

Using a Webcam as a Surveillance Camera in a Smart Home



Nathan Ramoly
Amel Bouzeghoub

`nramoly@ilyeum.com`
`amel.bouzeghoub@telecom-sudparis.eu`

Website:
nara.sfunt.fr

Objective

The aim of this lab is to use a webcam, embedded in a laptop or external one, to gather context data about the environment, including motions. The objective is to use these data, alongside a context ontology carried a “Smart Home Server”, to detect intrusions and trigger warnings.

Material

- A computer with a webcam
- OpenCV library

Documentation

- Java specification: <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>
- Jena documentation: <https://jena.apache.org/documentation/ontology/>
- Python reference: <https://docs.python.org/3/reference/>
- OpenCV documentation: <https://docs.opencv.org/4.0.1/>

Outline

I - Setting up

II - Motion Detection

III - Communications with Smart Home Server

IV - Semantic Analysis and Action

V - Further work

I - Setting up

Objective: Prepare the development environment, set up a first video acquisition script

This tutorial overviews all you need to do:

<https://www.learnopencv.com/install-opencv-4-on-windows/>

Install Python

If not installed on your machine, install Python.

You can use Anaconda, a Python distribution: <https://www.anaconda.com/> .

You can use any IDE. Visual Studio Code is a good option: <https://code.visualstudio.com/>

Be aware of the version of Python you are using.

Install OpenCV

Use pip in console to install OpenCV (Anaconda prompt, Powershell, terminal, etc.):

```
pip install opencv-python
```

Test a Program

Try a simple program to see if installation was done properly:

```
import cv2
cv2.__version__
```

Once it works, try to acquire a video stream ! Use the program described in the following link:

https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_video_display/py_video_display.html

If a window appear with a video of the classroom, you managed to acquire a video stream !

II - Motion Detection

Objective: Create a Python program that detect a motion from a video stream

OpenCV provides numerous tools to handle and analyze image. In this section, we aim to detect motions by observing variation in the video stream. There are multiple options to achieve that, you are free to try and use all tools at your disposal.

A basic process could be the following:

1. Save a reference picture (Refresh periodically if necessary)
2. Acquire the current picture
3. Transform both picture from color to grayscale (more accurate comparison)
4. Perform a difference of the reference and current grayscale pictures with a threshold
5. If the difference (number of different pixels for example) is beyond a limit, a motion is detected

Of course, this process should be refined. ;)

You can check examples and tutorials, as well as OpenCV documentation:

<https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>

<https://drive.google.com/file/d/1KMQq8m-2EShRcp8n6BKbpNHV2IU3UVsx/view?usp=sharing>

<https://github.com/RobinDavid/Motion-detection-OpenCV>

III - Wifi Communications

Objective: Signal the motion detected in your motion detector to a smart home server

In this section, we will establish communication between your motion detection script, in Python, and a Java program, that will be the root of a smart home server. To do so, we will design a client/server interaction (TCP or UDP).

If you are not familiar with network programming, you can rely on tutorials and examples, including:

<https://www.baeldung.com/a-guide-to-java-sockets>

<https://wiki.python.org/moin/TcpCommunication>

IV - Semantic analysis and action

Objective: Design a context ontology, integrate sensor data, apply reasoning and act accordingly

Scenario

In this lab, we will consider a simple scenario of surveillance.

A camera is configured to detect motion in a room and provide the information to smart home server. A motion can be a normal or an intrusion according to the context: *if a motion is detected while the user is away or sleeping, the detected motion is likely related to a trespassing, and the smart home should trigger an alarm.*

Ontology Creation

In order to store the context data about the environment, we will rely on a dedicated Ontology. With Protégé¹, design a context ontology that carries the following information:

- Individuals (example: Pierre, Maeva, Nono...)
- Rooms (example: living room, kitchen)
- Sensors
- Context data (example: luminosity, temperature, location, movement, etc...)
 - Relation towards entities
- User habit and/or agenda (example: working from 7am to 5pm)
- Anything you think is important to store

The structure and the general design remain to your concern. Don't forget that you will use it after, thus, think about your needs and use the scenario to warrant your design choices.

Java Integration with Jena

Based on the previous sections, create a program that:

1. Determine if a motion is observed, and insert this information in the ontology
2. Apply rules to reason on the ontology
 - a. Design and insert rules that match the scenario
3. Trigger a notification and/or warning to the user (email, phone, sound, etc.)

And test it out. If the system behave and react as expected through your test, congratulation, you now have a simple smart home application running. :)

¹ <https://protege.stanford.edu/>

V - Further work

Objective: Improve your work on multiple axis and make the application (more) usable.

Improvements

In the case you finished this lab early, don't worry, this was just the beginning. For now, you just did a very simple scenario with a few case. The next step is to create a real and complete application. Based on what you learnt, you have to slightly improve the system following multiple axis:

- Gather other information than motion
 - Direction of movement
 - Identification of user²
 - Gesture recognition
 - Object recognition
- Share and interact with other groups
- Consider and propose a (or more) complete and realistic scenario (not only a single use case !)
- Improve the context ontology and its rules accordingly

In this step, you are (more) autonomous in your directions, choices and implementation. Have fun. ;)

Projects

The work performed in this lab can be used for bigger scale projects. Here are some features that can be addressed as part of project, each of them having its problematics and challenges:

- Freedomotic integration
 - Freedomotic is a framework allowing to create, test and manage smart space
 - Website: <http://freedomotic.com/>
 - Integrate your service in Freedomotic
- Connection to the network (and Internet)
 - Make the sensor data accessible over the network (REST Api)
 - Be aware of privacy and security issues
- Uncertainty management
 - Some sensor are not accurate, impacting the reasoning
 - Modeling and solving uncertain is a strong issue
- Any awesome idea you may have :)

² https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html