

# MID-TERM REPORT GOOGLE SUMMER OF CODE 2012

## OPENCV MOBILE VISION APP DEVELOPMENT

AUTOR: EDUARD FEICHO <EDUARD.FEICHO@RWTH-AACHEN.DE>

### WEEK 1 (14.05 - 20.06)

I was contacted by my mentor Vadim Pisarevsky if I still would be available to GSoC, because I hadn't been accepted officially and due to certain circumstances I could still be accepted to GSoC after all.

### WEEK 2 (21.05 - 27.05)

Planning the project and talking to my mentor while waiting for being officially accepted as GSoC student. I was officially accepted on 25.05 as GSoC student.

### WEEK 3 (28.05 - 03.06)

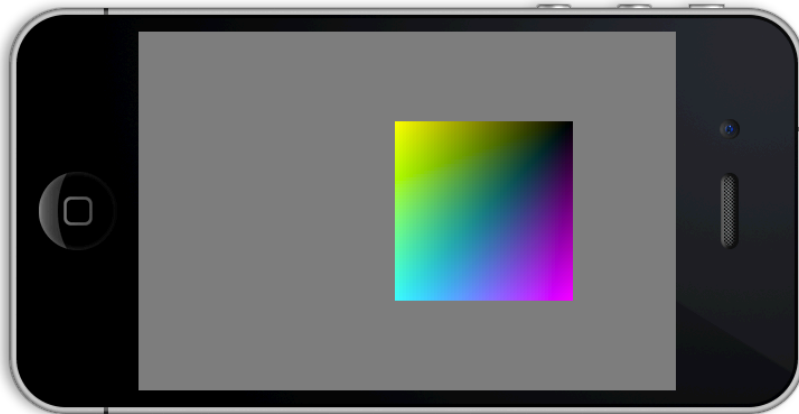
#### TASKS ACHIEVED

- Put up a blog (<http://duffycola.wordpress.com/>) and git repository (<https://github.com/Duffycola/gsoc2012>).
- Compiling OpenCV according to following tutorial: [http://docs.opencv.org/doc/tutorials/introduction/ios\\_install/ios\\_install.html](http://docs.opencv.org/doc/tutorials/introduction/ios_install/ios_install.html)
- Thinking about an image stitching App.
- **Added project IntroCamera:**
  - Example use of internal photo library, photo camera and video camera. Screenshots:



- **Added project IntroOpenGL:**

- It's an example project with a basic ViewController that implements all the relevant functions to have a simple OpenGL main loop up and running. The project files were from an online tutorial which I renamed. However, after the rename the project didn't work anymore. Have to look at it again, once we actually will need OpenGL.



- **Added project IntroOpenCV:**

- Just an example project with no function at all, except setup of the compiler and linker flags to include OpenCV. A number of problems were encountered when following the basic compile instructions at [http://docs.opencv.org/doc/tutorials/introduction/ios\\_install/ios\\_install.html](http://docs.opencv.org/doc/tutorials/introduction/ios_install/ios_install.html):
- OpenCV/iPhoneOS/lib/pkgconfig/opencv.pc lists folders with missing libraries.
- Had to manually copy /trunk/release/lib/RELEASE/\*.a and /trunk/release/3rdparty/lib/RELEASE/libzlib.a to the output library folder.
- The simulator build was compiled for arm architecture, which means it's not the simulator build, but the device build (incorrect).

## WEEK 4 (04.06 - 10.06)

### TASKS ACHIEVED

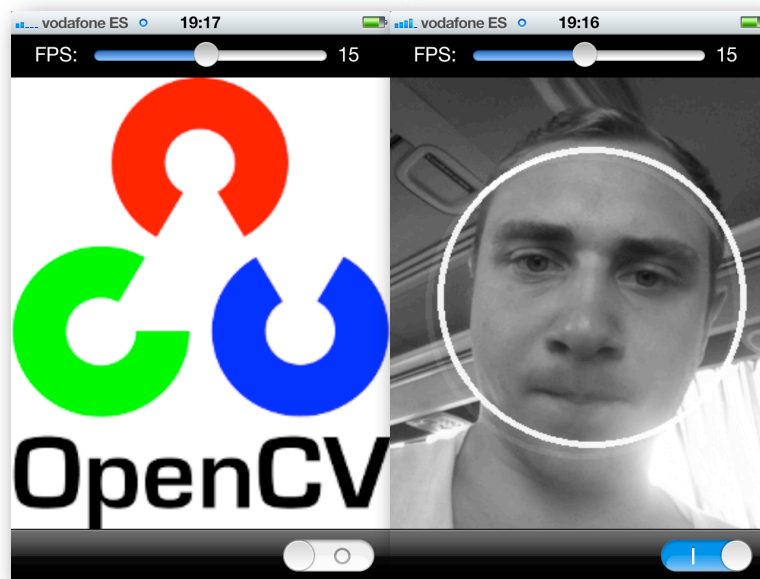
- **Compile project IntroOpenCV**

- Thanks to Charu Hans, the GSoC student that is also working for the iPhone port, I noticed that I placed wrong flags as compiler/linker flags. Eventually, I was able to compile a project with OpenCV support.

- **OpenCV framework**

- My mentor Vadim suggested to create an OpenCV private framework to avoid users having to set up compiler and linker flags manually.
- I remembered an article at <http://aptogo.co.uk/2011/09/opencv-framework-for-ios/#using> and finally found that users have already created build scripts:
- <https://github.com/aptogo/OpenCVForiPhone/blob/master/opencvbuild.sh>
- <https://github.com/BloodAxe/OpenCV-iOS-build-script/blob/master/BuildOpenCV.sh>
- Finally, Vadim put up a build script that fixed some of the aforementioned bugs and created an OpenCV framework that can be included in projects using drag-and-drop.

- **Added project FaceDetectSimple:** Detecting faces (and eyes) of VideoCamera using Haar cascades.
  - Code from: [http://docs.opencv.org/doc/tutorials/objdetect/cascade\\_classifier/cascade\\_classifier.html#cascade-classifier](http://docs.opencv.org/doc/tutorials/objdetect/cascade_classifier/cascade_classifier.html#cascade-classifier))
  - Basic setup:
    - Input images of size 640x480
    - OpenCV gray scale conversion
    - Down-Scaling the image with a factor of 2
    - Face Detection
    - Creating a CGContext and drawing circles (face, eyes)
    - Runs at 2 FPS :-/
  - Looked at SquareCam face detector from Apple, but some of the code was too specific to use it as a general opencv example.
  - Screenshots:



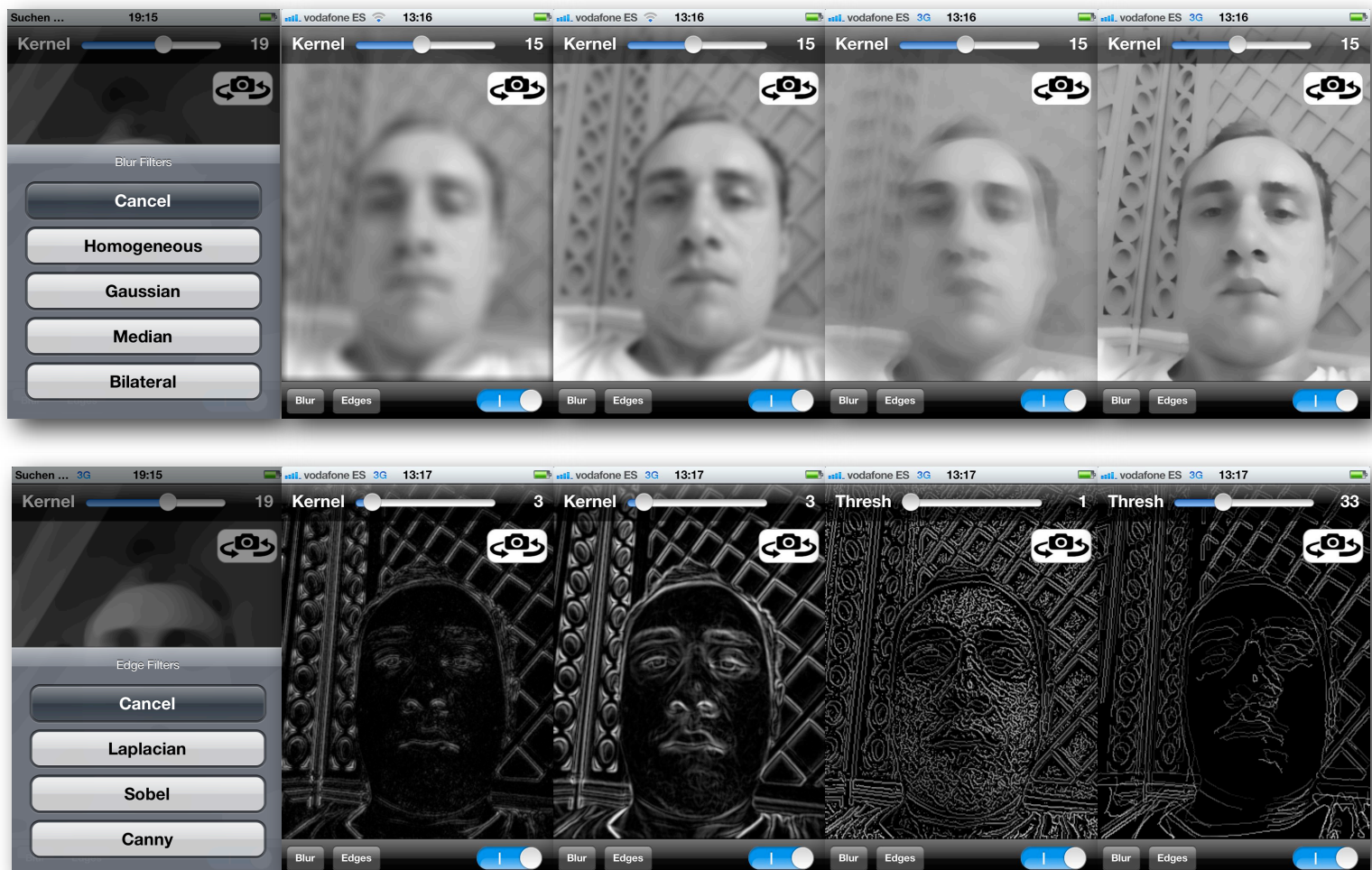
#### TASKS PLANNED

- Improve performance of face detector
  - Don't detect eyes.
  - Use YUV colorspace to avoid RGB to grayscale conversion.
  - Avoid scaling operations by setting up proper video output size.
  - Reduce unnecessary image copy operations according to suggestions from Vadim.
  - Use LBP cascades.

## WEEK 5 (11.06 - 17.06)

### TASKS ACHIEVED

- Migrated code to OpenCV repository: <http://code.opencv.org/svn/gsoc2012/ios/trunk/>
- A common folder structure (~/.work/ocv, ~/.work/ocv.build, ~/.work/ocv.ios\_samples)
- YUV video camera output, create Mat directly from pixel pointer without copying data
- Enhance camera controller code, provide properties for some settings (capturing video/single still image, front/back camera, video quality, default orientation of video output)
- Project FaceDetectSimple runs at 4-5 FPS (LBP cascade and video size of 352x288)
- **Added project VideoFilters:**
  - Code from [http://docs.opencv.org/doc/tutorials/imgproc/table\\_of\\_content\\_imgproc/table\\_of\\_content\\_imgproc.html#table-of-content-imgproc](http://docs.opencv.org/doc/tutorials/imgproc/table_of_content_imgproc/table_of_content_imgproc.html#table-of-content-imgproc)
  - Blur + Edge filters on video camera frames
  - Blur (Homogeneous, Gaussian, Median, Bilateral)
  - Edge Filters (Laplacian, Sobel, Canny)



### TASKS PLANNED

- Either continue to create samples copied from the tutorials section or create a single very good and elaborate sample, like an Image Stitching App.

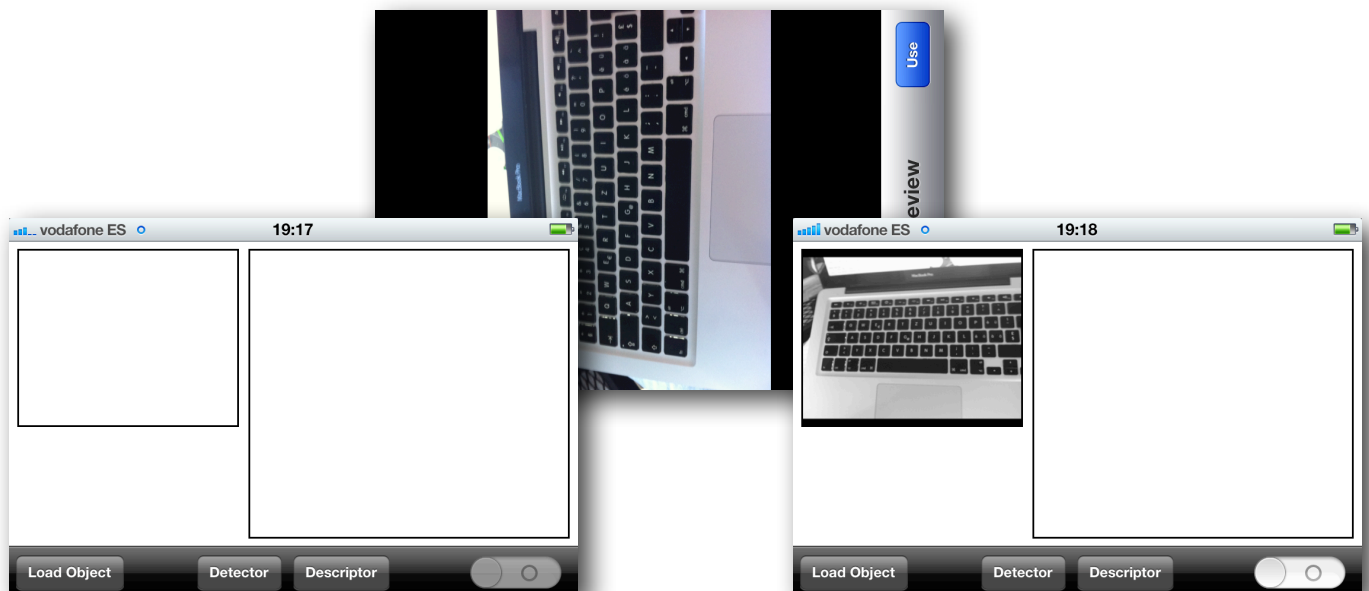
## WEEK 6 (18.06 - 24.06)

I didn't proceed much because of exams, just cleaned up the code a little.

## WEEK 7 (25.06 - 01.07)

### TASKS ACHIEVED

- Filed a number of bug reports regarding VideoCamera orientation
- VideoFilters: Homogeneous Blur with Acceleration framework (ARM-optimized DSP)
- **Added project FindHomography**
  - Code from [http://docs.opencv.org/doc/tutorials/features2d/feature\\_homography/feature\\_homography.html#feature-homography](http://docs.opencv.org/doc/tutorials/features2d/feature_homography/feature_homography.html#feature-homography)
  - Step 1, "Load object": Make a photo of an object that you want to detect in video.

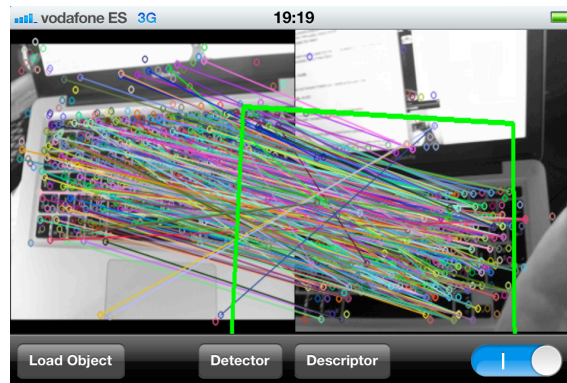


- Step 2: Optionally change the used feature detector or descriptor:





- Step 3: Enable processing using the UISwitch. If a homography can be computed, the result is drawn into the camera frame. (image based on drawMatches and manually drawing homography).



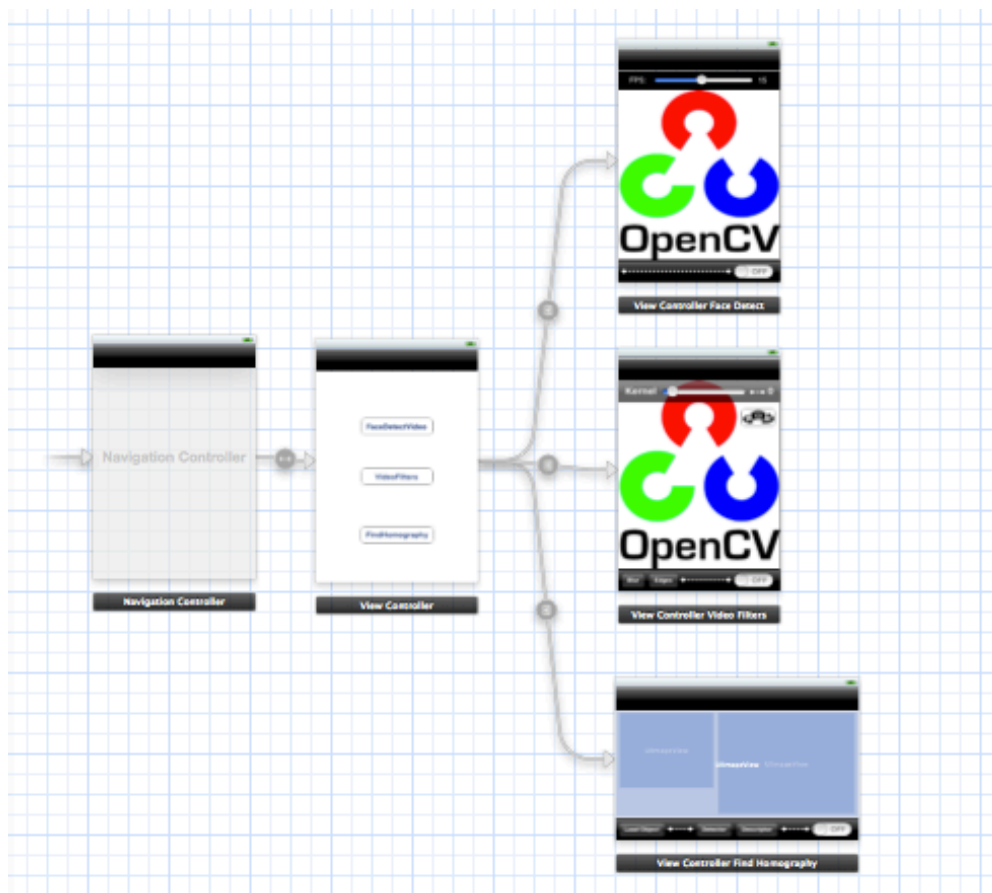
### TASKS PLANNED

- Since VideoFilters project is in landscape mode, issues with the video camera orientation were encountered. However, the source of the problem remained unclear. This is important and needs to be fixed.
- FindHomography project
  - Images rendered are transposed and mirrored (supposedly due to the orientation issues).
  - Check which combinations of descriptor+detector are allowed (undocumented), because some combinations crash. Can also be a bug of the application, though.
- VideoFilters project:
  - Try to replace more of the filters with a variant that uses Acceleration framework.
- Cleanup VideoCamera Controller:
  - Make an minimal and easy understandable interface.
  - Try to make the interface homogeneous (sometimes we work with raw pixel buffer, cv::Mat, CGContextRef, CALayer and UIImage...)
  - Maybe make two flavors of the camera, separating single static image camera and video camera into different classes.
- Start ImageStitching project, because we want to have an example that looks more like a complete App, which can be submitted to the App Store.

## WEEK 8 (02.07 - 08.07)

### TASKS ACHIEVED

- Cleanup of the code, especially VideoCamera
  - ViewControllers were renamed and moved to /shared/ViewControllers/ so that they can be reused in other projects, namely the full-scale demo app.
  - Rename project FaceDetectSimple to FaceDetectVideo
  - Rename files in /shared/ to have a fixed OpenCV prefix: "Cv", because they are likely to be used in other projects
- Coping with VideoOrientation issues
  - The problem not directly relates to the setup of the video orientation within the camera itself, but the parent view is incorrectly rotated and thus the video camera's preview layer (CALayer\*) relies on that parent view.
  - See also new project VideoOrientation, to test the orientation behaviour.
- Added project FullscaleDemoApp:
  - Project that demonstrates our demos in one complete demo app.
  - Simply three buttons for demos FaceDetectVideo, VideoFilters, FindHomography
  - Can be made more pretty later.
  - I used the storyboards feature for the first time to layout the navigation logic:



### PROBLEMS ENCOUNTERED

- I switched to XCode 4.5 beta 2 for iOS 6 beta 2, which resulted in incompatible project files (especially with the InterfaceBuilder and storyboards feature). I switched back to XCode 4.3.3 and will wait until we officially will support iOS 6.

#### TASKS PLANNED

- Fix the video orientation bugs, in order to properly support portrait and landscape mode.
- Include CvVideoCamera in opencv framework, to make it easily available to users (for example as port of highgui module).
- Find out if it is possible to include automatically with our framework all the Apple frameworks that we depend on (Acceleration, AVFoundation, CoreGraphics, CoreImage, CoreMedia, CoreVideo, QuartzCore).

#### WEEK 9 (09.07 - 15.07)

... Preparing mid-term evaluation ...