Qt, multimedia, and computer vision

Laszlo Agocs
Senior Software Engineer
The Qt Company, Oslo, Norway

Qt WORLD SUMMIT 2015

What’s this about?

- Qt is great for UI and multimedia (and lots of other things).
- Let’s combine it with computer vision libraries.
- Integrating into Qt apps is usually straightforward.
    - OpenCV: QImage/QVideoFrame <-> cv::Mat
- But what about video in QML and Qt Quick?

Video filters

- Qt 5.5 introduces video filters in Qt Multimedia.
- Allow inspecting and optionally replacing the video frames.
- Integrated with VideoOutput. Filter implementation in C++.
- Works also for platforms where the video frames reference GPU resources.

The hardware

freescale
i.MX 6 Series SABRE for Smart Devices
i.MX
Your Interface to the World

Freescale Sabre SD

- ARM Cortex-A9 1 GHz quad-core, 1 GB RAM, Vivante GC2000 GPU, Freescale i.MX 6 SoC

The software

- Qt for Device Creation 5.5
- Reference image based on Yocto Dizzy
  - Includes some core OpenCV 2.4 libs

- On i.MX6-based devices Qt Multimedia provides
  - camera support both for MIPI and USB
  - accelerated video decoding

Pulling in OpenCV

- conf/distro/include/imx6qsabresd.conf

```
MACHINE_EXTRA_INSTALL = ”\
 libopencv-core \
 libopencv-imgproc \
 libopencv-objdetect \
 …
MACHINE_EXTRA_INSTALL_SDK = " \
  opencv-dev \
  ...
```

Let’s use the camera

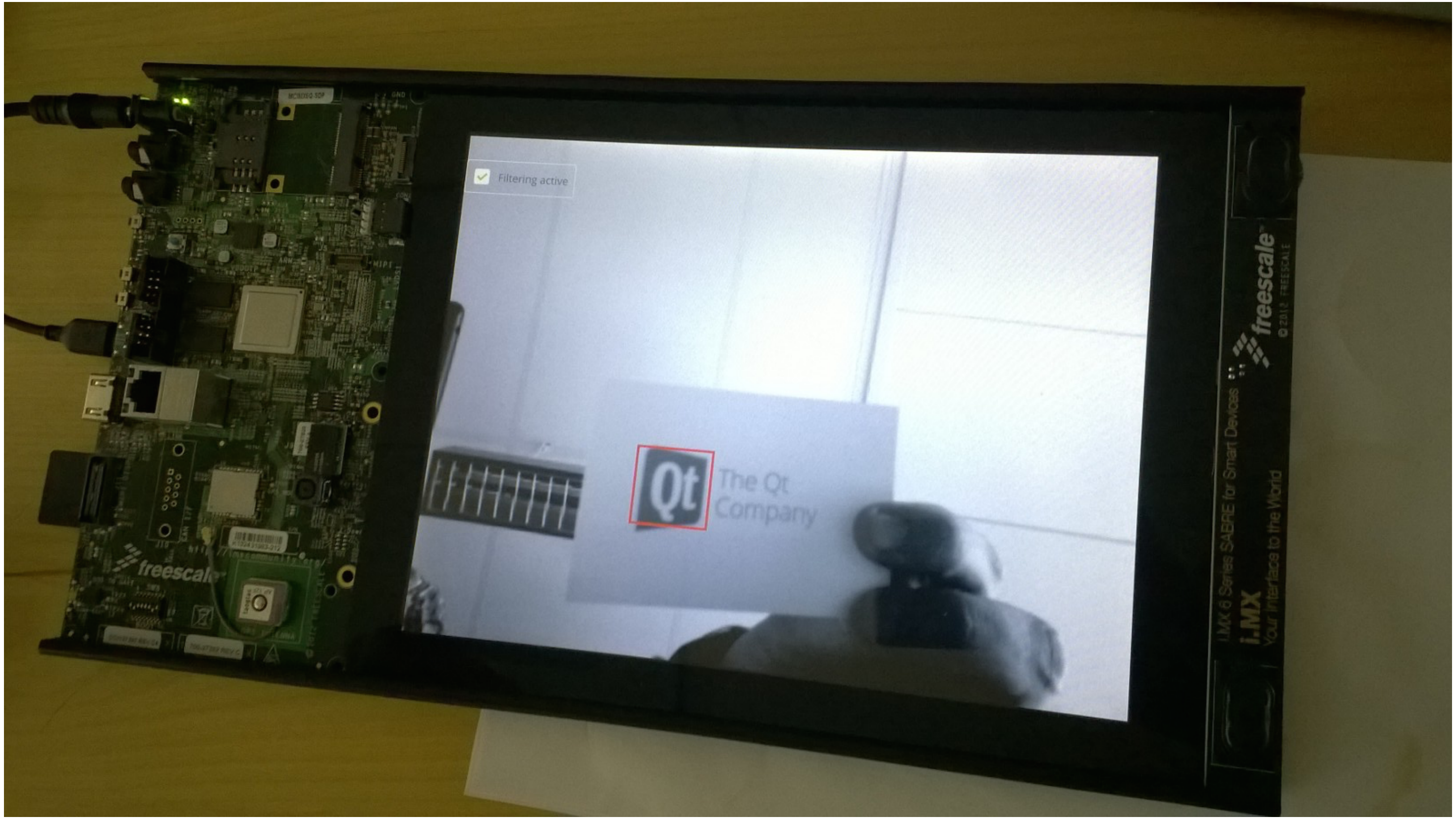
```
import QtQuick 2.4
import QtMultimedia 5.5

Item {
    Camera {
        id: camera
   }
   VideoOutput {
        source: camera
        anchors.fill: parent
    }
}
```

- That was easy.

- Now let’s add some object detection using OpenCV.

The result


☑ Filtering active
The Qt Company
freescale
i.MX 6 Series SABRE for Smart Devices
i.MX Your interface to the World

Thank you!

http://www.qt.io