FRAMEWORK for ATC/CTA visual analysis on sensor networks

Authors: Antonio Canclini, Alessandro Redondi (Politecnico di Milano)

github address: https://github.com/greeneyesproject/atc_cta_framework

CONTENTS

The repository is composed of three parts, namely:

1. The framework itself, contained in the folder **testbed_framework**, which includes the C++ classes for visual analysis on sensor networks

2. A simple example showing how to use the components of the framework. The example is structured in three folders:

   - **example_sender** → this is the application to be installed and run on the BeagleBone (BB). It tests both the ACT and the CTA paradigms. After taking a picture through a webcam connected to the BB, the application transmits on the sensor network the (encoded) visual features (ATC) or the JPEG bitstream (CTA)
   - **example_receiver** → this application is run on the main controllore (e.g., a PC). It receives the data transmitted from the BB and displays the received image and the keypoints.
   - **example_sender_local** → this is a replica of the *example_sender* that can be run locally on a PC. Since the build process on the BB is quite time-demanding, the local application is useful for a rapid test of the system.

3. A demo for object recognition on a visual sensor networks, operating with both ATC and CTA paradigms. This is the demo presented at MMSP 2013. Again, it is structured in three folders:

   - **DEMO_MMSP_sender** → the application running on the BB (takes the picture and sends visual data on the network)
   - **DEMO_MMSP_receiver** → receives visual data and performs the object recognition task
   - **DEMO_MMSP_sender_local** → this is a replica of the sender, which can be run locally on a PC

INSTALLATION on the BeagleBone

Requirements: BeagleBone with Ubuntu + OpenCV 2.4.5

1. Copy the folders *testbed_framework* and *example_sender* on the BeagleBone

2. Open the makefile contained at example_sender/Release/makefile, and adjust the following paths:

```
FRAMEWORK_FOLDER = ../../testbed_framework
OPENCV_INCLUDE_FOLDER = /usr/opencv/include
OPENCV_LIB_FOLDER = /usr/opencv/lib
USR_LIB_FOLDER = /usr/local/lib
INSTALL_FOLDER = /home/ubuntu/TESTBED_SENDER
```

3. Access to the folder *example_sender/Release* and type:

```
make clean
make all
make install
```

4. Run the executable from the installation folder. You must specify the USB port at which the wireless sensor is connected (e.g. /dev/ttyUSB0)

```
cd /home/ubuntu/TESTBED_SENDER
./example_sender /dev/ttyUSB0
```

The procedure for installing the MMSP_DEMO is similar.

INSTALLATION on the PC

Requirements: openCV 2.4.3 / 2.4.4 / 2.4.5
For the MMSP demo, the following libraries must be installed:

- lib_allegro
- wx_widgets

Installation guide for the *example_receiver*:

1. Copy the folders *testbed_framework* and *example_receiver* on the PC
2. Adjust the paths in the makefile at *example_receiver/Release/makefile*
3. Access to example_receiver/Release and type:

```
make clean
make all
make install
```

4. If needed, assign the full permission to the USB port at which the wireless sensor is connected (e.g. /dev/ttyUSB2):

```
chmod 777 /dev/ttyUSB2
```

5. Run the executable from the installation folder, specifying the USB port at which the wireless sensor is connected (e.g. /dev/ttyUSB2):

```
./example_receiver /dev/ttyUSB2
```

For the installation of the *DEMO_MMSP_receiver*, the procedure is analogous.