DISCUSSION AND ANALYSIS OF THE OBTAINED RESULTS

CASE 1: BRISK/ORB USING LK OPTICAL FLOW

Time analysis

We compared the time taken by BRISK detector and the ORB detector to detect the keypoints in a frame using **clock()** command in C++. We've divided the program execution into three parts-

a. The time taken for each iteration/frame when detector functions (BRISK/ORB) aren't called.
b. The time taken to detect Keypoints initially (by BRISK/ORB) for the first frame.
c. The time taken for each frame to calculate the new Keypoints from the previous Keypoints using LK Optical Flow method.

For the **'a' part**, the time taken is independent upon the Keypoint detector used. The average time per frame in this case is about **30 msec**.

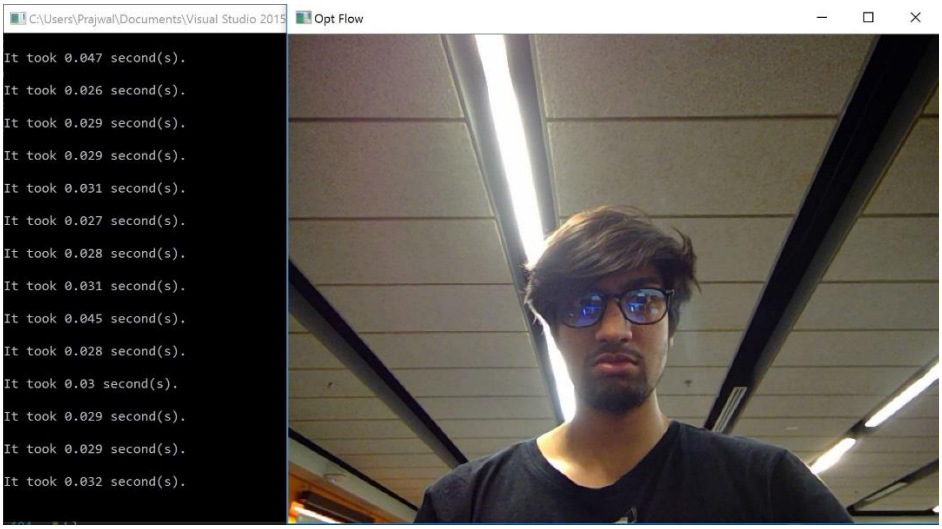

Figure 1.1. The time taken per frame without calling Keypoint detector (ORB).

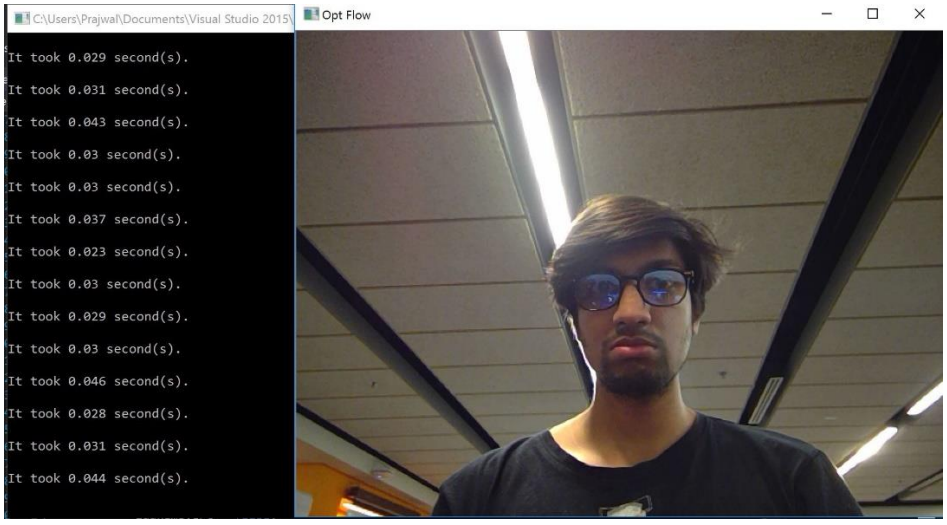

Figure 1.1. The time taken per frame without calling Keypoint detector (BRISK).

For the **'b' part**, the time taken solely depends upon the Keypoint detector used. In this case, when BRISK is used to calculate the Keypoints in the start frame, it takes about **2.7 seconds** while ORB takes about **80 msec**.

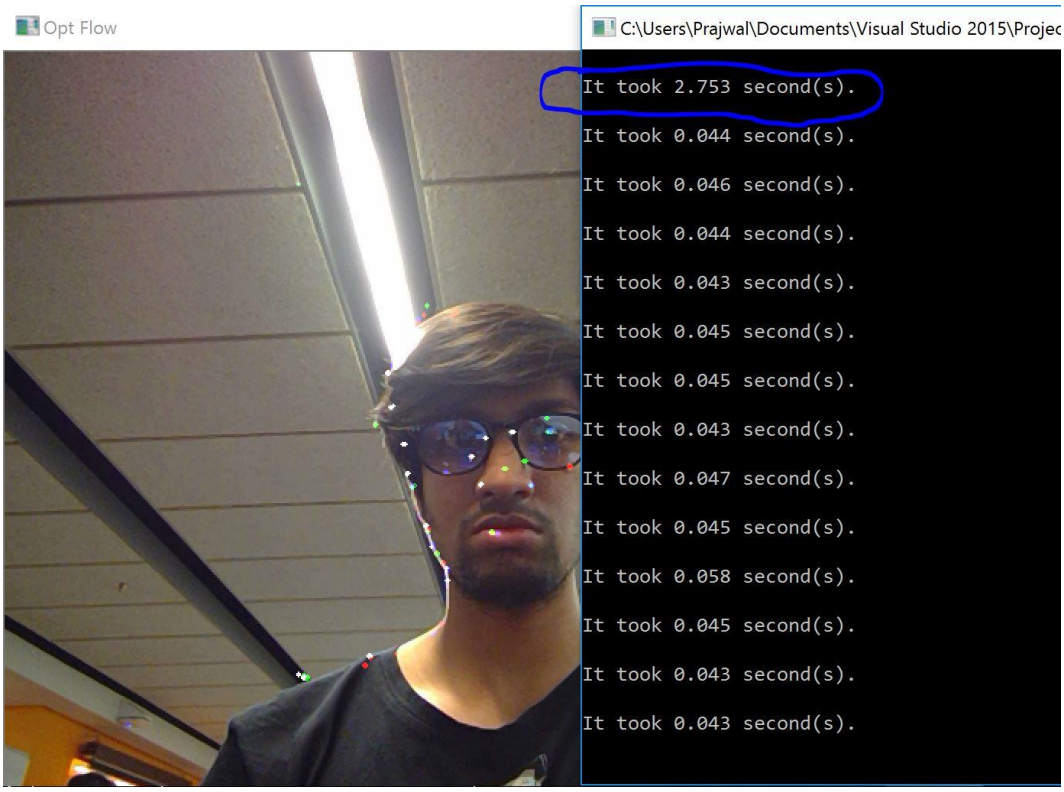

Figure 2.1. The time taken when BRISK is used to detect Keypoints in the start frame.

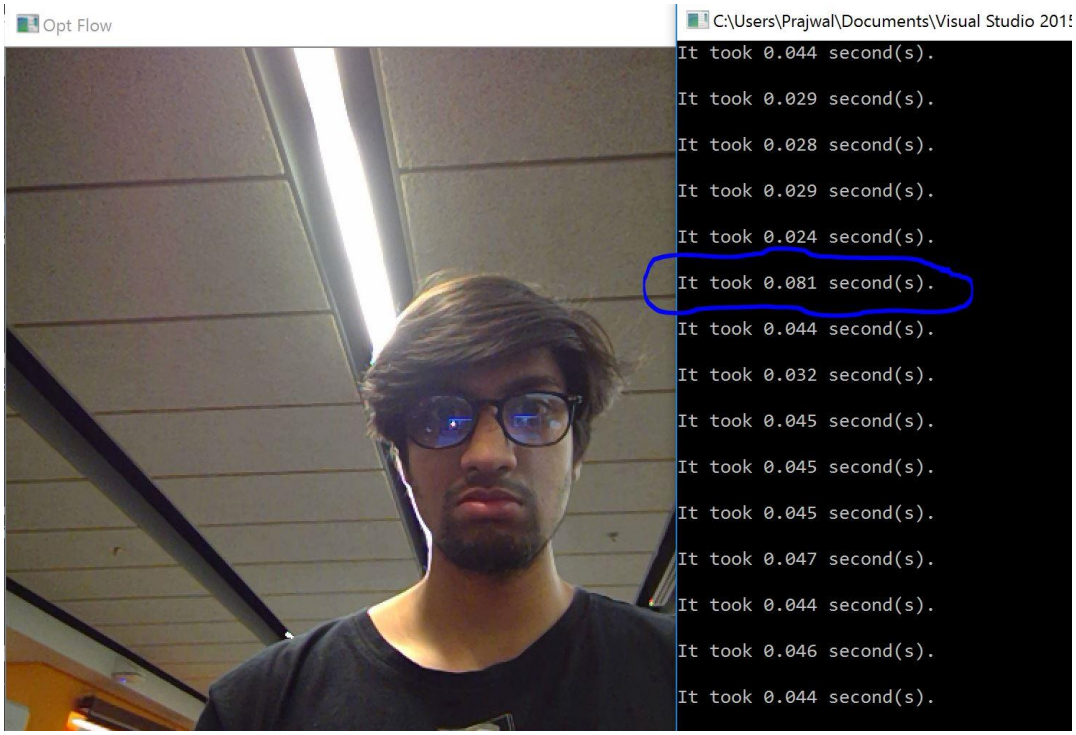

Figure 2.2 The time taken when ORB is used to detect Keypoints in the start frame.

When we see the imges in **Figure 2.1** and **Figure 2.2**, we see that BRISK takes about **35 times** more time to detect the Keypoints in the image when compared to ORB. But, as we see the images, BRISK detects more Keypoints in the image for the same threshold value when comapred to ORB. Because of this, the accuracy of motion estimation is high using BRISK but at the cost of time of execution. Hence, while using any detector, one must understand that there's a trade off between the accuracy of estimation and the time taken for Keypoint detection.

Tabulating the results discussed above, the following are the conclusions.

Method used	Time taken to detect Keypoints	Number of Keypoints detected
BRISK	2.7 Seconds	27
ORB	0.081 Seconds	6

For the **'c' part**, the time taken per frame depends solely upon the LK Optical flow function's response time. As it doesn't depend upon the method of Keypoint detection used, it is almost same in both the cases below.

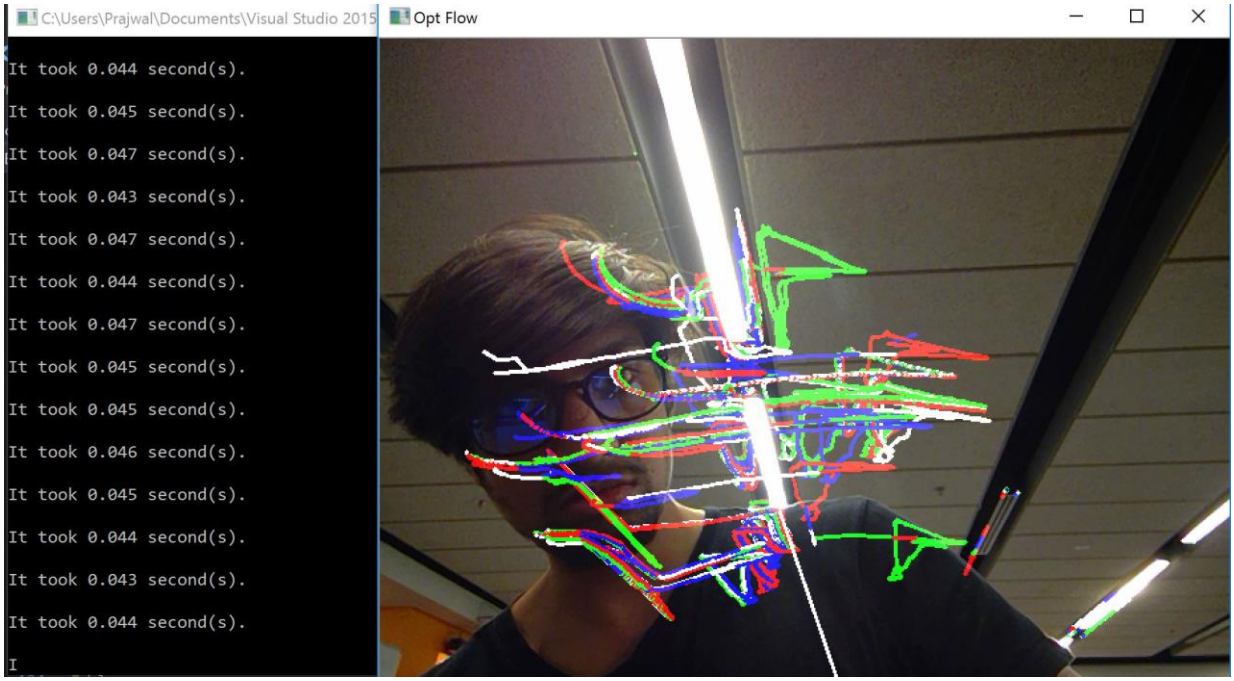

Figure 3.1. Time taken per frame to iteratively calculate the Keypoints using LK method after detecting Keypoints using BRISK.

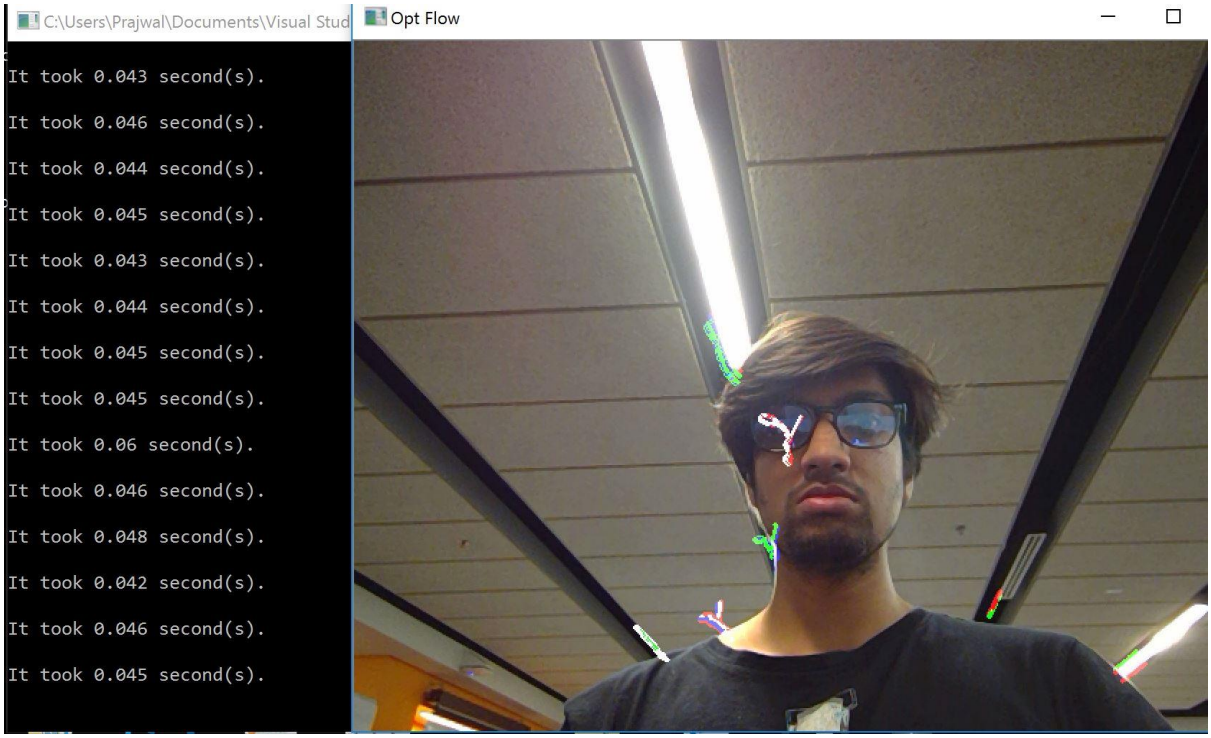

Figure 3.2. Time taken per frame to iteratively calculate the Keypoints using LK method after detecting Keypoints using ORB.

Outputs for SPL.mp4 file using BRISK and ORB + LK Optical Flow

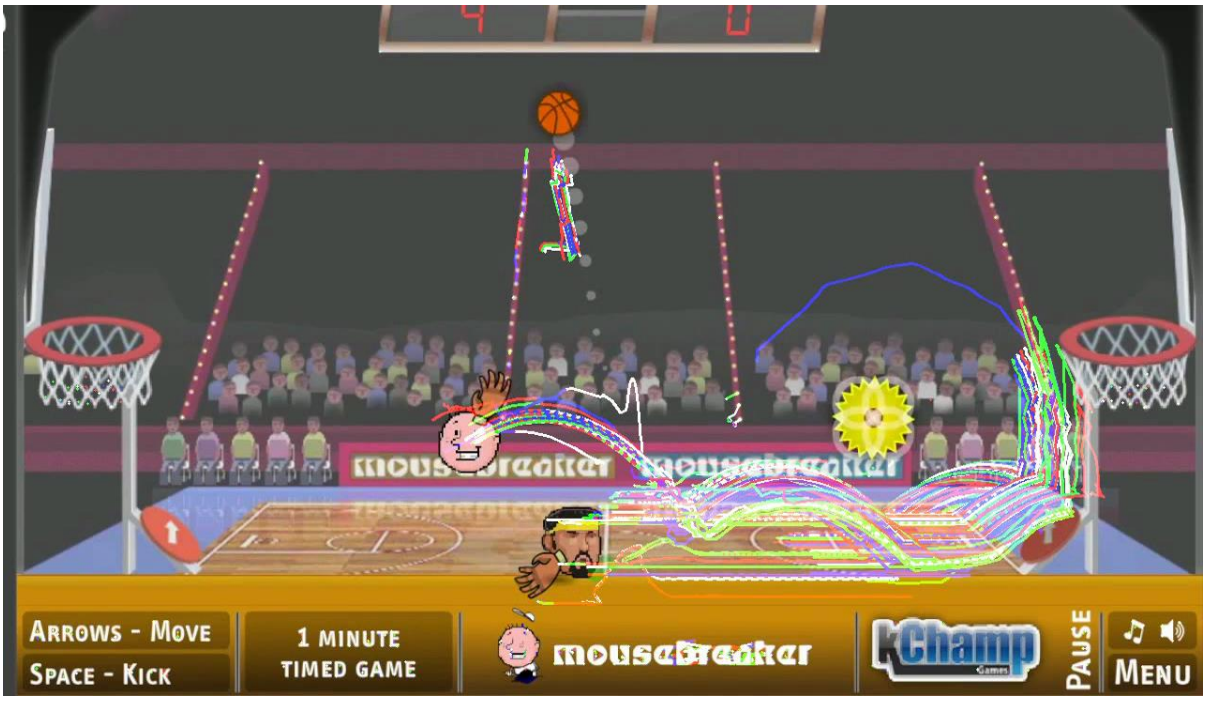

Figure 4.1. Optical Flow using LK method by iteratively tracking the keypoints obtained from BRISK

Figure 4.2. Optical Flow using LK method by iteratively tracking the keypoints obtained from ORB

CASE 2: BRISK/ORB WITH DESCRIPTOR MATCHING

Time analysis

We compared the time taken by BRISK detector and the ORB detector to detect the keypoints in a frame using **clock()** command in C++. Here, the total time taken is the sum of time taken for detectors to compute descriptors and the time taken by Flann matching to match the descriptors of previous frame to current frame. i.e.,

$T = t_d + t_m$

T → Total time taken per each frame

t_d → Time taken by detectors to compute descriptors

t_m → Time taken by Flann matcher to match the descriptors from previous frame to current frame.

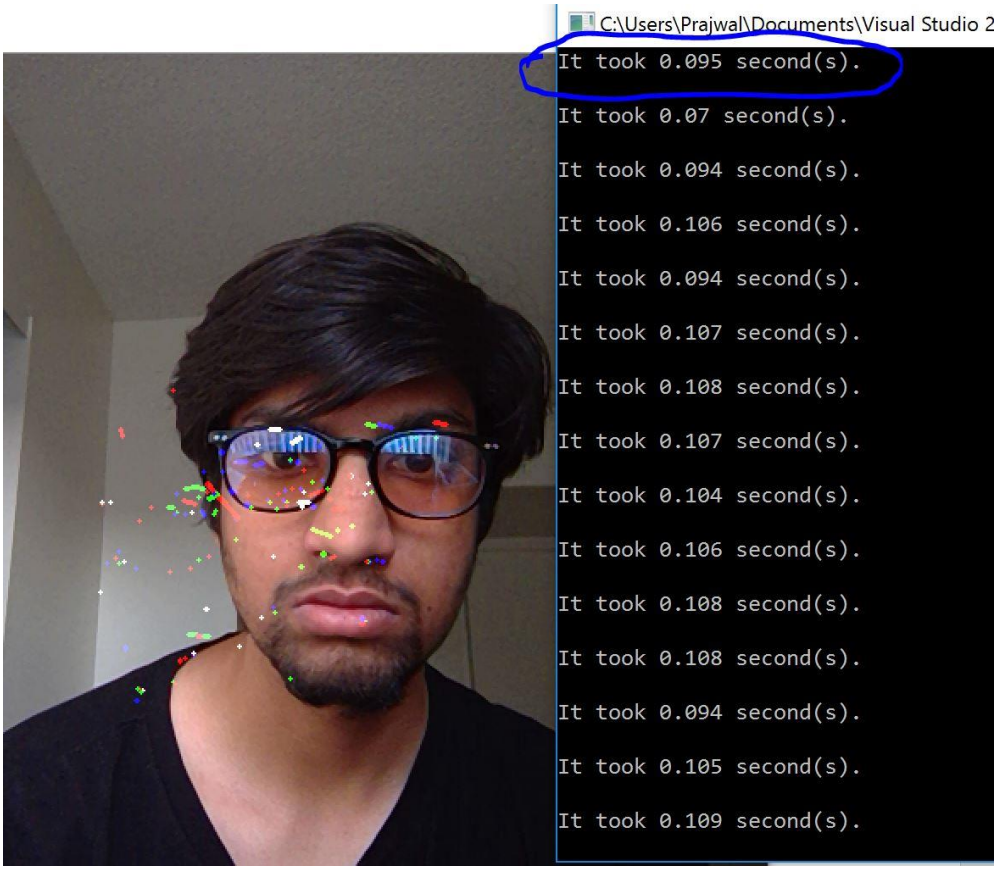

Figure 5.1. Total time taken (T) per frame when BRISK is used to detect Keypoints

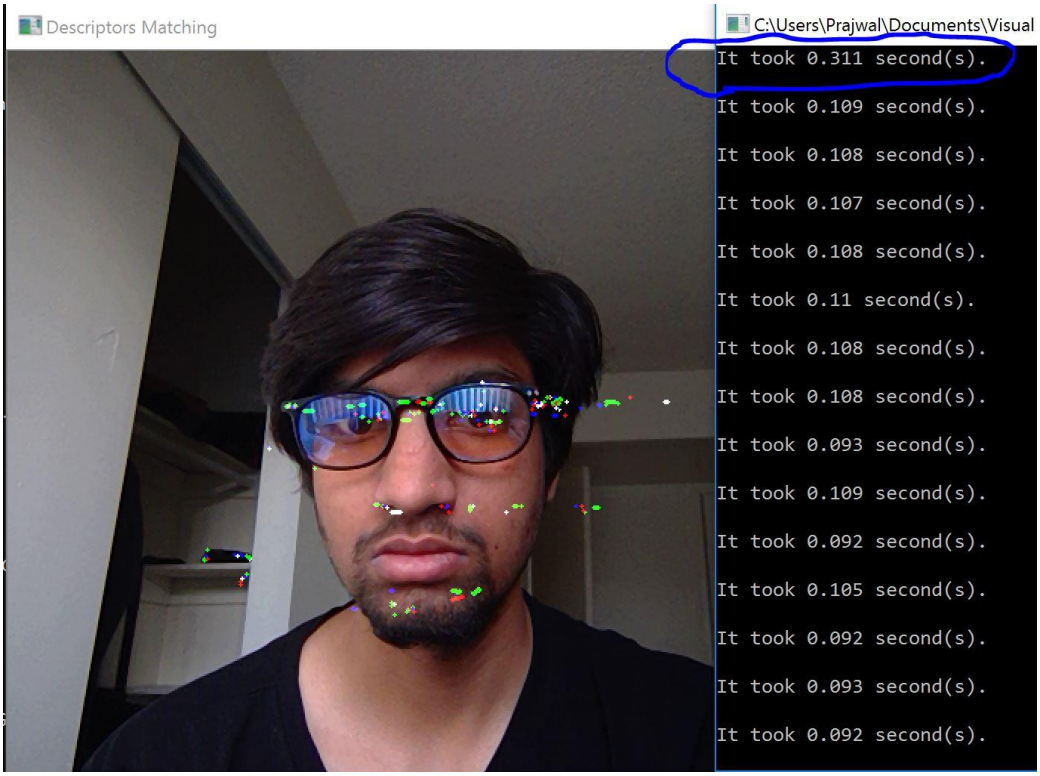

Figure 5.1. Total time taken (T) per frame when ORB is used to detect Keypoints

From Figure 5.1 and Figure 5.2, we see that only for the first frame, there's a huge difference in the total time taken. We see that BRISK is faster when compared to ORB given that we have same number of Keypoints detected by both the detectors. The time taken by BRISK to process 23 matchings in first frame is approximately 95 msec while the time taken by ORB to process 19 matchings in first frame is approximately 311 msec. Tabulating the above results, we have the following.

Method used	Time taken to detect Keypoints	Number of Keypoints detected
BRISK	95 Milli-Seconds	23
ORB	311 Milli-Seconds	19

From this, the inferences are:

- BRISK is faster when compared to ORB when the detected Keypoints are almost same in number.
- ORB gives less number of Keypoints for the same threshold value when compared to BRISK.

Outputs for SPL.mp4 file using BRISK and ORB + Descriptor matching using Flann matcher

Figure 6.1. Optical Flow using Flann matching of descriptors computed for BRISK Keypoints

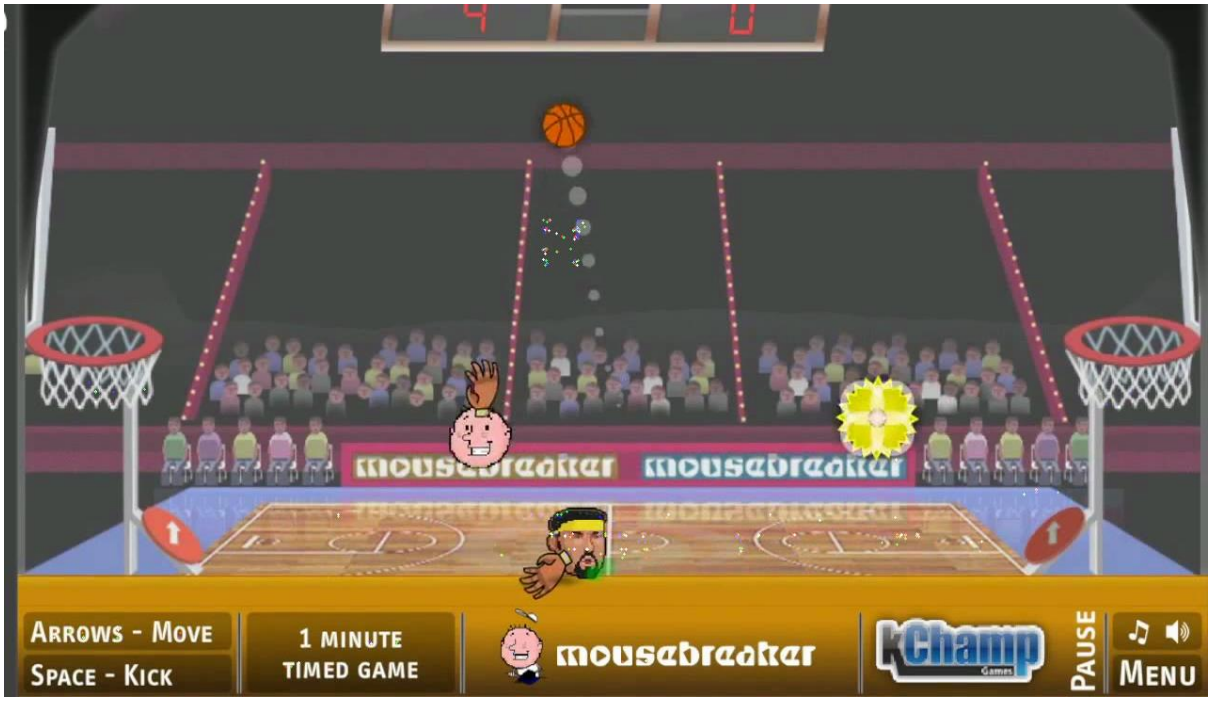

Figure 6.2. Optical Flow using Flann matching of descriptors computed for ORB Keypoints

COMPARISION OF CASE 1 & CASE 2

- From the Figures 4.1, 4.2, 6.1 and 6.2, we see that Optical Flow using LK method results in smooth optical flow lines while the descriptor matching resulted in discrete flow.
- Also, the time taken for each frame to compute Optical Flow using LK method is less when compared to the time taken for each frame to compute Optical Flow using descriptor matching.

CONCLUSION AND DISCUSSION OF UNSOLVED PROBLEMS

A few conclusions which we derived from this project are:

- BRISK is faster (Approximately 3 times) when compared to ORB when same number of Keypoints are detected by both the detectors.
- ORB computes less number of Keypoints in a frame when compared to BRISK for the same threshold value.
- Optical Flow using LK method results in a continuous flow while the descriptor matching gives us a discrete flow. This happens because of the varying number of matches between two consecutives frames in case of descriptor matching.
- LK method is twice as fast as descriptor matching. So, it is preferred to use LK method for optical flow instead of descriptor matching.

A few extensions that can be made to this project are-

- After calculating the lines of Optical Flow, we can calculate the speed with which object is moving based upon the time taken per frame, displacement of the object and camera's position.
- Results can be compared by using different descriptor matching techniques other than Flann matching to see the time taken and the continuity of the Optical Flow.
- This method along with predictive models can be used to predict the object's location in near future based on its previous trajectory.