LEARNING MY ROBOT TO GRASP
WITH PROMISES AND PUSHTSTREAMS

TIM VERBELEN
Senior Researcher
imec – Ghent University
MY ROBOT
TEACHING MY ROBOT
AND ... IT WORKS ... NOT :-(
AND ... IT WORKS ... NOT :-(
LET’S TAKE CONTROL!

Intel Realsense D435

NVIDIA Jetson TX2

Franka EMIKA
Controlling the robot arm
public interface Arm {

  // set position for an arm joint
  void setPosition(int joint, float position);

  // set positions for all joints
  void setPositions(float... positions);

  // move arm tip to a point in cartesian space
  void moveTo(float x, float y, float z);

  ...
}
public interface Arm {

    // set position for an arm joint
    Promise<Void> setPosition(int joint, float position);

    // set positions for all joints
    Promise<Void> setPositions(float... positions);

    // move arm tip to a point in cartesian space
    Promise<Void> moveTo(float x, float y, float z);

    ...
}
A PROMISING ROBOT ARM API

@Component
public class MyController {

    @Reference
    private Arm arm;

    public void doSomething(){
        arm.openGripper()
            .then(p -> arm.moveTo(0.3f, 0.0f, 0.25f))
            .then(p -> arm.setPosition(4, 1.57f))
            .then(p -> arm.moveTo(0.3f, 0.3f, 0.09f))
            .then(p -> arm.closeGripper());
        // immediately returns a Promise
    }
}
Processing camera frames
THE CAMERA

public interface Camera {
    // capture a frame with the camera
    Frame getFrame();
}

public interface CameraListener {
    // callback with new frame
    void newFrame(Frame f);
}
List<Person> guests = getWeddingGuestList();

// How much wine will we need to buy?
long adults = guests.stream()
    .map(Person::getAge)
    .filter(age -> age > 21)
    .count();

public interface Camera {
    // stream camera frames
    ?? stream();
}

example from https://blog.osgi.org/2018/06/osgi-r7-highlights-push-streams-and.html
OSGI PUSH STREAMS

PushStream<Person> guests = getWeddingGuestList();

// How much wine will we need to buy?
Promise<Long> adults = guests
    .map(Person::getAge)
    .filter(age -> age > 21)
    .count();

public interface Camera {
    // stream camera frames
    PushStream<Frame> stream();
}

example from https://blog.osgi.org/2018/06/osgi-r7-highlights-push-streams-and.html
NEURAL NETWORK
A DATA-HUNGRY PATTERN DETECTOR

Neural Network

“cat”
CLOSED-LOOP CONTROL GRASP

Camera camera;
NeuralNetwork nn;
ArmController controller;

def publicPromise<Promise<Boolean>> grasp()
{
    return camera.stream() // returns PushStream<Frame>
        .timeout(Duration.ofSeconds(10)) // timeout after 10 seconds
        .map(frame -> nn.forward(toTensor(frame))) // process camera frame with neural network
        .map(r -> controller.update(r)) // use result to update controller
        .anyMatch(s -> s); // short circuit in case of success
}
Training a neural network
DATA AUGMENTATION

positive samples

negative samples
TRAINING IN THE CLOUD WITH REMOTE SERVICES

Intel Realsense D435

Intel Realsense D435

Remote Services

NVIDIA Jetson TX2

NVIDIA Jetson TX2

Franka EMiKA

Franka EMiKA
THANK YOU

Special thanks to:

- **Tim Ward** for Promises and Push Streams
- **Pieter Van Molle & Elias De Coninck** for feeding the robots
REFERENCES

Promises
- Specification https://osgi.org/specification/osgi.cmpn/7.0.0/util.promise.html
- Download http://central.maven.org/maven2/org/osgi/org.osgi.util.promise/1.1.0/org.osgi.util.promise-1.1.0.jar

Push Streams
- Specification https://osgi.org/specification/osgi.cmpn/7.0.0/util.pushstream.html
- Download http://central.maven.org/maven2/org/osgi/org.osgi.util.pushstream/1.0.0/org.osgi.util.pushstream-1.0.0.jar
embracing a better life