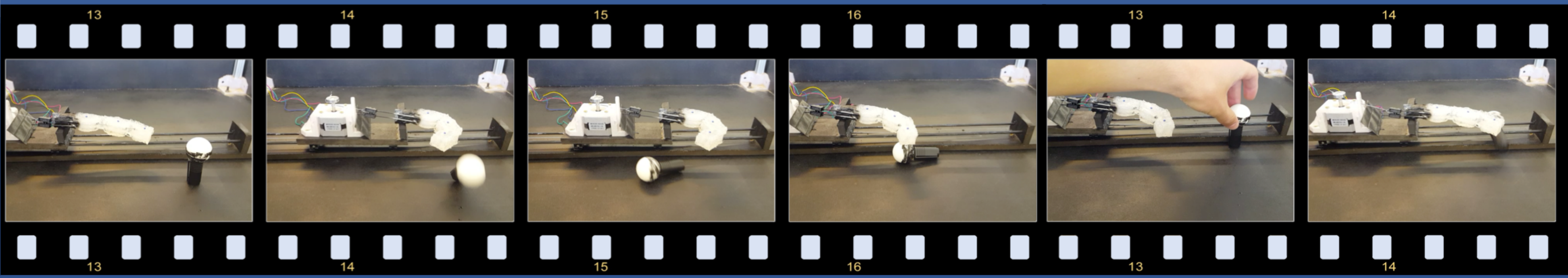
Data Driven Soft Robotics

Fredrik Holsten[1], Sune Darkner,[1] Morten Pol Engell-Nørregård[2], Kenny Erleben[1]
[1]Department of Computer Science, University of Copenhagen, Denmark
[2] Evil Hippie Design



INTRODUCTION: Soft robots are attractive because they have the potential of being safer, faster and cheaper than traditional rigid robots. If we can predict the shape of a soft robot for a given set of control parameters, then we can solve the inverse problem: to find an optimal set of control parameters for a given shape.

WHAT IS THE THEORY BEHIND IT?

1) SHAPE VECTOR EXTRACTION FROM POINT CLOUD

- Visual markers on surface of robot
- Multiple depth cameras for 360-view

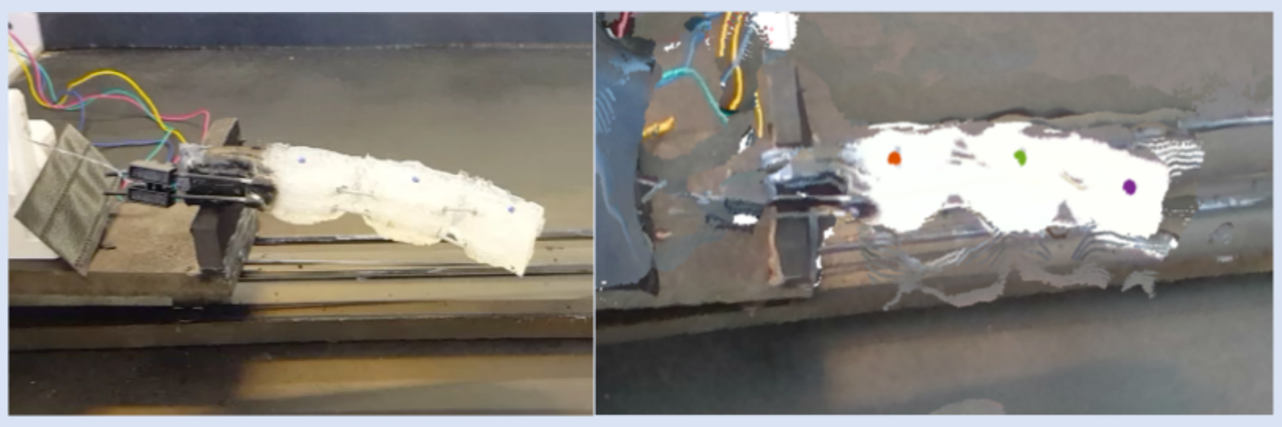


Figure 1: Sample robot shapes for different control parameters

2) LEARN SHAPE FUNCTION

Shape function $\vec{s}(\vec{\alpha}) \approx \vec{s}_0 + J\Delta\vec{\alpha}$

Control parameters $\vec{\alpha}_k = [\alpha_0^k, \alpha_1^k, \ldots, \alpha_P^k]^T$

Shape vector $\vec{s}_k = [x_0^k, y_0^k, z_0^k, x_1^k, y_1^k, z_1^k, ..., x_N^k, y_N^k, z_N^k]^T$

Displacement vector $\vec{u}_i = \vec{s}_i - \vec{s}_0$

$$A = [\vec{a}_1 \vec{a}_2 \cdots \vec{a}_{K-1} \vec{a}_K] \quad U = [\vec{u}_1 \vec{u}_2 \cdots \vec{u}_{-1} \vec{u}_K]$$

Jacobian $J = UA^T(AA^T)^{-1}$

- Trivial to increase order of approximation
- Split configuration space into disjoint regions
- Embarrassingly parallel
- 5-fold cross-validation for optimal approximation order vs number of local models

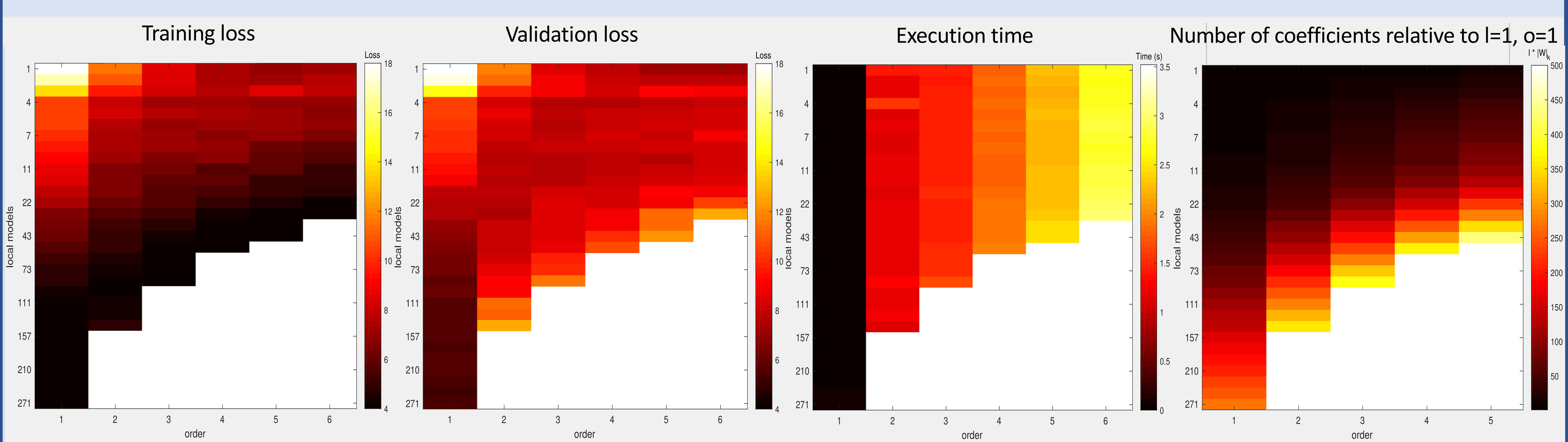


Figure 2: Many low ordered, local models show promising results in terms of time complexity and validation loss

3) INVERSE KINEMATICS

Solve inverse kinematics to find optimal control parameters for desired shape $\vec{s}_{goal}$

$$\vec{\alpha}_{goal} = \arg\min_{\vec{\alpha}} \frac{1}{2} \|\vec{s}(\vec{\alpha}) - \vec{s}_{goal}\|^2$$

HOW DO WE EXECUTE IT?

THE LEARNINGCUBE

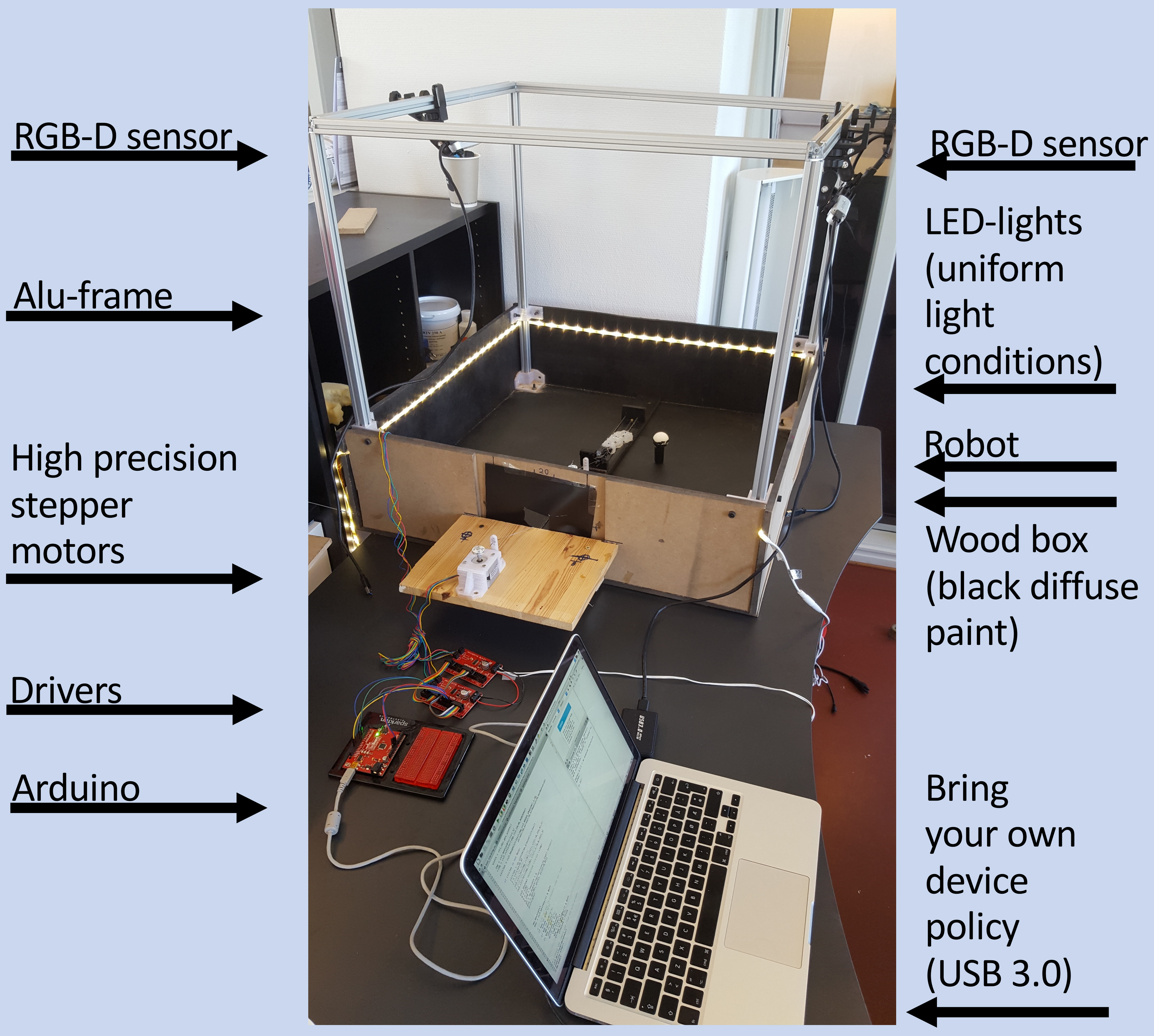


Figure 3: Overview of hardware setup

pySoRo

- Software for communicating with motors and cameras
- Camera calibration and noise analysis
- Postprocessing of data
- Various methods for learning the configuration function
- real time interaction of robot with environment, using learned function
- Documented
- Publicly available on GitHub!*
- github.com/erleben/pySoRo

INFO

- price for complete setup: < 1300 EUR
- contact us if you are interested

THE ROBOTS

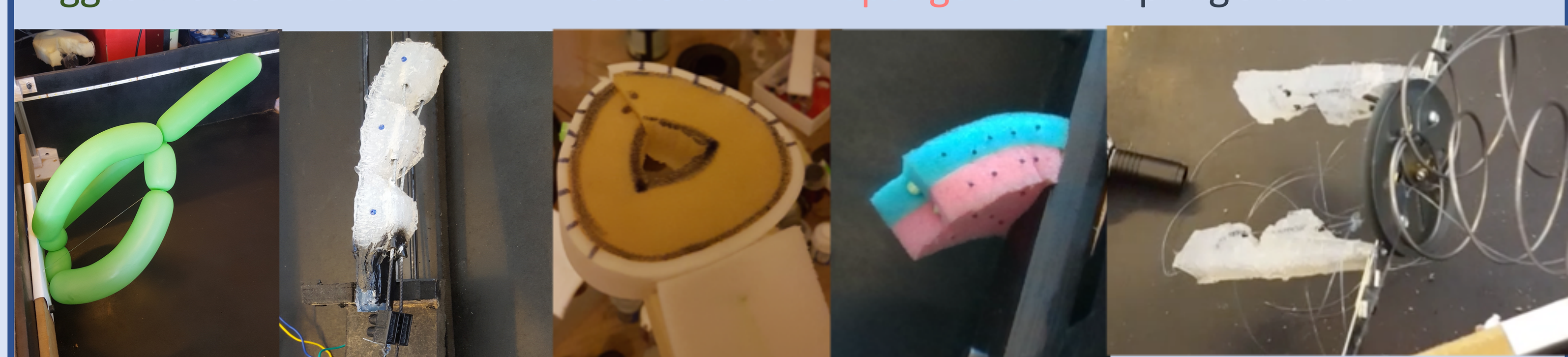


Figure 4: The robots used

UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Scan QR-code to see the robots in action on YouTube!

*Scan QR-code to check out pySoRo on GitHub!

Fredrik Holsten
Master student, DIKU
npb574@alumni.ku.dk

FUTURE WORK:

- Increase complexity of robots
- Time integration and path planning
- Task execution