MAKING OPEN SOURCE ROBOTICS APPROACHABLE -- THE FUTURE IS NOW!

James Ketrenos

Embedded Linux Conference, Dublin, 2015

I, ROBOT: A Pseudo History

A little bit about me...

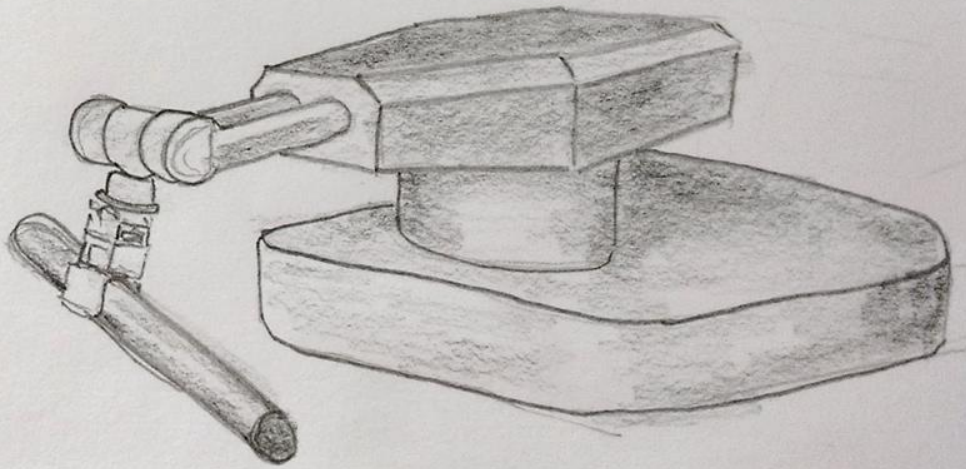
Intel
Open Source Technology Center
Portland, Oregon

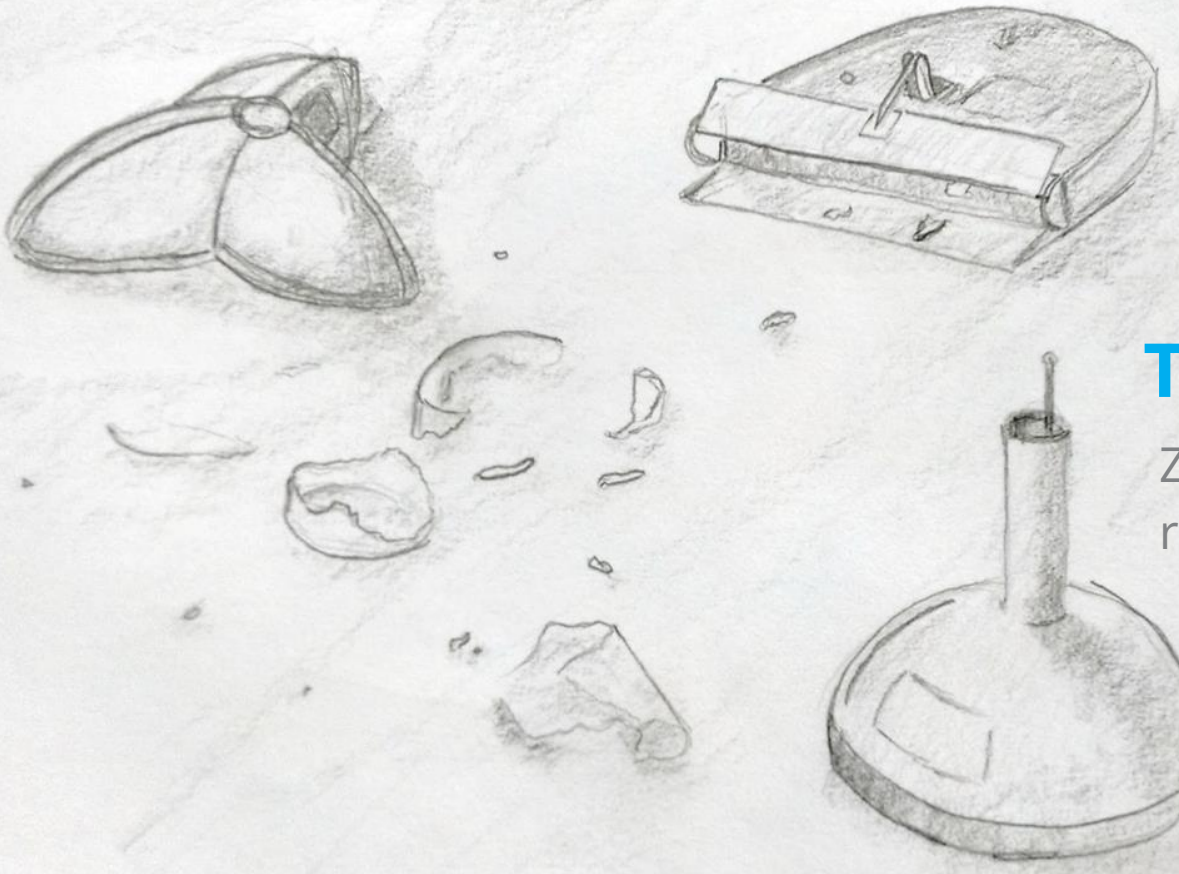
Where are all the robots?

“A robot is machine capable of carrying out a complex series of actions automatically.” - the Internet

1950s saw the first industrial robot by Unimate.

- Programmed with complex interactions
- Weighed 4000 pounds
- Retail price of $25,000. Adjusted for inflation, that would be 200,000 US dollars, or 180,000 euros today.

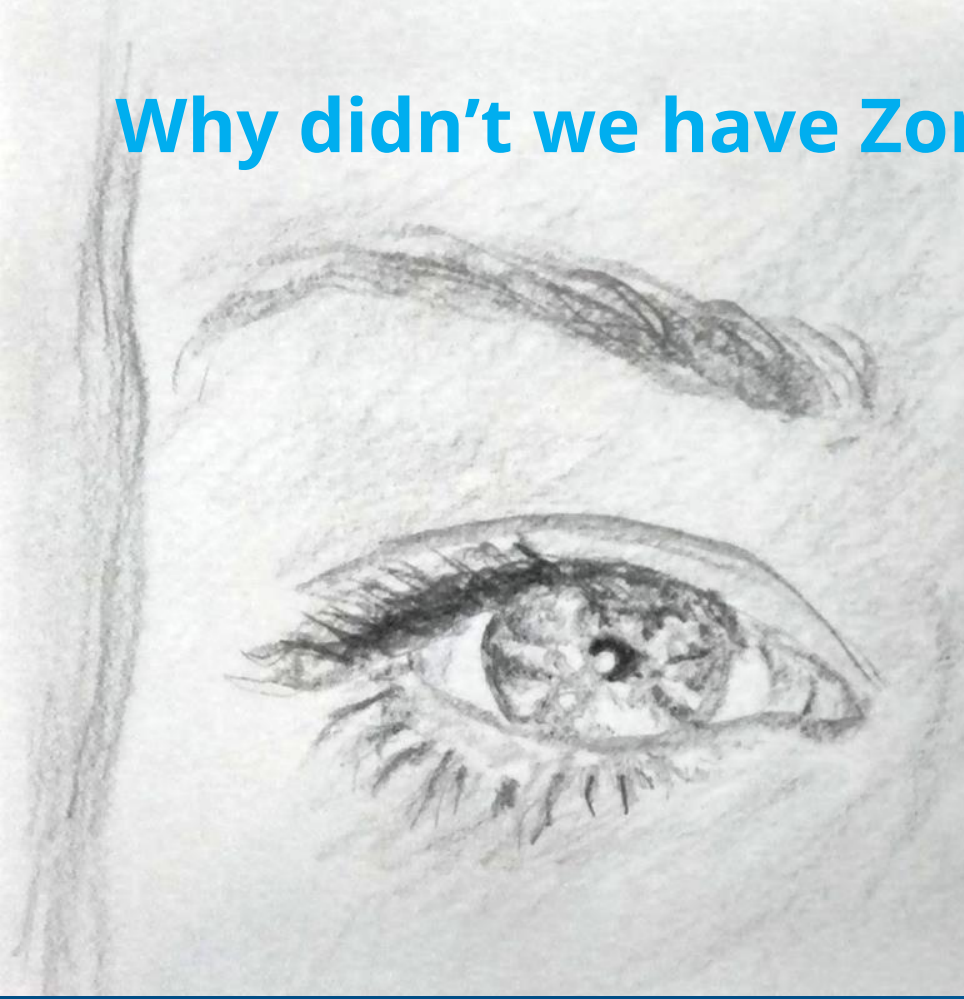




The Fifth Element

Zorg's automatic cleaning robots

Why didn’t we have Zorg robots in the 90s?

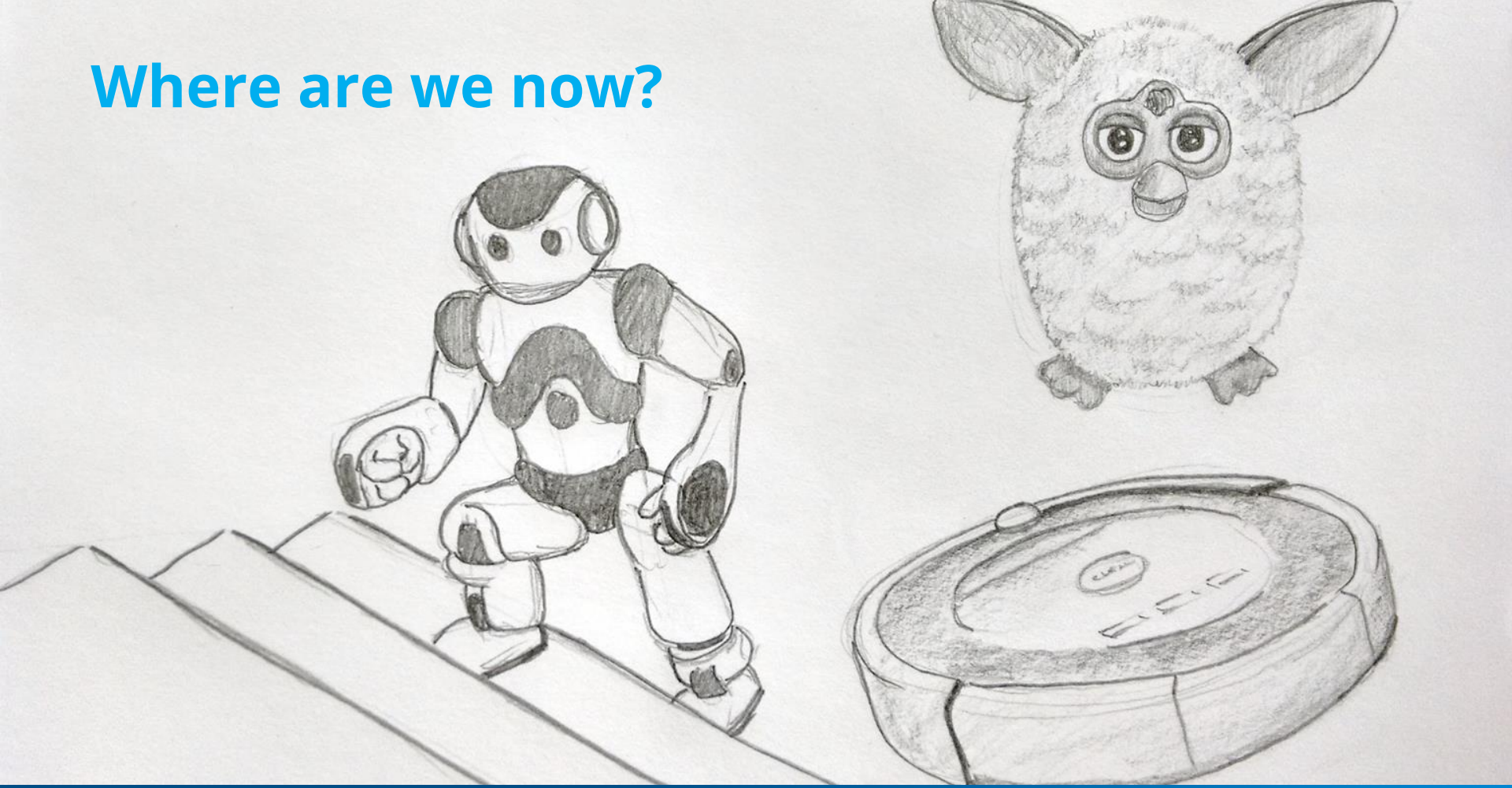


Robots were not aware of their environment--they lacked **vision**.

- Mobile computers weren’t fast enough
- Viable software algorithms didn’t exist

That started changing as we entered the 21st century.

Where are we now?



THE FUTURE IS NOW.
(LET'S BUILD A ROBOT)

What do you want to work on?

Innovate in areas you are interested in.

Build from scratch
$100 - $1,000s

- Electrical skills
- Mechanical skills
- Kernel drivers
- User space software

Robot kit
$100 - $1,000s

- Some electrical skills
- Some kernel drivers
- User space software

Functional robot
$80 - $30,000

- User space software

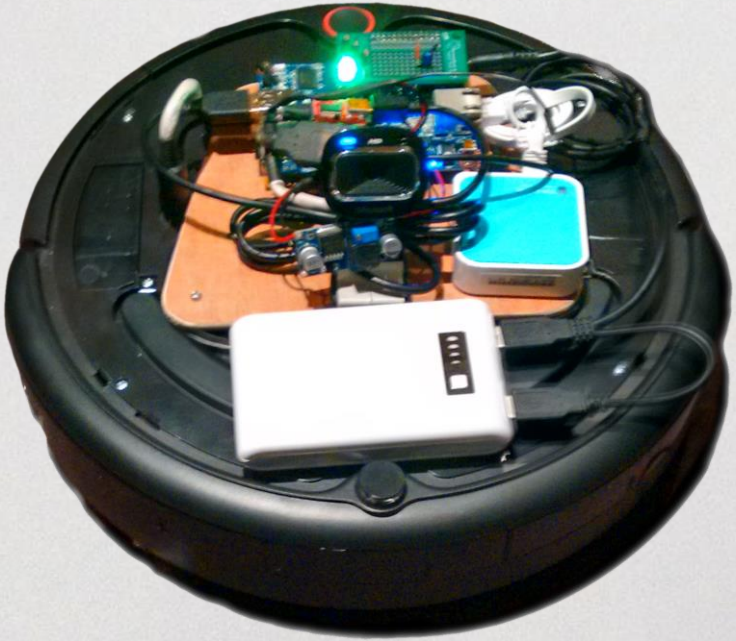
Expanding on success is more rewarding than fighting with failure. Start easy, get something working, and then expand into new areas.

Vision Robot Usages

There are many examples of vision related functionality on the Internet. Vision combined with a robot enables usages like the following:

- **Track Me** -- follow a face around; try to keep it centered
- **Ball Chase** -- track a ball on the ground; attempt to hit it (into a goal?)
- **Mapping** -- move around; create a point cloud of the environment

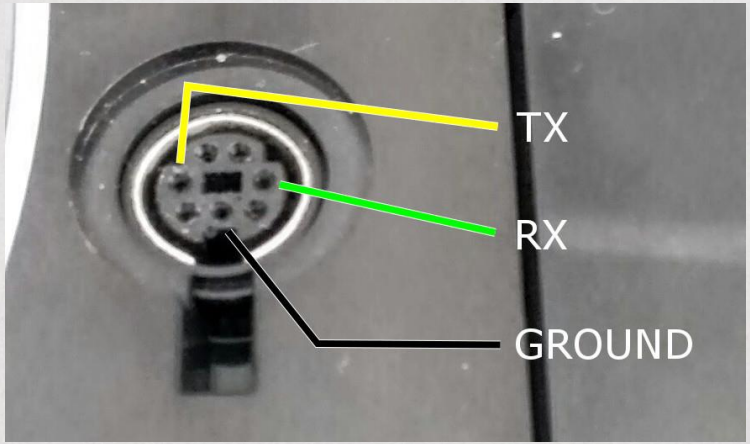
Computer Vision Robot



- MinnowBoard Max*
- iRobot* Roomba*
- SparkFun* USB-FTDI breakout board
- RGB web camera
- USB charger (capable of 2.5A output)
- Double-sided sticky tape
- MicroSD card (16G)
- Wireless Access Point**
- DC-DC voltage converter**
- Tadpole Lure**
- Mounting platform**

*Other names and brands may be claimed as the property of others
** Optional components

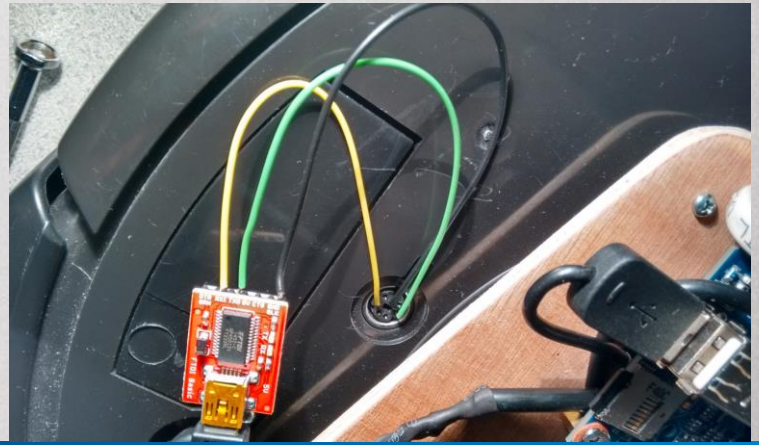
Wiring to the Roomba



Pins on Roomba are +5V!

- TX: Pin 4 => FTDI RXI
- RX: Pin 3 => FTDI TXO
- GROUND: Pin 6 => FTDI GND

MinnowBoard Max is **NOT** 5V tolerant! You must shift your voltages, or use a USB adapter like below.

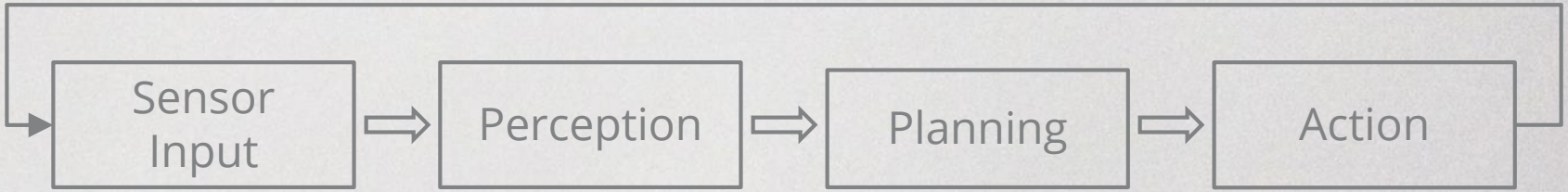


Operating System

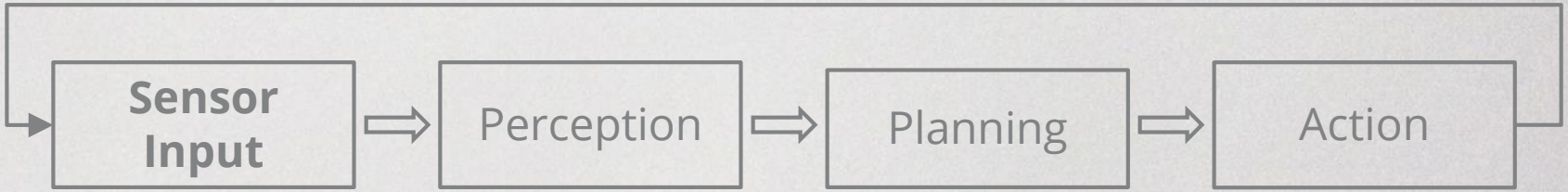
Ubuntu 14.04 LTS

- Supported by lots of software packages
- Initial install using desktop version was easy and “just worked”
- Pruning back and using an attempted non-graphical (server) install ran into issues
- Robot OS (ROS) supports Ubuntu
- Lots of community support available

Robot Functional Flow



Robot Functional Flow



- Roomba sensors
- Mono RGB

- irobot.js
- OpenCV

Sensor Input

Node.js snippet showing camera and sensor initialization

```
var cv = require(‘opencv’), ir = require(‘irobot’);
var cam = new cv.VideoCapture(0);
cam.setWidth(320);
cam.setHeight(240);
...
var robot = new ir.Robot('/dev/ttyUSB0', { baudrate: 115200 }); /* spec say 57600!! */
robot.on(‘ready’, function() {
     cam.read(detectFaces);
});
robot.on('sensordata', function () {
     updatePlan({sensors: robot.getSensorData()});
});
```

Robot Functional Flow

- Javascript state machine
- Searches for faces in image
- Smoothing / filtering on world data

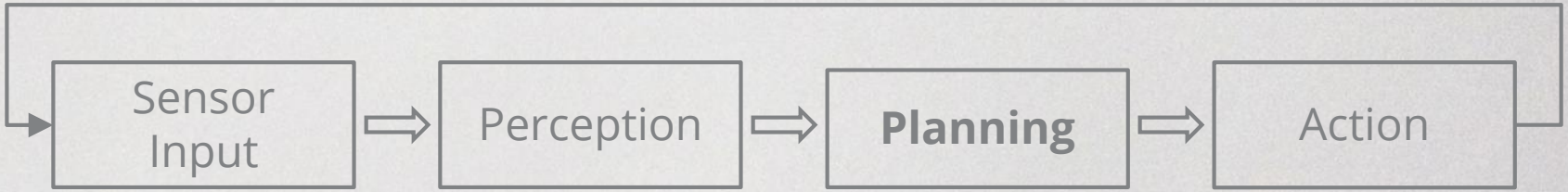
- node-opencv

Perception

Node.js snippet showing face detection

```
function detectFaces(err, image) {
    image.detectFaces(cv.FACE_CASCADE, {}, function (err, faces) {
          if (err) throw err;
          var largest = null;
          faces.forEach(function (face) {
               face.size = face.width * face.height;
               if (!largest || face.size > largest.size) largest = face;
          });
          updatePlan({face: largest});
          cam.read(detectFaces);
    });
});
```

Robot Functional Flow



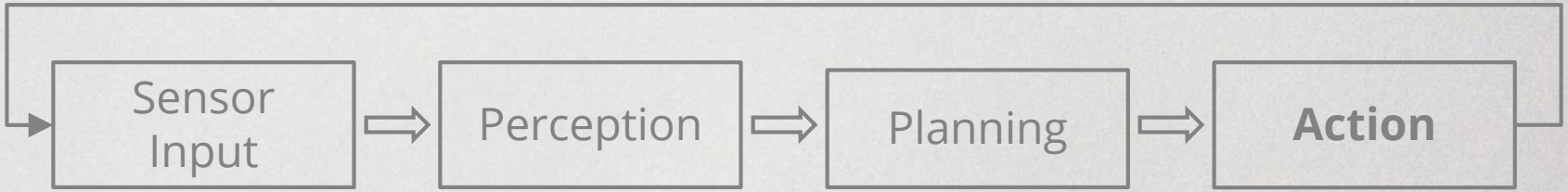
- Javascript state machine

Planning

Node.js snippet showing planning

```
function updatePlan(update) {
        ...
        if (!update.face) {
            requestSpeed(0, 0); /* This should go into “seek” mode instead */
            requestLED(0); /* No face; turn off the LED indicator */
            return;
        }
        /* calculate how far ‘face’ is from center of image and determine optimal
         * motor speeds to move robot in direction to center face */
        ...
        requestSpeed(left, right);
        ...
```

Robot Functional Flow



- Javascript state machine
- Send commands to Roomba
- Turn on LED

- irobot.js
- mraa

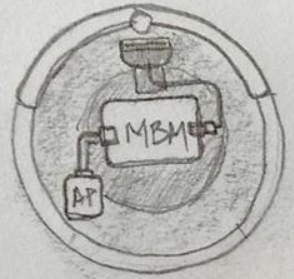
Action

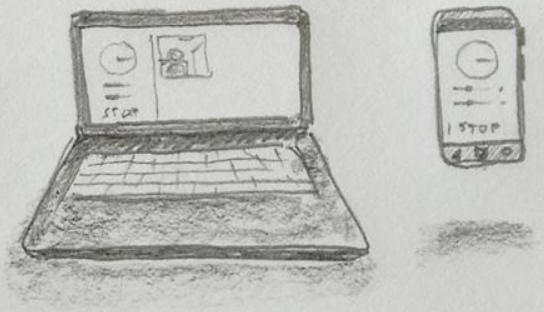
Node.js snippet showing requested plan being pushed to robot

```
function requestSpeed(left, right) {
          /* Validate requested speeds make sense given current
           * sensor wall and cliff values, adjust accordingly */
          …
          /* Send the new speed values to the robot */
          robot.drive({left: left, right: right});
}
```

DEMO

Remote control





MinnowBoard Max

- Access Point (AP)
- DHCP
- Node.js server
- nginx web server

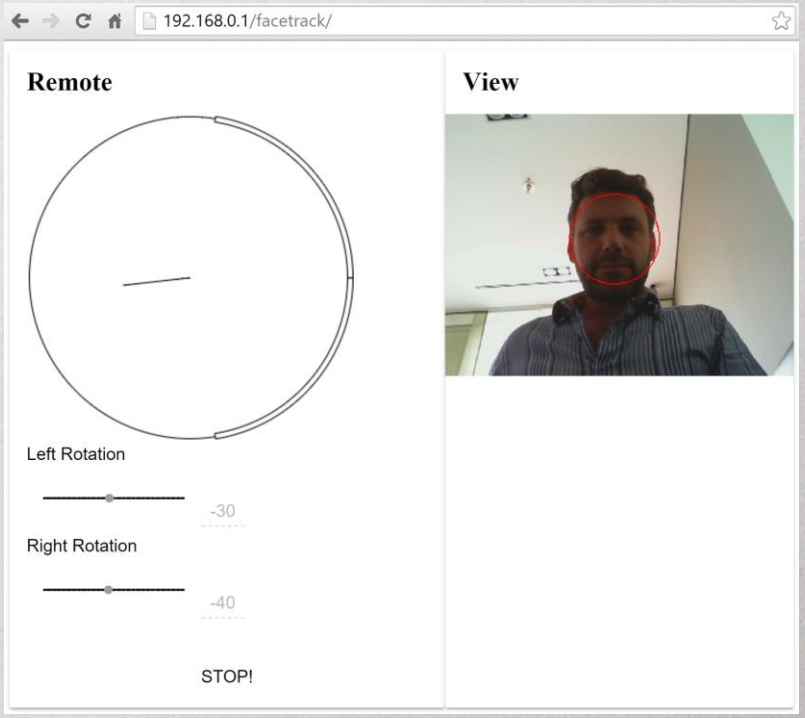
Remote client (laptop, phone, etc.)

- Connects to AP
- Opens http://192.168.0.1/facetrack

Remote control client

HTML5 Application

- Uses websockets to communicate with server on MinnowBoard Max
- Receives frames and displays them
- Has a simple interface for manual remote

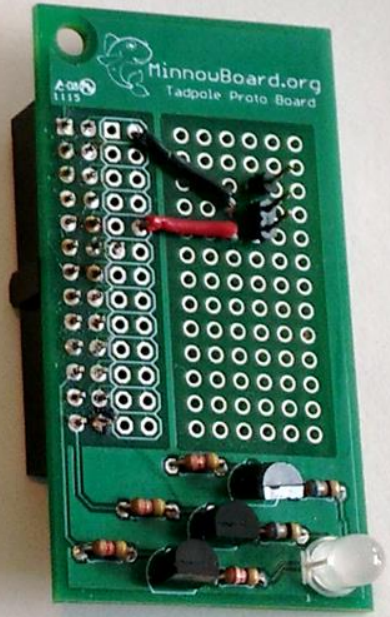


LEARNINGS

What's your IP address?

When developing over Ethernet, you might not know the robot's IP address. I built a small circuit to tell me the IP address.

- At system boot, begin watching a GPIO pin
- If pin is jumpered, begin encoding the IP address into a series of blinks
- Implemented as a bash script
- Built on a Tadpole lure, but circuit is simple enough to use wire twists and tape
- Same LED is used to indicate when face is found



The lag is killing me!!!

When running the face tracking application, there was a 5 second lag!

- Video frames are captured into buffers @ 30FPS
- 2 to 16 buffers, frequently 10
- OpenCV and Video4Linux2 pull from buffers as FIFO
- Face tracking reduces frame consumption rate to ~2FPS
- A 10 frame buffer will take 5 seconds to consume

Solution?

- Patch OpenCV (or node-opencv) to operate as a LIFO
- Background thread consumes frames from camera, dropping old frames

A logic analyzer is your friend

The Roomba wasn't communicating. It was only through the use of a logic analyzer that I realized the Roomba was transmitting at 115200 baud, not the documented 57600.

USB logic analyzers can be found for around $100US, maybe even less.

I used the Saleae Logic 4.

Cameras

A few things to look for in a web camera:

- Wide angle is better; too narrow a field of view means you will need to stand directly in front of the robot, or it will have to seek a lot more
- Rolling shutter vs. global shutter
- Exposure and dynamic range

Calibrate your camera and correct frames before processing.

WHAT'S NEXT?

Ball chasing

Use color and shape to track a ball.

- OpenCV has all the capabilities you need to detect a ball
- node-opencv doesn't expose all of OpenCV
  - Requires native development (C/C++)
- Use the same Planning and Action to chase the ball
- Track a moving ball and add target prediction
- Sew shark costume for cat
- Drive cat around kitchen

PTAM: Parallel Tracking and Mapping

Calculates the camera pose in the world allowing routing and navigation

- SLAM technique that uses a single RGB camera
- Current open source (GPLv3) version built and tested on robot
- Suffers the same lag problem as node-opencv

Robot OS

ROS combines many libraries and capabilities into a rich robot framework.

- SLAM mapping
- Point Cloud
- Navigation and planning
- C, C++, and Python

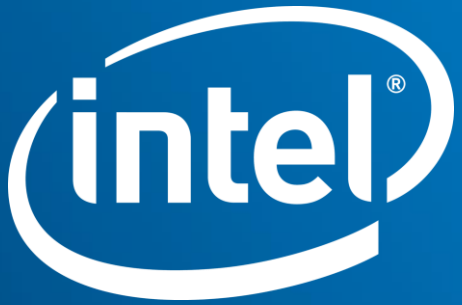
Join the Journey

https://01.org/developerjourney

- Step-by-step guide to build this robot
- Developer posts describing problems encountered, trials, errors, and solutions
- Links to source code
- Opportunity to ask questions or provide feedback
- Free cookies
- (that was an Internet joke)

THANK YOU :)

james.p.ketrenos@intel.com


intel®

Abstract

Robots and multi-rotor coptors have made their way into our lives. Whether it's a robotic vacuum cleaner or a first-person-view quad-copter racing through trees, today's devices are the toys we always dreamed of having as kids. Computational performance, power utilization, thermals, and weight have all reached a tipping point where we can now feasibly build and deploy intelligent robotic devices to improve our lives: they can now see, hear, and interact with the world.

Attend this presentation to get a introduction to the general problems in maker robotics and learn about the open source projects which have emerged over the last few years in an effort to bring robotics to the masses. Building a robot is no longer something that takes years of research--the technologies and capabilities previously only available in science fiction are now at your fingertips.