

SECTION 1: Implemented feature detectors/descriptors/methods

In code1: OptFlowCommentedCode.cpp (For Part B under Motion Estimation)

BRISK and ORB: Used for detecting Keypoints in the frame.

Optical Flow LK Method: Used to find updated Keypoints from the previous Keypoints using LK method.

In code2: OptFlowDescriptorMatching.cpp (For Part C under Motion Estimation)

BRISK and ORB: Used for detecting Keypoints and Descriptors in the frame.

Flann Matching: To match the descriptors in previous frame with the current frame's descriptors.







Note: Detailed explanation of applications and theory of each method has been explained in report. Here, we're just mentioning the methods used and their application in our Project.

SECTION 2: Code Citations

- Optical Flow using LK method: <https://github.com/opencv/opencv/blob/master/samples/cpp/lkdemo.cpp>
- For Flann Matching:
http://docs.opencv.org/2.4/doc/tutorials/features2d/feature_flann_matcher/feature_flann_matcher.html

Note: Rest of the code has been programmed by us and to the best of our knowledge, we haven't utilized any other code templates except for the above two.

SECTION 3: Code folder organization

 Code	05-04-2017 00:26	File folder	
 Inputs	05-04-2017 08:13	File folder	
 Outputs	05-04-2017 08:13	File folder	
 eee508s17_Project1_GradingSheet.pdf	05-04-2017 08:16	Adobe Acrobat Docu...	12 KB
 ReadMe.pdf	05-04-2017 00:21	Adobe Acrobat Docu...	910 KB
 Report.pdf	05-04-2017 08:09	Adobe Acrobat Docu...	1,628 KB

Code folder has two cpp files.

1. OptFlowCommentedCode.cpp
2. OptFlowDescriptorMatching.cpp

The **first code** corresponds to **part b** under the motion estimation while the **second code** corresponds to **part c** under the motion estimation.

Input folder has a video **spl_cut.mp4** which can be used as input.

Output folder has saved outputs for part b as well as part c.

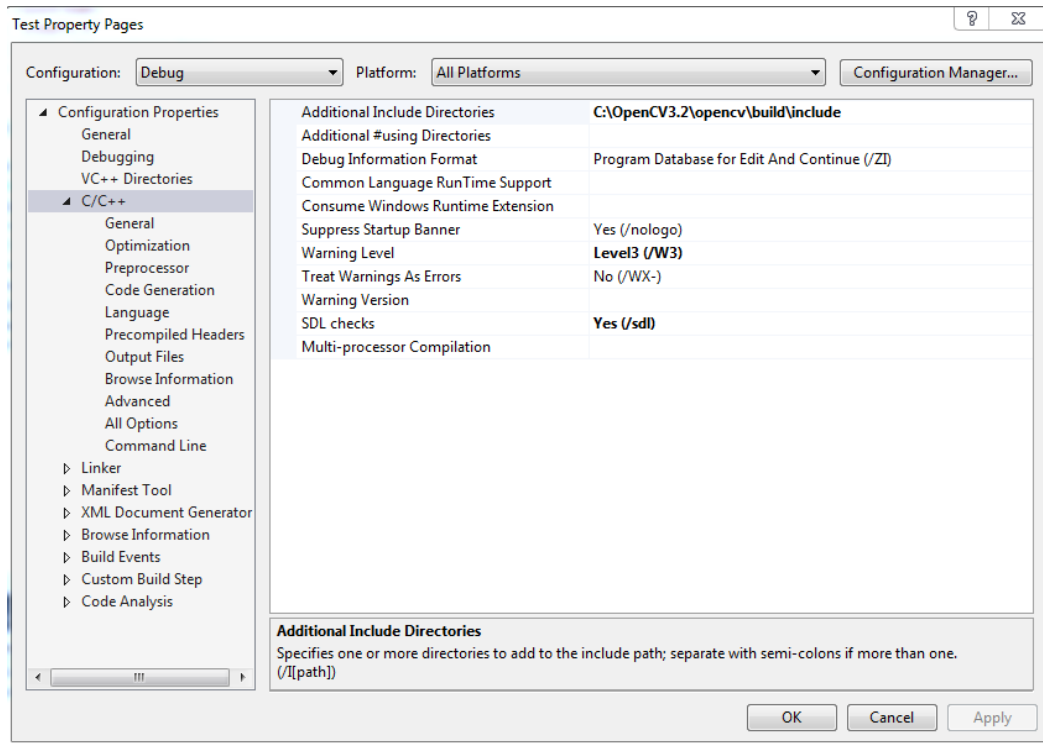
SECTION 4: Steps for execution of the codes

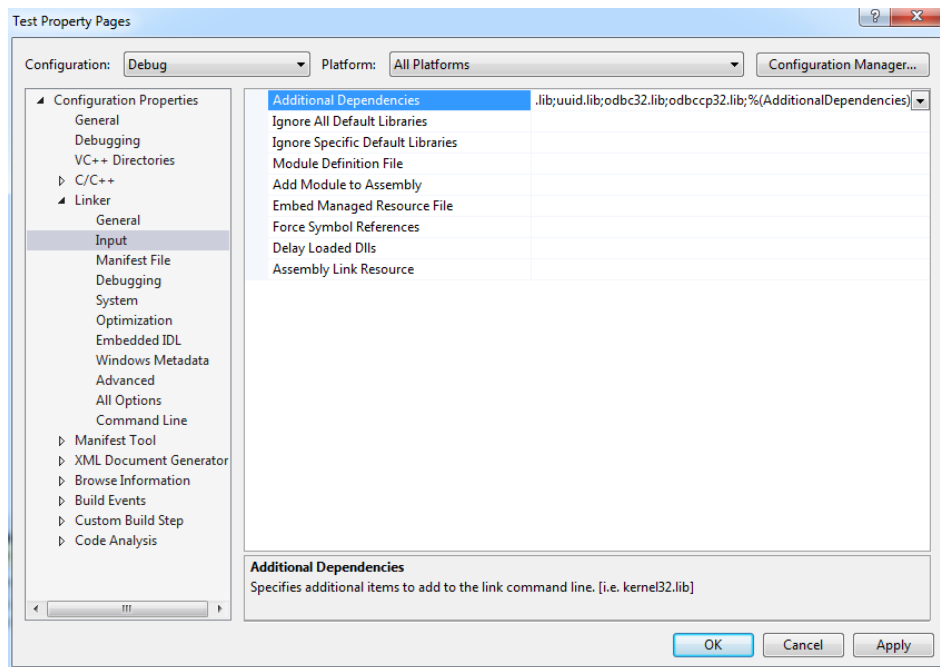
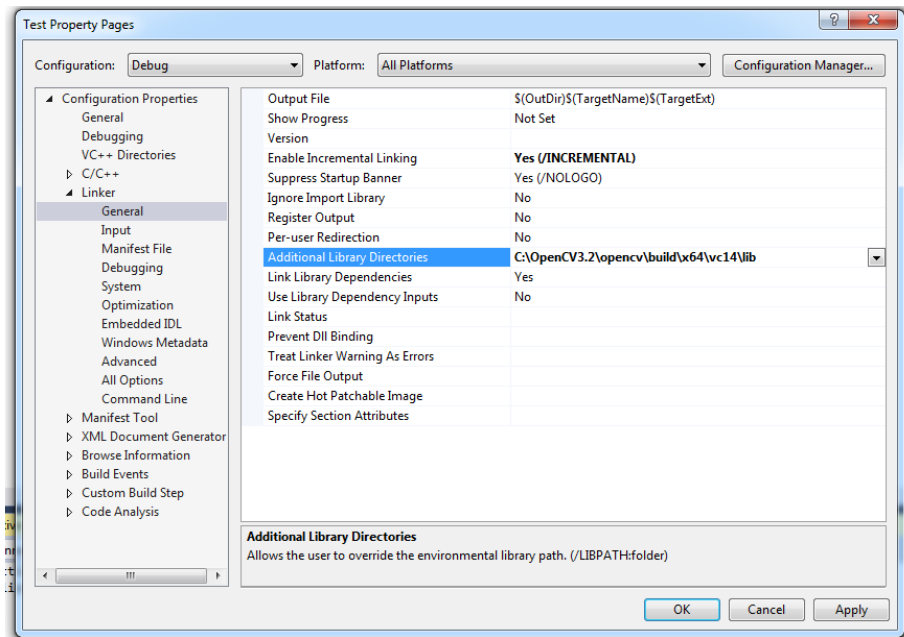
STEP 1: Create a new VS C++ project in Visual C++

- Go to start and Open VS 2015.
- On VS window create a new VC++ project by selecting File → New Project and choosing "Win32 Console application"
- Enter the name of the project and solution files and the location in which the project folder will be created.
- The wizard will create a project folder with a project file (.vcproj), a solution file (.sln), and three source files .cpp.
- If the project name is Test, then you can open the **Test.cpp** file and then replace the code there with the code in **OptFlowCommentedCode.cpp** (For part b under motion estimation) or **OptFlowDescriptorMatching.cpp** (For part c under motion estimation).
- Click on Next and in the Application Settings Select Empty project, and then click on Finish.

STEP 2: Including libraries

- Right click on the solution file and click on **Properties**.
- Check if everything is set as per the **Install OpenCV tutorial** on blackboard.
- After the libraries are included properly, you wouldn't see any red colour underlines in your **cpp** file.





Under additional dependencies, paste the following:

kernel32.lib;user32.lib;gdi32.lib;winspool.lib;comdlg32.lib;advapi32.lib;shell32.lib;ole32.lib;oleaut32.lib;uuid.lib;odbc32.lib;odbc32.lib;odbc32.lib;opencv_world320.lib;opencv_ts320.lib;%(AdditionalDependencies)

Note: It is assumed that you've new build files corresponding to the xfeatures2d library. If not, follow the instructions mentioned in the appendix of the project description file.

http://lina.faculty.asu.edu/eee508/eee508s17_Project1_FeatureDetectorDescriptor.pdf

STEP 3: Making necessary modifications in the code (Optional)

In OptFlowCommentedCode.cpp

- If you want to test the optical flow on a video file, but not using your web camera, change the line 60 in code from **cap.open(0)** to **cap.open("FilePath/file.mp4")**.

```
59     if (input.empty())
60         cap.open(0); //Opens up the default camera
61     // In the above line, replace argument with the file path/file name if you have a video
62     // For example, cap.open("C:/Users/Prajwal/Desktop/SampleVideo.mp4");
```

- Make sure you change the File path of the destination where you want your output video to get saved.

```
79     // The below line saves the video on to my Desktop. Make sure you change the path before execution
80     VideoWriter oVideoWriter("C:/Users/Prajwal/Desktop/MotionEstimation_partB.avi", CV_FOURCC('P', 'I',
```

- **Uncomment line 106** and **comment line 109** for using **BRISK** as the feature detector. If you want **ORB** as the feature detector, **uncomment line 109** and **comment line 106**.

```
103     // automatic initialization
104
105     // For BRISK to detect keypoints, keep the following line.
106     //Ptr<Feature2D> f2d = BRISK::create(50, 4, 1.0f);
107
108     // For ORB to detect Keypoints, keep the following line.
109     Ptr<Feature2D> f2d = ORB::create(100, 1.2f, 8, 31, 0, 2, ORB::HARRIS_SCORE, 31, 80);
110
```

In OptFlowDescriptorMatching.cpp

- If you want to test the optical flow on a video file, but not using your web camera, change the line 23 in code from **cap(0)** to **cap("FilePath/filename.mp4")**.

```
21     string filename;
22
23     VideoCapture cap(0); //In place of cap(0), replace with cap("C:/Users/./FileName.mp4") for video file as input
24
25     if (!cap.isOpened())
26     {
27         cout << "ERROR: Cannot open the video file" << endl;
28         return;
29     }
```

- Make sure you change the File path of the destination where you want your output video to get saved.

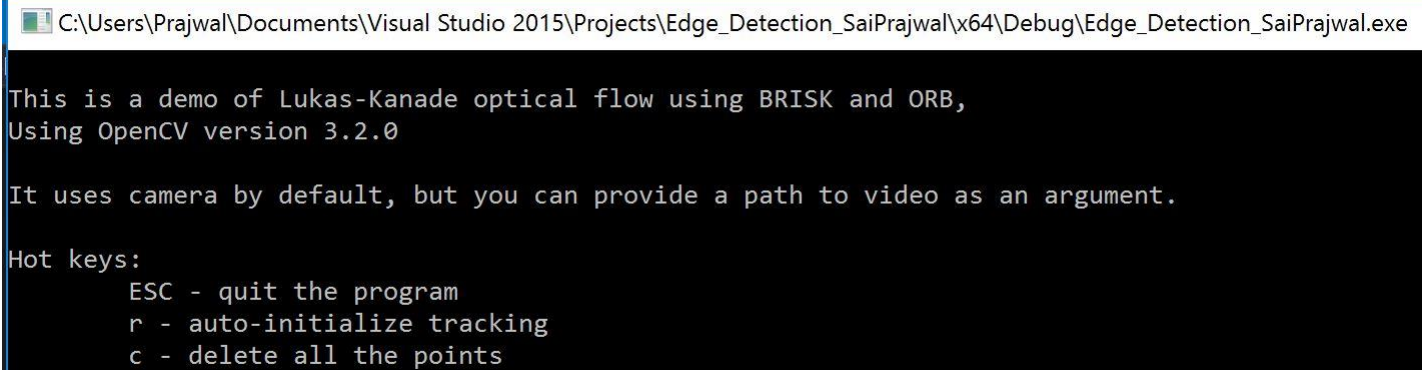
```
35     // Don't forget to change the file path in the line below.
36     VideoWriter oVideoWriter("C:/Users/Prajwal/Desktop/MotionEstimation_partC.avi", CV_FOURCC('P', 'I',
37     if (!oVideoWriter.isOpened()) //if not initialize the VideoWriter successfully, exit the program
```

- **Uncomment line 46** and **comment line 49** for using **BRISK** as the feature detector. If you want **ORB** as the feature detector, **uncomment line 49** and **comment line 46**.

```
45     // For BRISK to detect keypoints, keep the following line.
46     Ptr<Feature2D> f2d = BRISK::create(50, 4, 1.0f);
47
48     // For ORB to detect Keypoints, keep the following line.
49     //Ptr<Feature2D> f2d = ORB::create(100, 1.2f, 8, 31, 0, 2, ORB::HARRIS_SCORE, 31, 50);
50
```

STEP 4: Steps to run the code

For OptFlowCommentedCode.cpp



```
C:\Users\Prajwal\Documents\Visual Studio 2015\Projects\Edge_Detection_SaiPrajwal\x64\Debug\Edge_Detection_SaiPrajwal.exe

This is a demo of Lukas-Kanade optical flow using BRISK and ORB,
Using OpenCV version 3.2.0

It uses camera by default, but you can provide a path to video as an argument.

Hot keys:
    ESC - quit the program
    r - auto-initialize tracking
    c - delete all the points
```

- Once the necessary changes are made in the code as per Step 3, **build the solution** from the build drop down menu and then drop down the debug menu and click on **Start debugging**.
- Time taken for every frame can be seen on the command prompt.
- **To start the optical flow**, click on the Opt Flow window and then **press r** on your keyboard. You'll be able to see the tracking once you press r.
- **To clear the points, you can press c** on your keyboard. **To restart the tracking again**, you can **press r**.
- **To quit debugging, you can press Esc key**.
- Once the execution is finished, you can see the output video file in the file location which you set up in step 3.

For OptFlowDescriptorMatching.cpp

- Once the necessary changes are made in the code as per Step 3, **build the solution** from the build drop down menu and then drop down the debug menu and click on **Start debugging**.
- Time taken for every frame can be seen on the command prompt.
- **The optical flow** starts automatically from the first frame itself unlike the previous code.
- **To clear the points and restart tracking, you can press c** on your keyboard.
- **To quit debugging, you can press Esc key**.
- Once the execution is finished, you can see the output video file in the file location which you set up in step 3.

SAI PRAJWAL KOTAMRAJU
SAI PRATYUSHA GUTTI