LIFFT - Stage 3 Documentation

The objective for this stage was to deliver 30% of our user stories. After deliberating and re-evaluating our JIRA stories, our group achieved this goal with the following breakdown:

Graphical User Interface (New user story) - 3 pts

Object tracking in Video (Required) - 5 pts

Track Objects with Digitally Placed Anchors (Wishlist) - 13 pts

This documentation will address the following items:

1. Running the program

2. Areas for improvement

3. Sources

**(1) Running the program:**

Our project uses open source libraries provided by OpenCV, a library of programming functions aimed at real-time computer vision. Download the latest stable version of OpenCV from here: https://www.opencv.org/releases.html

Once you have downloaded the latest stable version you should be able to compile our cpp source code files. Before moving on, please download the mp4 file to your local machine. Remember the location of this file as you will need to edit a variable in our program to include the abolute path of the location of this file. In order to run our application you must compile all our source code files, namely: mainwindow.cpp, mainwindow.h, videotracker.cpp and videotracker.h. Compilation instructions for the respective files are below.

Note: mainWindow.cpp takes approximately 20 seconds to compile, please be patient.

Compiling and running mainwindow.cpp:

Compile with the following instruction: g++ main.cpp mainwindow.cpp mainwindow.h -o program `pkg-config gtkmm-3.0 --cflags --libs`

Once compilation is complete there will be an executable file named 'program'. Run 'program' from the present working directory. Running program will present you with a simple menu UI. Availble options will be to load a file or to quit. Click 'Load file', then proceed to select the mp4 file you downloaded to your local machine. In the next iteration of our project, clicking 'Ok' will kickstart our videotracker program with the video as an input parameter. For now these items are left separate, and thus they are functionally isolated. Click 'Quit' to quit anytime.

Compiling and running videotracker.cpp:

Before you do anything, you MUST edit the stringVideoFile variable so that it holds the absolute path of the mp4 file downloaded. Compile with the following instruction: g++ videotracker.cpp -o app `pkg-config --cflags --libs opencv

Once compilation is complete there will be an executable file named 'app'. Run 'app' from the present working directory.

**(2) Areas for Improvement:**

Our initial scoring scheme for the user stories proved to be inaccurate of the workload of the respective tasks. For example our required 'Object Tracking in Video' user story score dropped by 4 points because of its seamless integration with the functionality of the wishlist feature 'Track Objects with Digitally Placed Anchors'. However we realize that this is a part of the agile process and will make adjustments accordingly.

**(3) Sources:**

In order to implement our tracking software, we used the following OpenCV modules and functions:

OpenCV Modules - <https://docs.opencv.org/3.4.3/>

To track objects frame to frame - calcOpticalFlowPyrLK() from <https://docs.opencv.org/3.4.1/dc/d6b/group__video__track.html#ga473e4b886d0bcc6b65831eb88ed93323>

For video captures in OpenCV - <https://docs.opencv.org/3.1.0/d8/dfe/classcv_1_1VideoCapture.html#a949d90b766ba42a6a93fe23a67785951>

We also used available functions provided to us through the OpenCV library. For guidance in code base and logic, we received help from tutorials from the following sources:

<https://docs.opencv.org/3.3.0/d2/d0a/lkdemo_8cpp-example.html?fbclid=IwAR12ZmCLZyWklIoQadHgMv9HynU_F36StRev91rLZw_KRNriWd7el-9X_Q8>

<https://www.docs.opencv.org/3.1.0/d2/d0a/tutorial_introduction_to_tracker.html>

<https://docs.opencv.org/2.4/doc/tutorials/introduction/display_image/display_image.html>