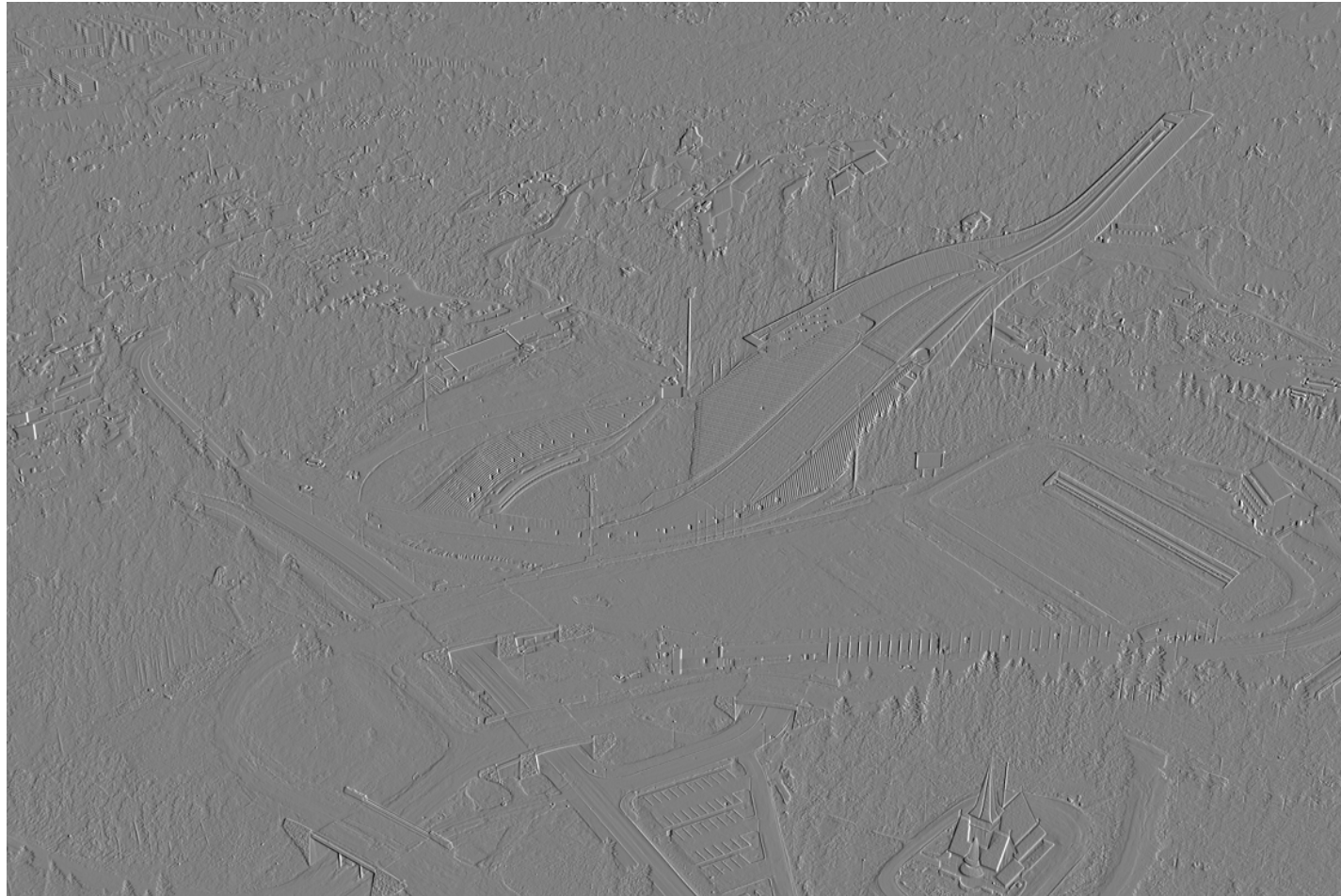
The difference between neighbouring pixels



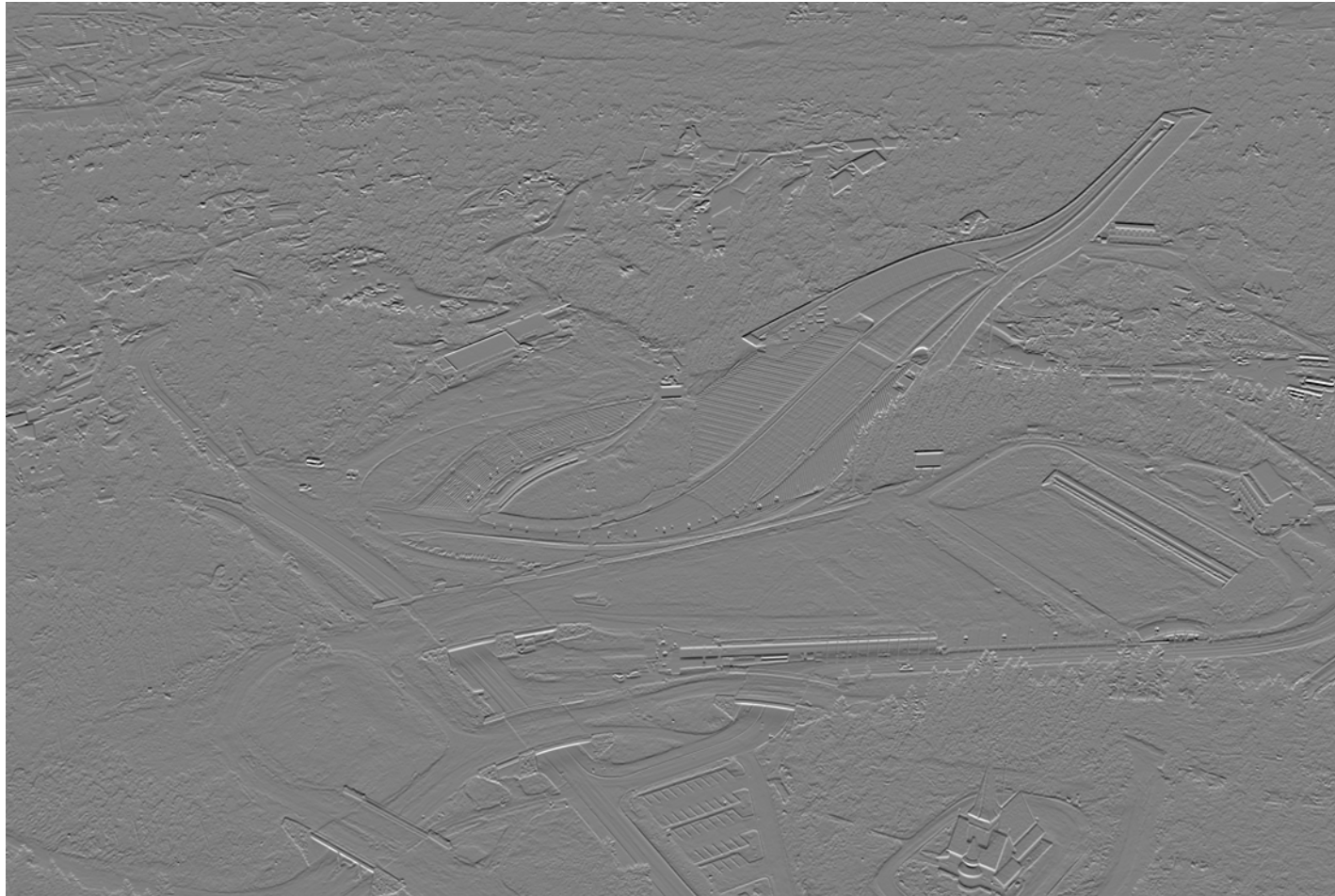
The difference between neighbouring pixels



Horizontal differences



Vertical differences



Edges and corners



Significant corners



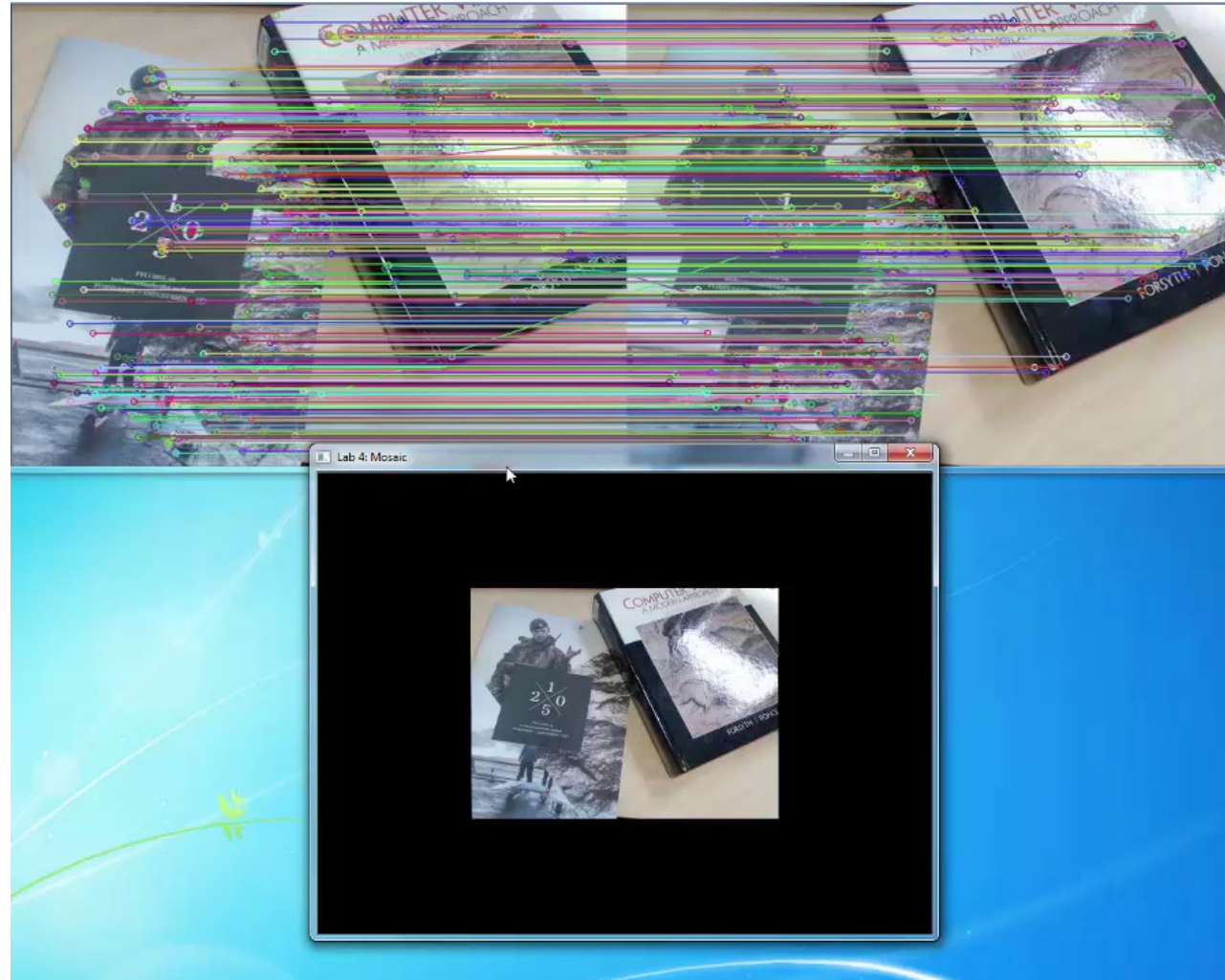
Significant corners



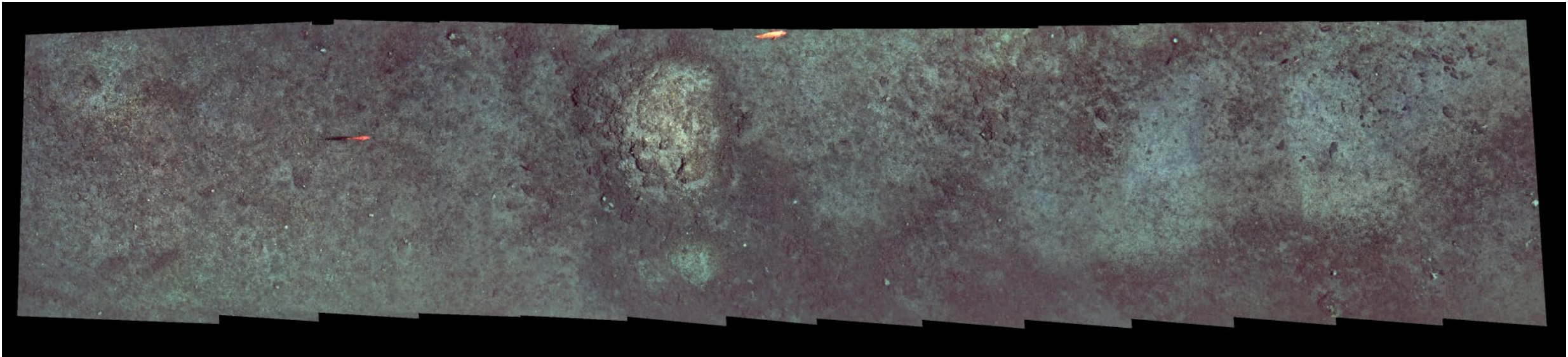
Significant corners



Example: Coregistering images

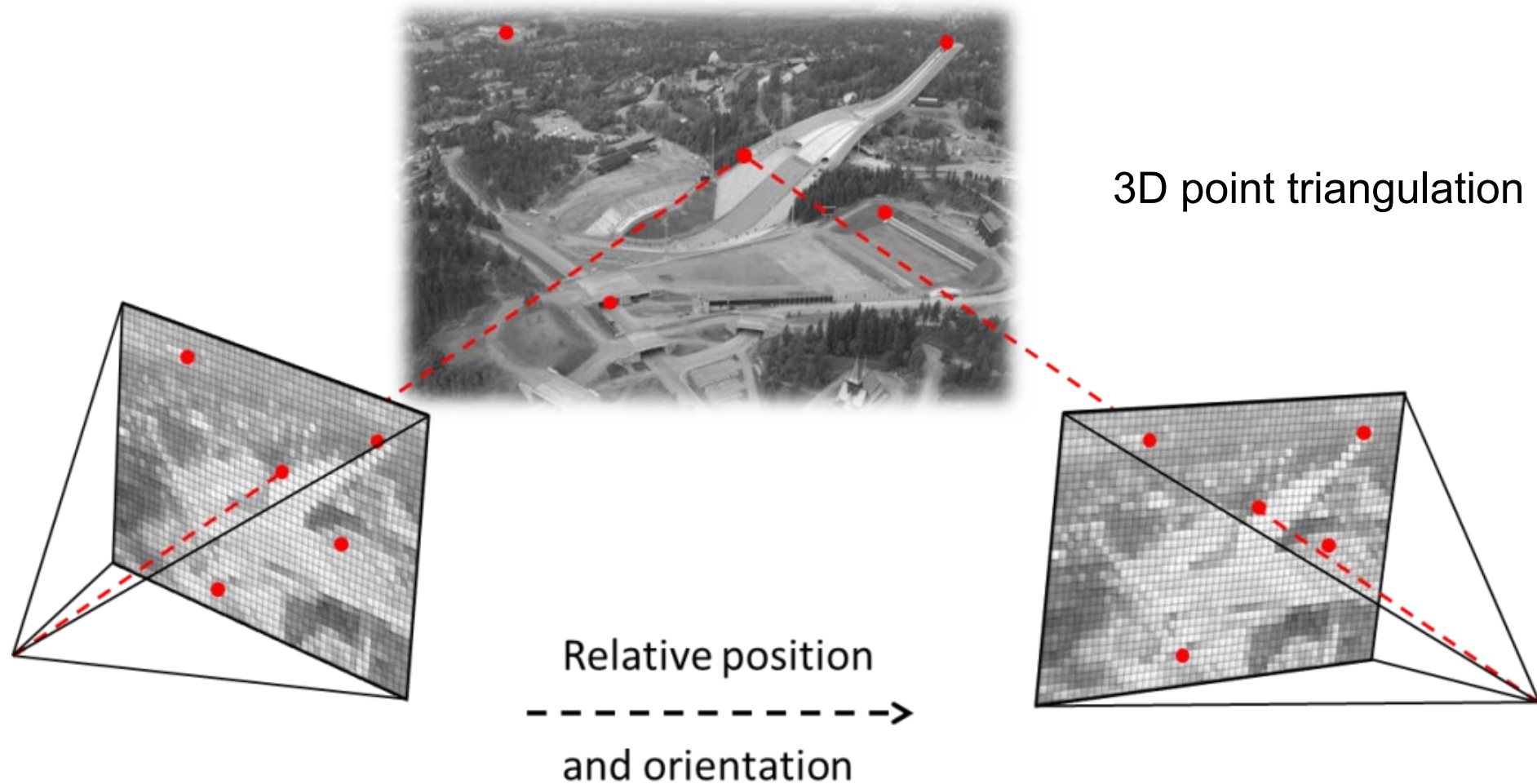


Application: Underwater visual mapping from HUGIN AUV

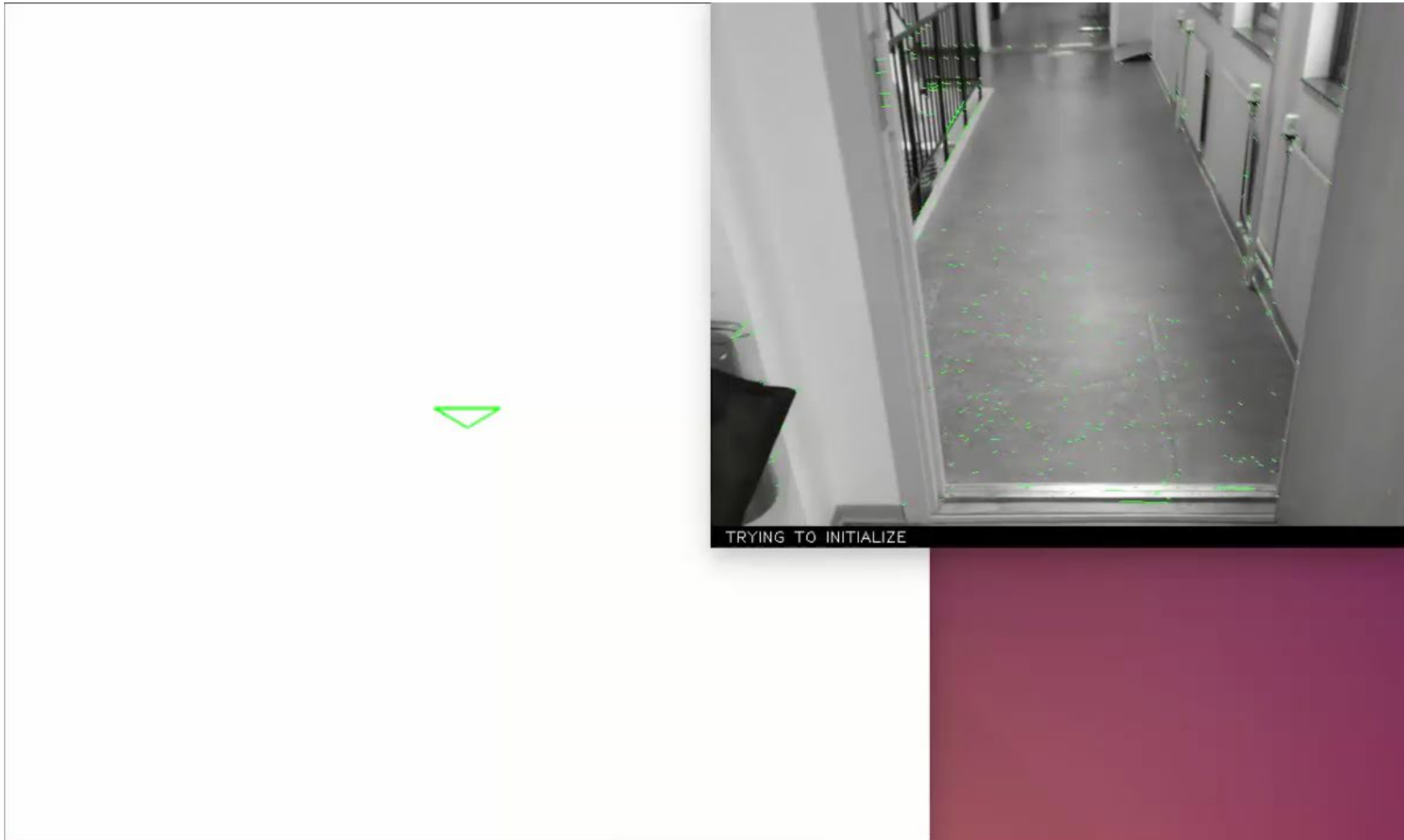


TEK5030

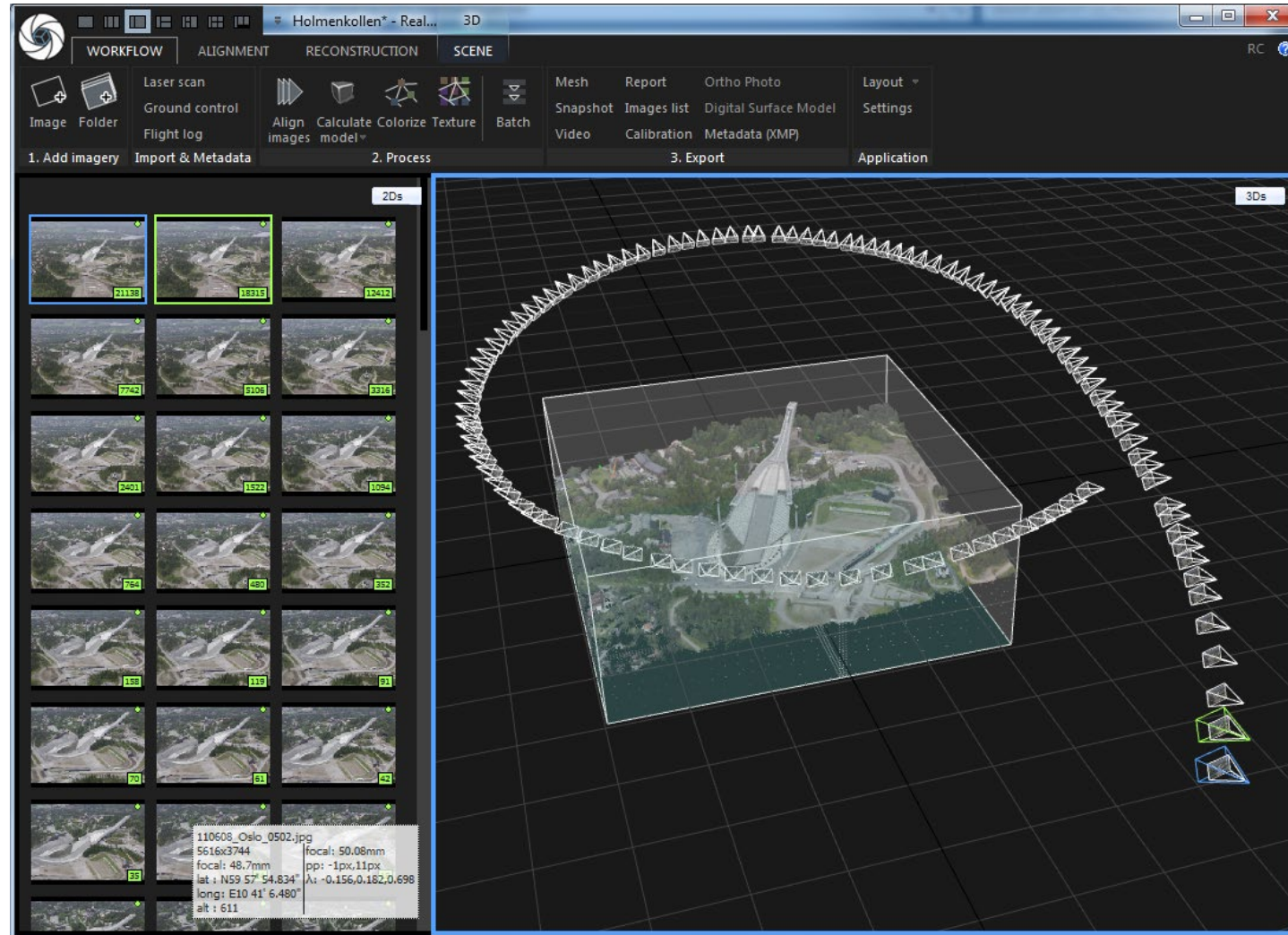
Extracting geometric information from images



Visual navigation



3D reconstruction from images



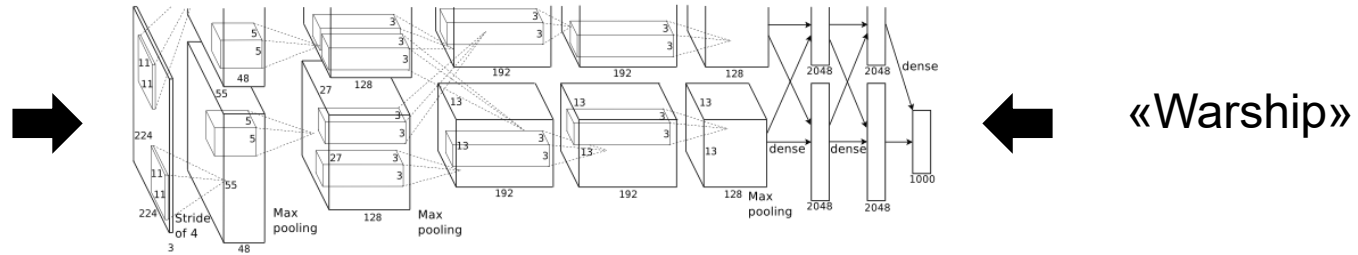
A detailed 3D surface in colors!



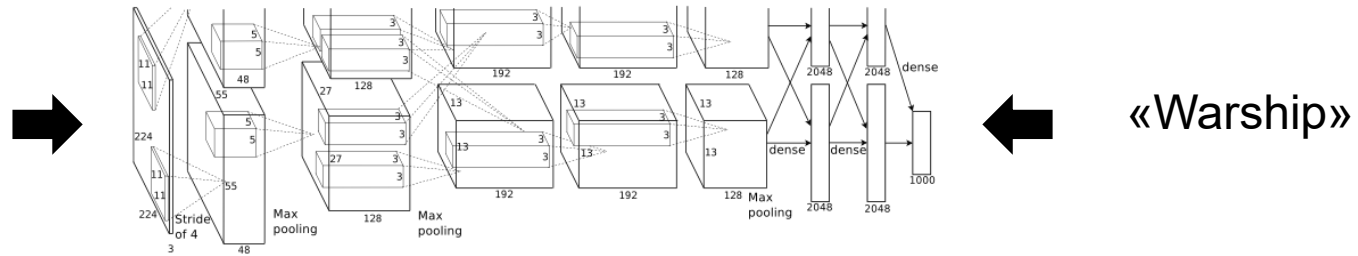
Recognise the shape of a ship



Let the machine learn a better representation itself!

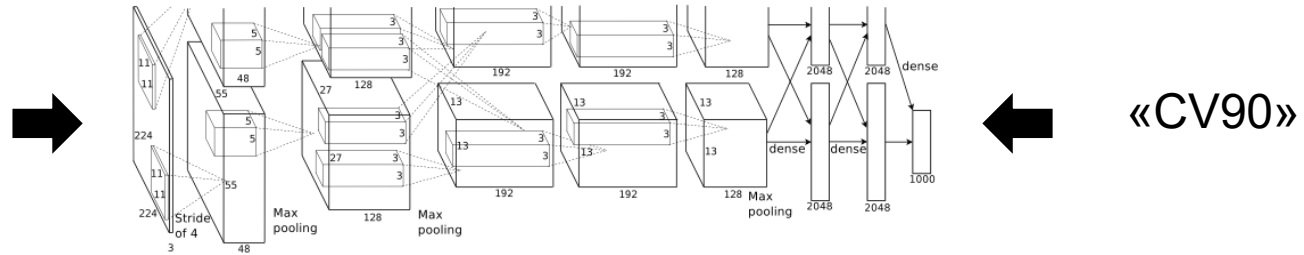


Let the machine learn a better representation itself!



«Warship»

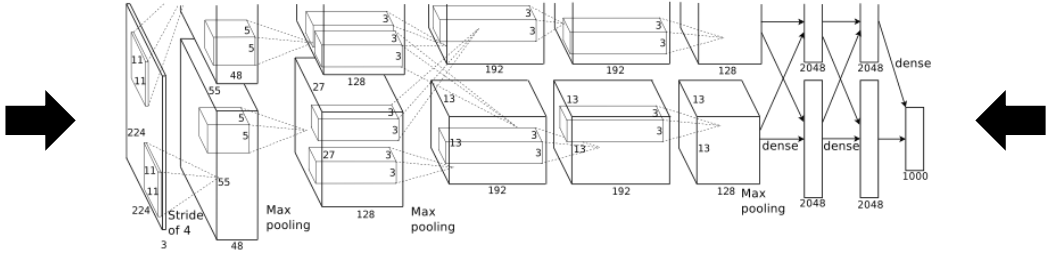
Let the machine learn a better representation itself!



Let the machine learn a better representation itself!

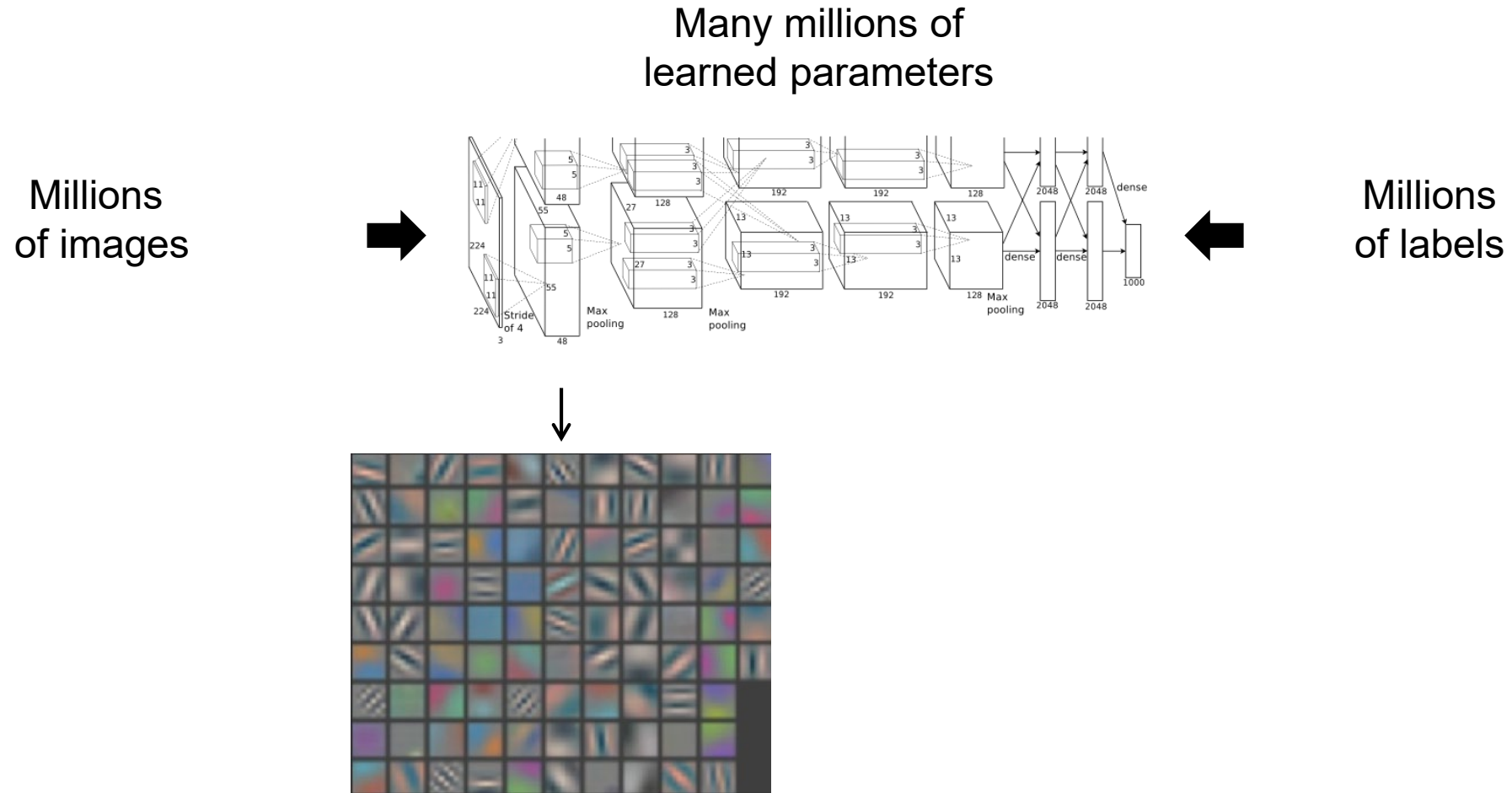
Many millions of learned parameters

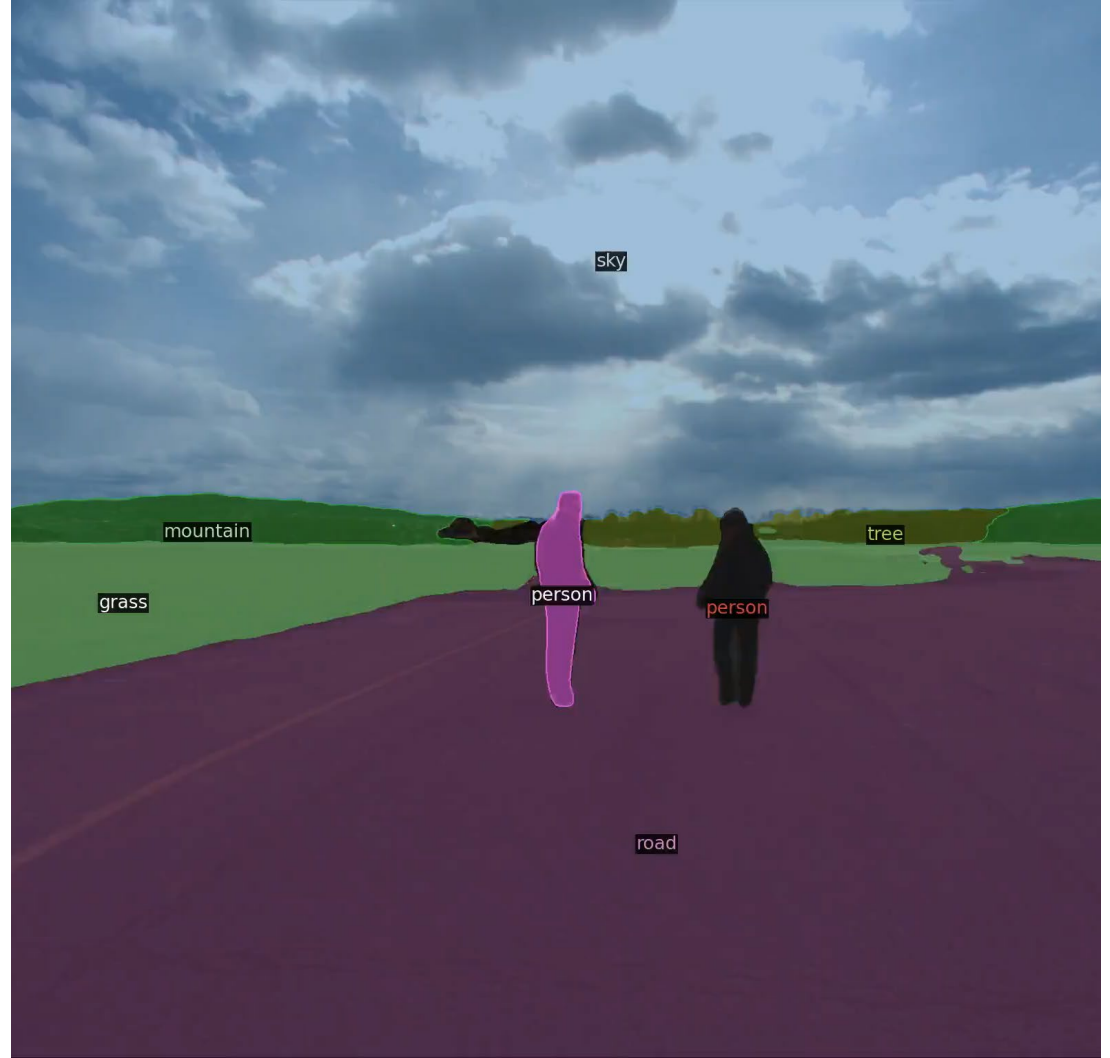
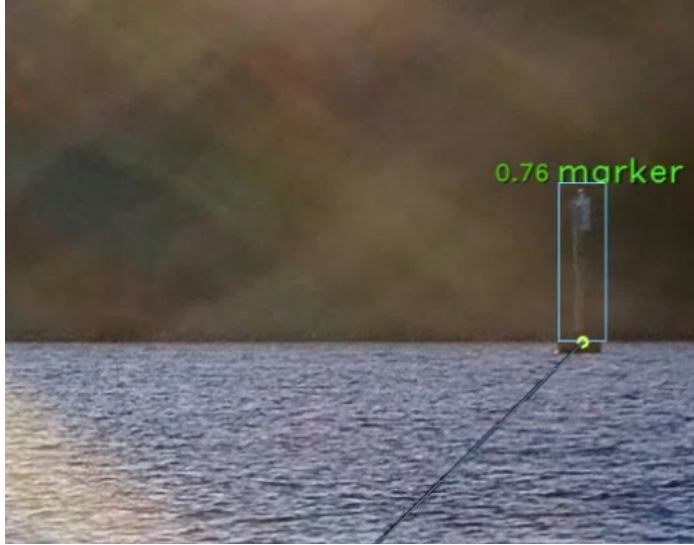
Millions of images

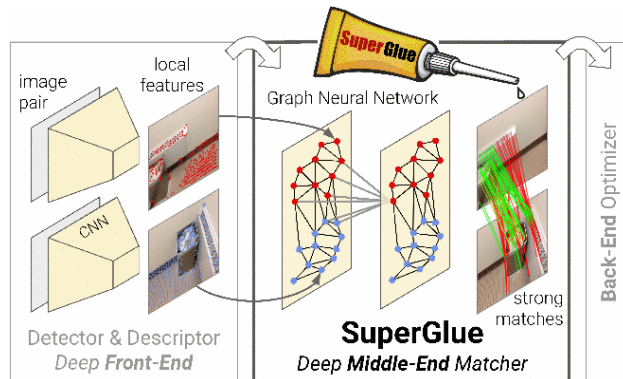
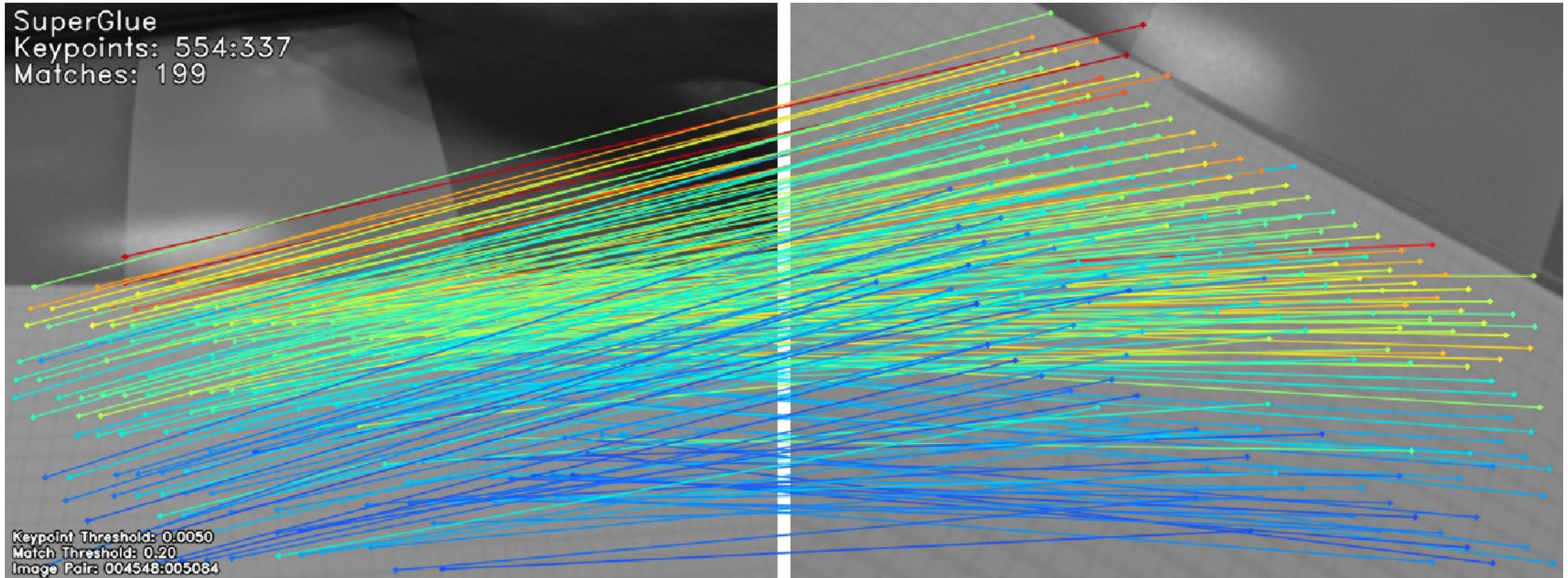


Millions of labels

Let the machine learn a better representation itself!

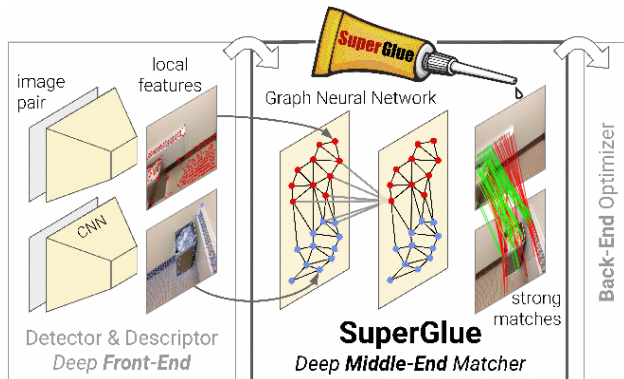
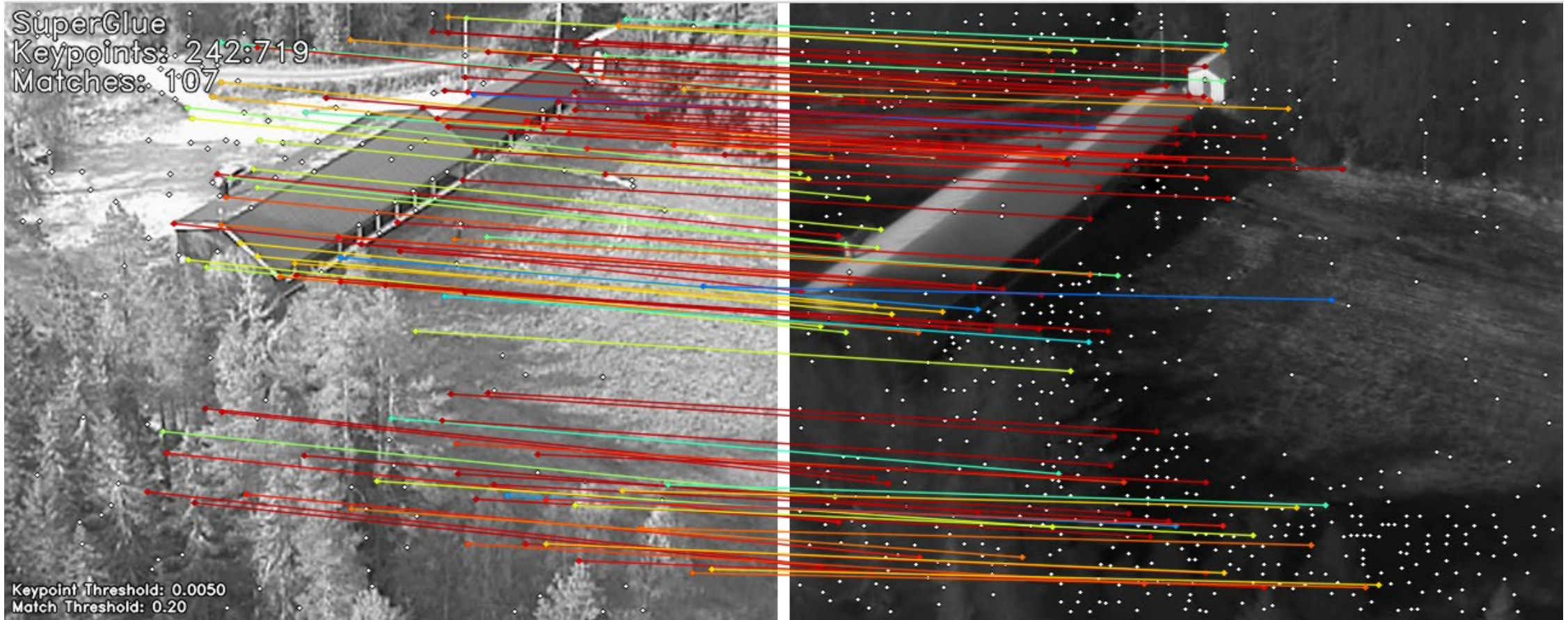






Sarlin, P. E., Detone, D., Malisiewicz, T., & Rabinovich, A. (2020).
SuperGlue: Learning Feature Matching with Graph Neural Networks.
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4937–4946.

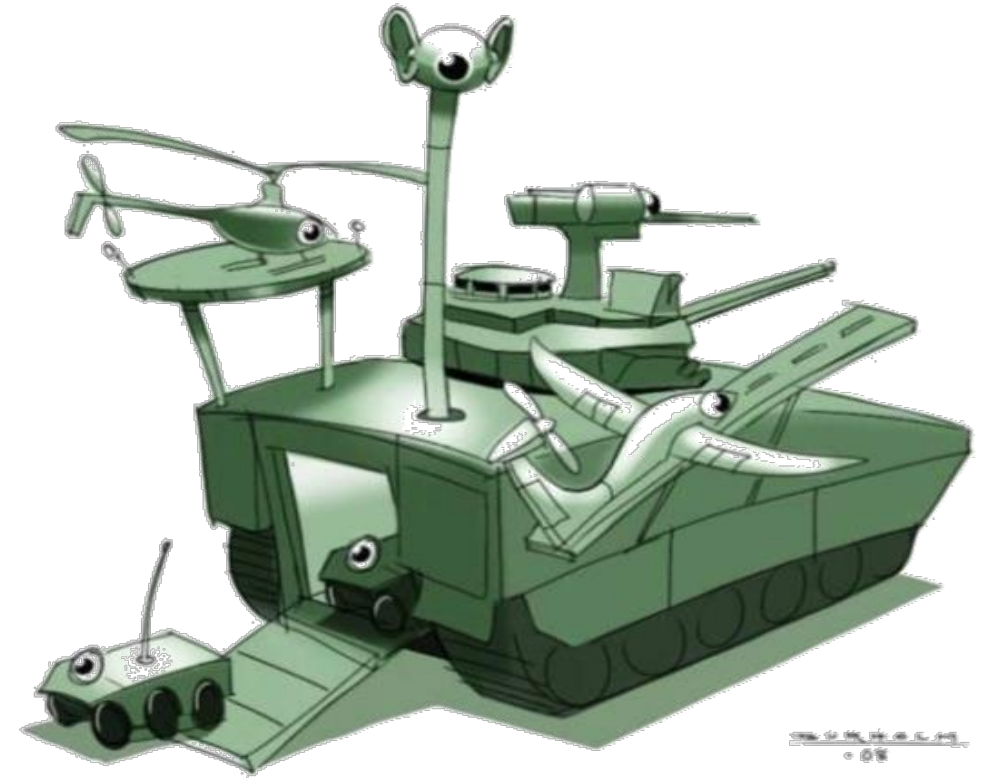
<https://psarlin.com/superglue/>



Sarlin, P. E., Detone, D., Malisiewicz, T., & Rabinovich, A. (2020).
SuperGlue: Learning Feature Matching with Graph Neural Networks.
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4937–4946.

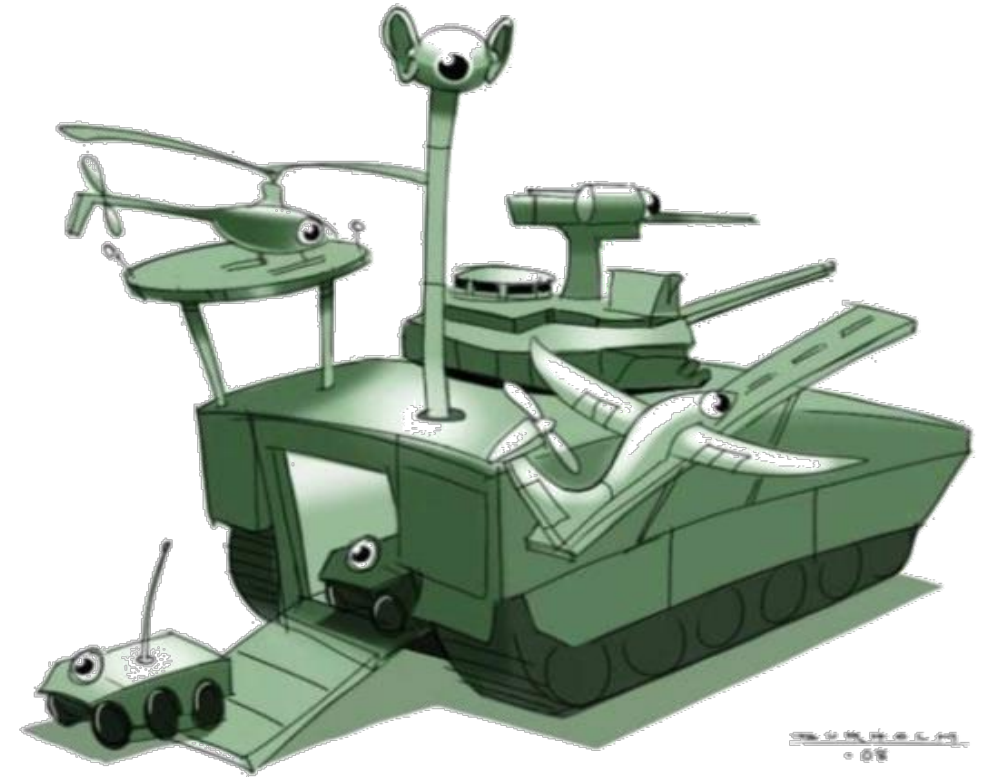
<https://psarlin.com/superglue/>

What is computer vision?



What is computer vision?

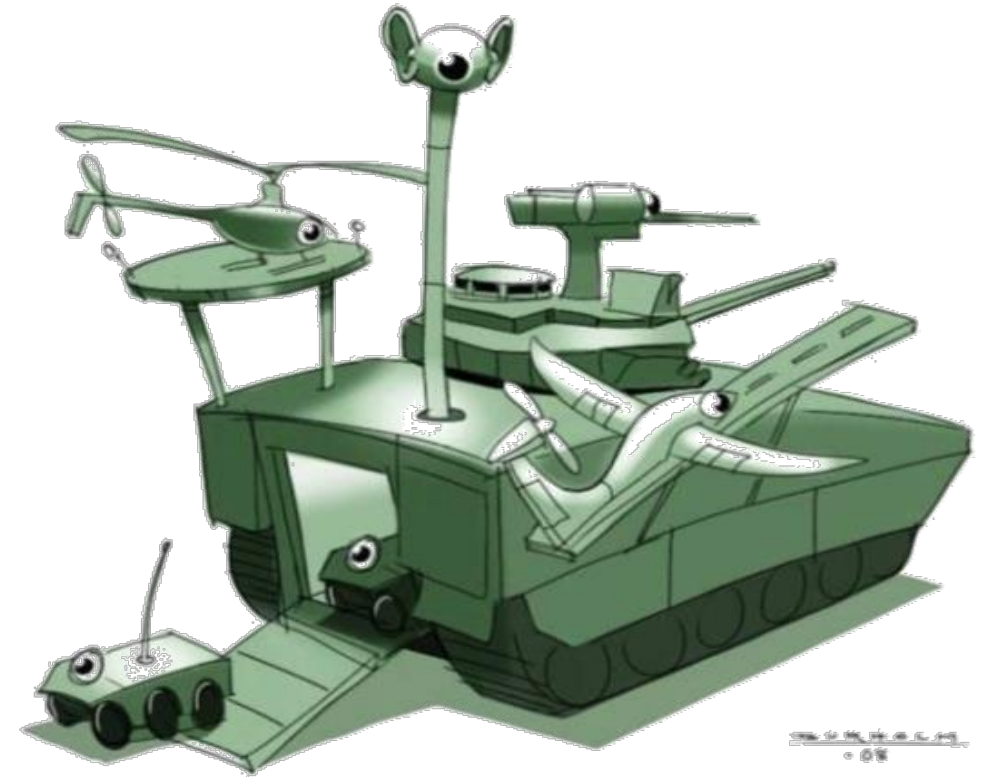
The study of how a machine can interpret and understand its surroundings from images



What is computer vision?

The study of how a machine can interpret and understand its surroundings from images

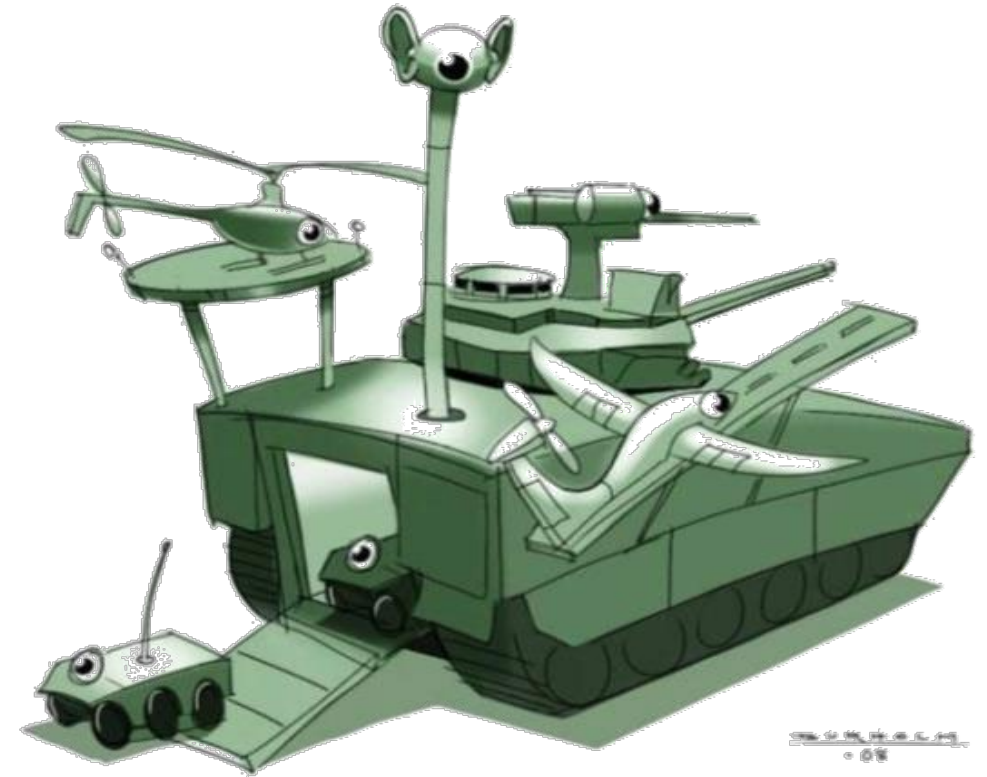
- “Enabling computers to see”



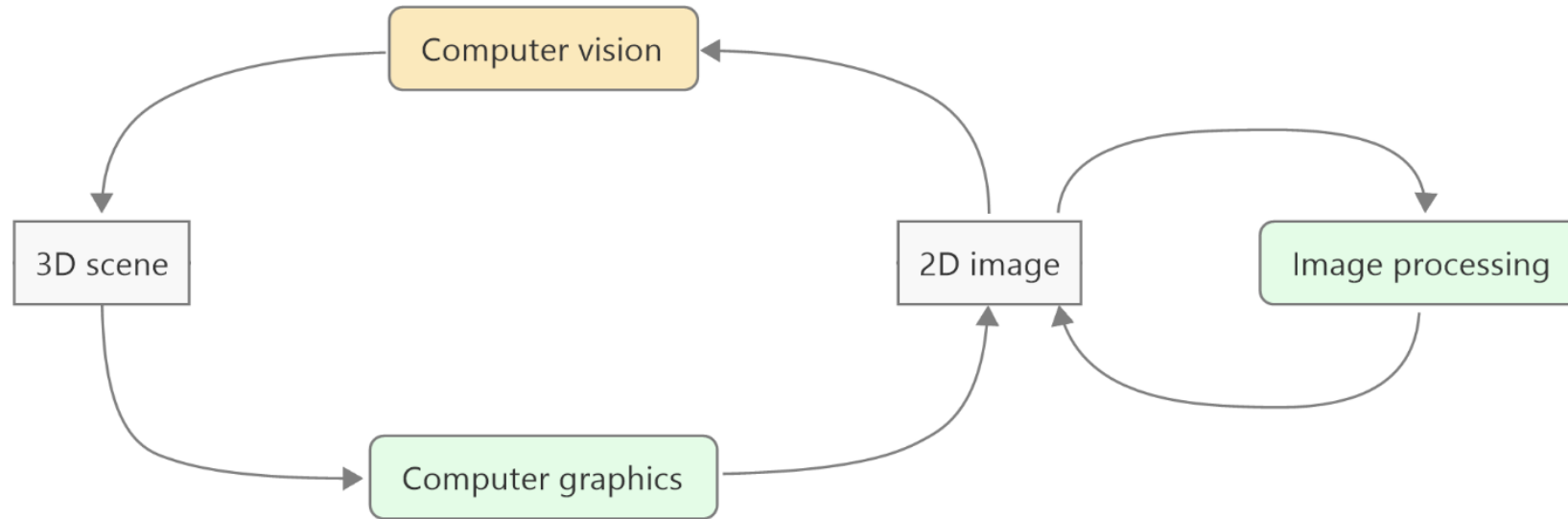
What is computer vision?

The study of how a machine can interpret and understand its surroundings from images

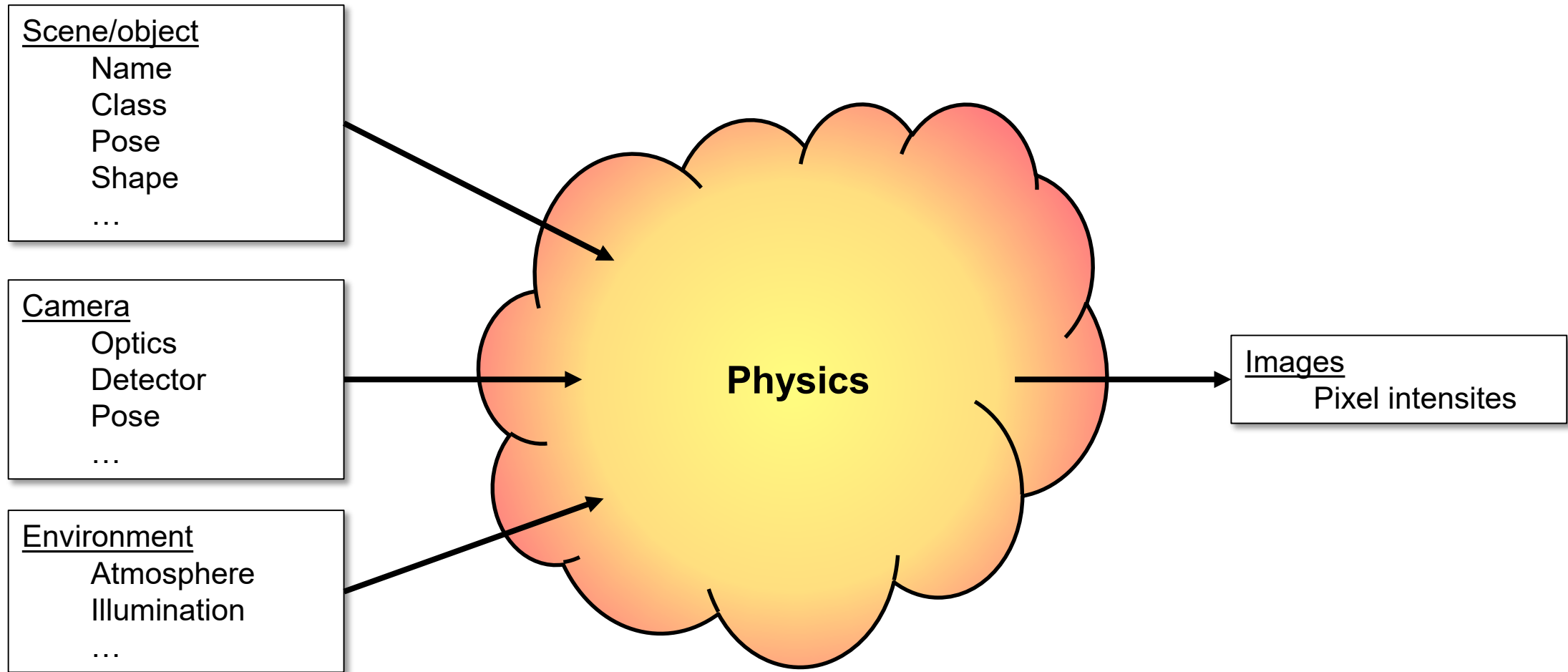
- “Enabling computers to see”
- Image analysis
- Robotic vision
- Perception
- Spatial AI



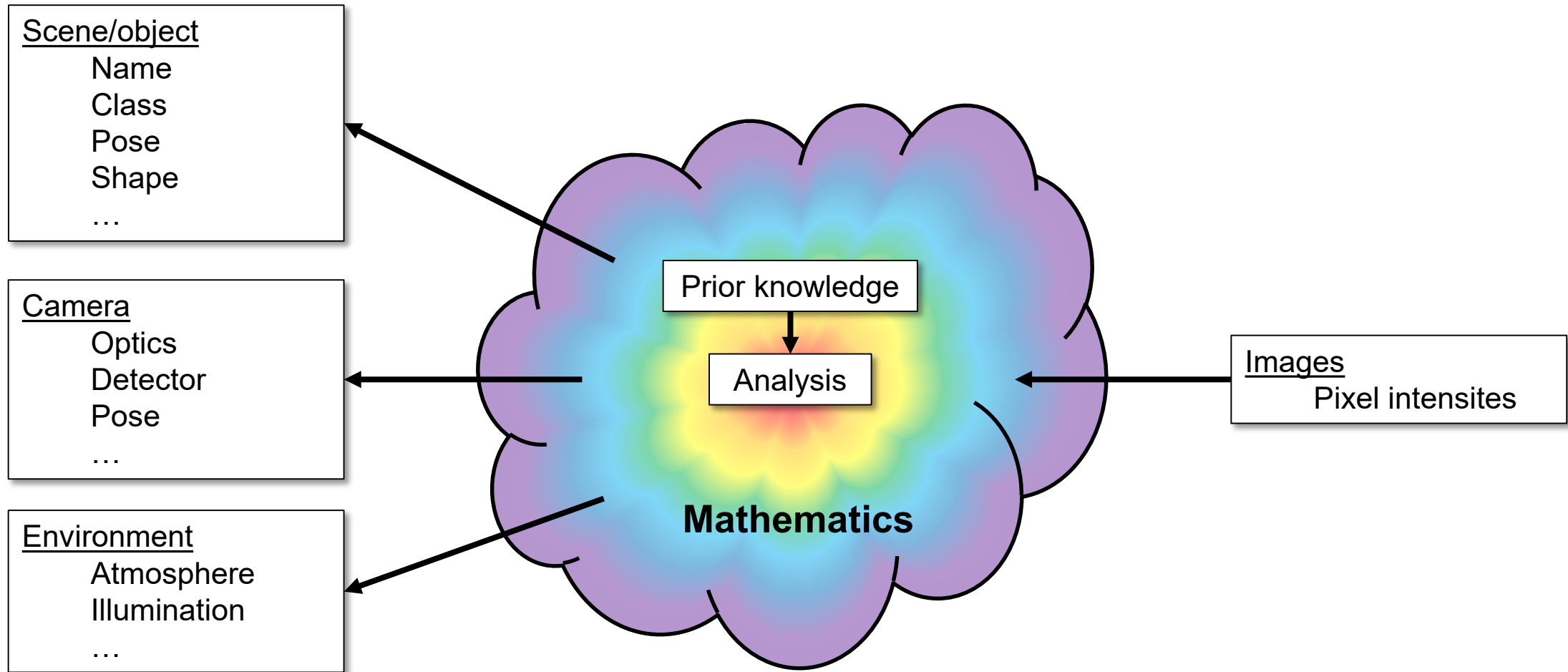
Computer vision is an *inverse problem*!



The *forward* imaging process



The *inverse* analysis process



About the course

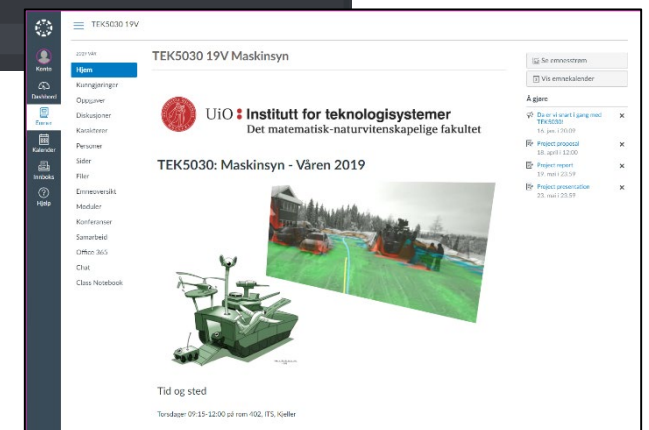
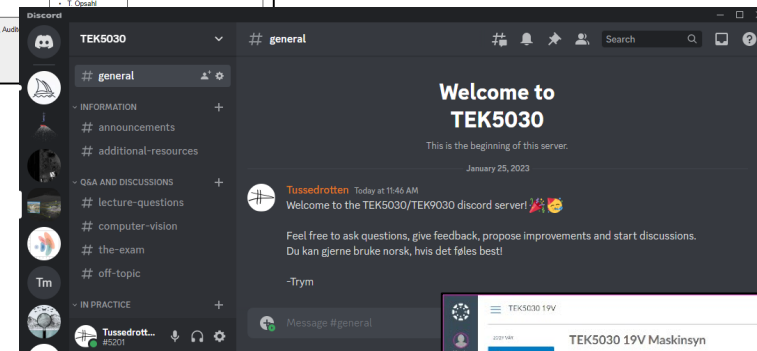
Course sites

- [The semester page](#)
 - Course plan
 - Lecture slides and videos
 - Lab exercises
- [Canvas](#)
 - Course overview
 - Zoom links
 - Project deliveries
- [Discord](#) (invitation sent to all)
 - Announcements
 - Questions
 - Discussions

TEK5030 - Vår 2023
Timeplan

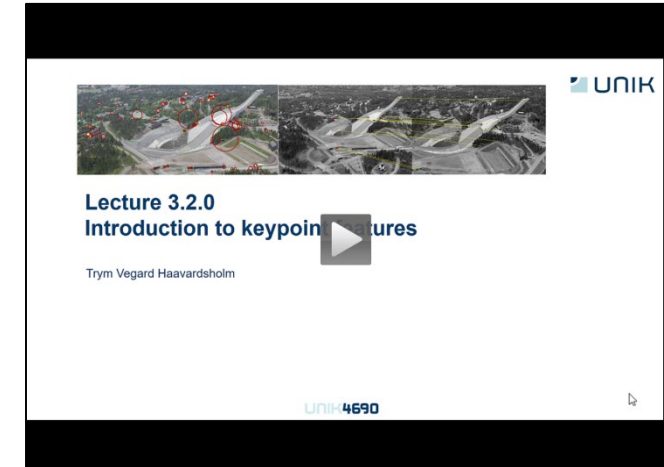
Fellesundervisning
Forelesninger - fre. 09:15-12:00 [Abonner på disse aktivitetene](#)

| Dato | Tid | Aktivitet | Sted | Foreleser | Resurser/pensum |
|------------|-------------|-----------------------------------|-----------------|--|--------------------------|
| 8. 27. jan | 09:15-12:00 | Live lecture and lab | GR19 Auditorium | • I. Dyrhaug • T. Haavardsholm • T. Oppahl | Week 1: Introduction |
| 9. 3. feb | 09:15-12:00 | Lab based on pre-recorded lecture | GR19 Auditorium | • I. Dyrhaug • T. Haavardsholm • T. Oppahl | Week 2: Image formation |
| 9. 10. feb | 09:15-12:00 | Lab based on pre-recorded lecture | GR19 Auditorium | • I. Dyrhaug • T. Haavardsholm • T. Oppahl | Week 3: Image processing |
| 9. 17. feb | 09:15-12:00 | Lab based on pre-recorded lecture | GR19 Auditorium | • I. Dyrhaug • T. Haavardsholm • T. Oppahl | |



«Flipped classroom»

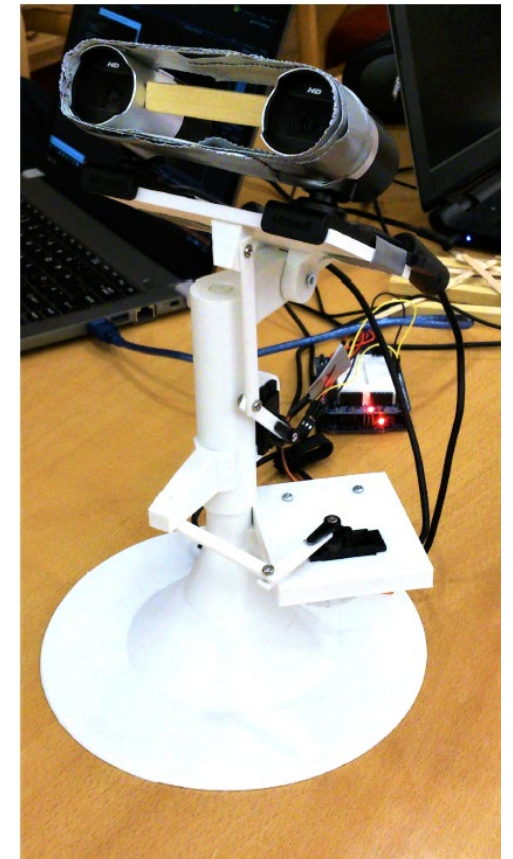
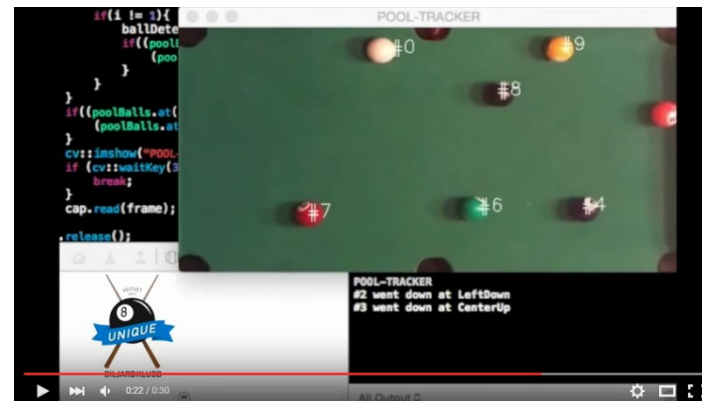
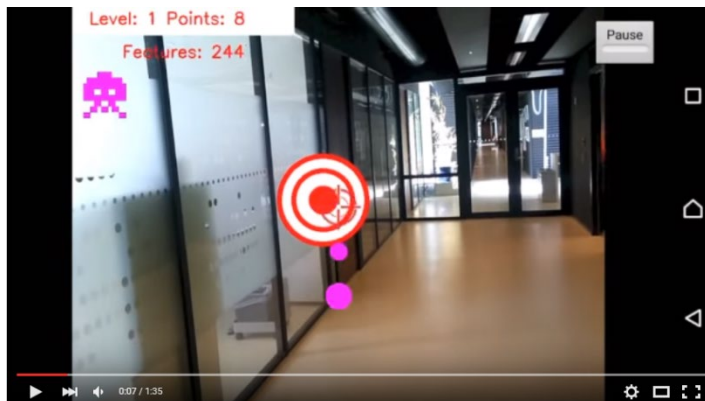
- Purpose
 - Get as much as possible out of a day at Kjeller
- Online
 - Pre-recorded lectures most weeks
- Fridays 09:15-12:00
 - ~20 min lecture summary and questions
 - ~2.5 hours programming lab



```
1 #include "opencv2/highgui.hpp"
2 #include <iostream>
3
4 int main()
5 {
6     cv::VideoCapture input_stream(0);
7
8     if (!input_stream.isOpened())
9     {
10         std::cerr << "Could not open camera\n";
11         return EXIT_FAILURE;
12     }
13
14     const std::string window_title = "Lab 0: Introduction to OpenCV";
15     cv::namedWindow(window_title, cv::WINDOW_NORMAL);
16
17     cv::Mat frame;
18
19     while(true)
20     {
21         input_stream >> frame;
22
23         if (frame.empty())
24             { break; }
25
26         cv::imshow("cam", frame);
27
28         if (cv::waitKey(15) >= 0)
29             { break; }
30     }
31
32     return EXIT_SUCCESS;
33 }
34
```

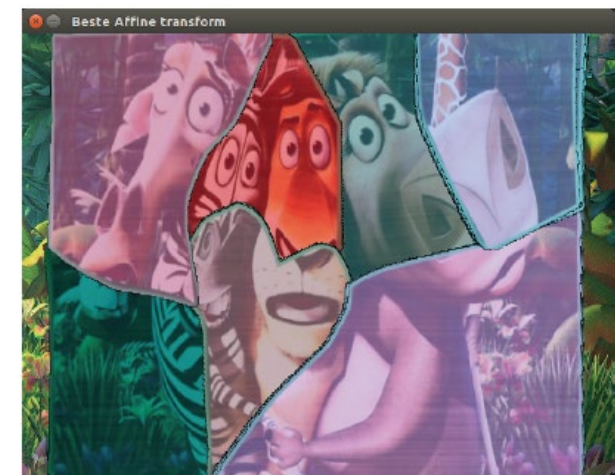
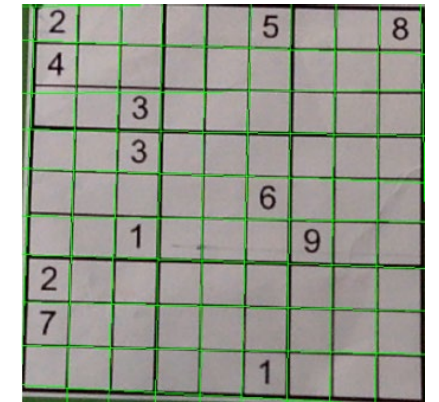
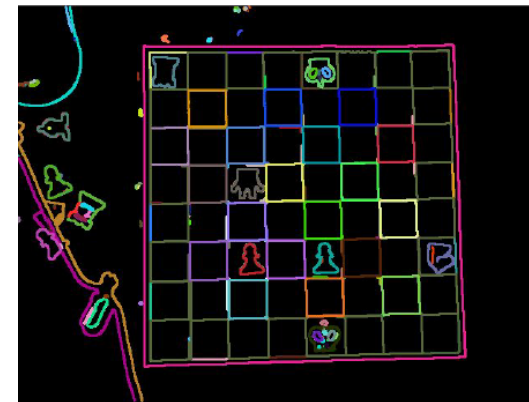

Student projects

- Develop a working computer vision system that does something interesting
 - Big: More than a month
 - Approved/Not approved
- Project topic of your own choice
- Preferably in groups of up to 3 persons



Student projects

- Develop a working computer vision system that does something interesting
 - Big: More than a month
 - Approved/Not approved
- Project topic of your own choice
- Preferably in groups of up to 3 persons



Exam

- Written or oral exam
- Mid June



Feedback

- We encourage feedback during the course
 - We are open to making adjustments!
- Please fill out and deliver the course evaluation form after the course!
- Any feedback or questions now?

