**Homework #5: Advanced Image Capture and Processing**

Design a Windows Phone to fulfill the following specifications. There are two separate applications requested, please submit both applications as part of the same solution.

**Application 1**: Develop a Windows Phone application that reads in an image stream from the camera, and outputs it to the user in real time. Analyze the image stream to find the location of a simple marker (in this case, a red square). Superimpose an image bundled with your application over this marker, effectively creating a simple augmented reality application. The choice of tracking algorithm is yours; this is meant to be a creative endeavor to obtain some familiarity with the challenges involved in tracking a complex object such as a face. Feel free to bounce ideas off of other students, and don’t be afraid to just try things to see if they will work. There are many attributes a tracking algorithm may strive for, some ones to keep in mind are rotation invariance, scaling invariance, efficiency and false positive rate.

**Application 2**: Develop a Windows Phone application that builds upon the preview application and tracks faces in the current window. A face detection library will be provided shortly. Perform the same action as before, superimposing an image on top of the faces in the current application.

The emulator has a synthetic camera input it provides to the application, so you can test **Application 1** using the emulator only. Note that as **Application 2** tracks faces, it will be much more difficult to do this using the emulator, although you may be able to read in images with faces on them and feed these images to the face recognition code instead of using the camera. This should only be for testing, and your final code should work by using the camera.

The libvideo camera wrapper library can be downloaded [here](http://sdrv.ms/XrGkAq).

The TextureGraph component can be downloaded [here](http://sdrv.ms/12xvSif).

The helper classes shown in class (Frame and Pixel) can be viewed [here](https://gist.github.com/staticfloat/4735957/raw/5f968db08d9de14d81bb6b3e76fb423bffefede2/FframeHelper.hpp).

The face detection library can be downloaded [here](https://catalyst.uw.edu/workspace/file/download/c44d49742a2cee44d70a960f050b26f3d848de6e31b04d44f7f34036ff6ef890).