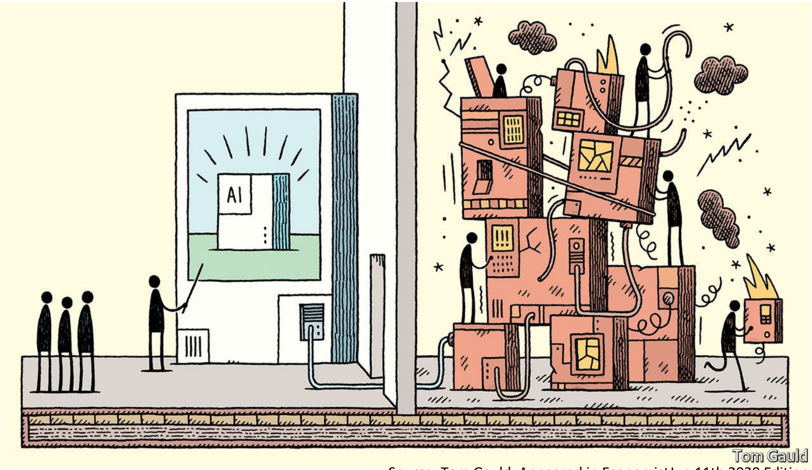
Deep Learning – the Path from Big Data Indexing to Robotics Applications



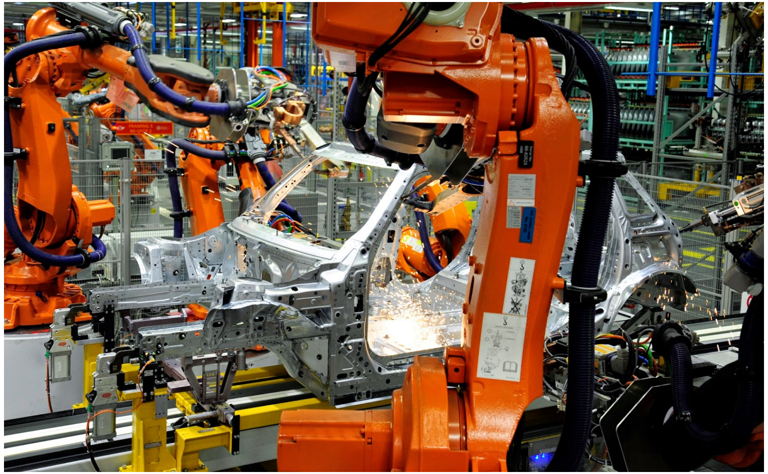
Source: Tom Gauld. Appeared in EconomistJun 11th 2020 Edition

Darius Burschka

Machine Vision and Perception Group (MVP)
Department of Computer Science

Technische Universität München

Computational Challenges in Robotics Applications

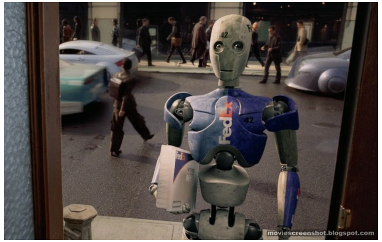


Source: Aytoindustry Newsletter

Complete knowledge about the environment – early adoption of robots in industrial apps

Geriatrics: Garmi Robot (MSRM)

Human-Robot Interaction: understanding human gestures, predictable behavior for acceptance



Source: "I, Robot"

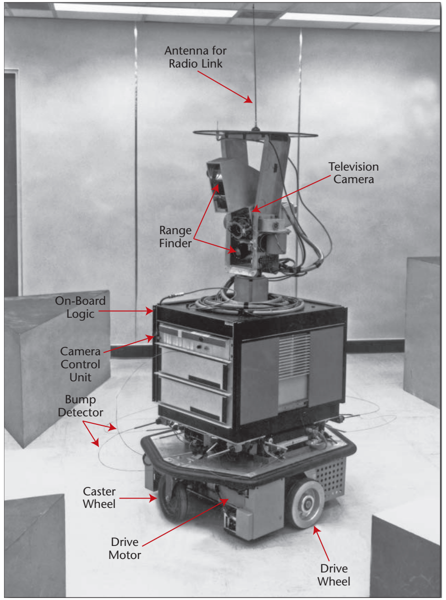
Understanding and Acting in Dynamic Environments: understanding human actions/behaviors, collision avoidance

Inherent Safety to Humans: Understanding injury parameters

Semantic Labeling of Scenes: Knowledge about functions of scene geometry

Early rule-based AI Approaches to Deal with the Challenges

“Shakey” (Stanford AI Lab) was the first mobile robot with the ability to perceive and reason about its surroundings. The first challenge to get funding before anyone knew about mobile robots
– development of “intelligence automata” for “reconnaissance applications.”



Shakey: 1966 to 1972

- Planex (Executive)
- STRIPS (Symbolic Planning System)
- Intermediate Level Actions
- Low Level Actions

Condition	Action
infrontof(door) /\eq(s,OPEN)	bumblethru(room1,door,room2)
near(door) /\eq(s,OPEN)	align(room1,door,room2)
near(door) /\eq(s,UNKNOWN)	doorpic(door)
eq(s,CLOSED)	return [fail]
T	navto(nearpt(room1,door))

Markov Table for GoThroughDoor (single action)

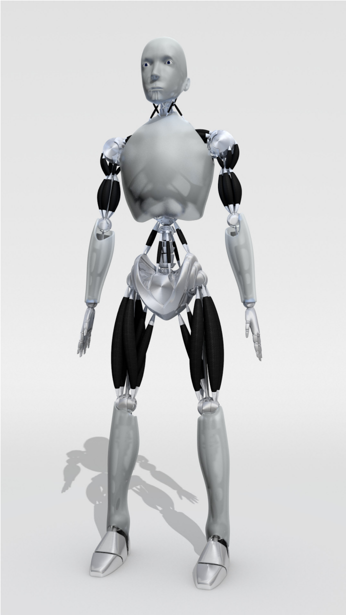
Rule-based AI systems are artificial intelligence models, which utilize the rule of if-then coding statements. The two major components of rule-based artificial intelligence models are “a set of rules” and “a set of facts”

Modern Rule-Based AI System

Rosie is a research robot that has four-fingered hands, an omnidirectional mobile base, and a wide variety of sensors. It's designed to undertake many of the household chores.

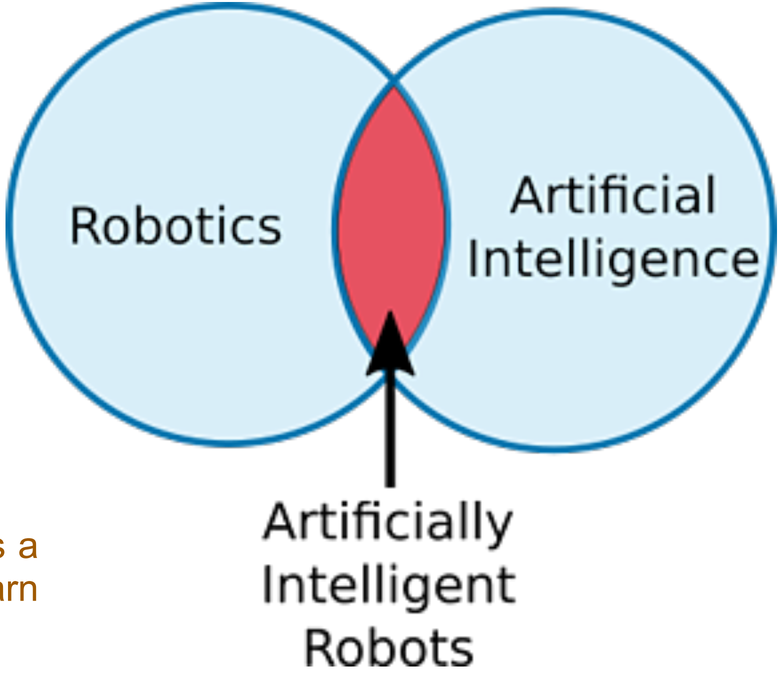
It uses an Internet database to parse recipes and generates a set of rules, how to accomplish the task.

Current Trend to Avoid direct Rule Programming - Learning Approaches



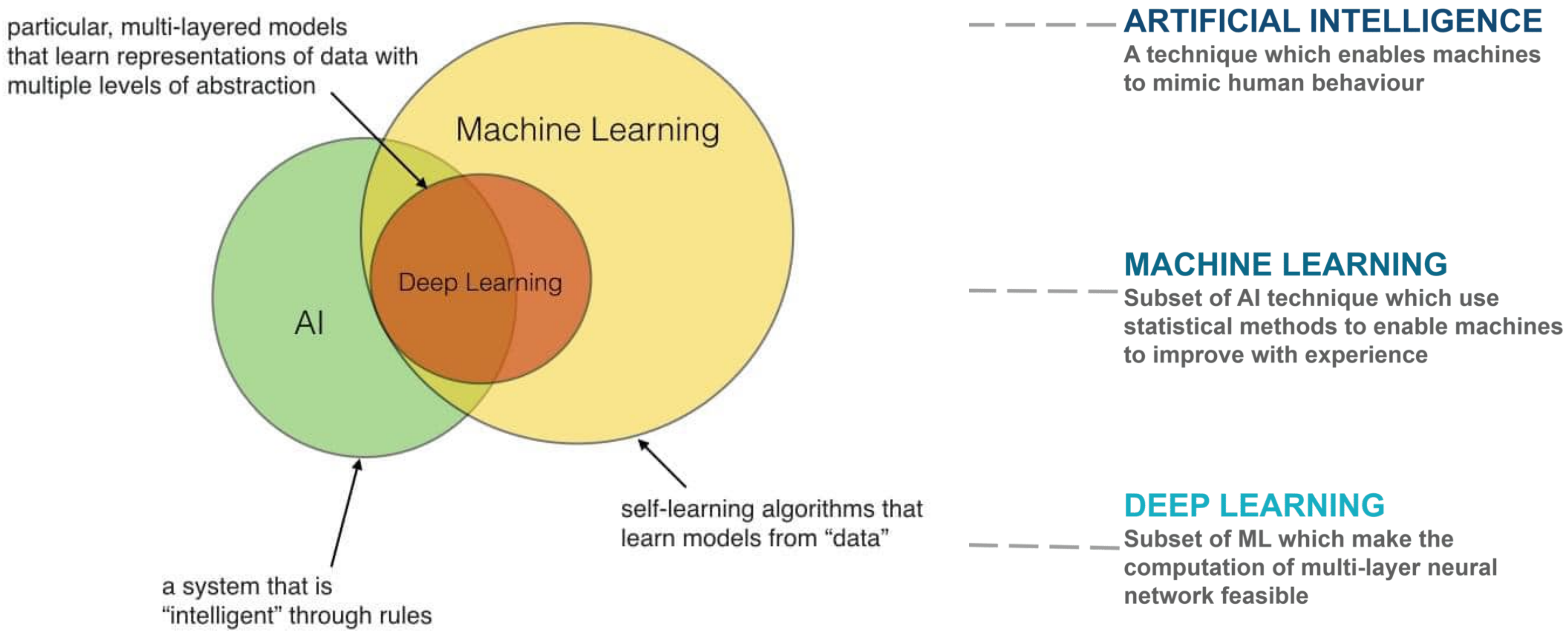
- Semantics (object recognition) – ImageNet,VGG16
- Action modelling – RNNs

…



The machine learning system uses a large number of examples to learn the rules from observation.

Misconceptions in current DL Research (DL ≠ AI)



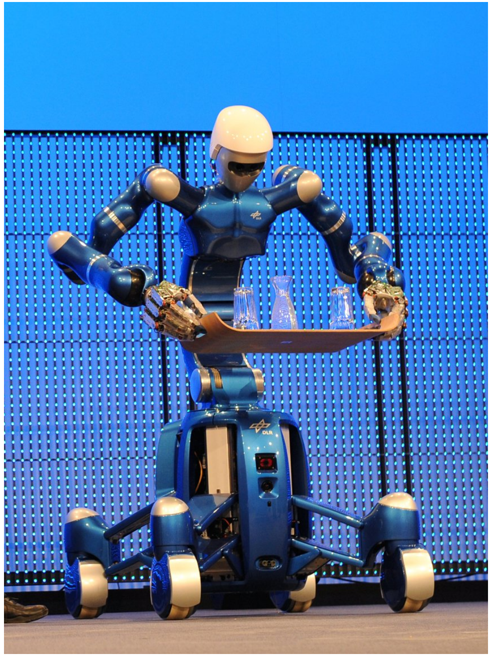
Emergence of Deep Learning – Big Data

Textual or visual query
to large database
Labeling/Categorization

Categorization of behavioral
models for advertising and news

Similarity measures and
shopping behavior modelling

What is different in Robotics compared to Big Data Queries?



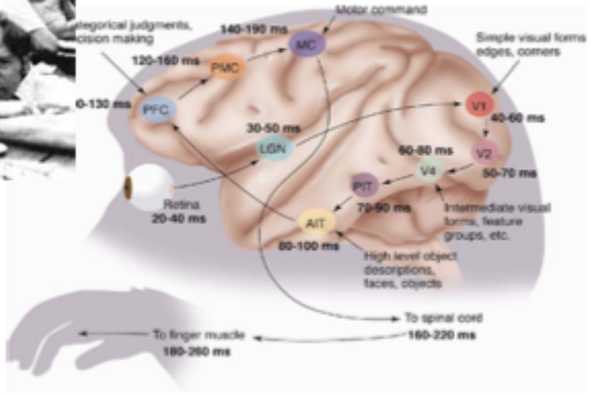
We need to know not only what is in the area around the robot, but also

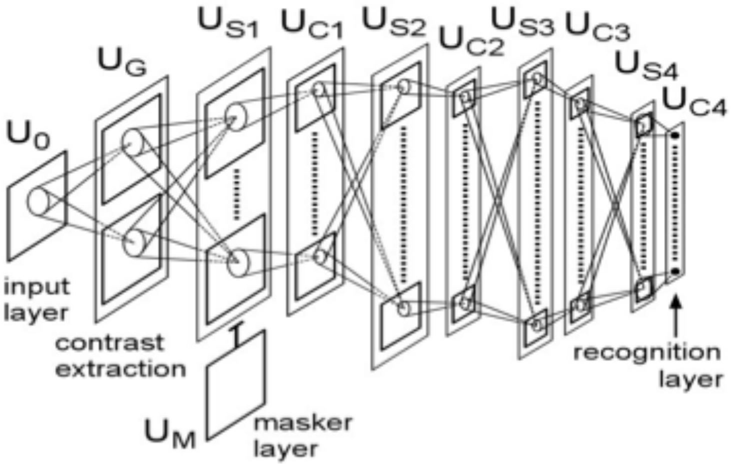
- How big is the confidence in the correctness of the observation? How much of the object was visible…
- How certain is the system to see a specific object (similarity to other similar ones)?
- Where it is relative to the robot?
- What is the dynamic state of the observed object?
- What is the accuracy of the metric observation?

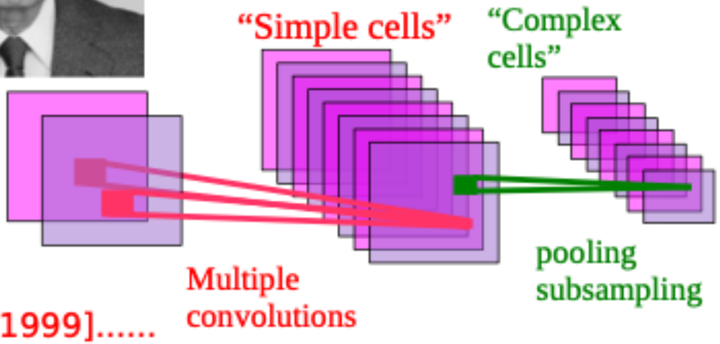
Categorization (What) Analogy in Visual Cortex

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a

[Thorpe & Fabre-Thorpe 2001]





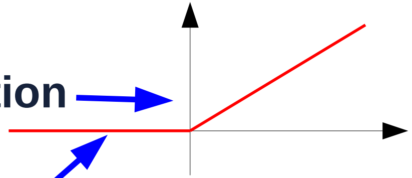


[Fukushima 1982][LeCun 1989, 1998],[Riesenhuber 1999]......

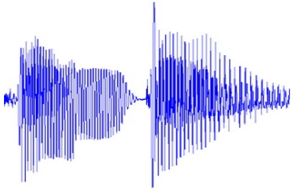
Deep Multiple Layer Neural Nets (Yann LeCun)

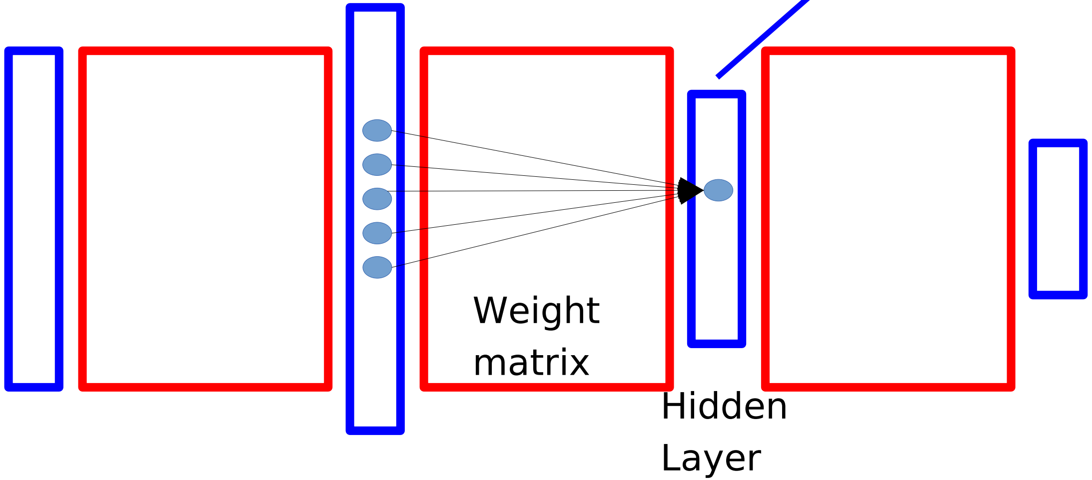
- Multiple Layers of **simple units**
- Each units computes a **weighted sum** of its inputs
- Weighted sum is passed through a **non-linear** function
- The learning algorithm changes the **weights**

$$ReLU(x) = max(x, 0)$$



Ceci est une voiture



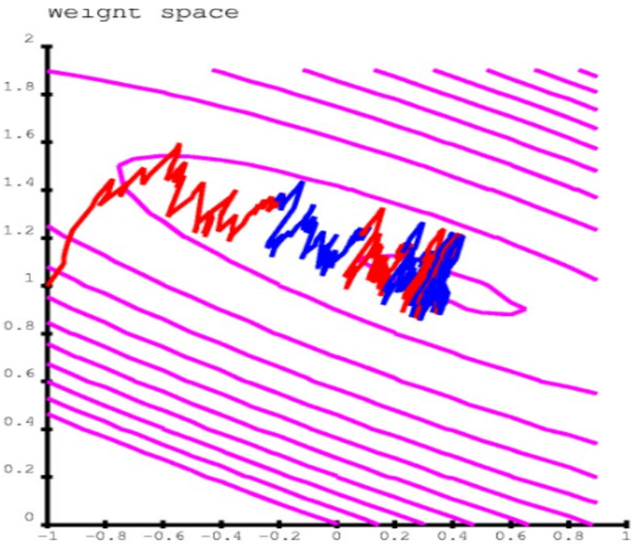


Learning as Parameter Optimization without explicit Rules and Features (Yann LeCun)

Function with adjustable parameters

Objective Function

traffic light: -1

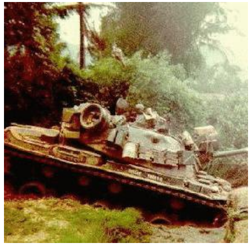


- **It's like walking in the mountains in a fog and following the direction of steepest descent to reach the village in the valley**
- **But each sample gives us a noisy estimate of the direction. So our path is a bit random.**

$$W_i \leftarrow W_i - \eta \frac{\partial L(W, X)}{\partial W_i}$$

- **Stochastic Gradient Descent (SGD)**

The NN anecdote from the 80s is still alive

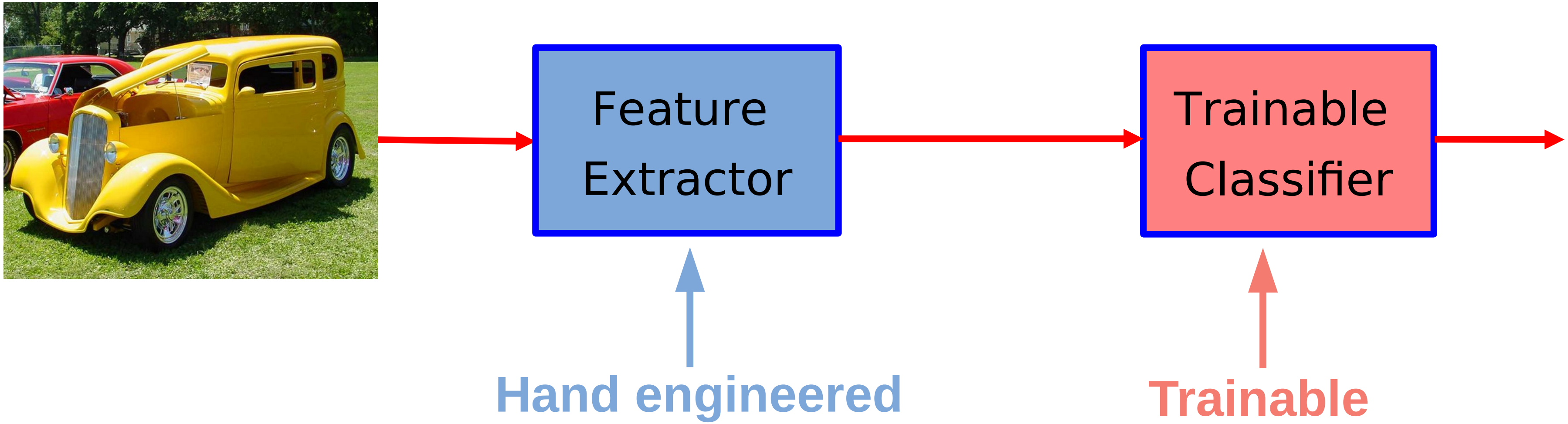


In the 1980s, the Pentagon wanted to harness computer technology to make their tanks harder to attack.The preliminary plan was to fit each tank with a digital camera hooked up to a computer. The computer would continually scan the environment outside for possible threats (such as an enemy tank hiding behind a tree), and alert the tank crew to anything suspicious. Computers are really good at doing repetitive tasks without taking a break, but they are generally bad at interpreting images. The only possible way to solve the problem was to employ a neural network.

The research team went out and took 100 photographs of tanks hiding behind trees, and then took 100 photographs of trees - with no tanks. They took half the photos from each group and put them in a vault for safe-keeping, then scanned the other half into their mainframe computer. The huge neural network was fed each photo one at a time and asked if there was a tank hiding behind the trees. The question was did it understand the concept of tanks vs. no tanks, or had it merely memorized the answers? So the scientists took out the photos they had been keeping in the vault and fed them through the computer. The computer had never seen these photos before -- this would be the big test. To their immense relief the neural net correctly identified each photo as either having a tank or not having one.

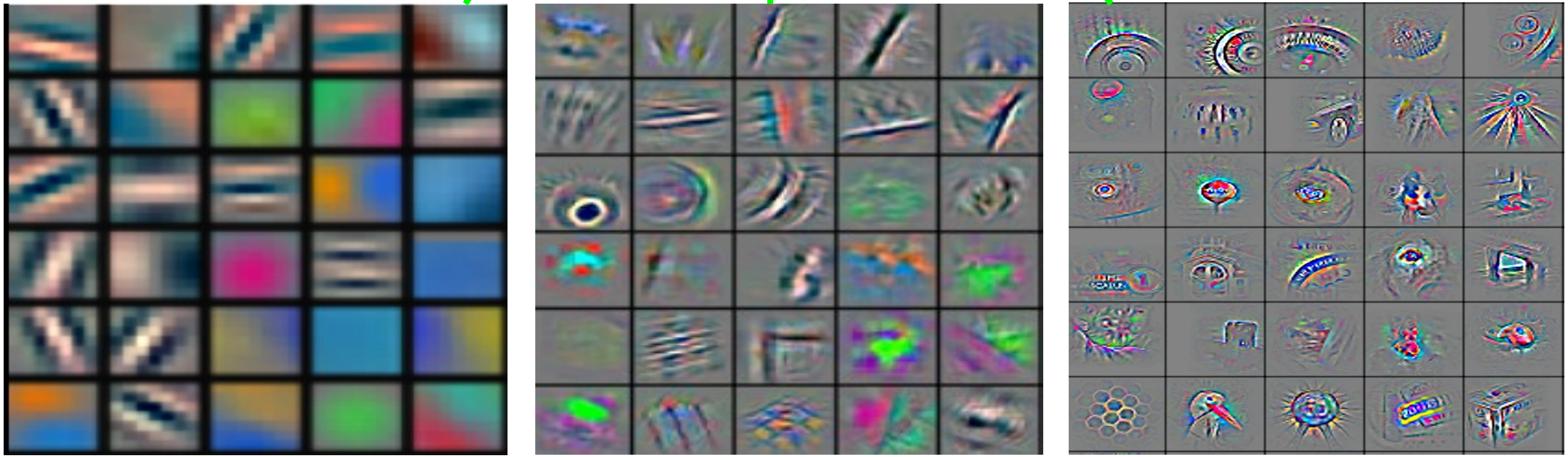
Independent team commissioned another set of photos (half with tanks and half without) and scanned them into the computer and through the neural network. The results were completely random. For a long time nobody could figure out why. After all nobody understood how the neural had trained itself. Grey skies for the US military - Eventually someone noticed that in the original set of 200 photos, all the images with tanks had been taken on a cloudy day while all the images without tanks had been taken on a sunny day. The neural network had been asked to separate the two groups of photos and it had chosen the most obvious way to do it -

Traditional Object Detection System



Deep Learning = Learning Representations

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier →



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Impressive DL Performance, but we lost the ability to understand, how it is done…

[Ribeiro et al. 2016]

Prediction wolf vs. husky

Only 1 mistake!

Explaining DL at CVision example is cheating…

Low-Level Feature → Mid-Level Feature → High-Level Feature → Trainable Classifier →

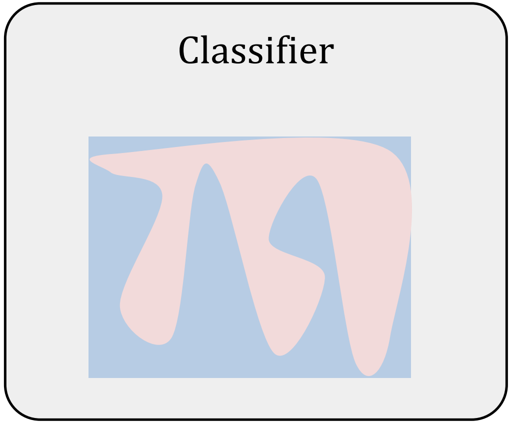
Computer Vision is a well-studied and well-understood problem. This is what helps to formulate all the explanations but… you notice that it is hard to understand DL net structures on an unsolved problem.

Toy example to understand the problem:
DL framework that predicts girlfriend's anger

After a successful training, you will end up with the system that may correctly predict the anger state, but it will not help you understand the process and involved cues in any sense (black box)

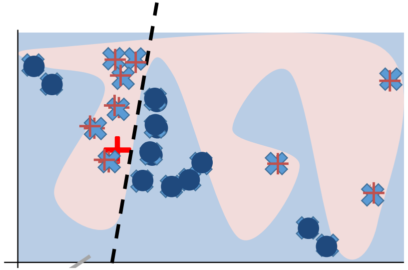
Can we understand the black box? → LIME

Global explanation may be too difficult



LIME: Sparse Linear Explanations

1. Sample points around x_i
2. Use model to predict labels for each sample
3. Weigh samples according to distance to x_i
4. Learn simple model on weighted samples
5. Use simple model to explain

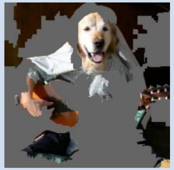


LIME Example - Images (Certainty)

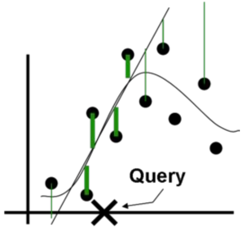
[Ribeiro et al. 2016]

Original Image

P(labrador) = 0.21

Perturbed Instances	P(Labrador)
	0.92
	0.001
	0.34

Maybe to a fault?



Locally weighted regression

Explanation

LIME is quite customizable:

- How to perturb?
- Distance/similarity?
- How *local* you want it to be?
- How to express explanation

Categorize Wolf vs Husky

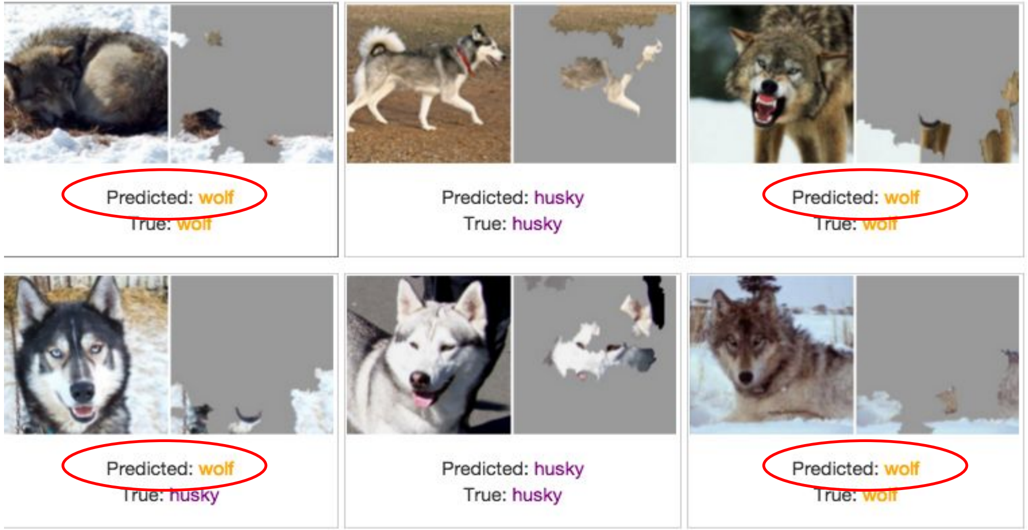
[Ribeiro et al. 2016]

Prediction wolf vs. husky

Only 1 mistake!

System learned the wrong classifier because of not enough diversity in the training set

[Ribeiro et al. 2016]



We've built a great snow detector…

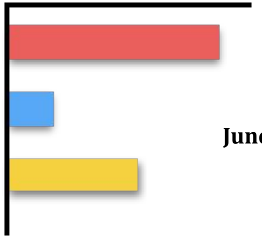
Explainability with Silency Maps

SmoothGrad

Input

Model

Predictions

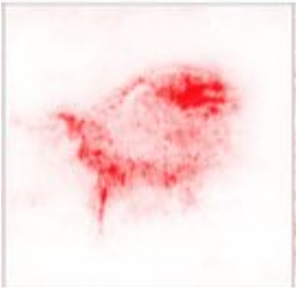


Junco Bird

SmoothGrad

$$\frac{1}{N}\sum_{i}^{N}\nabla_{(x+\epsilon)}F_i(x+\epsilon)$$

Gaussian noise



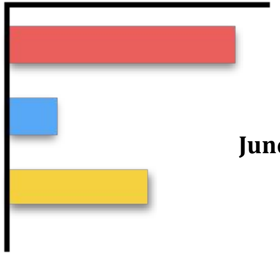
Average Input-gradient of ‘noisy’ inputs.

Layer Relevance Propagation (LRP)

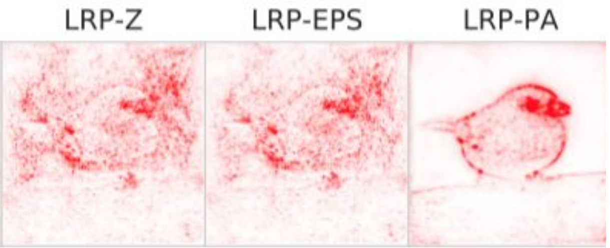
Input

Model

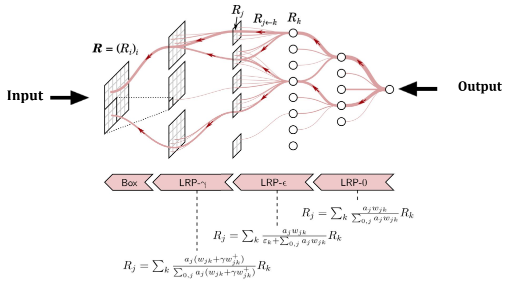
Predictions



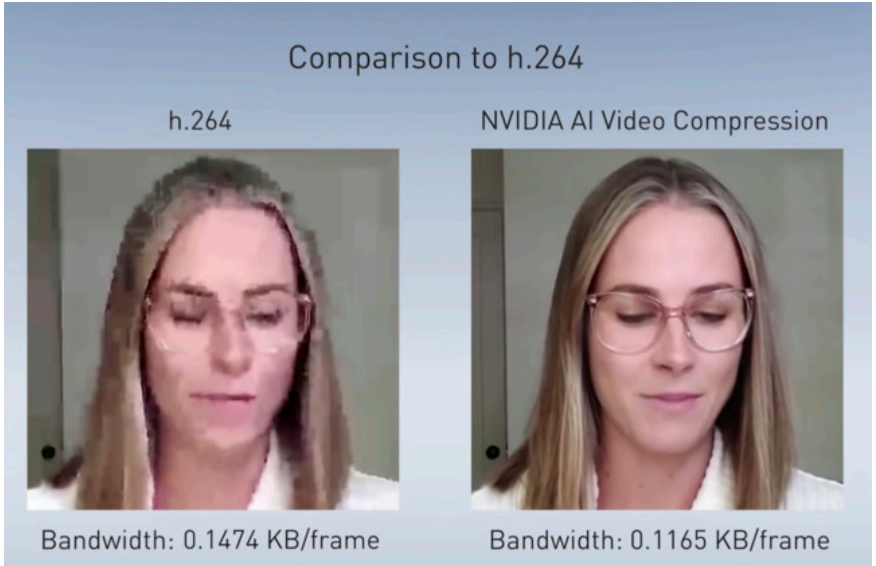
Junco Bird



Compute feature relevance iteratively and propagate. Different **propagation rules** can be specified.



“Hallucination” of Data in DL approaches can be dangerous in Robotics - ”Beautification” is not desired

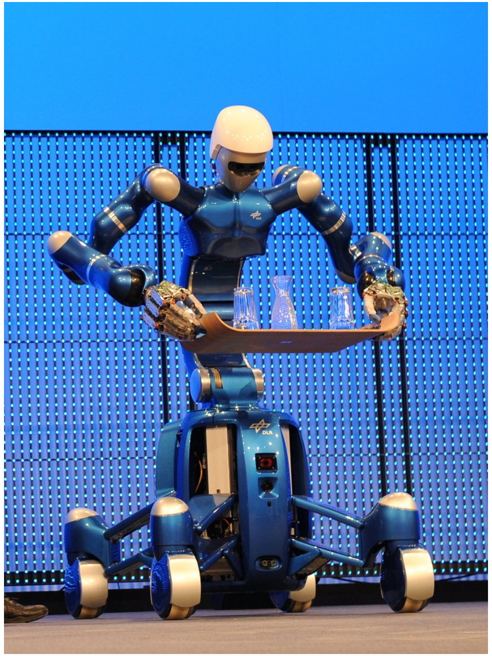


While image restauration in holiday photography is a desired feature, which helps to correct image acquisition errors, Filling gaps in 3D reconstruction using continuation of boundary information can fill gaps that may trap robot and in the medical domain important information about not detected tumors may be removed from the image or a non-existing tumor can be added to the image (hallucination).

95% accuracy of a system without confidence output means that the system runs 72min//day havoc without reporting it.

Decision systems need to know, which data was actually detected in the perception unit and where are gaps to act based on this!

What is different in Robotics compared to Big Data Queries?

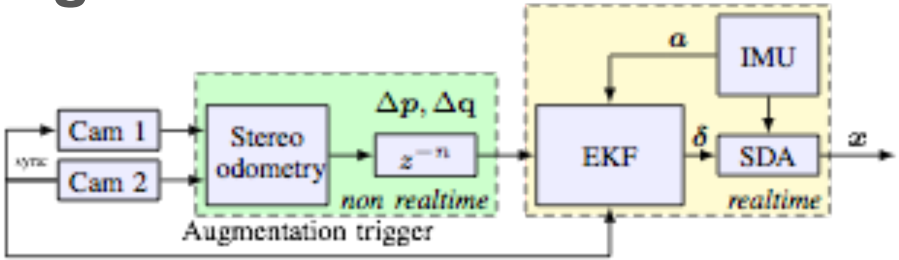


We need to know not only what is in the area around the robot, but also

- How big is the confidence in the correctness of the observation? How much of the object was visible…
- How certain is the system to see a specific object (similarity to other similar ones)?
- Where it is relative to the robot?
- What is the dynamic state of the observed object?
- What is the accuracy of the metric observation?

Navigation for Control – what have we learned from conventional systems? VINS filter design

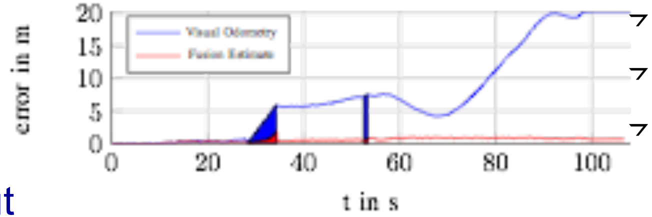
[Schmid et al. IROS 2012]



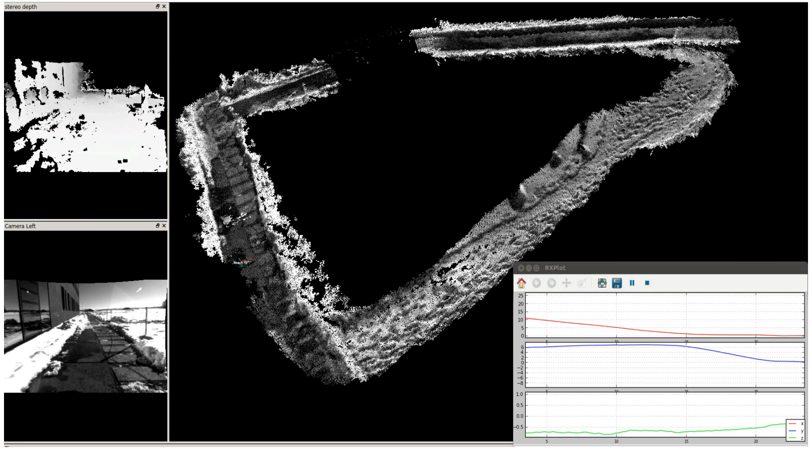
- Synchronization of real-time and non realtime modules by sensor hardware trigger
- Direct system state: $x = \left(p_{ob}^{o,T} \quad v_{ob}^{n,T} \quad q_{b}^{o,T} \quad b_{a}^{b,T} \quad b_{\omega}^{b,T}\right)^{T}$
- High rate calculation by „Strap Down Algorithm“ (SDA)
- Indirect system state: $\delta = \left(\delta_{p}^{o,T} \quad \delta_{v}^{o,T} \quad \delta_{\psi}^{o,T} \quad \delta_{b_a}^{b,T} \quad \delta_{b_\omega}^{b,T}\right)^{T}$
- Estimation by indirect Extended Kalman Filter (EKF)

VINS-Systems
Fusion of heterogeneous data with varying latencies (with DLR)

- 70 m trajectory
- Ground truth by tachymeter
- 5 s forced vision drop out with translational motion
- 1 s forced vision drop out with rotational motion

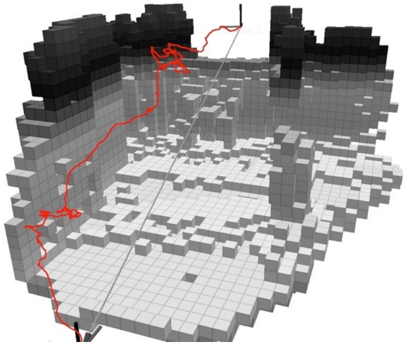


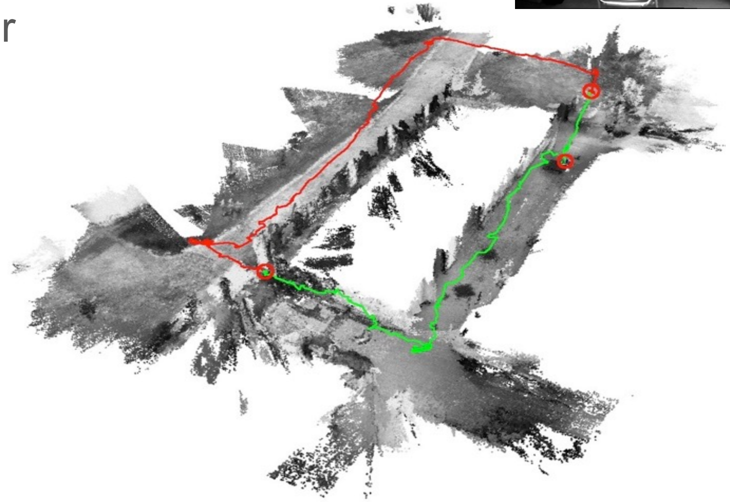
- Estimation error < 1.2 m
- Odometry error < 25.9 m
- Results comparable to runs without vision drop outs



Navigation under strong illumination changes

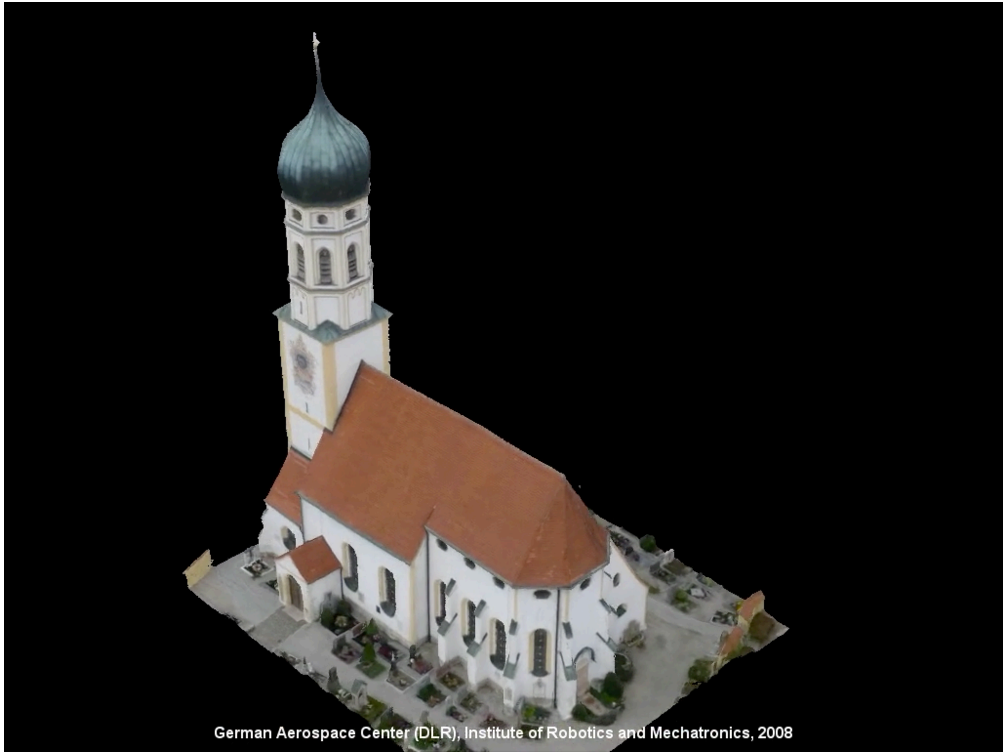
- Autonomous indoor/outdoor flight of 60m
- Mapping resolution: 0.1m
- Leaving through a window
- Returning through door





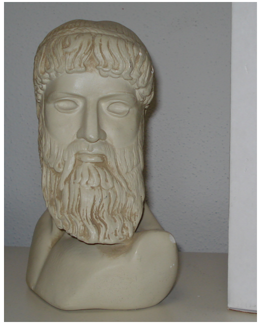
Real-Time Navigation Data from an Image Sequence

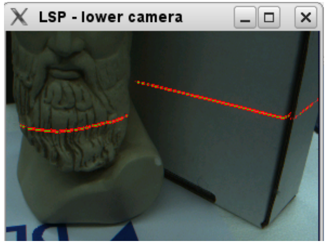
We used to reconstruct static scenes from monocular in 2007… (with DLR)

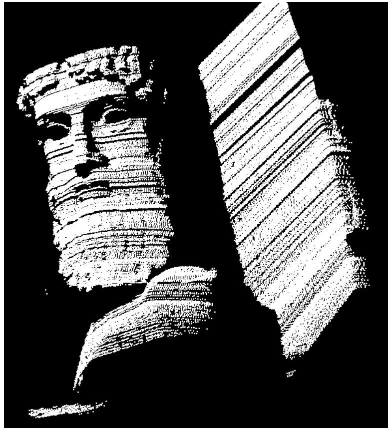


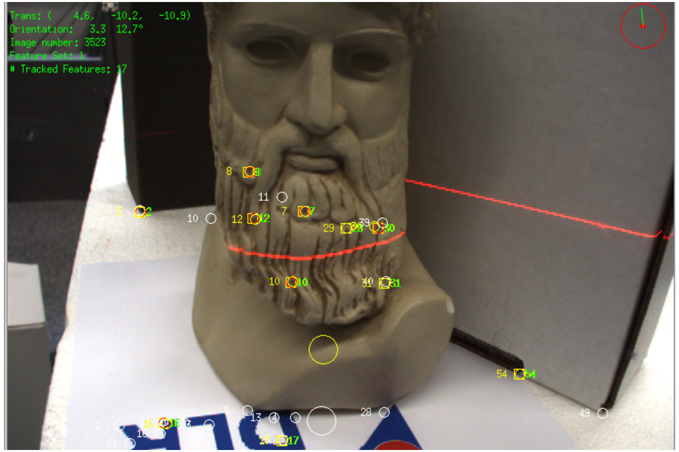
Accuracy:1.5cm

High Accuracy at Example of Light Section 3D Reconstruction

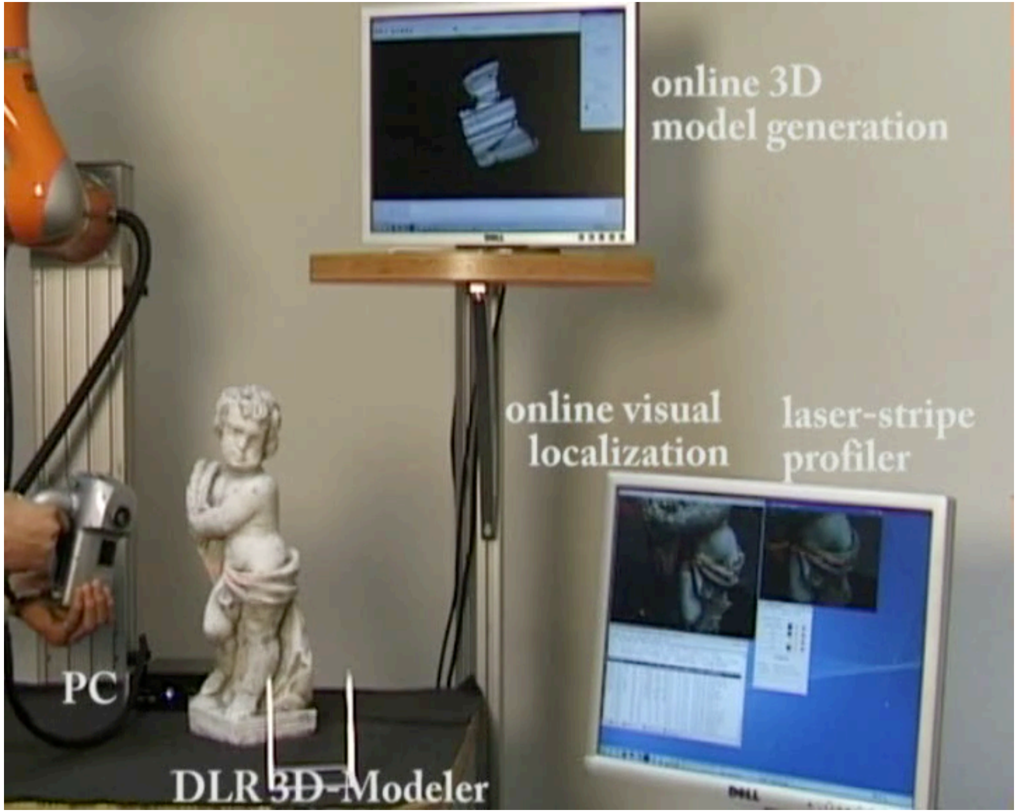






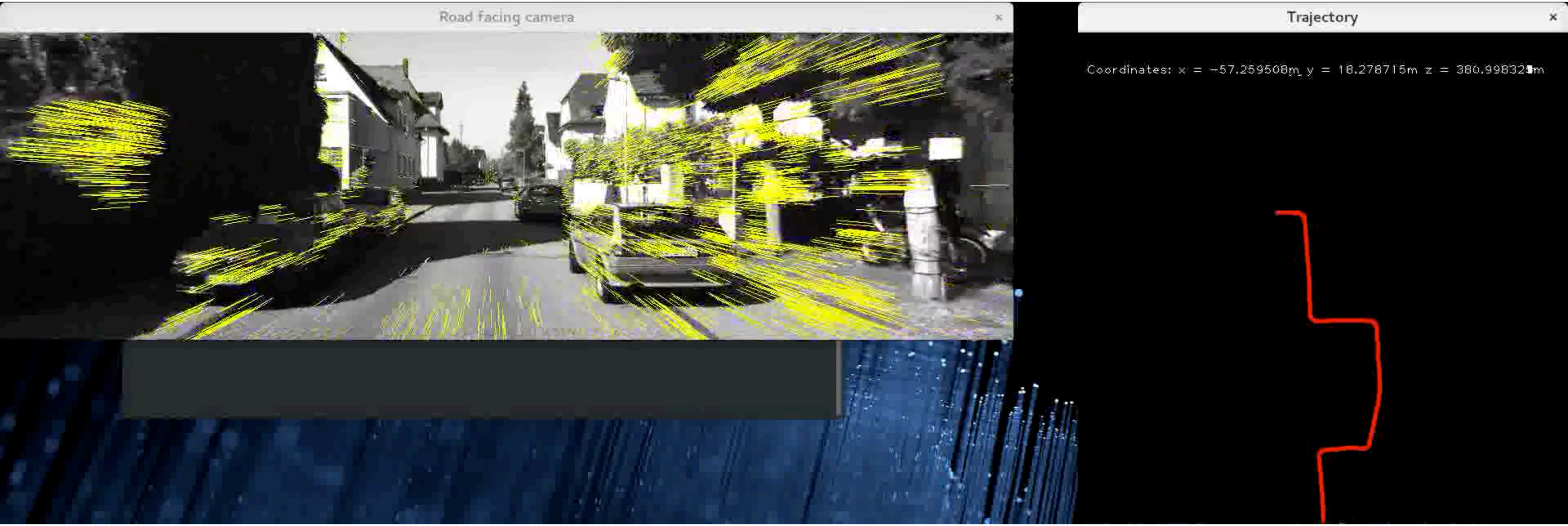


Accuracy of the system - Construction of 3D models (2008)



Camera localization accuracy allows direct stiching of the line responses from the light-section system

120fps Monocular Navigation from Sparse Optical Flow



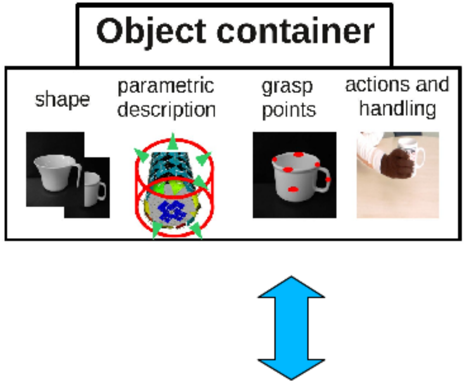
GPU implementation of sparse flow (feature-based OpenCV) system using only 10% of the resources

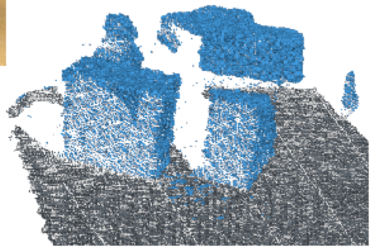
What is in the scene? (labeling)

Indexing of the Atlas information from 3D perception

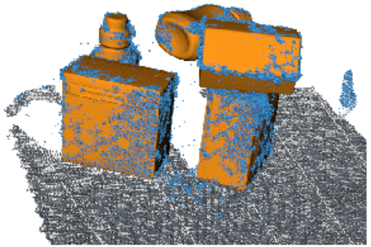
Real-world scenario

scene setup





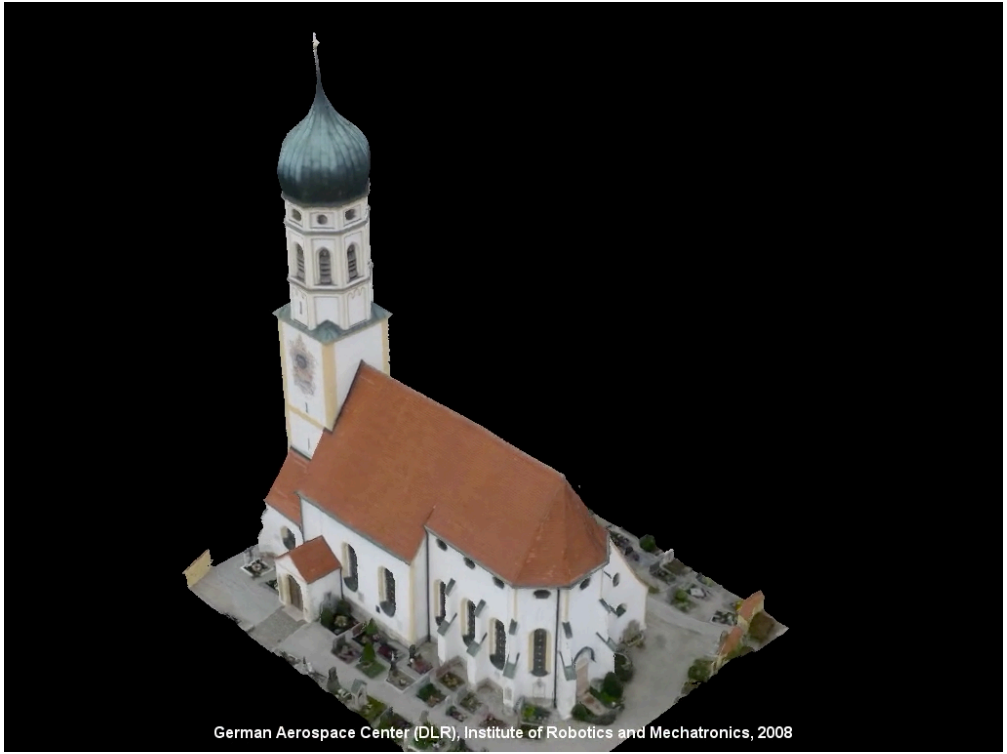
input point cloud



recognized models

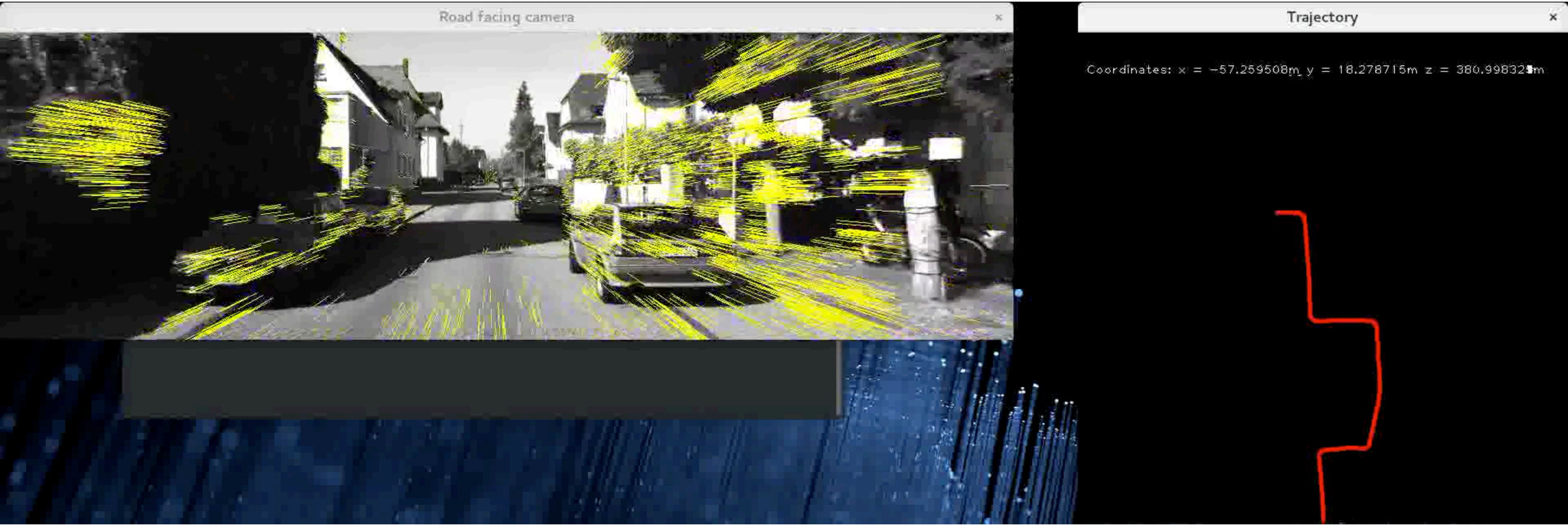
ObjectRANSAC system fitting 3D models into cluttered scenes (Papazov et al. 2010)

We used to reconstruct static scenes from monocular in 2007… (with DLR)



Accuracy:1.5cm

120fps Monocular Navigation from Sparse Optical Flow



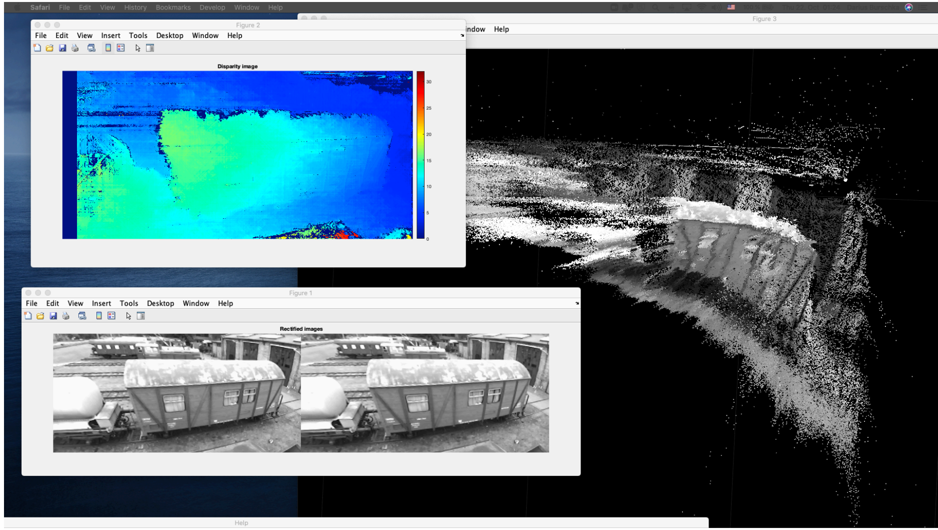
GPU implementation of sparse flow (feature-based OpenCV) system using only 10% of the resources

Benchmarks in the Age of Deep Learning…

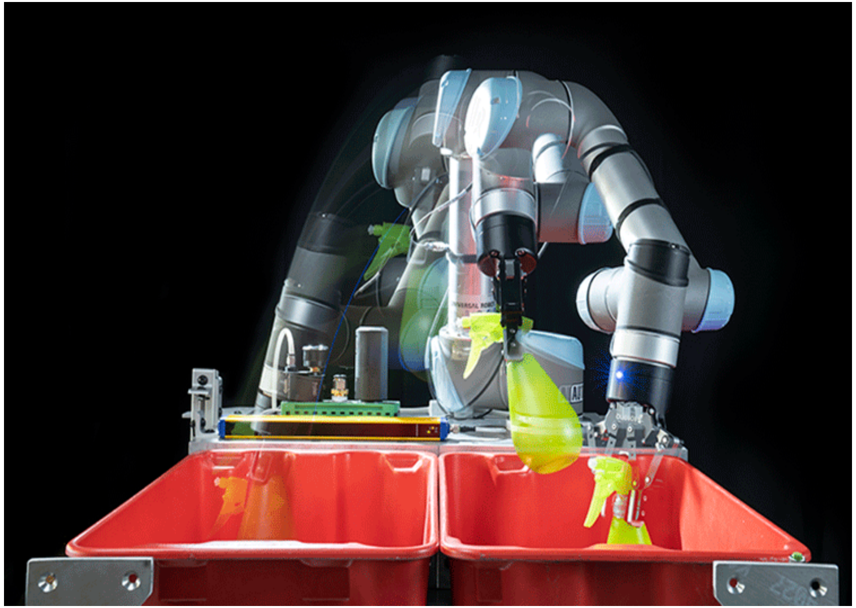
The conventional systems presented had a metric accuracy of a 1-3 cm and 0.1-0.2 degrees. The current deep learning systems switched to percent.

This means that a 95% accurate visual-navigation system appears very accurate at the first glance, but it is often based on a metric that the metric accuracy was 95% of the cases under 15cm and the rotational error was under 10degrees… The missing metric and angular value prevents any useful integration…

Visual qualitative evaluation replaced often a quantitative evalution.



Combination DL and Conventional Methods



Recent UC Berkley result

DL can deliver:

- useful path in an environment used in the training set
- good path approximation in similar environments, which still need to be refined
- No useful results in novel environments

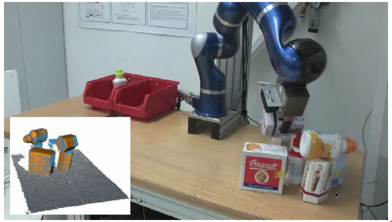
The metric high-accuracy refinement is done with conventional methods

Deep Learning is not AI but Artificial Experience

- Deep Nets provide a powerful method to compare current view to the previously seen examples in the training set. It is a powerful method to train the robot to operate in a specific know domain without specification of features.
- The network does not extrapolate well into regions without training samples. Variational extensions to improve the gradient field of, e.g., Variational Autoencoder (VAE) improve but only in local neighborhoods
- Deep Net can be compared to a librarian, who does know what are the most similar books to the currently parsed text, but who does not know how to apply the knowledge for novelty - not Intelligence, but Experience

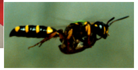
Research of the MVP Group

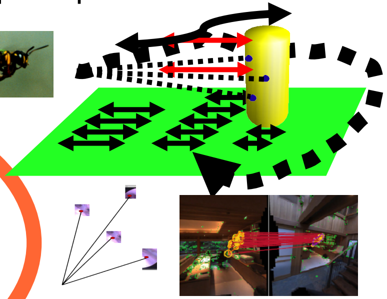
Perception for manipulation



Visual navigation

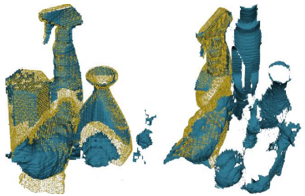
Biologically motivated perception





The Machine Vision and Perception Group @TUM works on the aspects of visual perception and control in medical, mobile, and HCI applications

Rigid and Deformable Registration

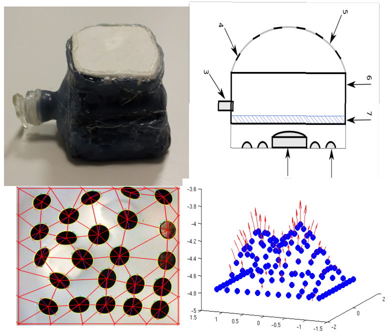


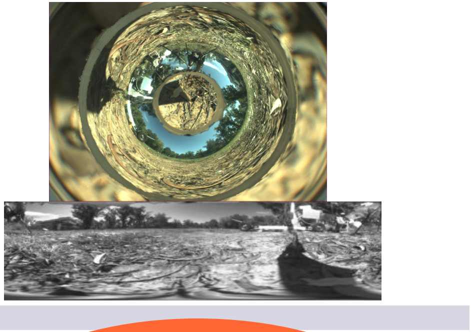
Photogrammetric monocular reconstruction

Visual Action Analysis

Research of the MVP Group

Sensor substitution





Development of new
Optical Sensors

The Machine Vision and Perception Group @TUM works on the aspects of visual perception and control in medical, mobile, and HCI applications

Multimodal Sensor
Fusion

Exploration of physical
object properties