Overview

- **Horizon 2020**, EU Framework Programme for Research and Innovation
- 2014-2020
Eyes of Things, H2020 Innovation Action

- Topic: Smart cyber-physical systems
- Innovation-Type Action
  
  R&I Action: “Novel solutions to old applications”

  **Innovation Action:** “Novel applications with old solutions”

- Started Jan 1st, 2015. Budget, 4M€
“Eyes of Things” Consortium

EOT
EYES OF THINGS

evercam.io
Movidius
THALES
AWAIBA
VISILAB
DFK
nvISO
fluxguide
Motivation and Context

Vision is our richest sensor

One of the most complex tasks for both humans and machines

Proven success in ‘machine vision’ context, i.e. factory automation, inspection,...
Motivation and Context

CV is going out-of-factory ...
  • Microsoft Kinect
  • Google Project Tango
  • Google Glass

Lots of new developments
  • Drones
  • ADAS
  • Multispectral, 3D, ...
  • Mobile imaging
  • Deep learning
• Many innovative ‘out-of-the-factory’ CV applications developed for smartphones, which are cheap, easy-to-use...
Motivation and Context

• But smartphones are not appropriate for many applications
  • Wearable for the EOT? No
  • Low power consumption? No
  • Optimized for CV? No

• No flexible open platform for mobile embedded vision is currently available

• New-platform needed
Other devices?

- IoT devices:
  - WaRPboard, Ambarella, Intel EDISON, Ingenic Newton2, Nixie...

- Most of these systems process data from scalar sensors
- Some can be attached to cameras, but not do CV
- NONE has been designed from bottom-up with vision in mind!
Challenges

Efficiency

Cost

Size

Flexibility
Challenges

Build a generic vision system that can be used standalone but also embedded in more complex artifacts

BOM < $15
Movidius Myriad2
Optimum performance for CV vs power consumption, size and cost
Awaiba’s NanEye camera
- World’s smallest digital camera
- Power consumption < 10mW
- Disposable
EoT Platform: Hardware

Texas Instruments CC3100
- 20x17 mm
- Low-cost, low-power
- Internet-on-a-chip: integrates all protocols
- Supports Station and Access Point modes
## EoT Platform: Software

<table>
<thead>
<tr>
<th>Library/API</th>
<th>Type</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenCV</td>
<td>Computer vision</td>
<td>Local and Cloud</td>
</tr>
<tr>
<td>MvCv</td>
<td>Computer vision</td>
<td>Local</td>
</tr>
<tr>
<td>Libccv</td>
<td>Computer vision</td>
<td>Local</td>
</tr>
<tr>
<td>Google Cloud Vision API</td>
<td>Computer vision</td>
<td>Local</td>
</tr>
<tr>
<td>Quirc</td>
<td>QR code recognition</td>
<td>Local</td>
</tr>
<tr>
<td>MvCNN</td>
<td>Convolutional Neural Networks</td>
<td>Local</td>
</tr>
<tr>
<td>MvBot</td>
<td>Robotics applications</td>
<td>Local</td>
</tr>
<tr>
<td>Opus</td>
<td>Audio</td>
<td>Local</td>
</tr>
<tr>
<td>RTSP</td>
<td>Video streaming</td>
<td>Local</td>
</tr>
<tr>
<td>MicroPython</td>
<td>Scripting</td>
<td>Local</td>
</tr>
<tr>
<td>MQTT</td>
<td>Messaging</td>
<td>-</td>
</tr>
<tr>
<td>Google Cloud Pub/Sub</td>
<td>Messaging</td>
<td>-</td>
</tr>
</tbody>
</table>
In EoT we selected **MQTT**

- Open lightweight publish/subscribe protocol
- Efficient 1-to-n communication mechanism

---

**TABLE I. MQTT vs HTTPS, Send Performance**

<table>
<thead>
<tr>
<th></th>
<th>3G HTTPS</th>
<th>3G MQTT</th>
<th>WiFi HTTPS</th>
<th>WiFi MQTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages/hour</td>
<td>1926</td>
<td>21685</td>
<td>5229</td>
<td>23184</td>
</tr>
<tr>
<td>% Battery/Message</td>
<td>0.00975</td>
<td>0.00082</td>
<td>0.00104</td>
<td>0.00016</td>
</tr>
</tbody>
</table>
MQTT Broker: Pulga

It is used for communication

(1) Download interface App

(2) Install CV application

(3) Configure CV application

(4) Runtime data communication

User

EoT device(s)
The broker is embedded in the EoT device!
• Computer Vision library **optimized for the EoT device**
  • Application example: Tracking
OpenCV

• Computer Vision library
  • Application example:
    • **Canny** Edge Detection + **RTSP** Streaming
Libccv

- Portable and Embeddable
- Modern Computer Vision Algorithms
- Clean Interface with Cached Image Preprocessing
  - Application Example: **Text Detection**

https://github.com/liuliu/ccv
QR recognition
• EoT devices can control other hardware
Rotation Invariant Face Detector
• EoT device programming language = C/C++
  • We have added a MicroPython console!
4 demonstrators will be developed using the EoT platform

1. Peephole Surveillance

1. Attach

2. Configure

3. Receive alarms
4 demonstrators will be developed using the EoT platform

2. Hands-free automatic museum audio guide
4 demonstrators will be developed using the EoT platform

3. Doll that recognizes child emotions
4 demonstrators will be developed using the EoT platform

4. Wearable lifelogging camera
Conclusions

- ‘Computer vision’ connected platform
  - Foreseen for September 2016

- Demonstrators
  - Foreseen end of 2017

- An important stake with potential numerous applications!
Thank you

http://www.eyesofthings.eu