# Report on Image-Stitching Challenge

1. The Approach

I adopted a feature based approach for image stitching. I implemented it using Python and OpenCV.

1.1 Steps involved in image stitching

- Extracting key frames from the video.
- Stitching Successive key frames to form the panaroma.

1.1.1 Extracting key frames form video

- Video frames are divided into batch_size of 10.
- A mean frame is calculated using these 10 frames.
- Then variance of each frame in the batch is calculated with respect to the mean frame.
- A frame with the minimum variance is selected as the key frame from the respective batch.
- I am considering only one key frame per batch.
- These key frames for every batch are appended in a list.

1.1.2 Stitching Successive key frames to form the panaroma

- Two succesive images from this list of key frames are used for processing.
- Features and descriptors for each image are calculated bsased on user's choice input.
- Features are matched in both the images using a "**Brute Force Matcher".**
- The matched key points are used to compute the **homography matrix**.
- Perspective correction is performed on one of the two images and the corrected image is appended to another image.
- This operation is carried out for consecutive frames in the list of key frames to obtain the resulting image.
- The result image is stored as a jpg file.

2. Algorithms Used

- OpenCV has 5 different implementations of detectors and descriptors. They are as follows:
  - SIFT (Scale Invariant Feature Transform)
  - SURF (Speedup Robust Feature) without Upright
  - SURF (Speedup Robust Feature) with Upright
  - STAR Detector with BRIEF descriptor
  - ORB
- I used Brute Force matcher to determine keypoint matches between two successive frames. Lowe's test is performed to find good keypoint matches. "ratio" parameter for the test is set to 0.75
- To compute homography between two successive frames RANSAC is used with reprojThresh= 4.0

3. Conclusion

- Different algorithms have different performance with respect to time complexity and quality of result image.
- SIFT gives good quality result image at the expense of computational time.
- While ORB is fastest amongst above mentioned algorithms it does not produce a good quality result image.
- A good trade-off can be obtained using SURF with Upright flag.
- The metric for key frame extraction would change according to the application.

4. References

- OpenCV3 documentation
- Image-stitching blog

Note:

Refer Readme.txt to execute the demo.