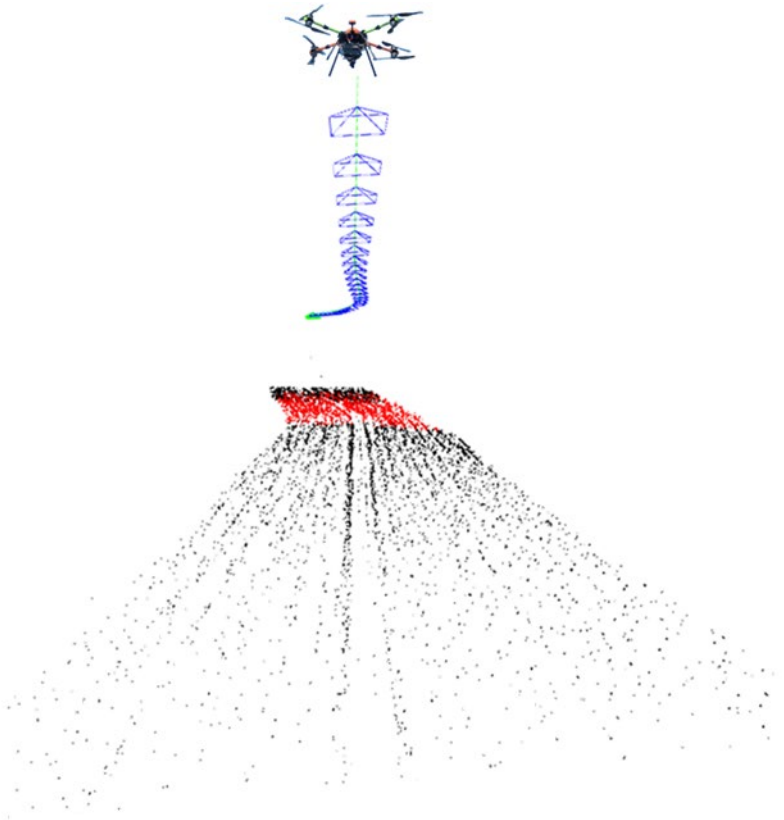
UiO : Department of Technology Systems
University of Oslo

Introduction to TEK5030 – Computer Vision

22.01.2025

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)



FFI Norwegian Defence Research Establishment

Situational awareness across collaborating autonomous systems

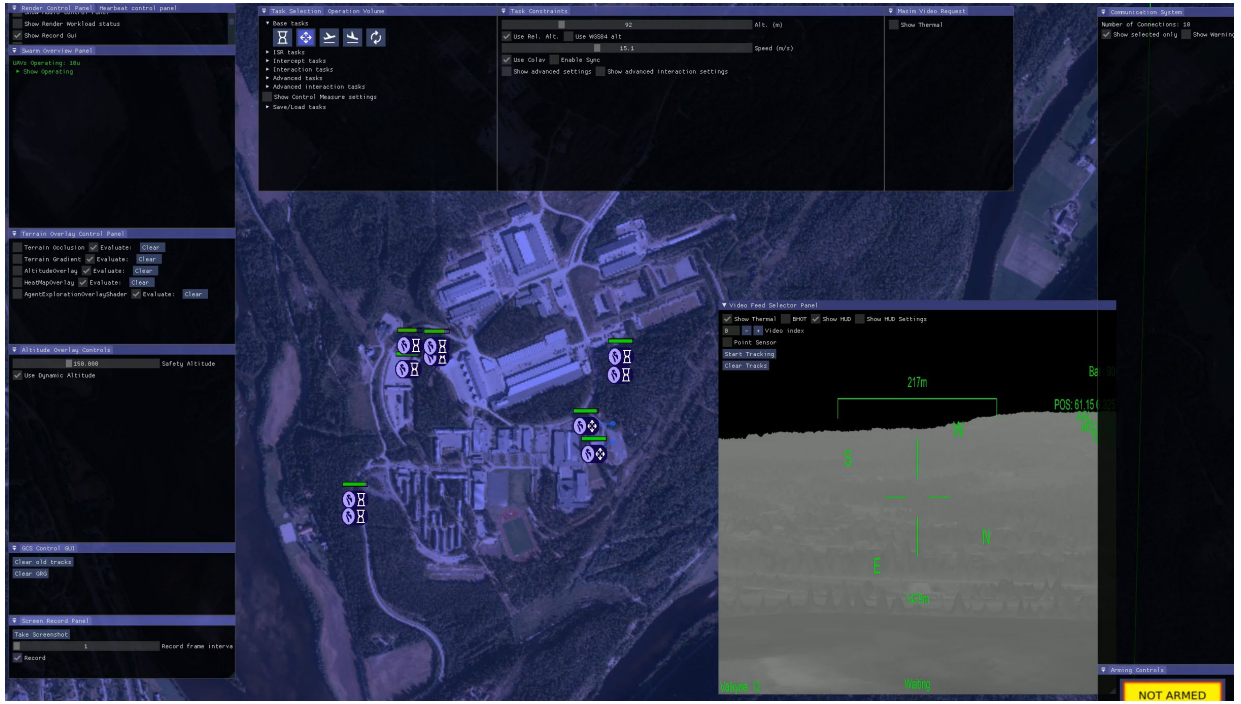
My day job at FFI

The autonomy project









Is this situational awareness?





Is this situational awareness?

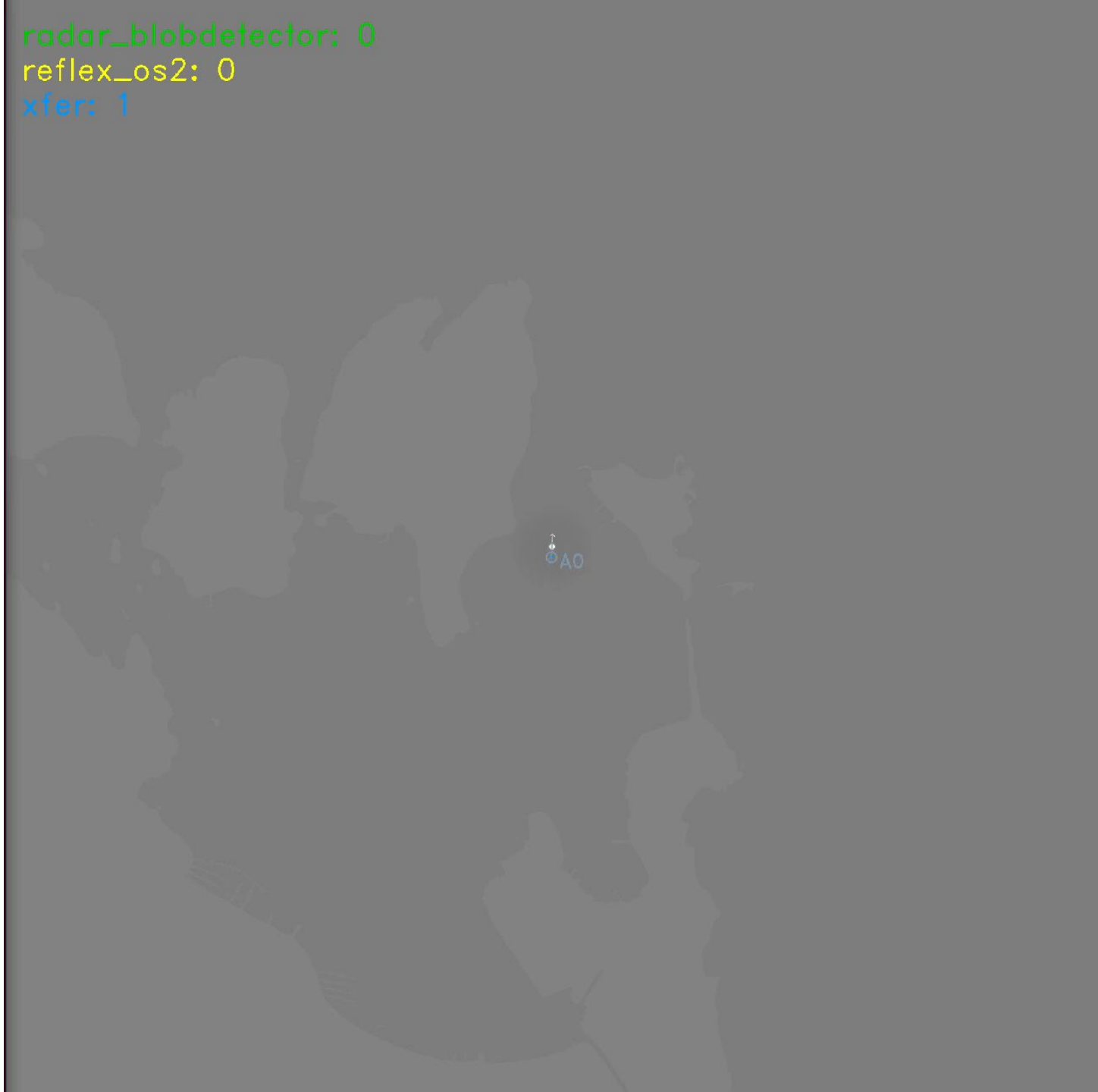








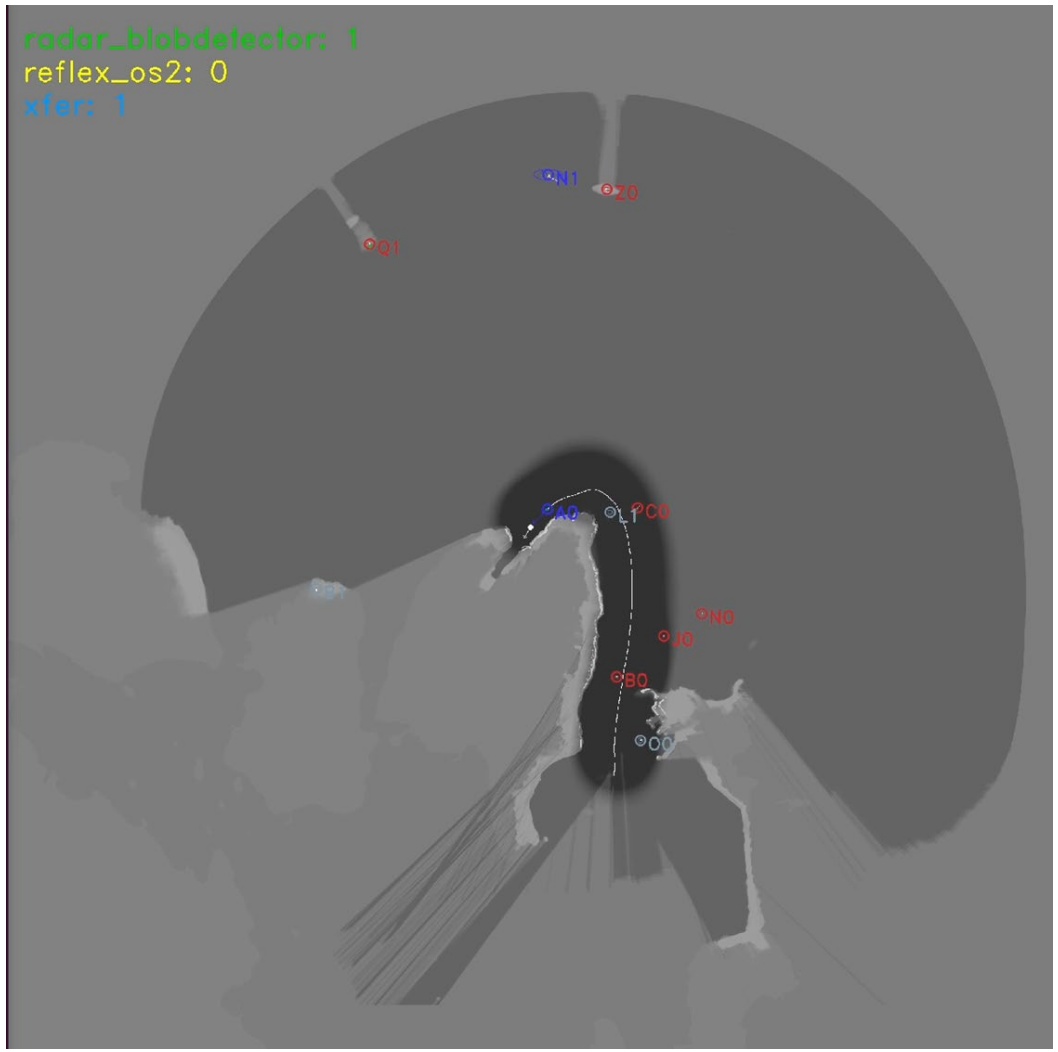
What about this?



The situational picture is a set of *states*

represented as probability distributions

$$X = \left\{ \right.$$

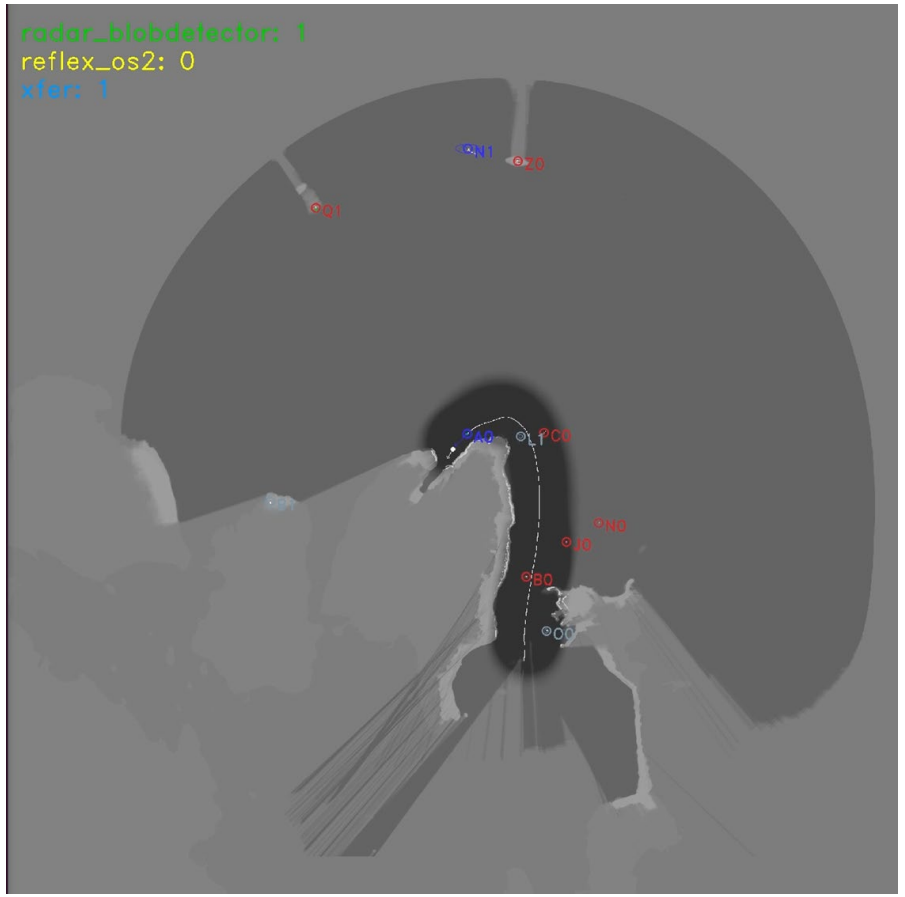


$$\left. \right\}$$

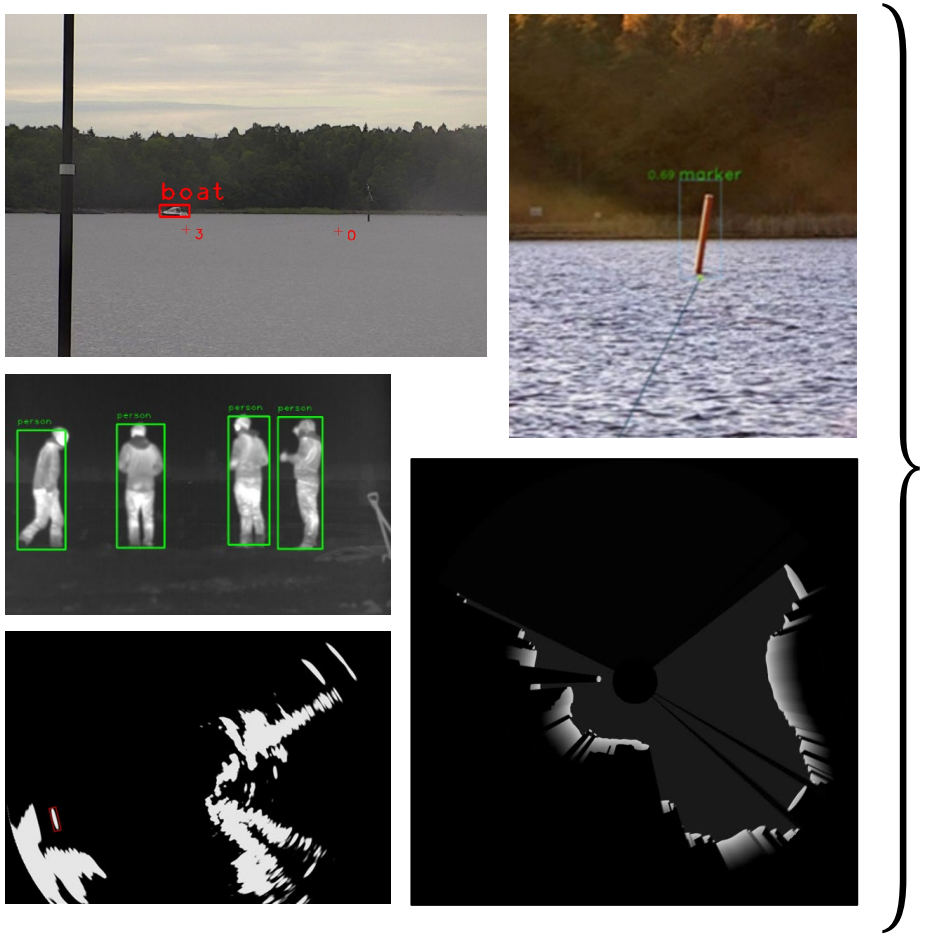
We *estimate* states from *measurements*

so that the states fit as good as possible with the measurements

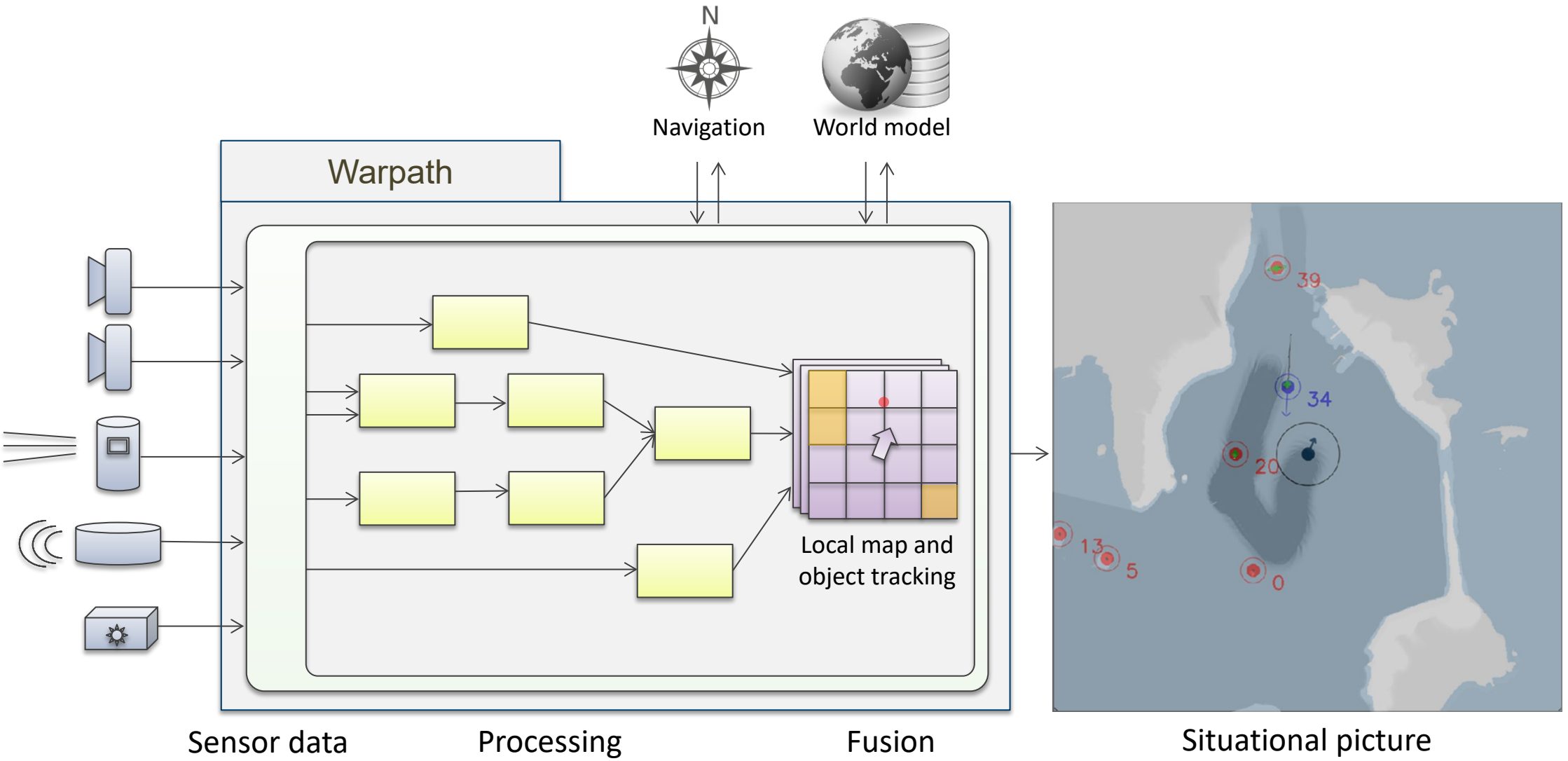
$$X = \left\{ \right.$$

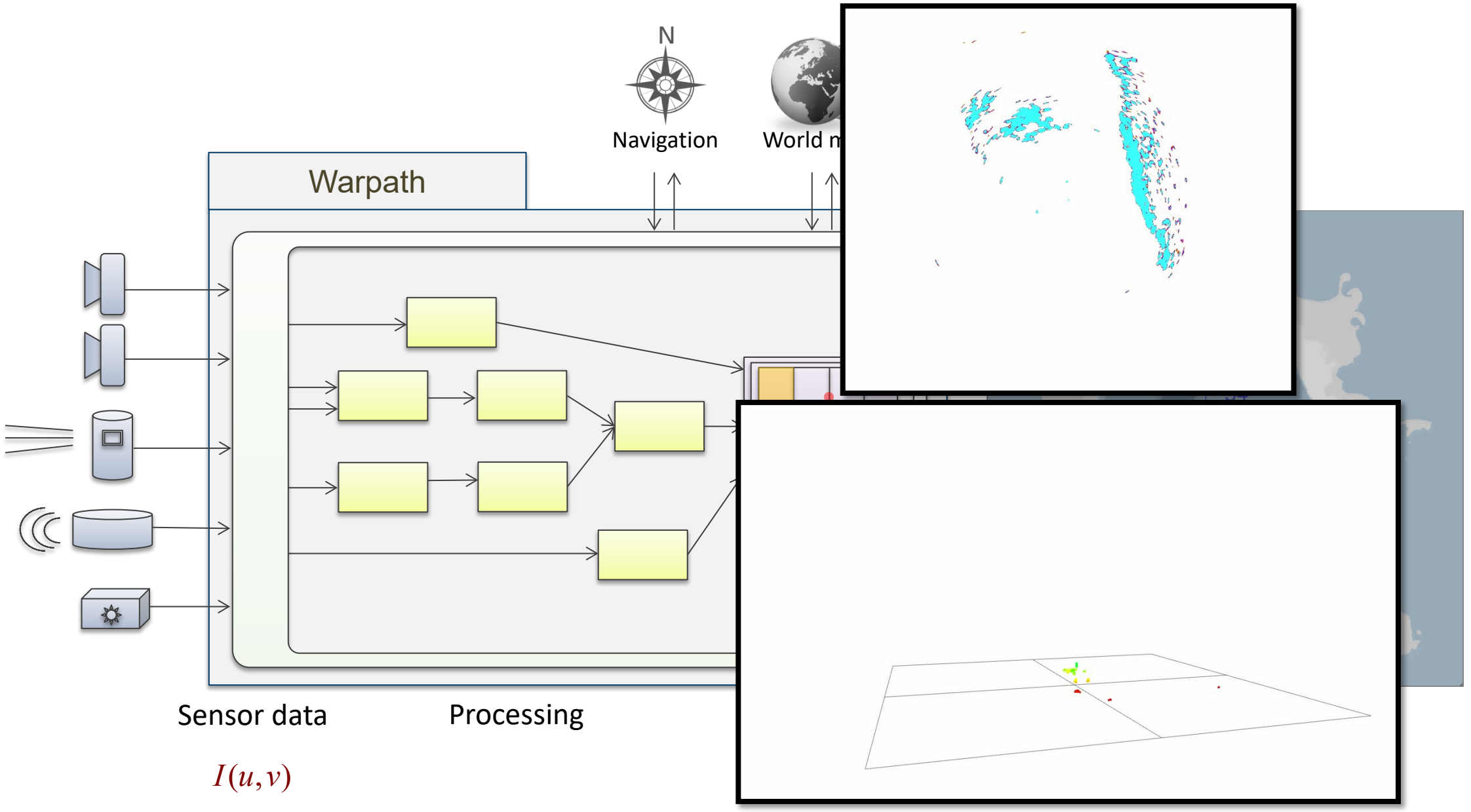


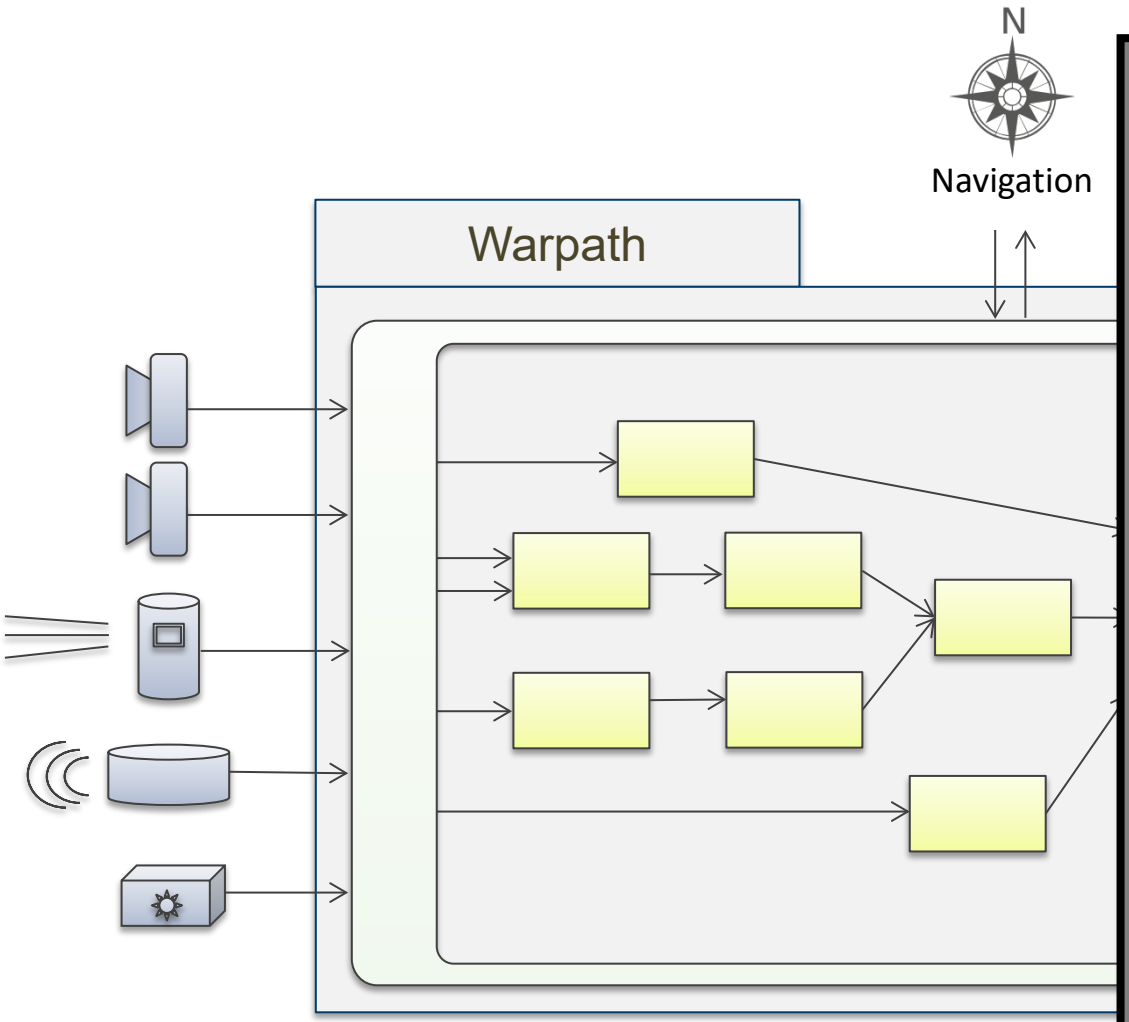
$$\left. \right\} \quad Z = \left\{ \right.$$

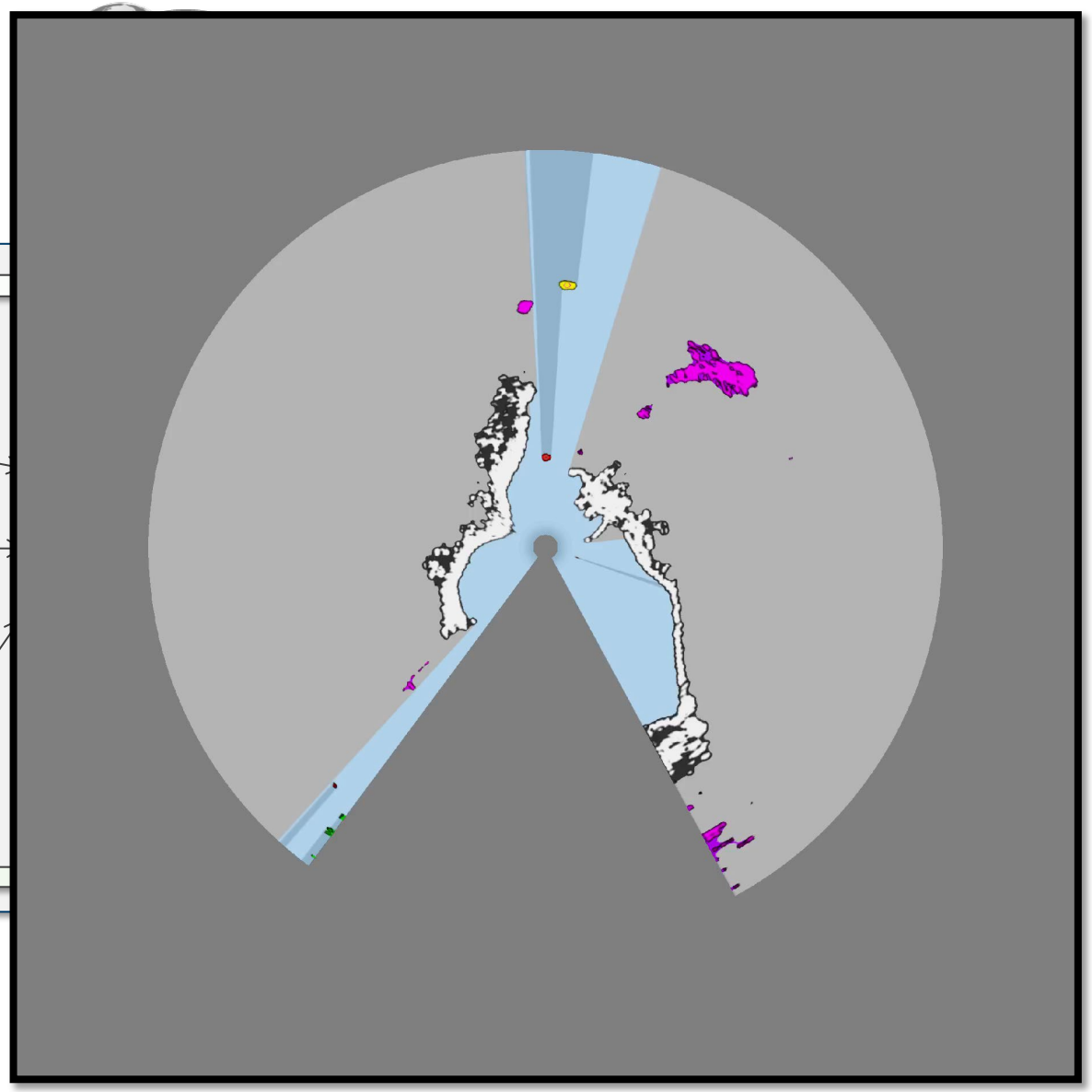


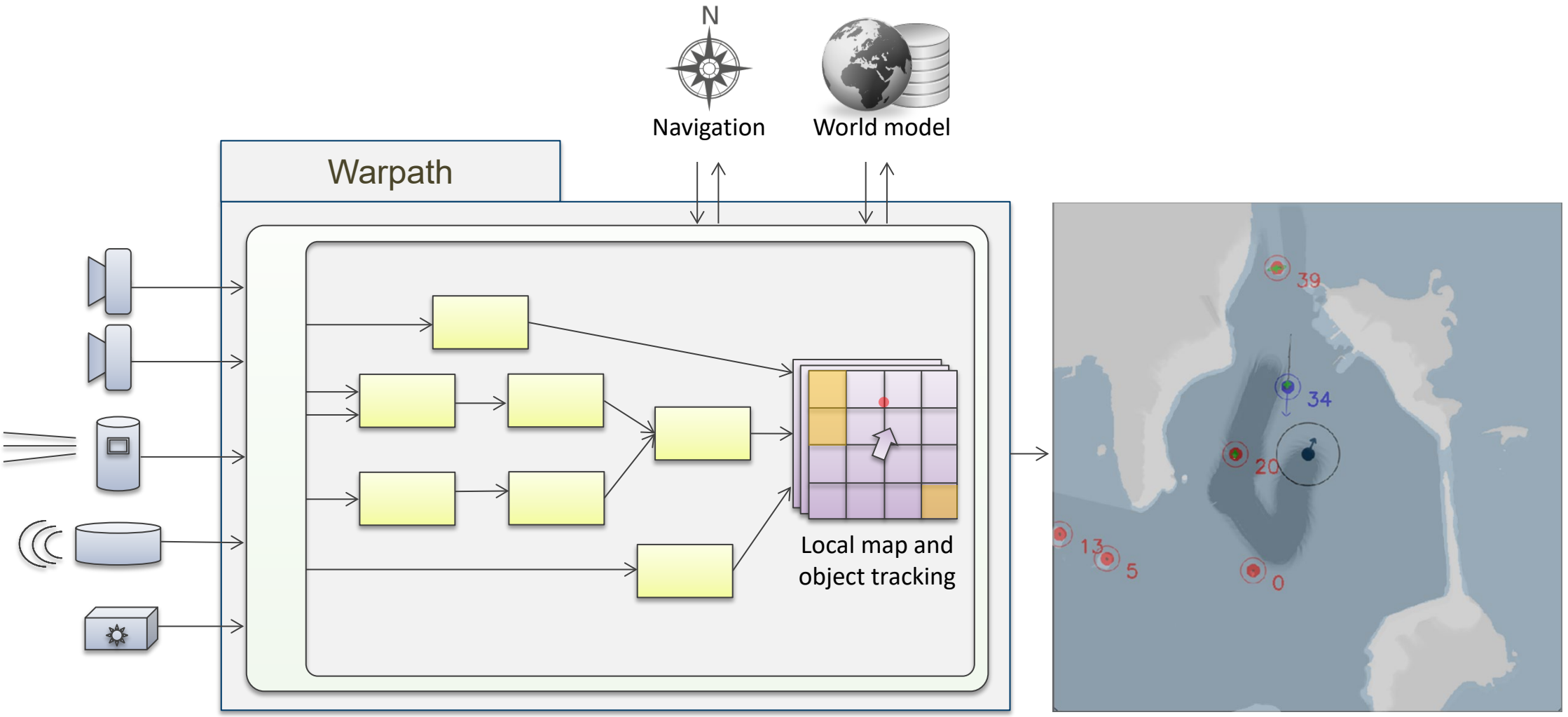
$$\left. \right\}$$


N
Navigation
World model
Warpath
Local map and
object tracking
39
34
20
13
5
0
Sensor data
Processing
Fusion
Situational picture


N
Navigation
World m
Warpath
Sensor data
Processing
$I(u,v)$


N
Navigation
Warpath
Sensor data
Processing
$I(u,v)$
$I(u,v) \rightarrow Z$





Sensor data

$I(u,v)$

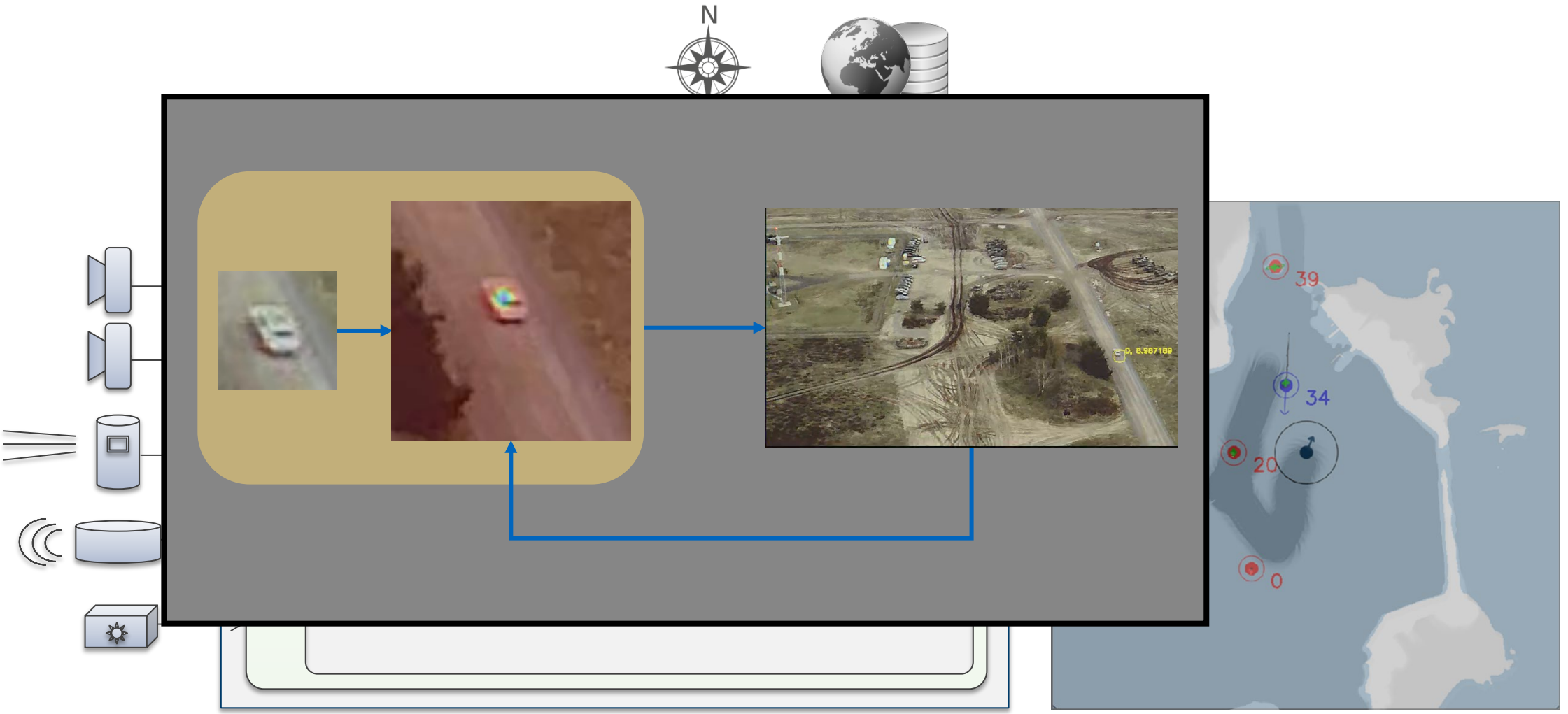
Processing

$I(u,v) \rightarrow Z$

Fusion

$$X^* = \arg\max_X p(X \mid Z)$$

Situational picture



Sensor data

$I(u,v)$

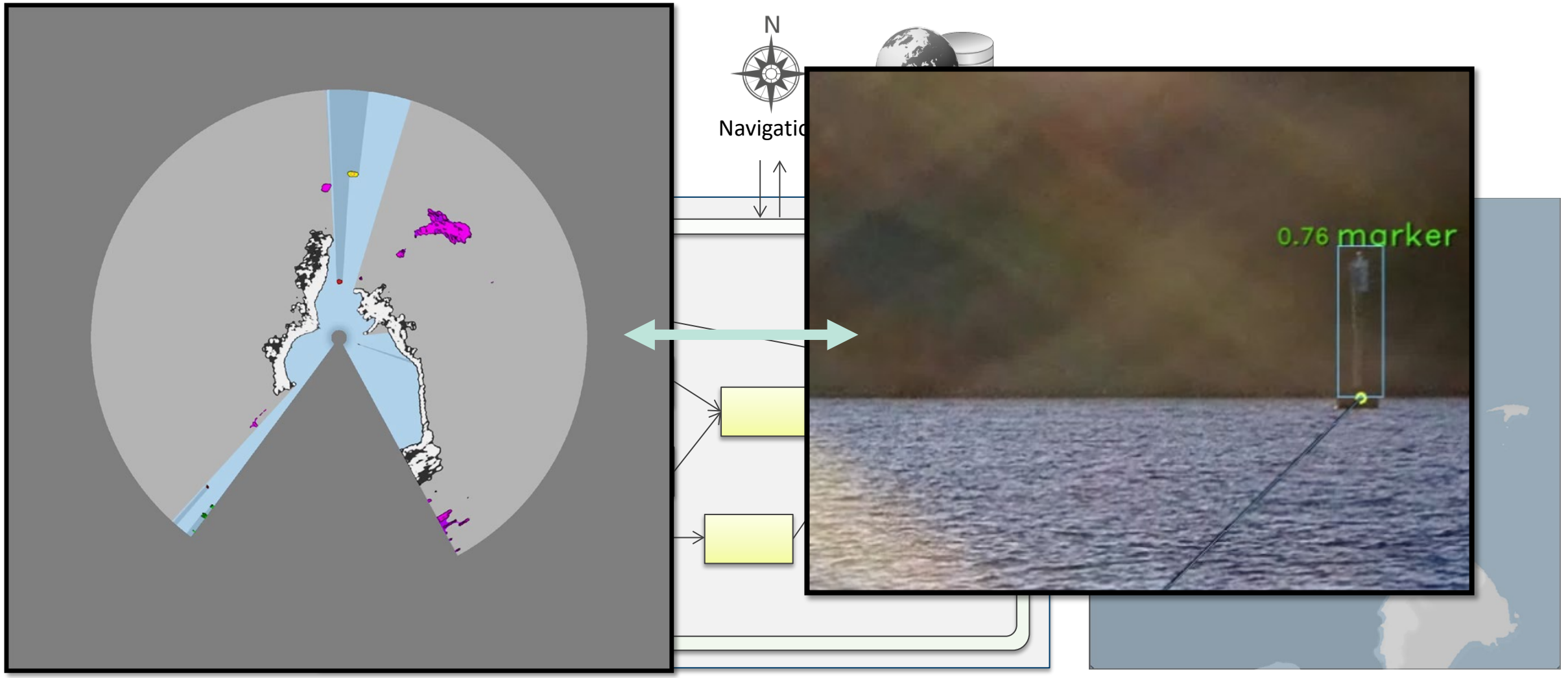
Processing

$I(u,v) \rightarrow Z$

Fusion

$$X^* = \arg\max_X p(X \mid Z)$$

Situational picture



Sensor data | Processing | Fusion | Situational picture

$I(u,v)$ $\quad I(u,v) \rightarrow Z \quad$ $X^* = \arg\max_X p(X \mid Z)$

Today

- A quick introduction to computer vision
- About the course

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



Illustration: Generated by Midjourney

«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



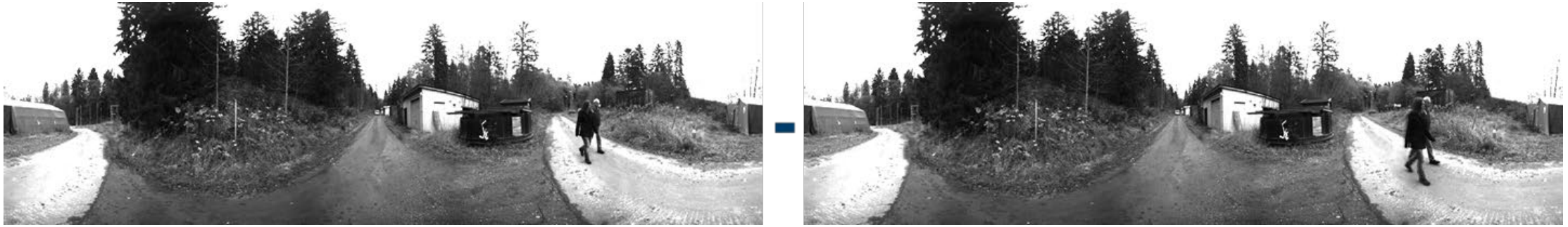
Images and pixels



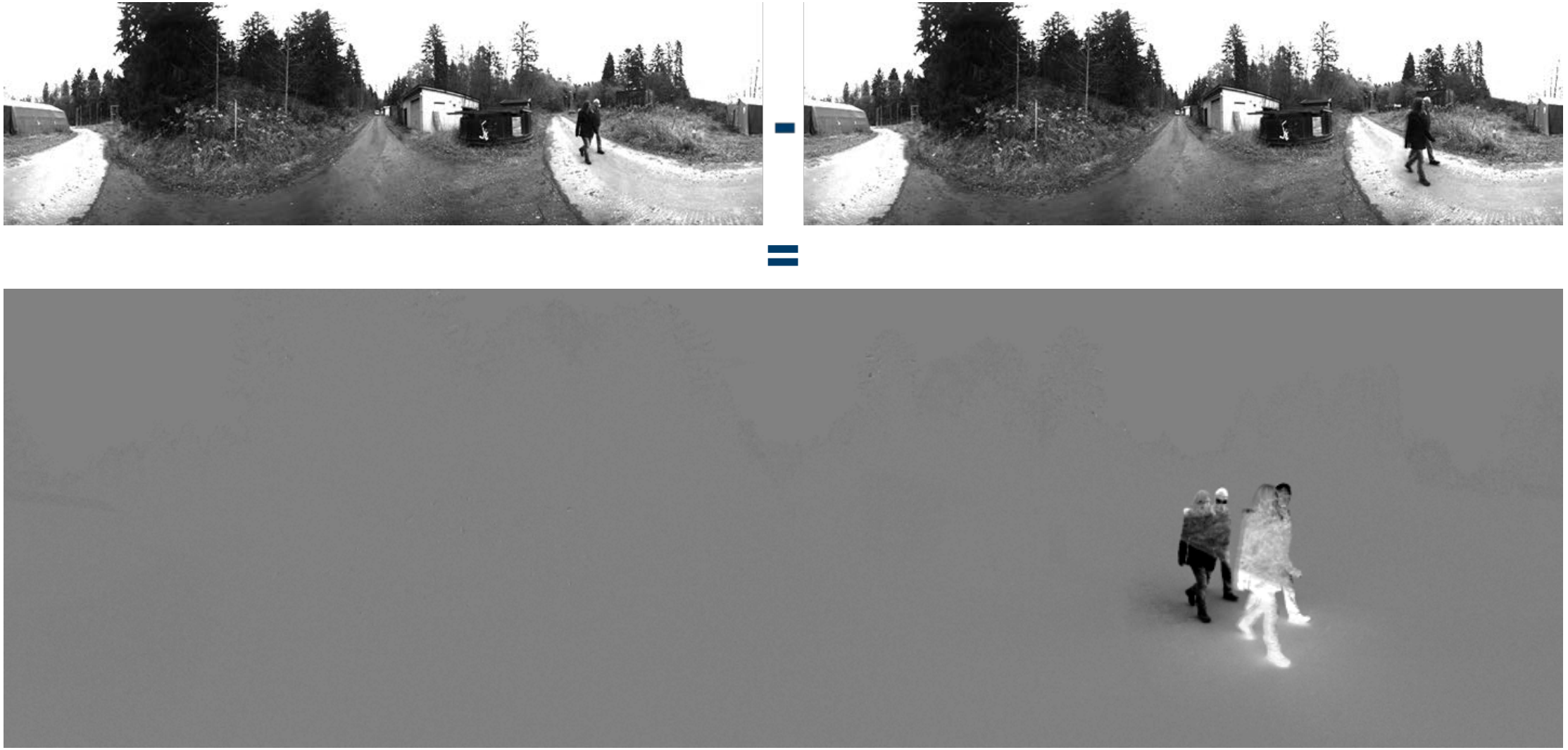
Images and pixels



It is easy to compute with images!



It is easy to compute with images!



How fast can a machine crunch pixel values?



Illustration: Generated by Midjourney

How fast can a machine crunch pixel values?

How many operations per second?

- Thousands?
- Millions?
- Billions?



Illustration: Generated by Midjourney

How fast can a machine crunch pixel values?

How many operations per second?

- 50 billion?
- 200 billion?
- 1000 billion?
- 3000 billion?
- 1000000000 billion?



Illustration: Generated by Midjourney

Floating point operations (FP32) per second (FLOPS)



Photo: Miiicihiaieil Hieinizilieir (CC BY-SA 4.0)

Raspberry PI 4B
48 GFLOPS

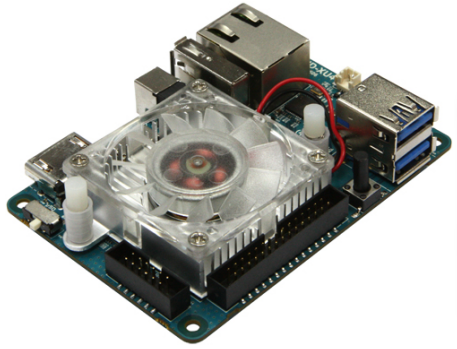


Photo: Alexandre.willame (CC BY-SA 4.0)

Odroid XU-4
214 GFLOPS

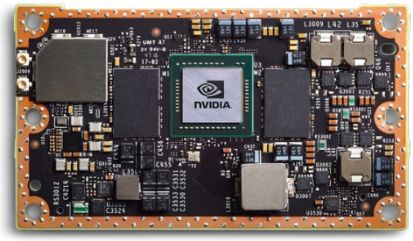


Photo: NVIDIA

Jetson TX2
782 GFLOPS



Photo: HP

My laptop
3 TFLOPS

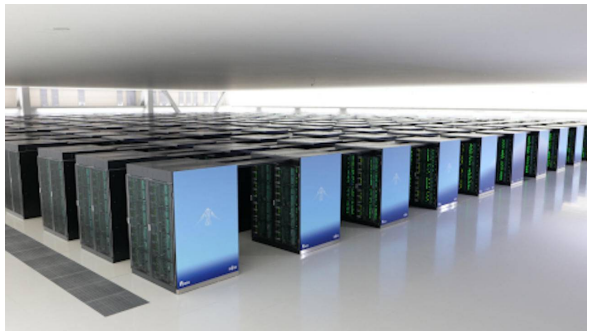
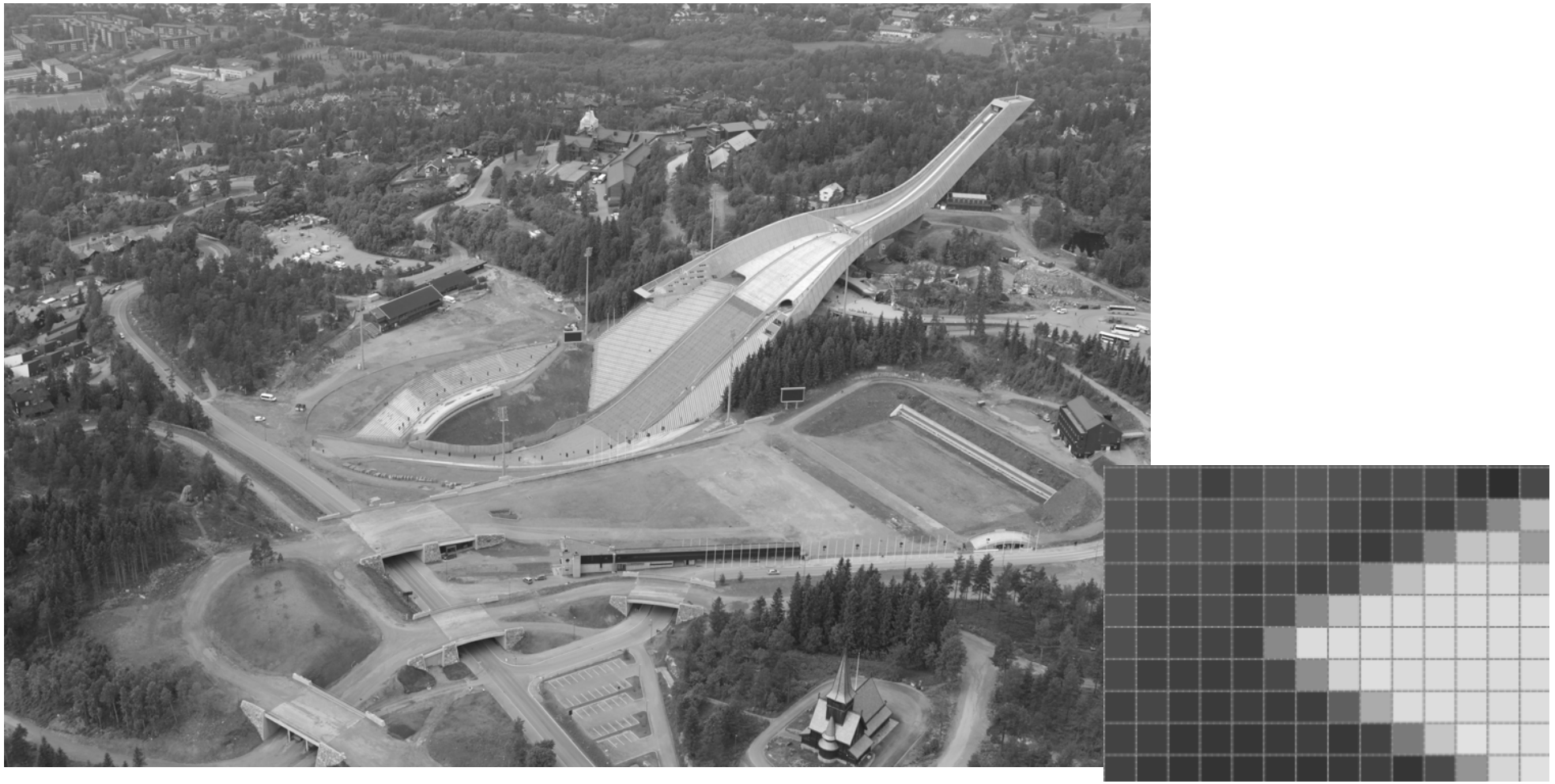


Photo: RIKEN

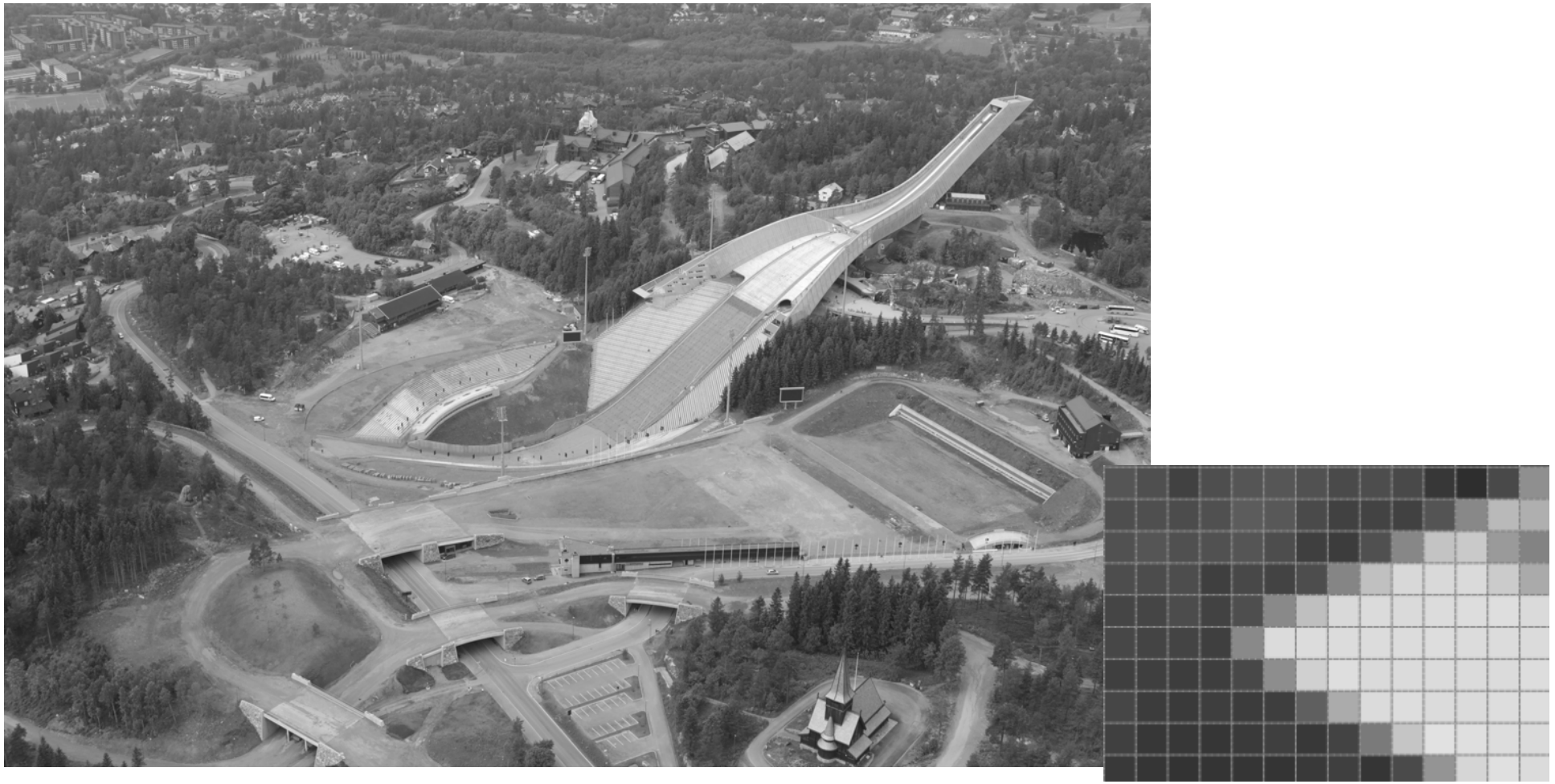
Fugaku supercomputer
>1000 PFLOPS

(theoretical *peak*)

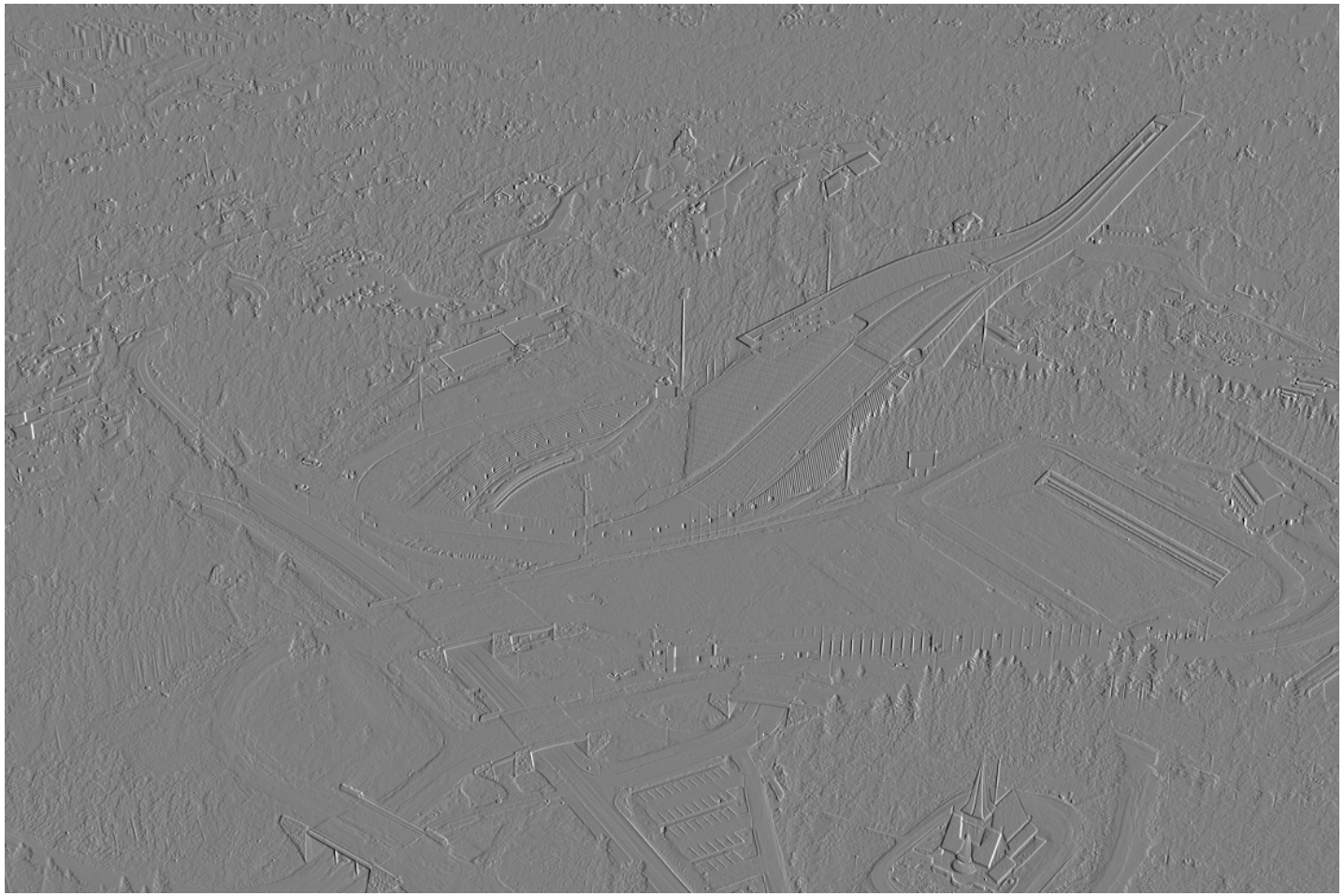
The difference between neighbouring pixels



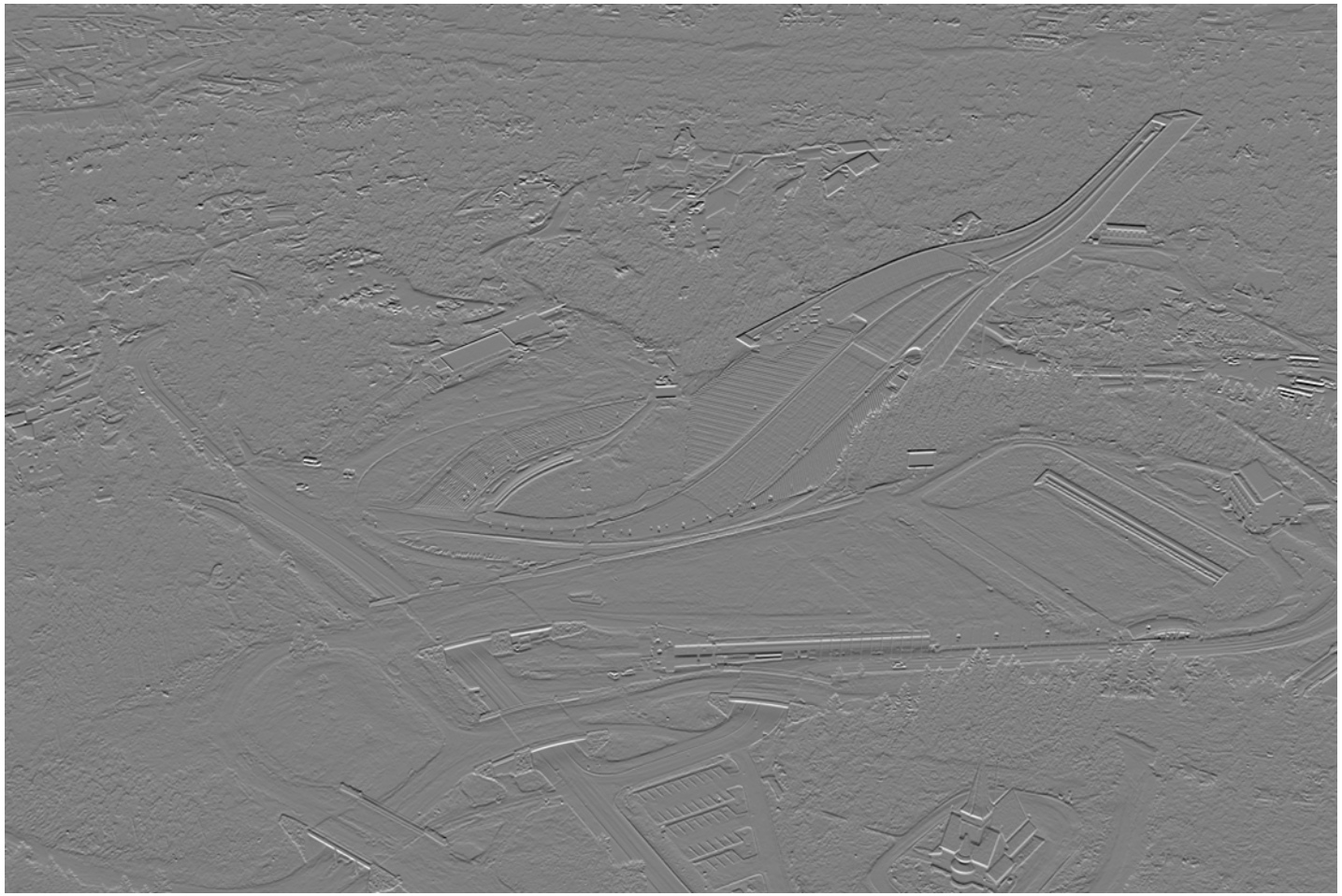
The difference between neighbouring pixels



Horizontal differences



Vertical differences



Edges and corners



Significant corners



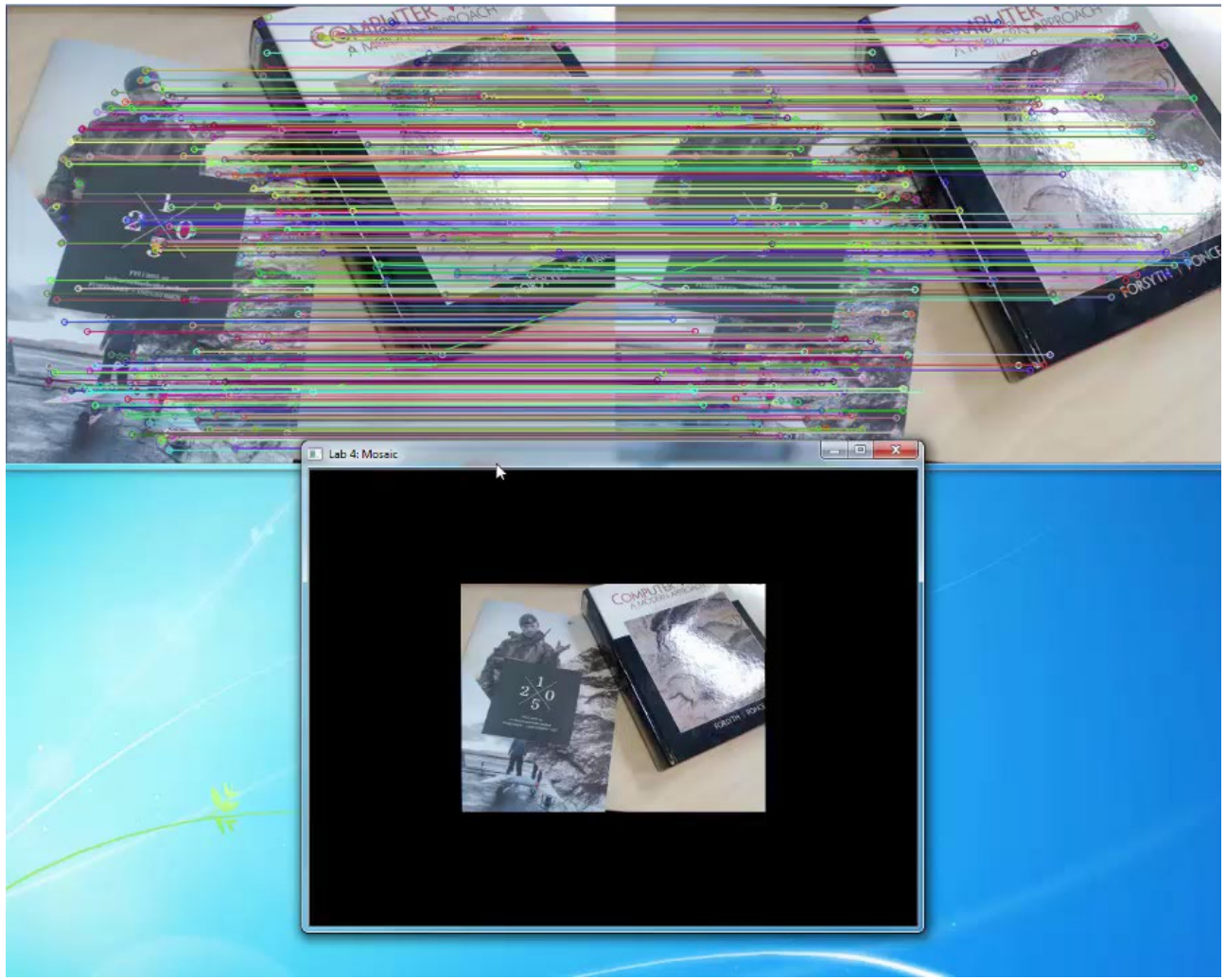
Significant corners



Significant corners

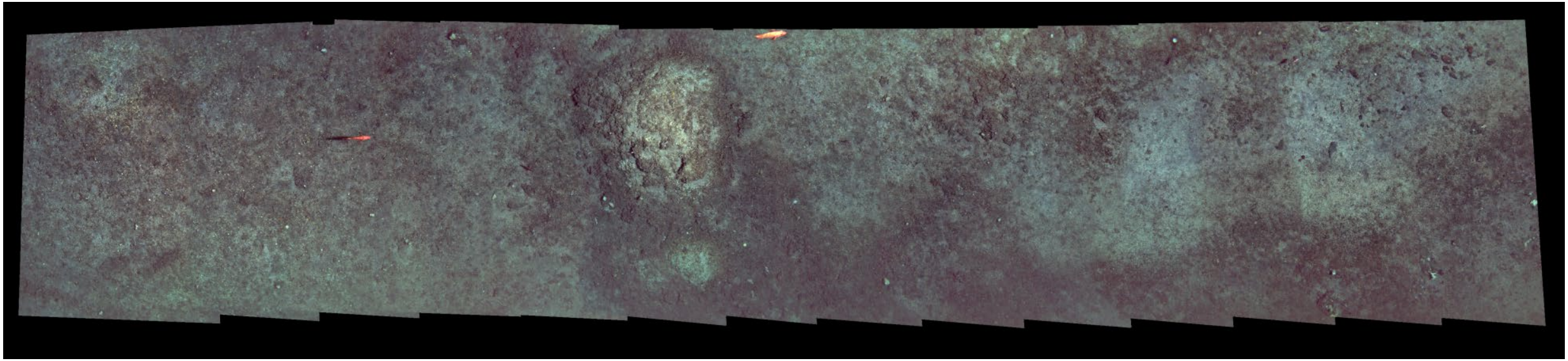


Example: Coregistering images

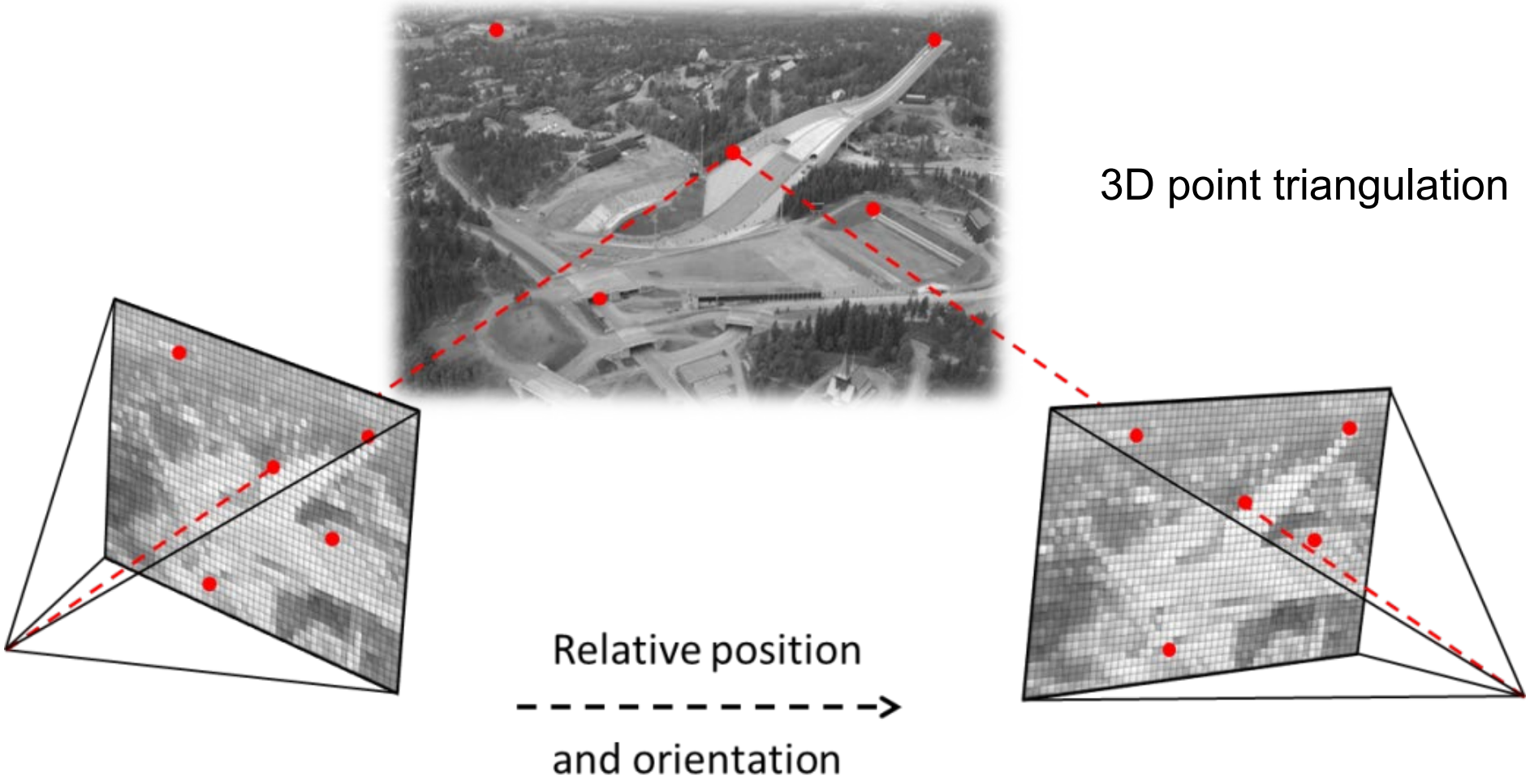


Application: Underwater visual mapping from HUGIN AUV

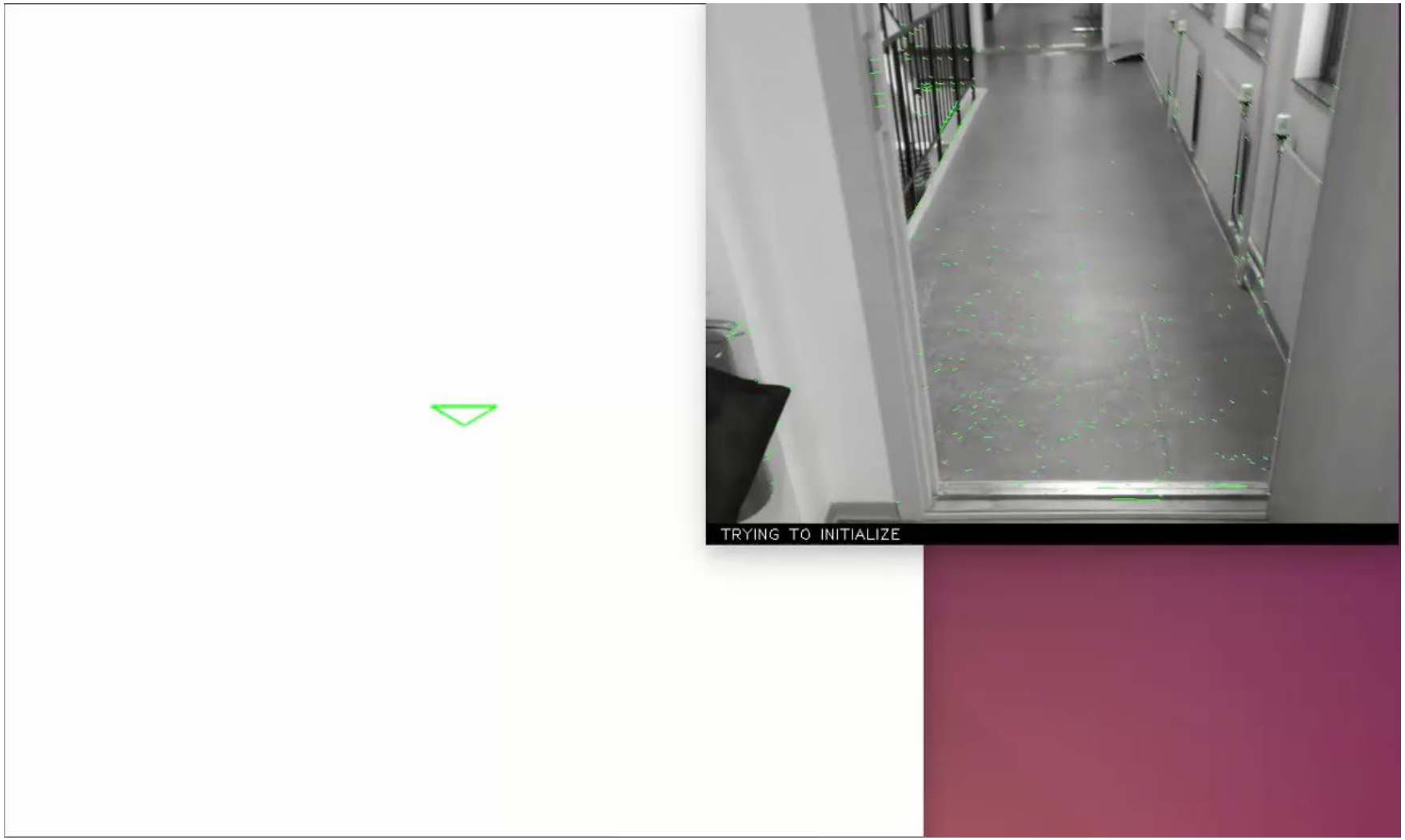




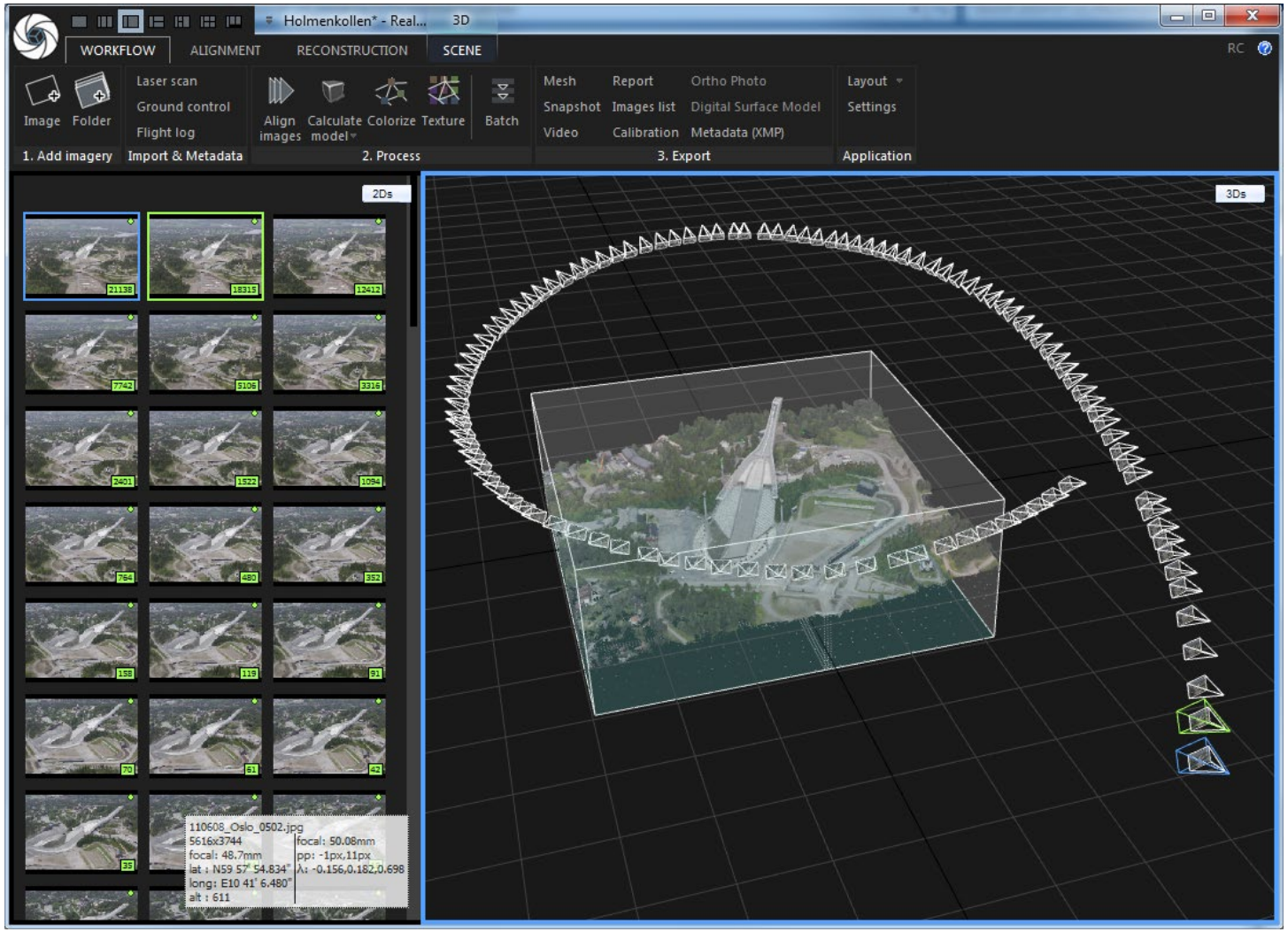
Extracting geometric information from images



Visual navigation



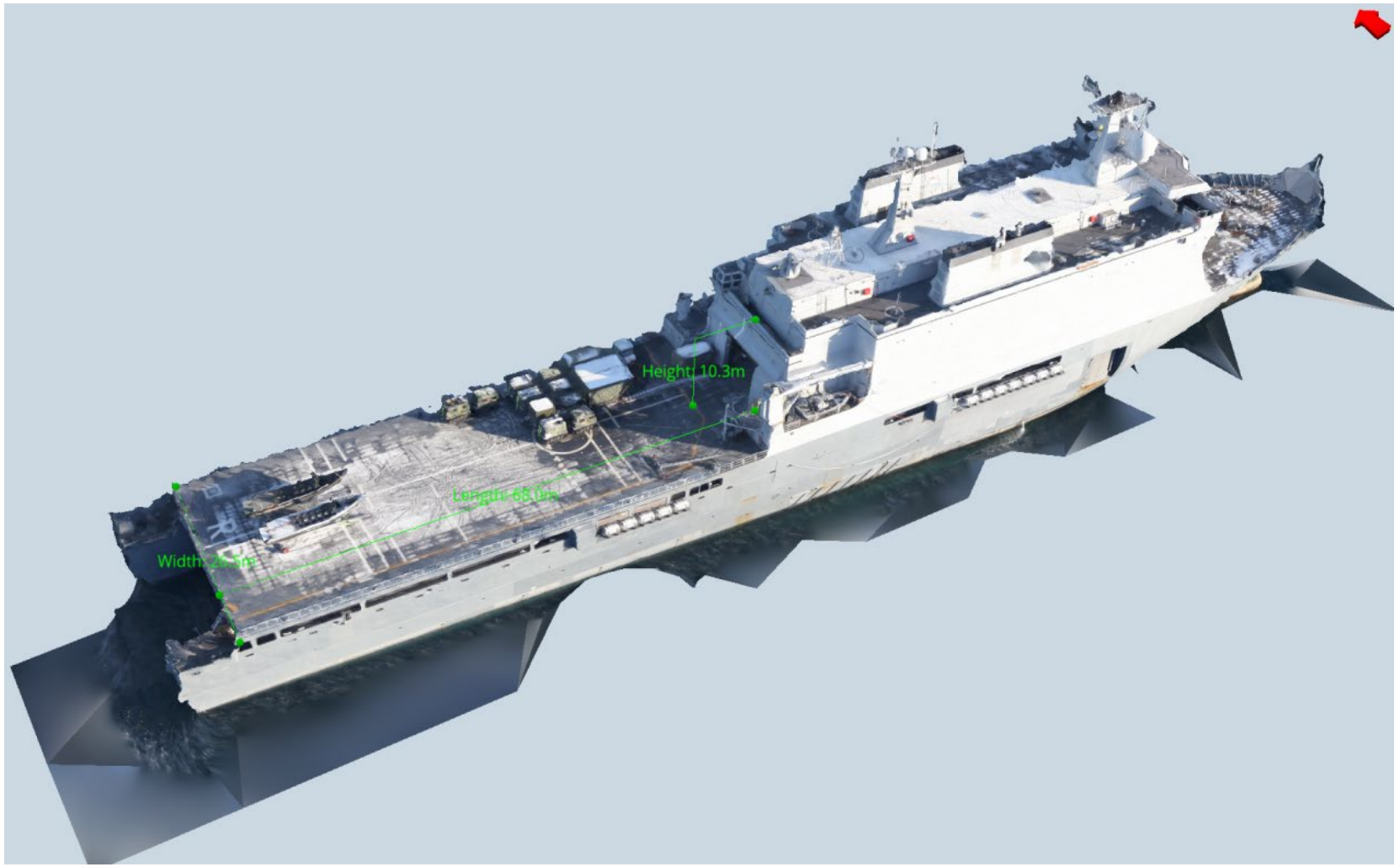
3D reconstruction from images



A detailed 3D surface in colours!

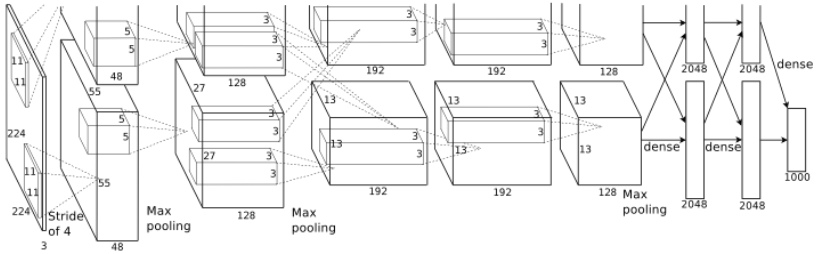


Recognise the shape of a ship



Let the machine learn a better representation itself!

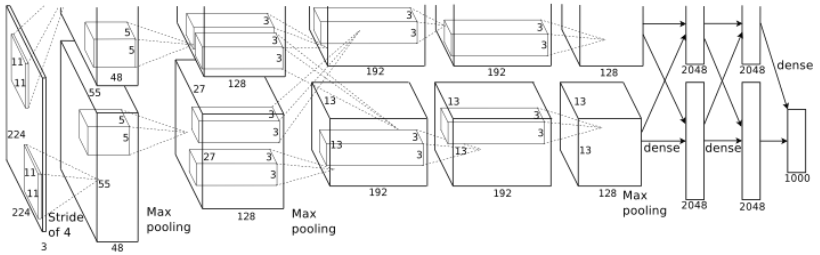




«Warship»

Let the machine learn a better representation itself!

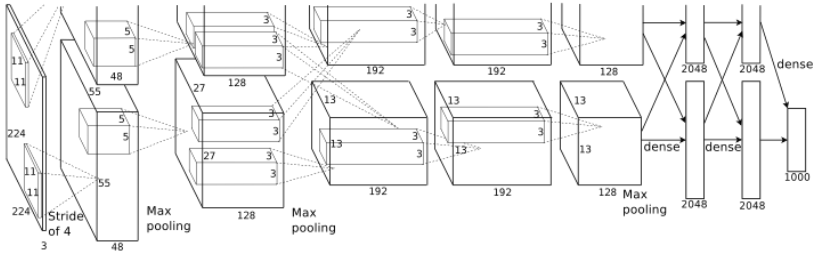




«Warship»

Let the machine learn a better representation itself!



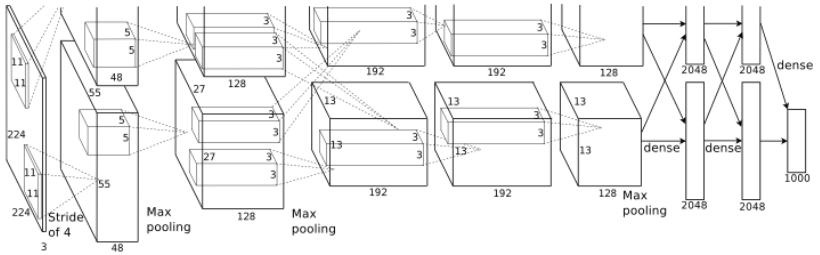


«CV90»

Let the machine learn a better representation itself!

Many millions of
learned parameters

Millions
of images

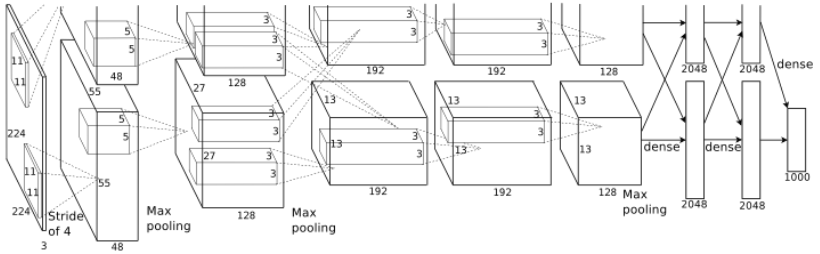


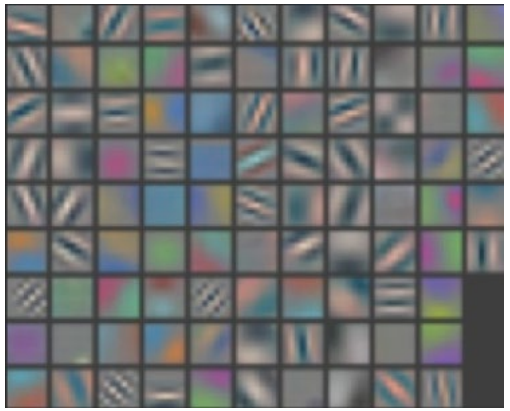
Millions
of labels

Let the machine learn a better representation itself!

Many millions of
learned parameters

Millions
of images



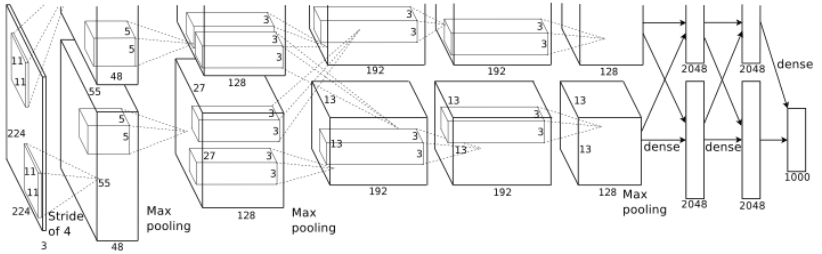


Millions
of labels

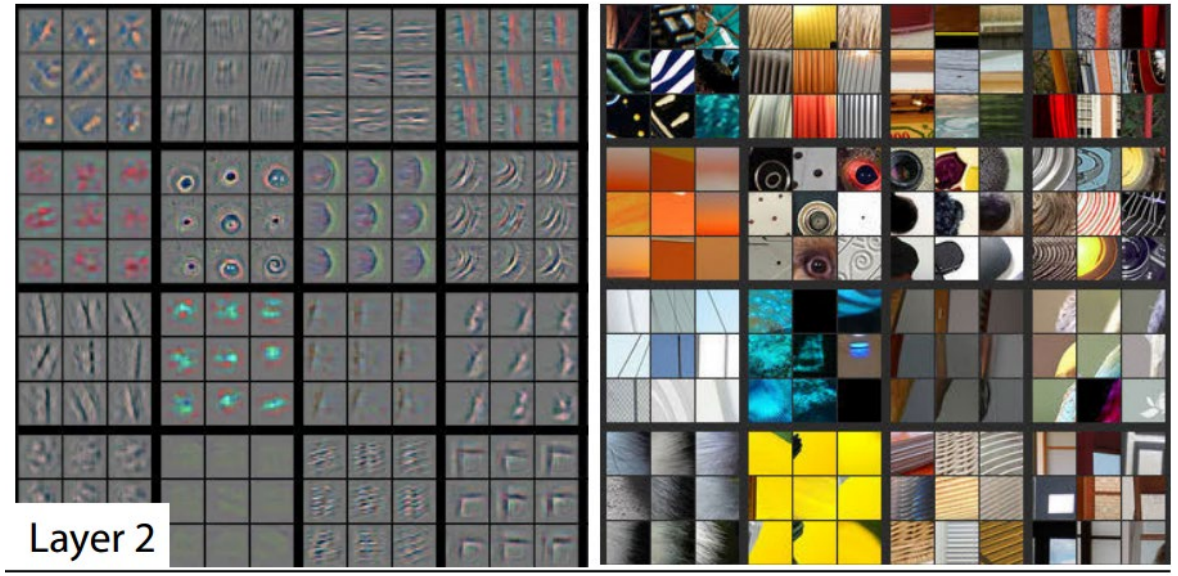
Let the machine learn a better representation itself!

Many millions of
learned parameters

Millions
of images



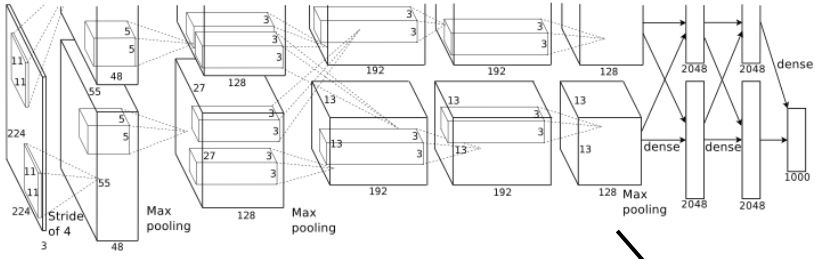
Millions
of labels



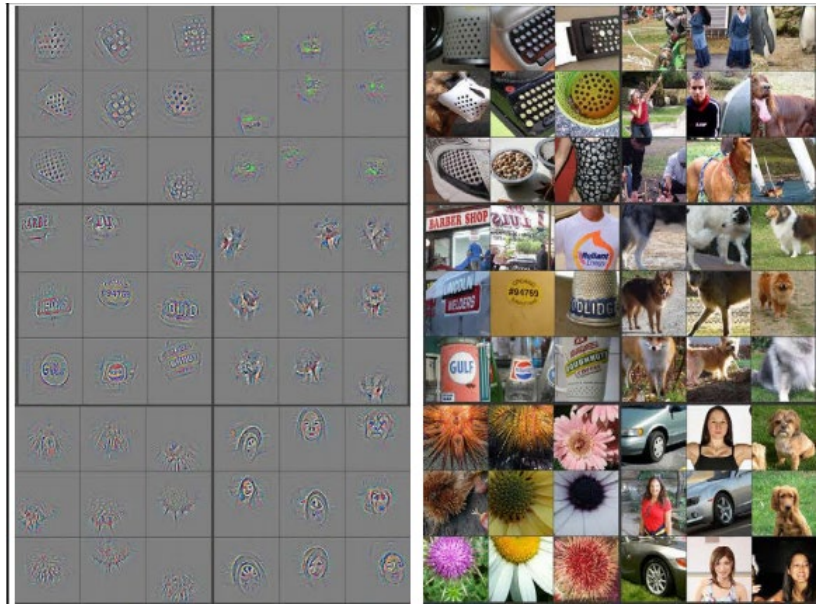
Let the machine learn a better representation itself!

Many millions of
learned parameters

Millions
of images



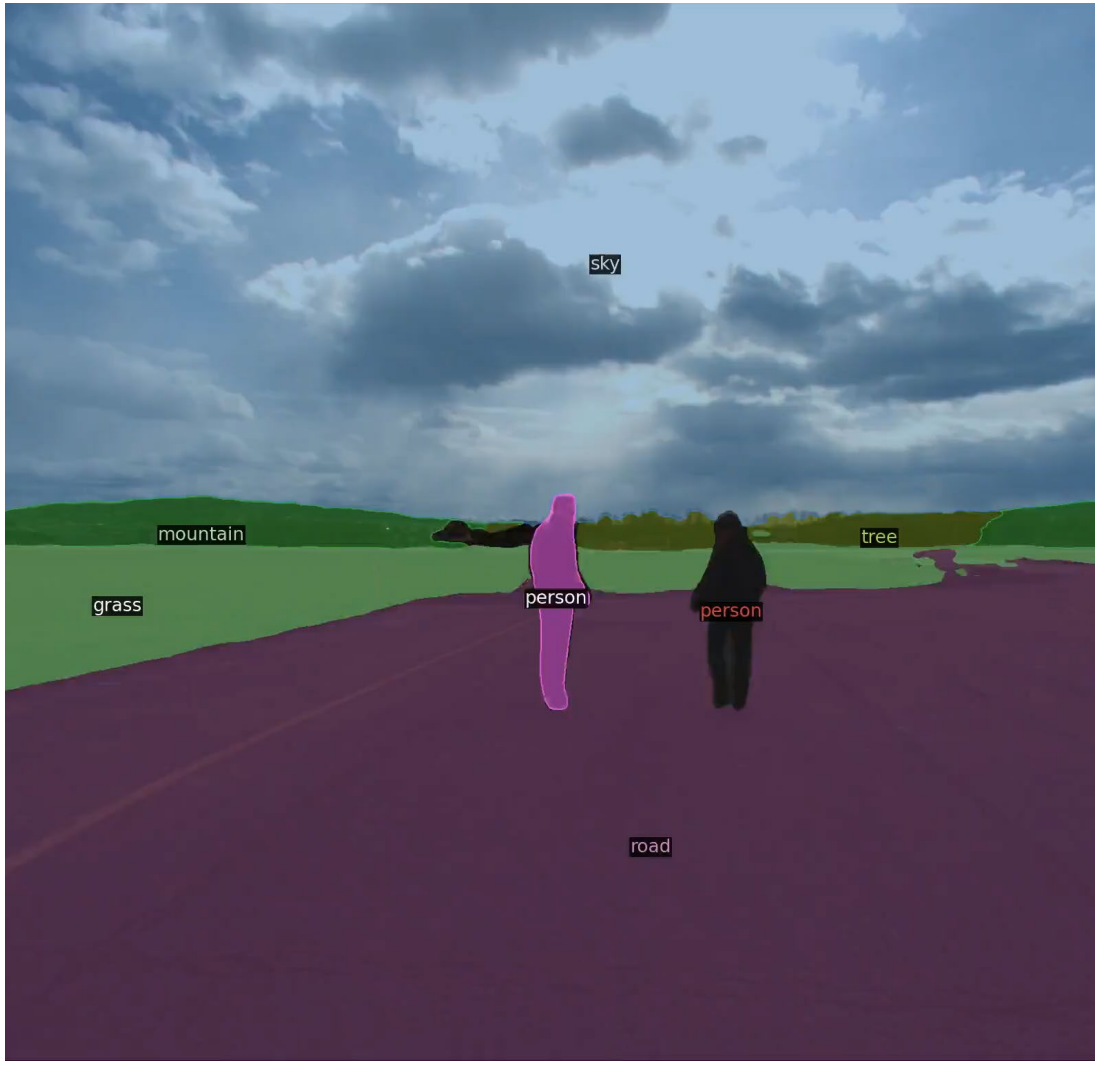
Millions
of labels




0.76 marker


boat
+3
+0

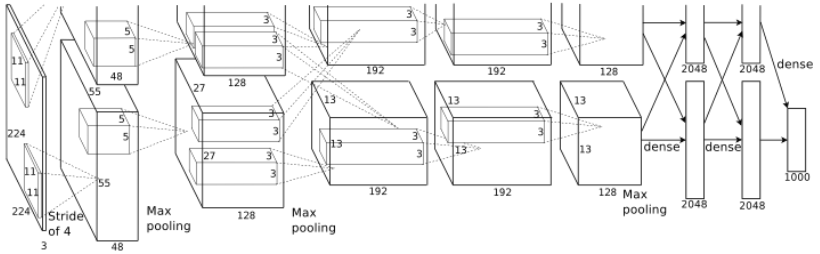



sky
mountain
tree
grass
person
person
road

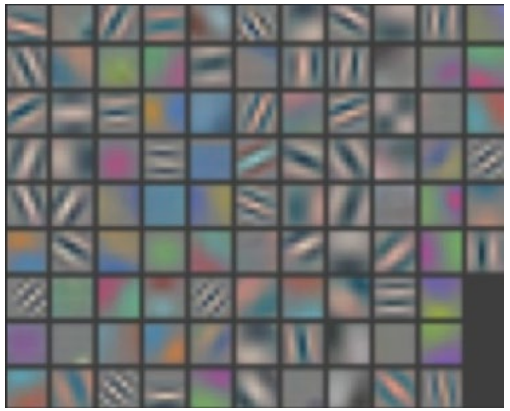
Convolutional Neural Networks (CNNs)

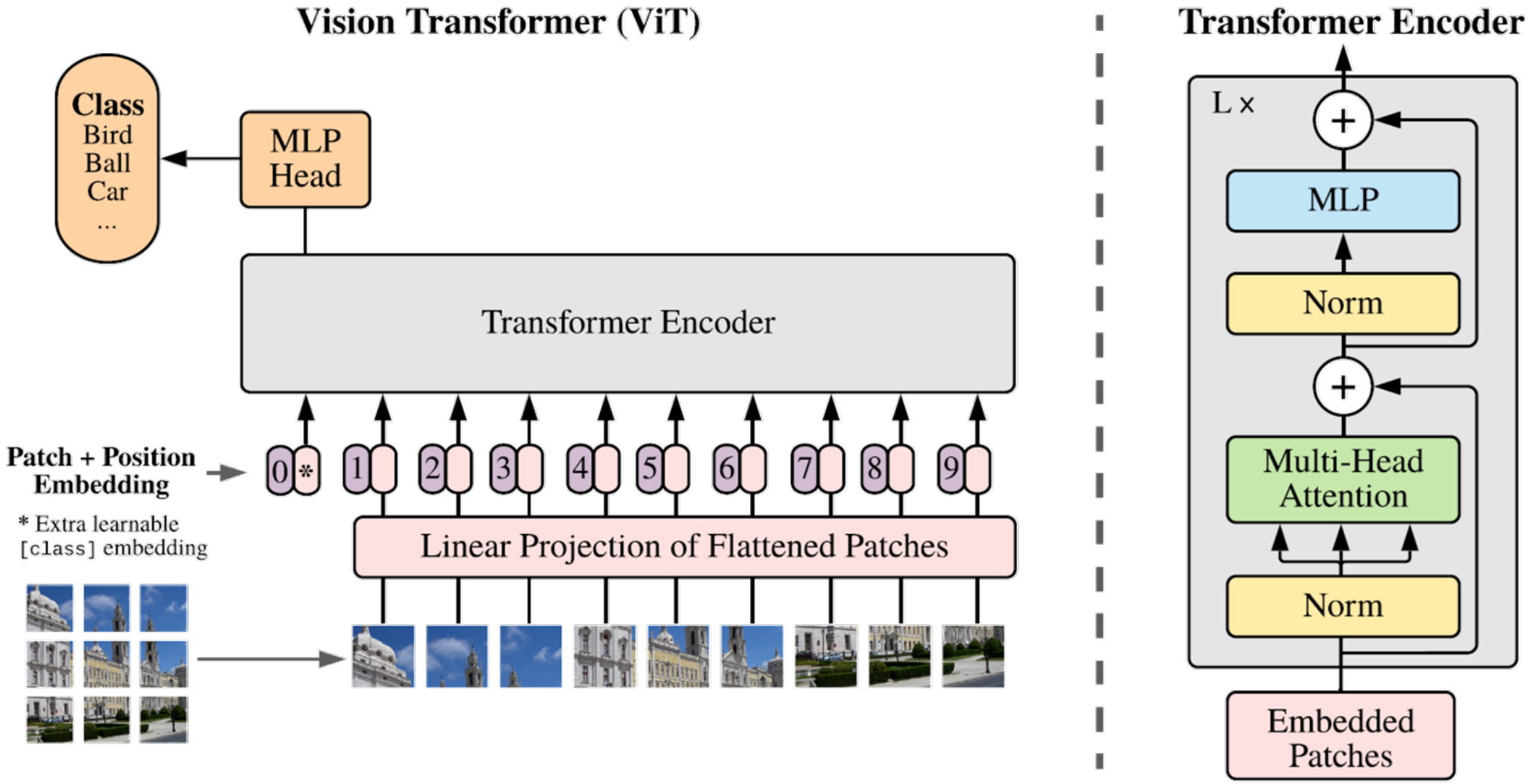
Many millions of
learned parameters

Millions
of images

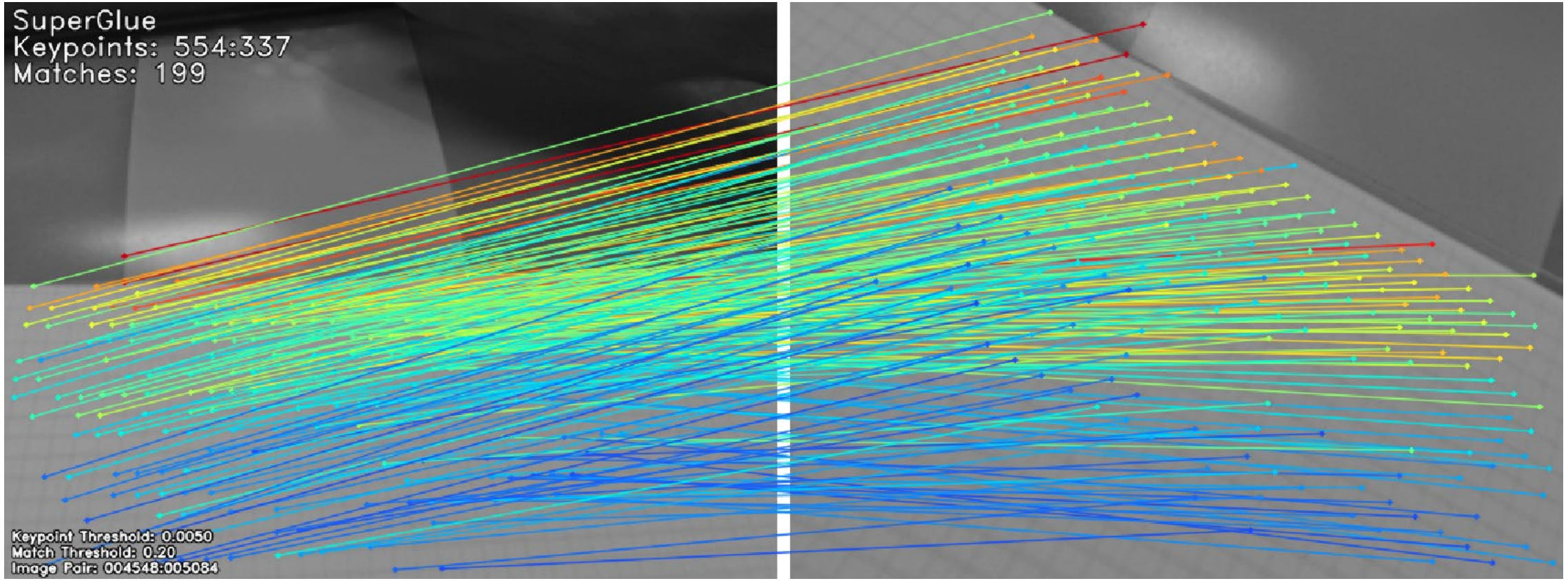


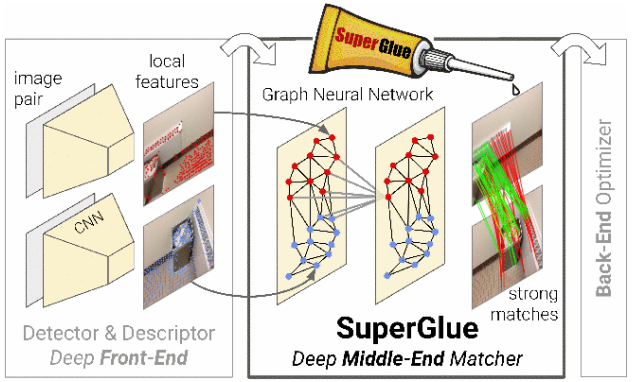
Millions
of labels





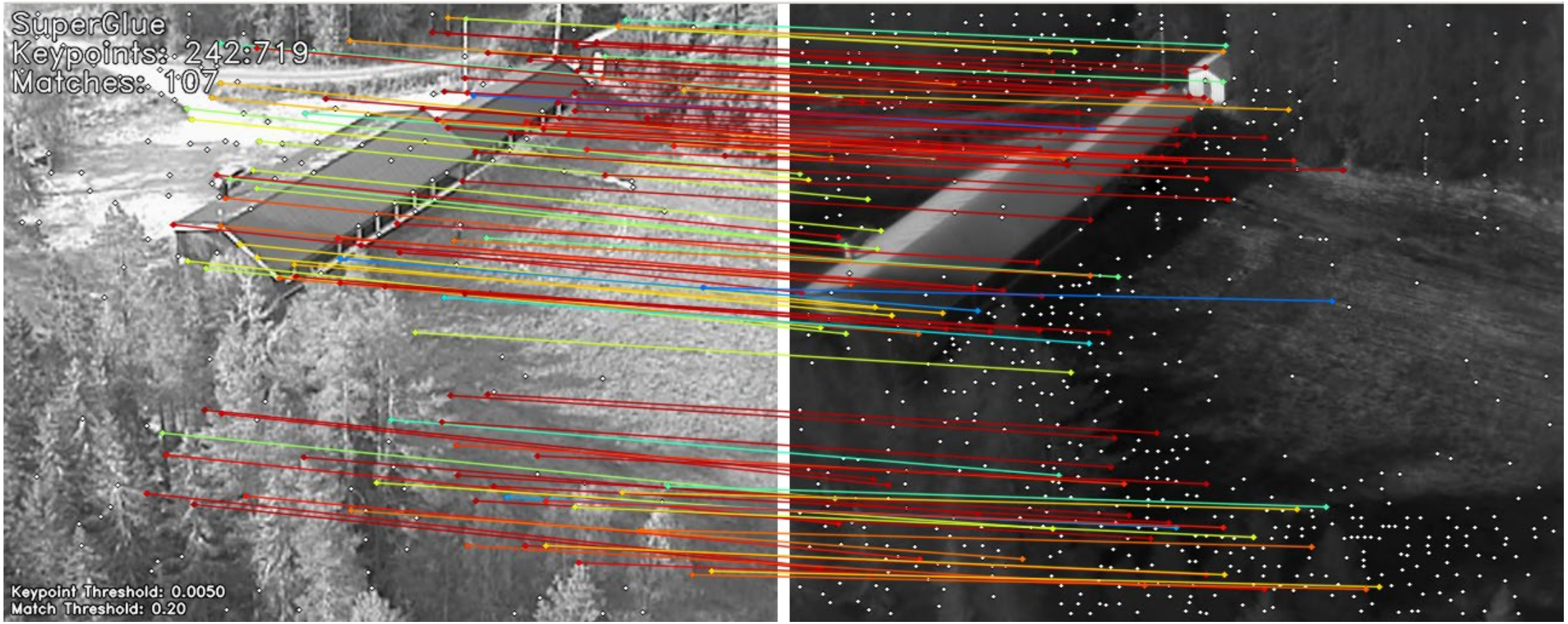
A. Dosovitskiy et al., 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale', Jun. 03, 2021, arXiv: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.

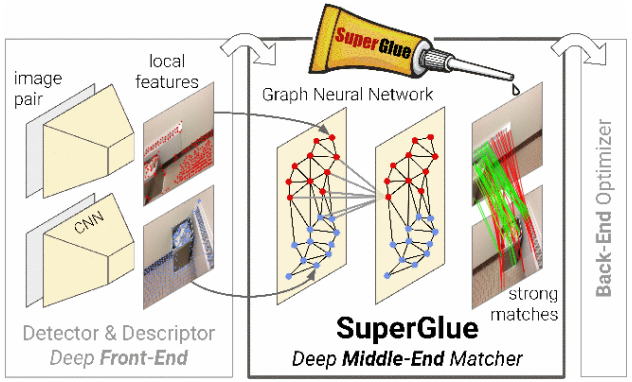




Sarlin, P. E., Detone, D., Malisiewicz, T., & Rabinovich, A. (2020).
SuperGlue: Learning Feature Matching with Graph Neural Networks.
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4937–4946.

https://psarlin.com/superglue/





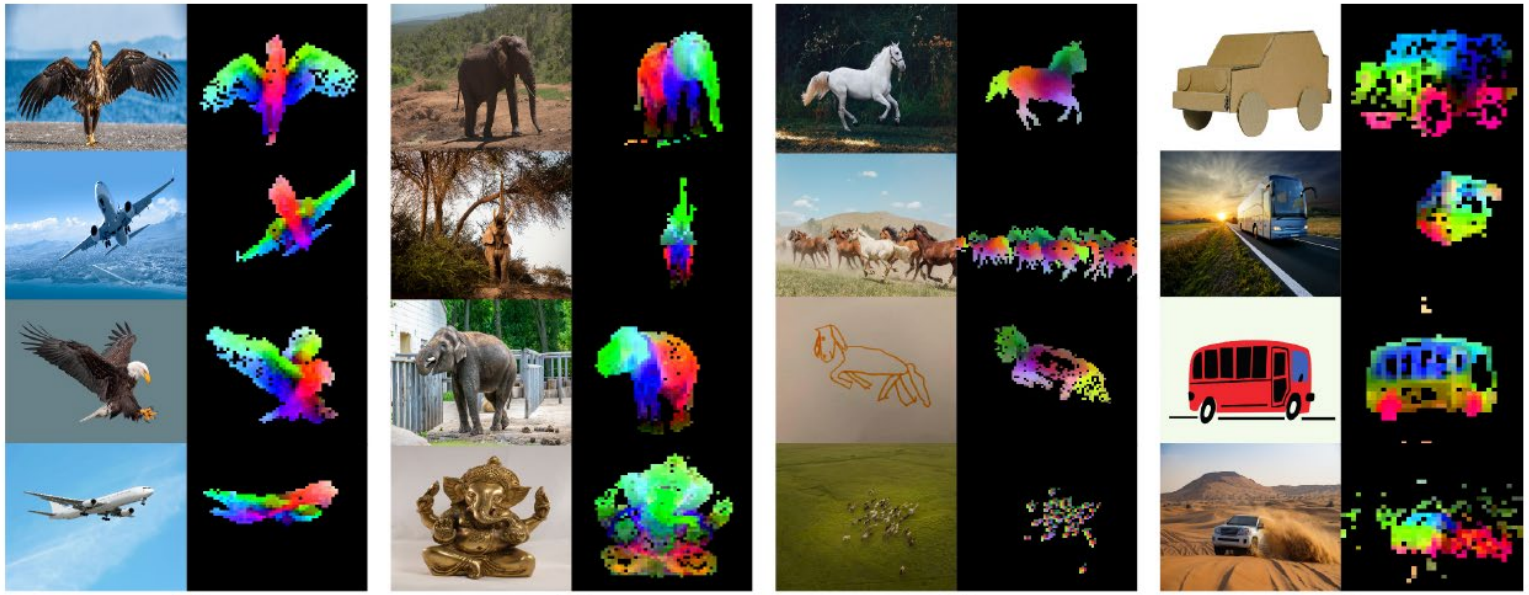
Sarlin, P. E., Detone, D., Malisiewicz, T., & Rabinovich, A. (2020).
SuperGlue: Learning Feature Matching with Graph Neural Networks.
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 4937–4946.

https://psarlin.com/superglue/

DINOv2: Learning Robust Visual Features without Supervision

Maxime Oquab**, Timothée Darcet**, Théo Moutakanni**,
Huy V. Vo*, Marc Szafraniec*, Vasil Khalidov*, Pierre Fernandez, Daniel Haziza,
Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba,
Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat,
Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal[1],
Patrick Labatut*, Armand Joulin*, Piotr Bojanowski*

Meta AI Research [1]*Inria*



(a) (b) (c) (d)

M. Oquab et al., 'DINOv2: Learning Robust Visual Features without Supervision', 2023, arXiv. doi: 10.48550/ARXIV.2304.07193.

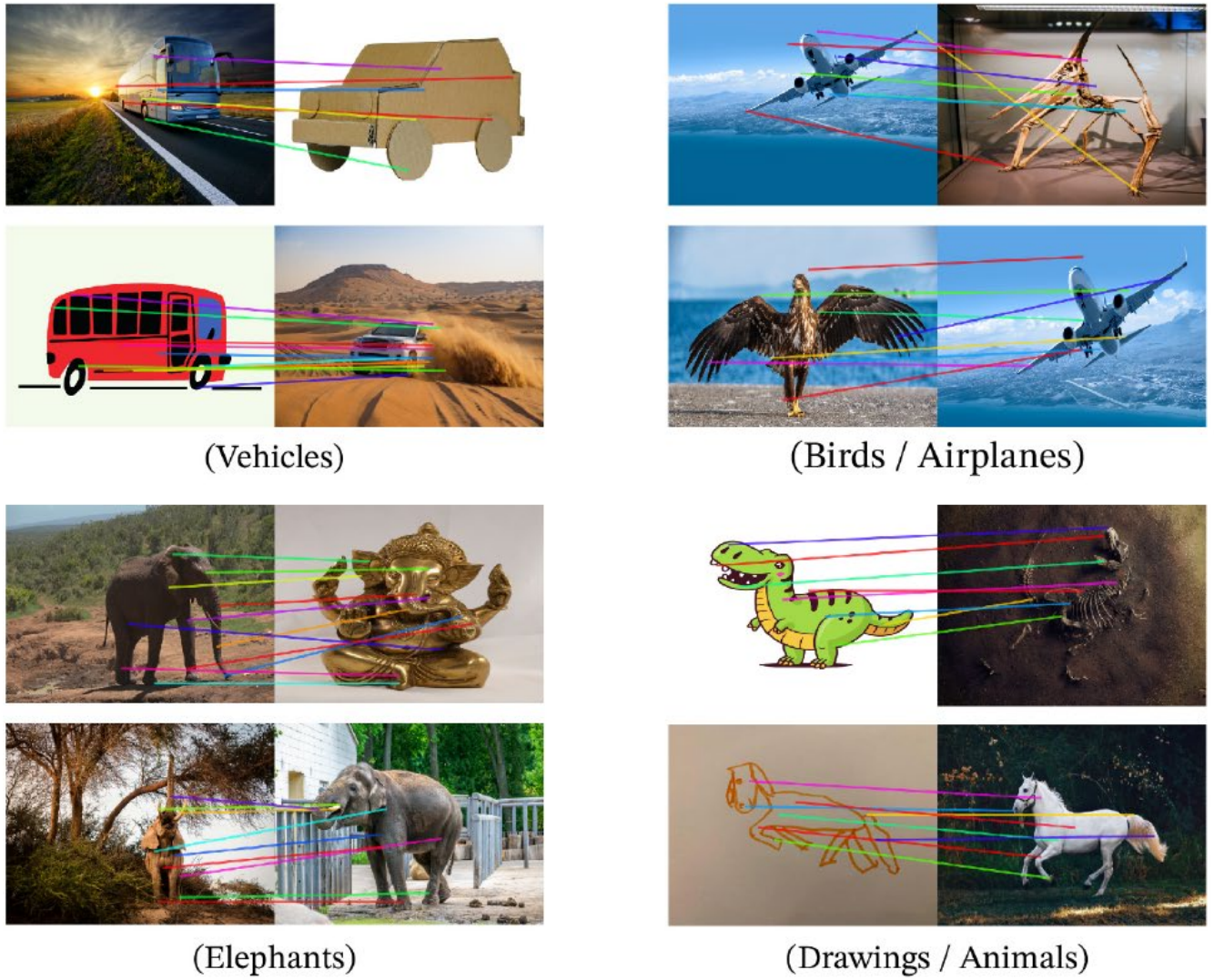
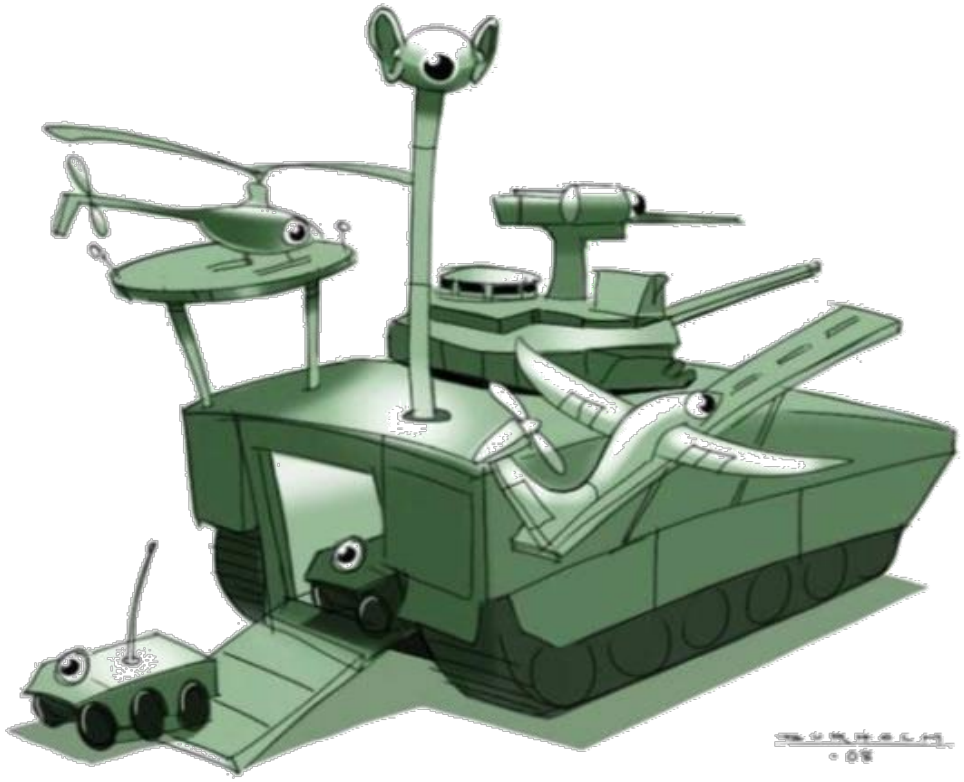


Figure 10: **Matching across images.** We match patch-level features between images from different domains, poses and even objects that share similar semantic information. This exhibits the ability of our model to transfer across domains and understand relations between similar parts of different objects.

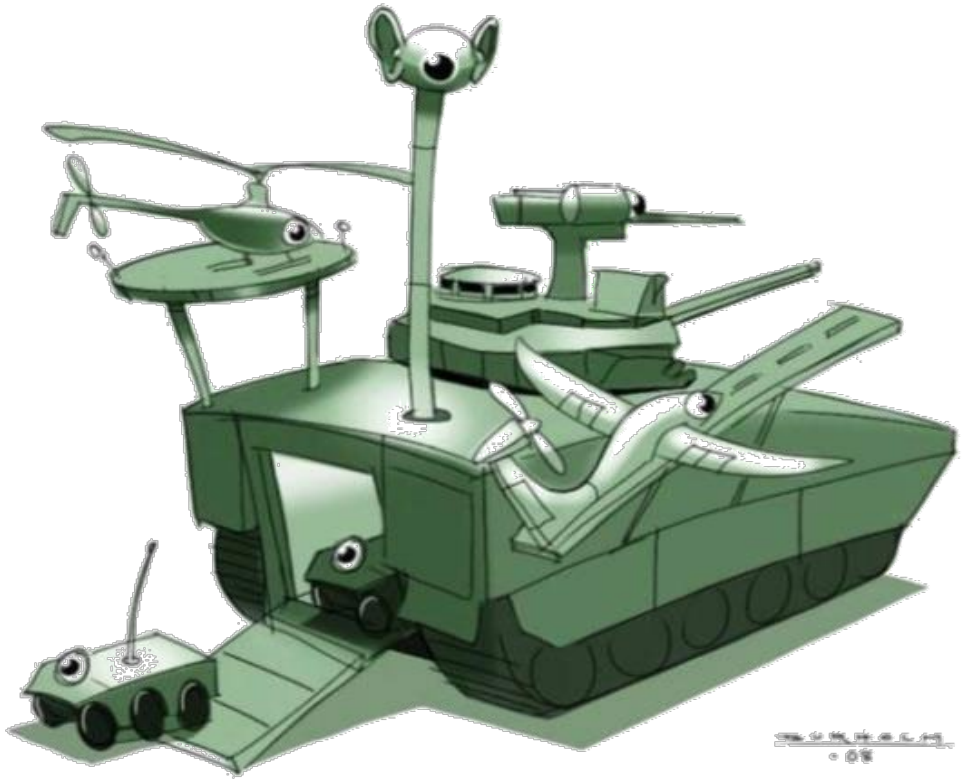
M. Oquab et al., 'DINOv2: Learning Robust Visual Features without Supervision', 2023, arXiv. doi: 10.48550/ARXIV.2304.07193.

What is computer vision?



What is computer vision?

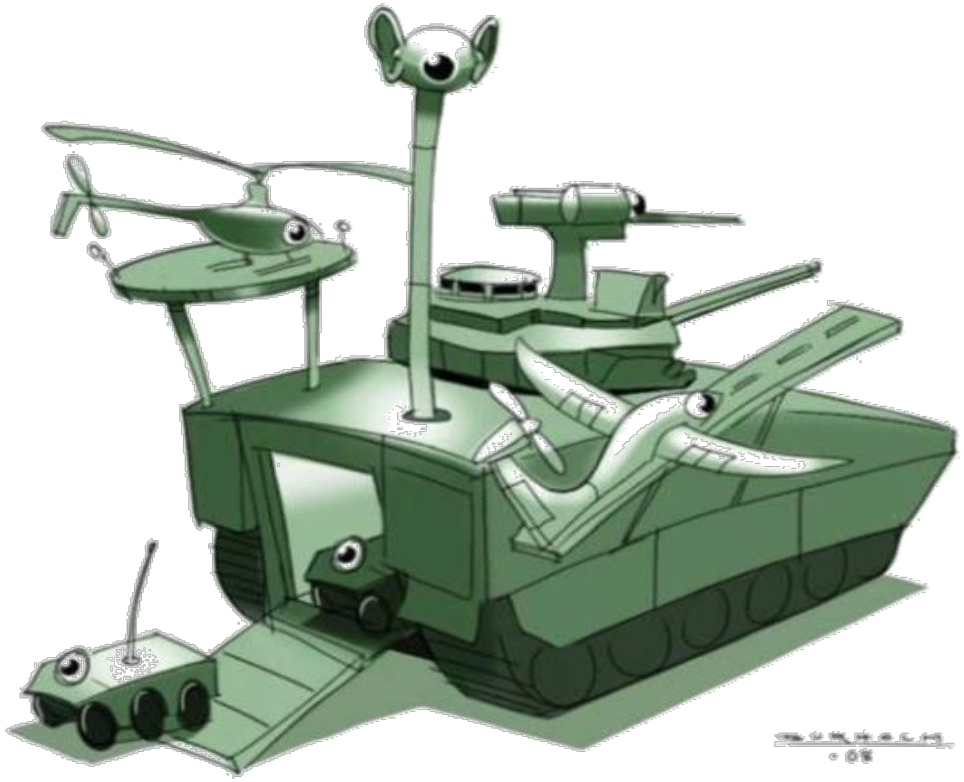
The study of how a machine
can interpret and understand its surroundings
from images



What is computer vision?

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

- "Enabling computers to see"

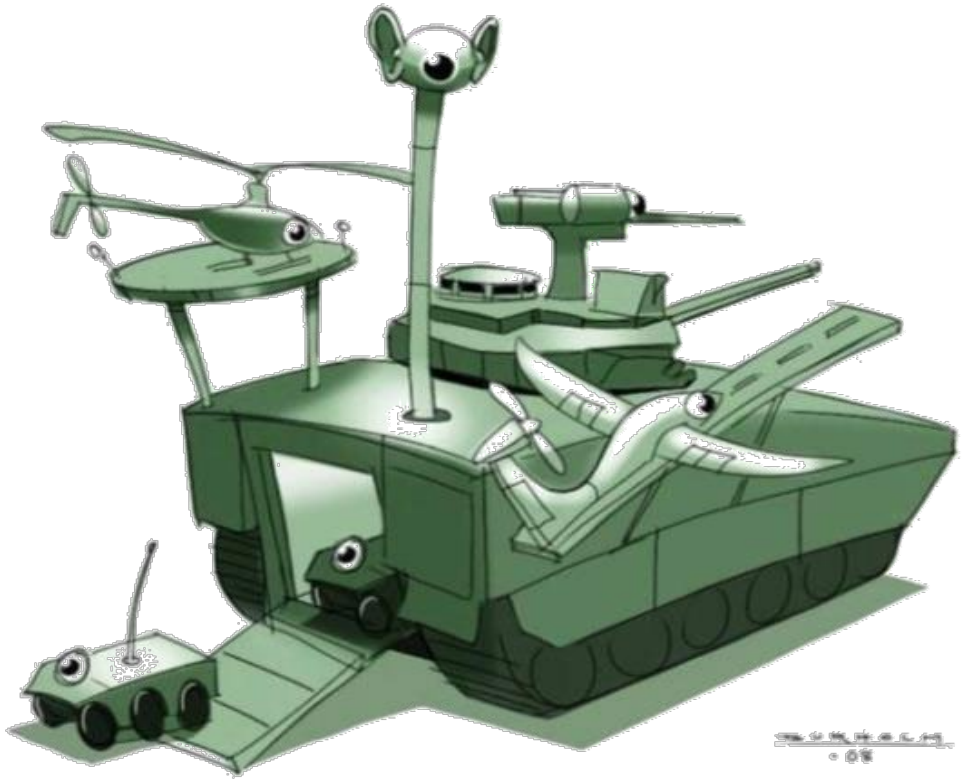


What is computer vision?

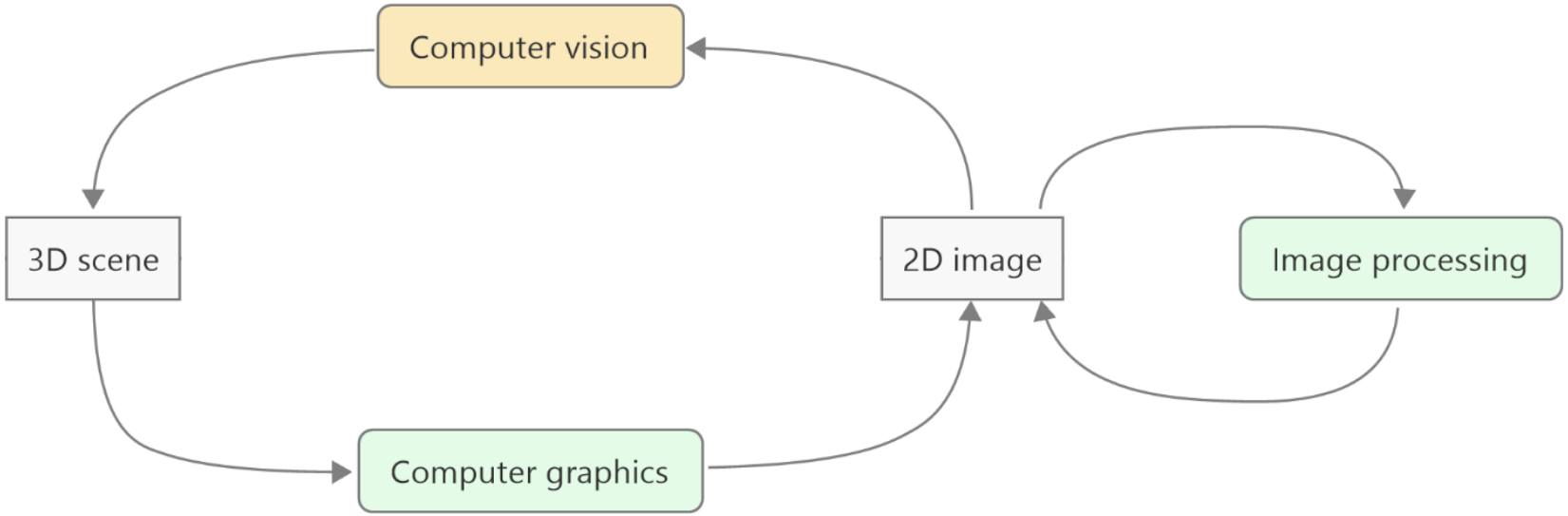
The study of how a machine
can interpret and understand its surroundings
from images

- “Enabling computers to see”

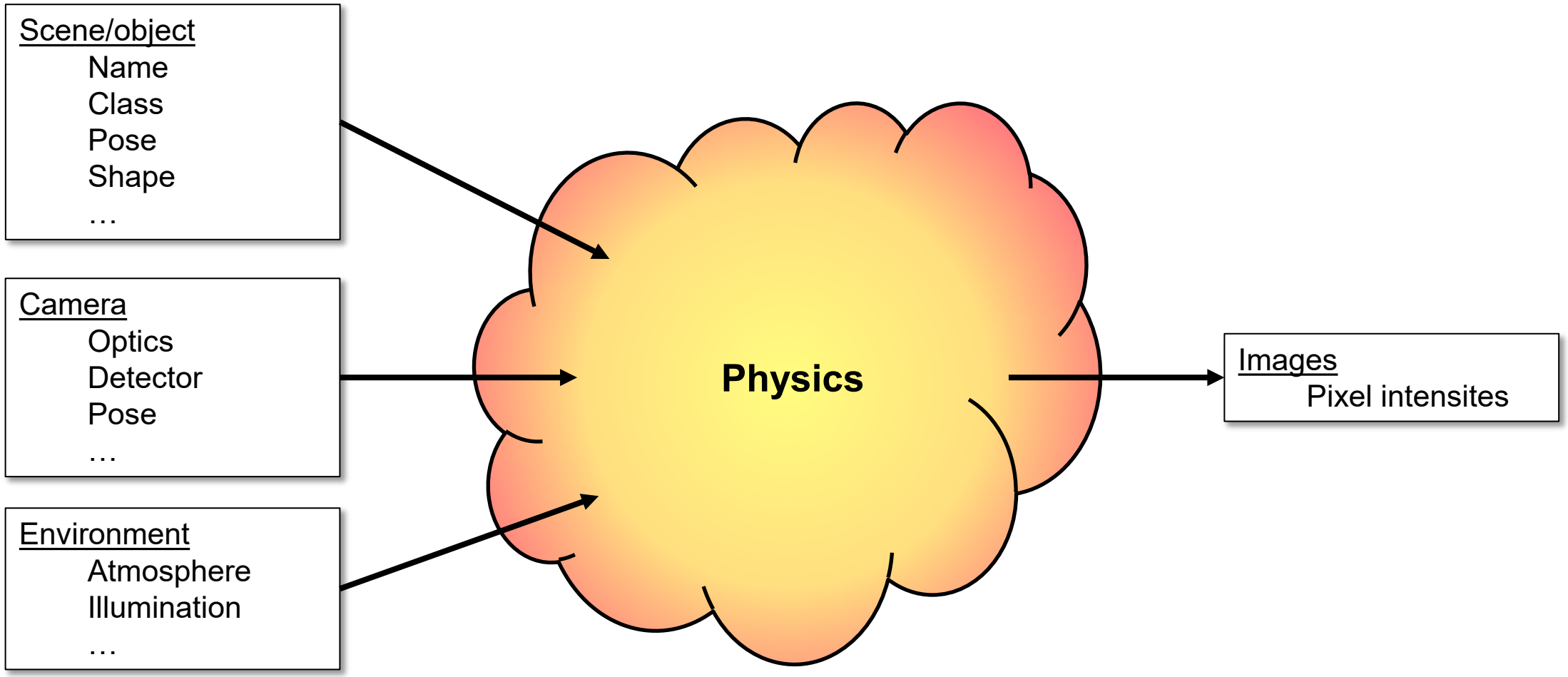
- Image analysis
- Robotic vision
- Perception
- Spatial AI



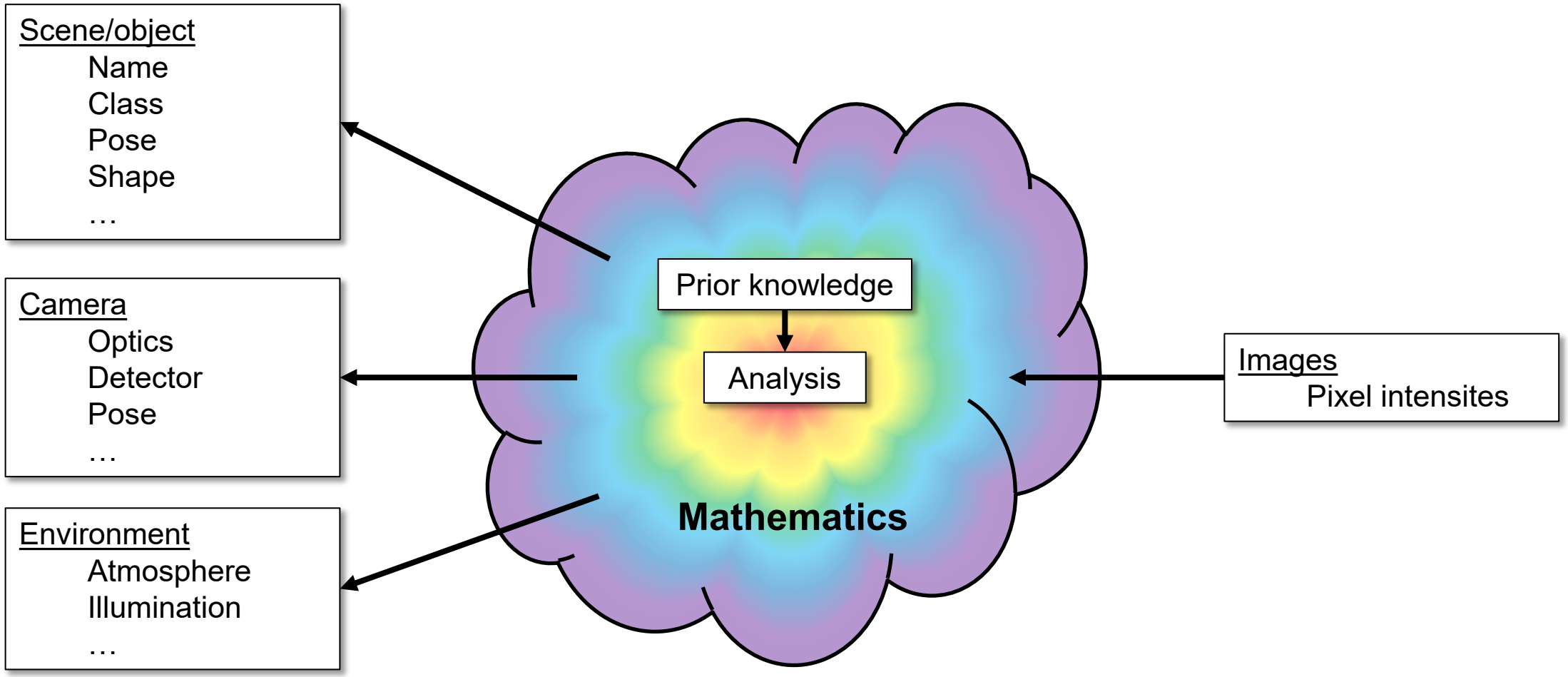
Computer vision is an *inverse problem*!



The *forward* imaging process



The *inverse* analysis process

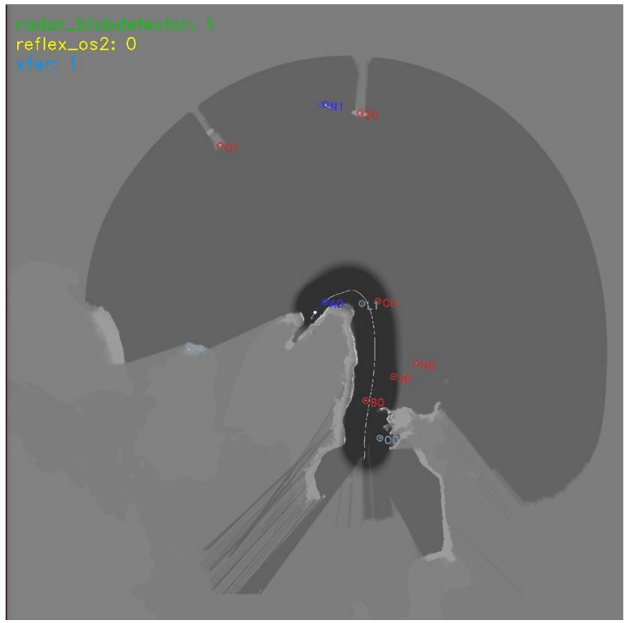


About the course

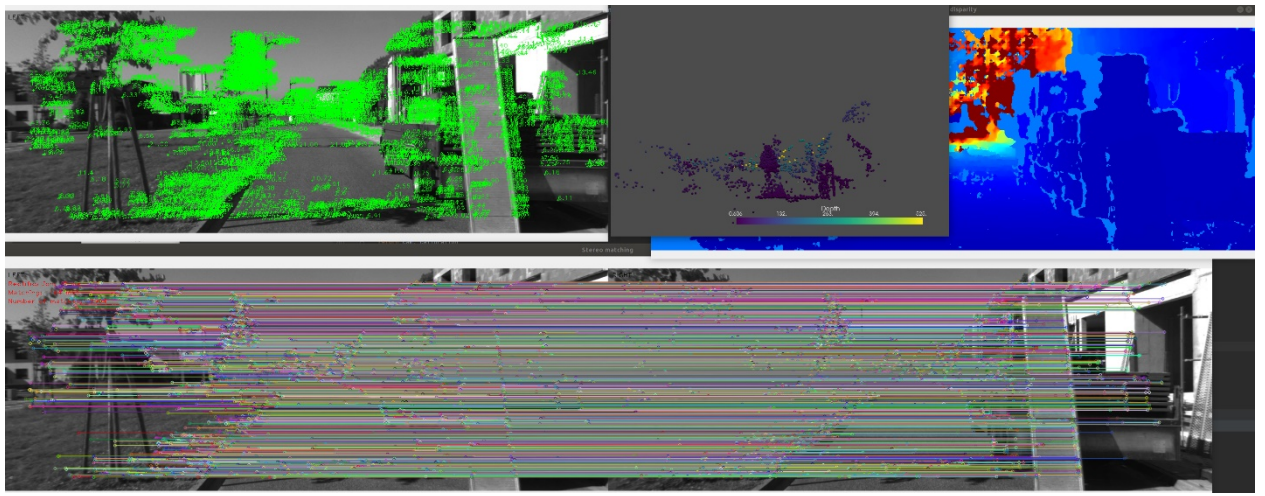
TEK5030 – Computer vision in practice!

- From pixel intensities to state estimation!
- Real-time video processing
- Supervised programming labs
- One large project

$$X = \left\{ \right.$$



$$\left. \right\}$$



«Flipped classroom»

- Purpose
  - Get as much as possible out of a day at Kjeller
- Online
  - Pre-recorded lectures most weeks
- Wednesdays 13:15-16:00
  - ~20 min lecture summary and questions
  - ~2.5 hours programming lab




```cpp
#include "opencv2/highgui.hpp"
#include <iostream>

int main()
{
    cv::VideoCapture input_stream(0);

    if (!input_stream.isOpened())
    {
        std::cerr << "Could not open camera\n";
        return EXIT_FAILURE;
    }

    const std::string window_title = "Lab 0: Introduction to OpenCV";
    cv::namedWindow(window_title, cv::WINDOW_NORMAL);

    cv::Mat frame;

    while(true)
    {
        input_stream >> frame;

        if (frame.empty())
        { break; }

        cv::imshow("cam", frame);

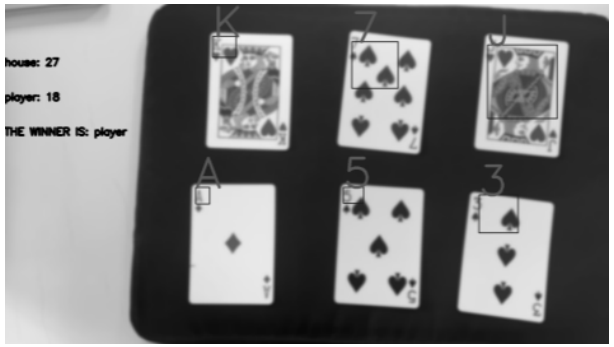
        if (cv::waitKey(15) >= 0)
        { break; }
    }

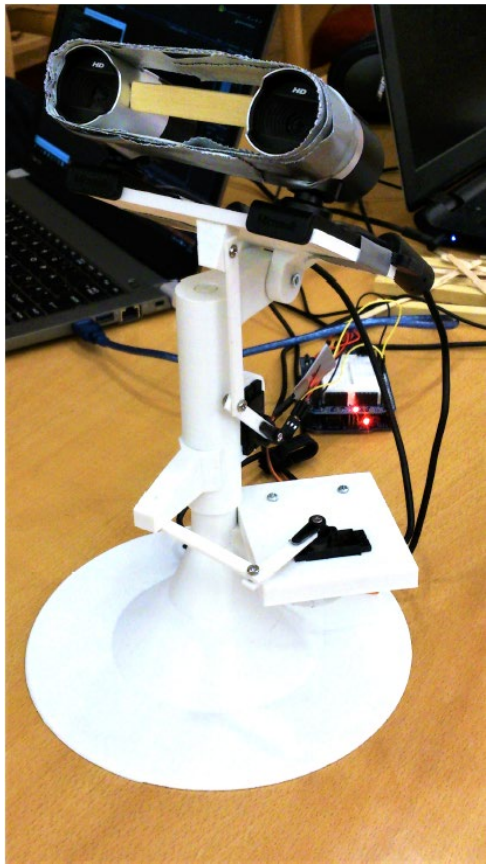
    return EXIT_SUCCESS;
}
```

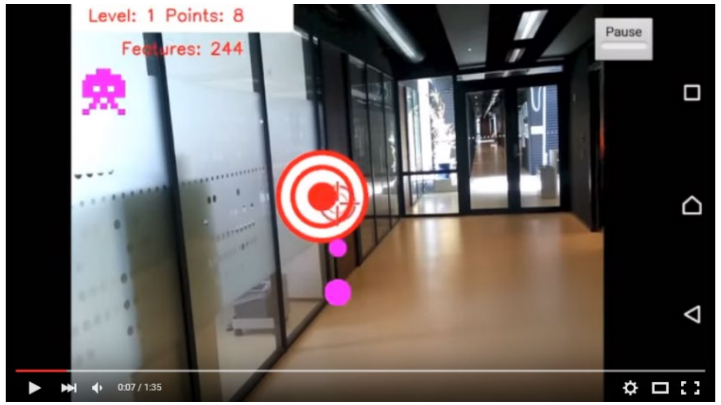
Student projects

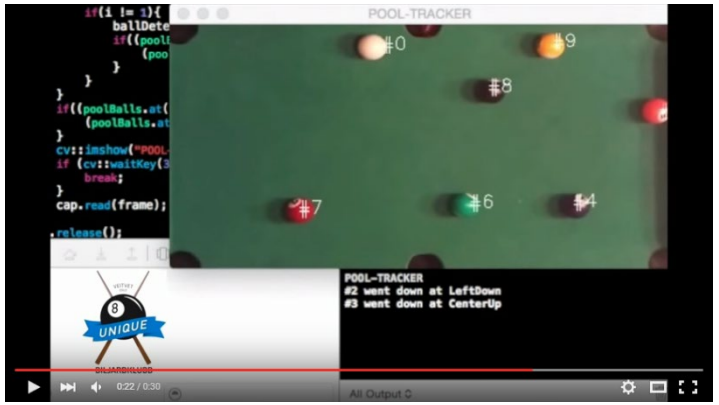
- Develop a working computer vision system that does something interesting
  - Big: More than a month
  - Approved/Not approved

- Project topic of your own choice
- Preferably in groups of up to 3 persons

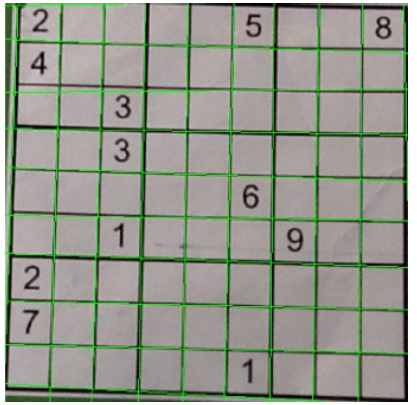




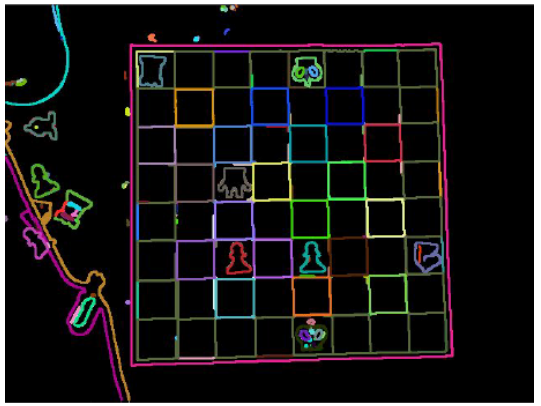




Student projects



- Develop a working computer vision system that does something interesting
  - Big: More than a month
  - Approved/Not approved



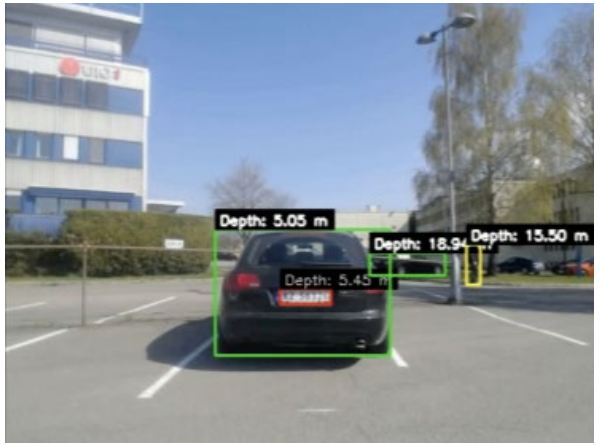


- Project topic of your own choice
- Preferably in groups of up to 3 persons



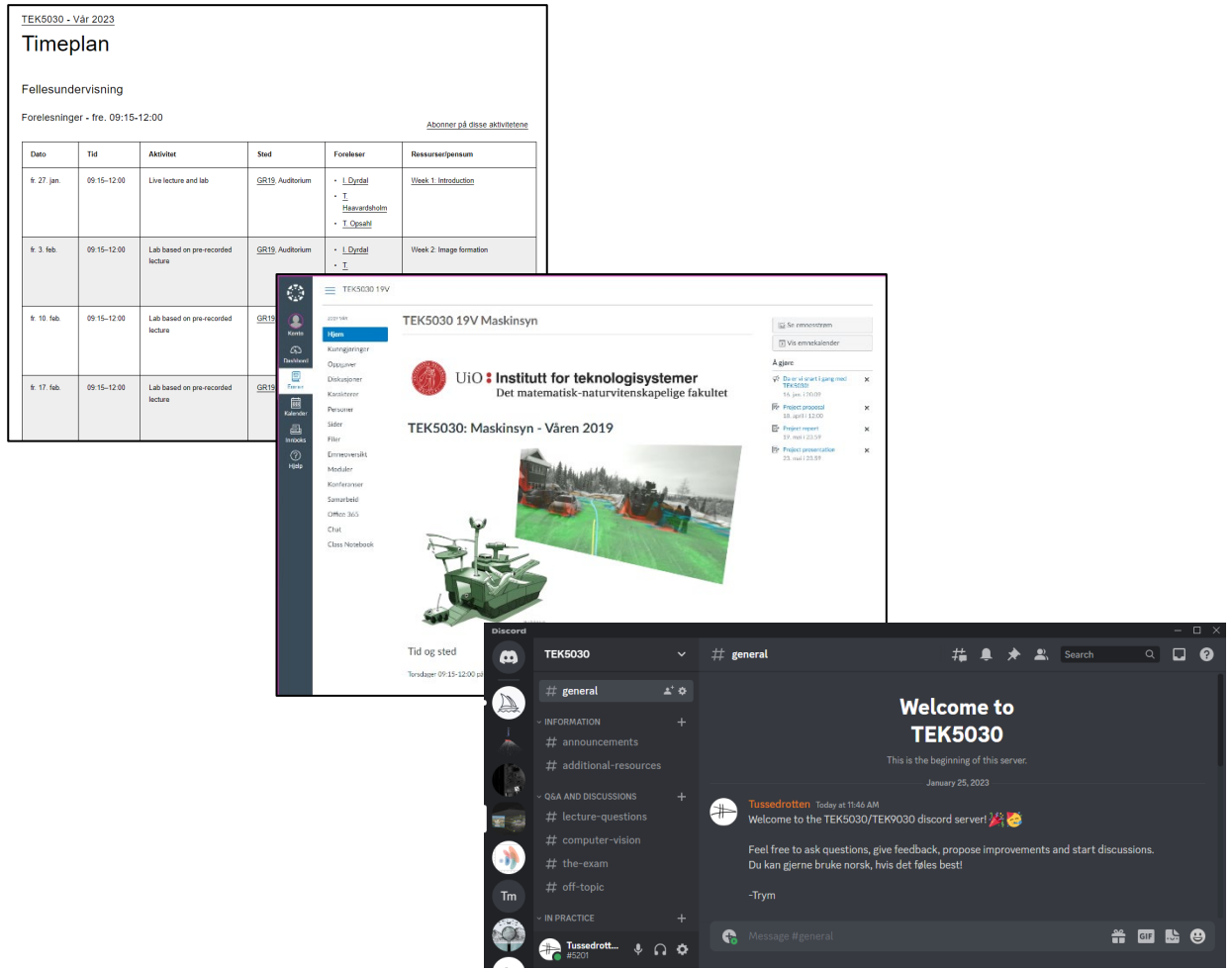






Course sites

- The semester page
  - Course plan
  - Lecture slides and videos
  - Lab exercises
- Canvas
  - Announcements
  - Course overview
  - Zoom links
  - Project deliveries
- Discord
  - Announcements
  - Questions
  - Discussions



Exam

- Written or oral exam
- Mid-June



Feedback

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

