Special Projects for ESP1415

Last modified: March 20, 2015

Each project can be selected by a single group of students unless otherwise noted. Collaboration among groups for such projects is not allowed in any case.

Are We There Yet?

Design and implement a software solution for mobile devices that listens to a piece of music and detects the current position inside the piece in real time. The software can rely on a recording of the whole piece available before the live performance. Two approaches are possible.

1. **[ASSIGNED]** Detection is performed by a personal computer, then the result of the detection process is sent to mobile devices. If you pursue this approach, it is required that the signal be distributed without wires; moreover, the solution must scale to hundreds of devices (e.g., spectators at a concert hall).
2. Detection is performed by the mobile device. In this case, there are no signal distribution or scalability issues; however, computing resources are limited and battery consumption may be a concern if the detector consumes significant CPU power for the whole duration of the piece.

The project will be carried out in collaboration with Antonio Rodà and Nicola Orio. An existing detector, developed by Nicola Orio and running on a personal computer, will be made available to the students as a starting point for their work.

Students are required to sign an NDA before accessing the code of the existing detector.

A maximum of two groups of students -- one for each approach -- can select this project.

"Correct Speech" Recognition

Design and implement a software for mobile devices that does more than recognizing speech: it also recognizes proper pronunciation. Students will move in a bottom-up fashion from the recognition of the simplest units of speech, i.e., letters and syllables, to more complex structures such as groups of syllables and words. The precise extent of the work will be determined in accordance with the instructor and taking difficulties encountered during the work into account.

The project will be carried out in collaboration with Antonio Rodà and the Department of Neuroscience, University of Padua (Cristian Leorin, Manuela Susigan).

Students are required to release the source code of their application under an open-source license.

What's New? [ASSIGNED]

Design and implement a portion of an app which analyzes a video taken from a standing (i.e., not moving) camera and detects "new" objects entering the frame. Note that

1. some portions of the image -- such as the background -- will never change;
2. some portions may change slightly (e.g., water in a river), but we are not interested in such minor changes;
3. some other portions can change significantly (e.g., a boat moving in the river), and these are the interesting changes.

Literature on computer vision algorithms for security cameras is definitely useful for this project. An example: http://dx.doi.org/10.1109/TSMCC.2010.2065803.
Literature on background subtraction (see the "Moving Objects in a Photo" project) is also relevant.

The project will be carried out in collaboration with Sergio Canazza.

Students are required to release the source code of their application under an open-source license.

Common Issues in ESP1415

The group of students that tackles this project will

1. interact with all the other groups of fellow students,
2. collect the issues and difficulties faced by their fellow students, trying to cluster them into homogeneous sets,
3. propose possible solutions,
4. write a report on the issues, the solutions, and the outcome of the solutions.

Issues and difficulties can be both organizational and technical, hence at least one (better: more than one) of the members of the group must be familiar with Android (better: familiar with both Android and software engineering).

Motion Detection

Design an implement an app that detects small (20-30 cm max) movements of the tablet with the maximum possible speed and precision. To improve precision, the estimate must be performed by integrating data from different sensors (front camera, accelerometer, gyroscope). As a demonstration of the precision of the system, the app must include an activity that displays an object on a monochrome background: the object must maintain its position when the tablet is moved (motion compensation).

Students are required to release the source code of their application under an open-source license.

Moving Objects in a Photo [ASSIGNED]

Design and implement a photo-taking app where moving, temporary objects (cars, passersby, …) disappear from the final shot. An existing app that offers such a function is Nokia Smart Camera (http://disq.us/8mctj2).

Alternatively, design and implement an app that removes the background from a photo and keeps only the moving subject (e.g., a person).

Useful papers:

- http://dx.doi.org/10.1016/j.cosrev.2014.04.001,
- http://dx.doi.org/10.1109/TCSVT.2002.800516.

See also:

- http://docs.opencv.org/trunk/doc/tutorials/video/background_subtraction/background_subtraction.html,
- http://mateuszstankiewicz.eu/?p=189.

However, mind that in your app the video acquired from the camera is usually shaky.

Camera Shake [ASSIGNED]

Analyze and implement the algorithm of Fergus et al. to remove camera shake from a single photograph. Reference paper: http://dx.doi.org/10.1145/1141911.1141956.

The algorithm is to be inserted into an app that provides two modes of operation.

- Offline mode: a photograph is selected from the device library, then the photo is de-shaken, shown, and the user can save it.

- Real-time mode: the first half of the screen shows the preview of the picture acquired in real time from the device camera, the second half shows the same picture after de-shaking.

Fall Detection

Perform a survey on fall detection algorithms, report on them, then implement some of them and compare performance.

Some recent papers on the topic:

- http://dx.doi.org/10.1109/PERCOMW.2010.5470652,
- http://dx.doi.org/10.1109/ComComAp.2012.6154019,
- http://dx.doi.org/10.1109/FSKD.2012.6234271.

More than one group of students can select this project.

Vision Problems with OpenCV

Design and implement a solution for one of the following computer vision problems discussed in Chapter 10 of the book "A Practical Introduction to Computer Vision with OpenCV":

- *Reading Notices* (Section 10.5),
- *Determining the Time from Analogue Clocks* (Section 10.14),
- *License Plates* (Section 10.18),
- **[ASSIGNED]** *Recognize Paintings* (Section 10.20).

These are toy problems; little assistance will be provided by the instructor, and the implementation must be flawless to get top marks.

More than one group of students can select this project.

De-noising

Apply various kinds of de-noising algorithms (book "A Practical Introduction to Computer Vision with OpenCV", Chapter 2) to an image; implement each filter yourself and compare with OpenCV implementation where possible. This project is simple but pretty boring. The implementation must be flawless to get top marks.