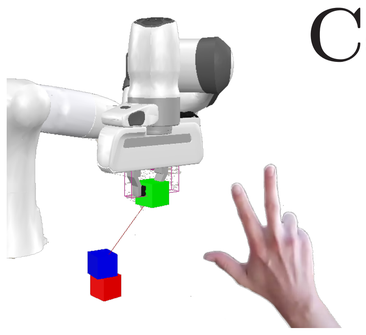


Controlling Robotic Manipulations via Bimanual Gesture Sequences

Petr Vanc, Jan Kristof Behrens, and Karla Stepanova

Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague

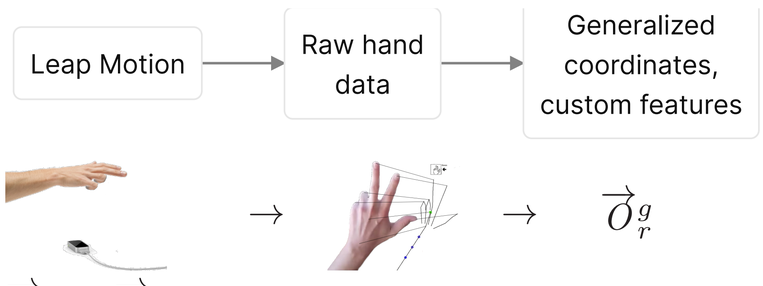
petr.vanc@cvut.cz, jan.kristof.behrens@cvut.cz, karla.stepanova@cvut.cz

CZECH INSTITUTE OF INFORMATICS ROBOTICS AND CYBERNETICS CTU IN PRAGUE

Abstract

- Probabilistic gesture detection framework enables using a static and dynamic gestures to literately parameterize and trigger robot actions.
- Interactive specification of new gestures and their grounding to robot motion skills or parameters (e.g., goal pose, speed, distance, etc.) supported by a GUI.
- Demonstration of the proposed system in simulated blocks world experiments.

Dataset generation



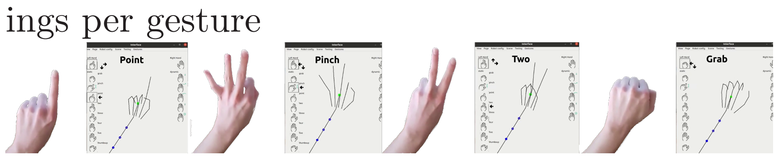
$\vec{D} = \{\vec{D}^g_r\}$, r goes over recordings per gesture, g goes over available gestures

$\vec{D}^g_r = [\vec{O}^g_{r1}, \ldots, \vec{O}^g_{rT}, l_g]$, l_g is label for the gesture g, T length of the recording.

Observation $\vec{O}^g_{rt}$ is a set of hand parameters:

- static gesture $\vec{O}^g_{rt} = [a_1, \ldots, a_n, d_1, \ldots, d_j]$, where $a_1, \ldots, a_n$ are bone angles of the hand, $d_1, \ldots, d_j$ are custom features
- dynamic gesture $\vec{O}^g_{rt} = [p_1, \ldots, p_m]$, where p_i is the hand position

Sample dataset: 8 static, 5 dynamic, 60 recordings per gesture



Robot Actions

Robot actions **R**: Represented as probabilistic motion primitives

- Learned from the demonstrations
- Conditioned via context variables (start/end, speed, etc.)

Observed state:

→ Used to parameterize robotic actions

- Focus object (e.g. specified by user)
- Robot state
- Specific parameters (e.g. *pinch* distance)

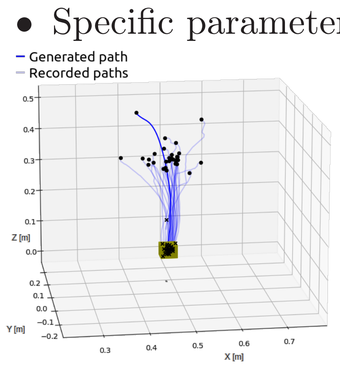


Fig. 9: *Touch*

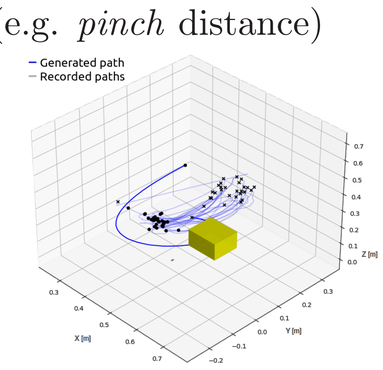


Fig. 10: *Kick*

Example: *Kick* Motion primitive condition (time $t = 0 - 100\%$):

- Start (t=0%) given by end-effector position
- Position and direction at time $t \in (65\%, 75\%)$ is given by the pose of the focused object and direction towards it

System for continuous Gesture Detection

- User defined mapping of gestures to robot actions
- Real-time estimation of the current state (evaluation of recognizers for all gestures)

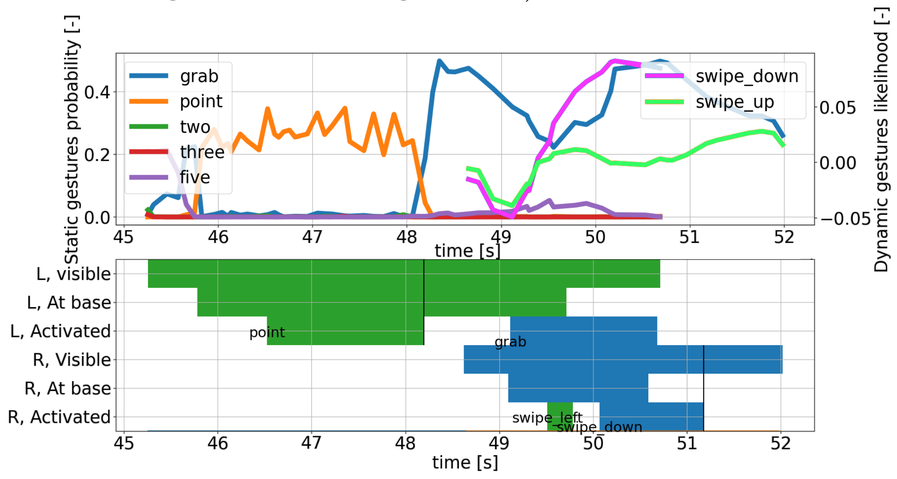


Fig. 1: Real recognition output from two sources of detection. (top) History of confidence level for individual gesture. (bottom) Activation of gestures using summed probability over time window and hand position. Static gesture *point* and dynamic gesture *swipe down* were detected and triggered. Gesture *grab* detected, connected robot action not triggered.

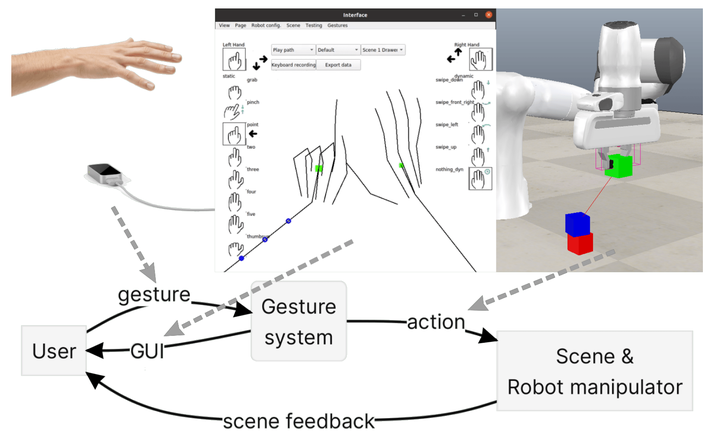


Fig. 2: Human-Robot control loop diagram.

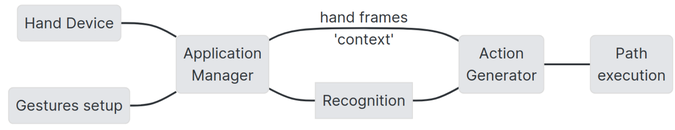


Fig. 3: The system diagram displayed in ROS. Implementation as independent nodes.

Set of Recognizers

Gestures **G**: We consider three types of hand gestures.

- **Static** (e.g. *grab*)
- **Dynamic** (e.g. *swipe right*)
- **Compound** (e.g. *grab* + *swipe right*)

Recognizers:

- Deterministic (static + dynamic)
- Convolution NN (static + dynamic)
- Probabilistic NN (static)
- Dynamic Time Warping (dynamic)

Recognizer benchmarks: From set of recognizers, we picked best performing method for static and dynamic type and constructed confusion matrix visualizing balanced accuracy.

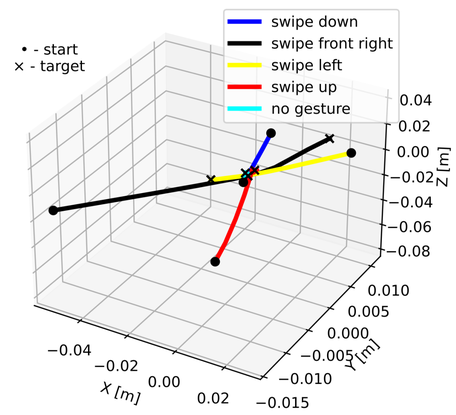


Fig. 4: ProMP representation (mean visualized) of dynamic swipe gestures (hand palm positions).

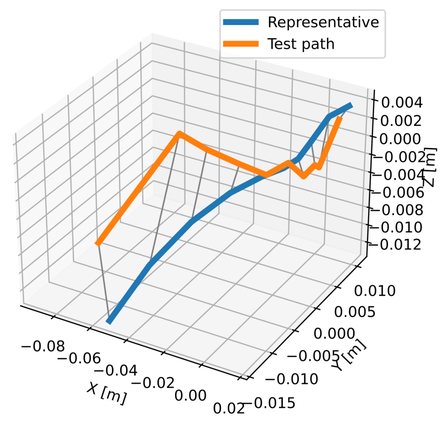


Fig. 5: Recognition of Dynamic gestures by DTW. Observed data are compared to gesture path representation.

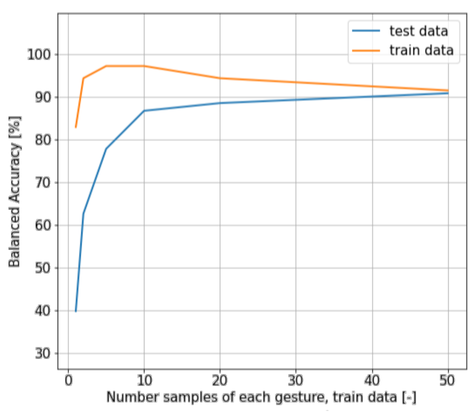


Fig. 6: **Learning progress** for static gestures (Probabilistic neural network).

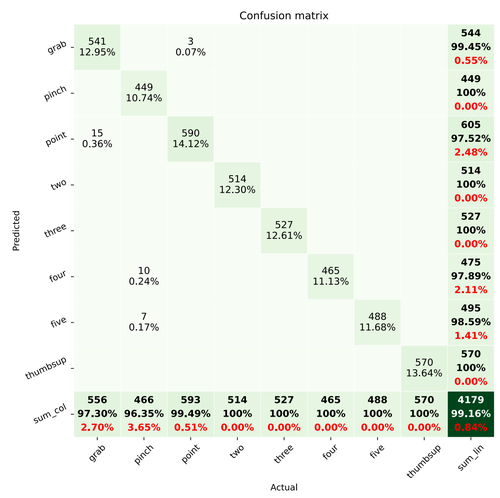


Fig. 7: **Static gestures** - Probabilistic neural network.

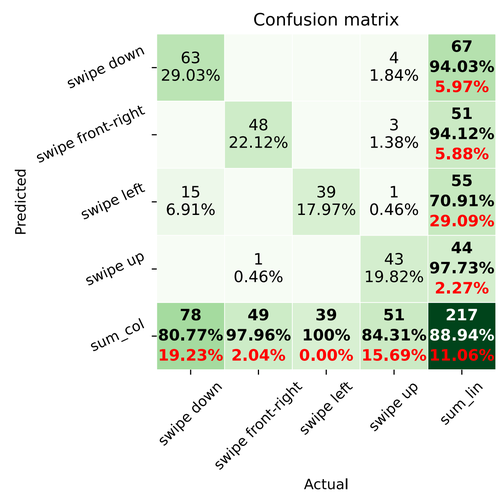


Fig. 8: **Dynamic gestures** - Dynamic Time Warping.

Gestures set & Mapping example setup

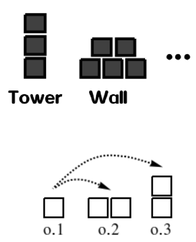
Custom mappings can define various scenarios. Sample build scenario mapping:

Left Hand:

- *Grab* → close *gripper*
- *Thumbs-up* → open *gripper*
- Open fingers (*point*, *two*, …, *five*) → make *focus* on object with (id="number of opened fingers")

Right Hand:

- *Swipe down* → *Touch* focused object (uses Structures helper class *)
- *Swipe up* → *Go to Home* position
- *Swipe front-right* → *Kick* focused object



*) Structures helper class decides based on context (scene objects, build configuration, attached object), target position

Project webpage & Acknowledgements

P.V. by CTU Student Grant Agency (reg. no. SGS21/184/OHK3/3T/37); J.K.B. was supported by the European Regional Development Fund under project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15 003/0000470); K.S was supported by the Czech Science Foundation (project no. GA21-31000S).

Gesture toolbox is open source on: https://github.com/imitrob/teleop_gesture_toolbox

Imitation Learning group: http://imitrob.ciirc.cvut.cz