Video Stabilization due to shake, jitter (TPA 11)

Sai Surendra Reddy.B *CS11B008* and Nikhil.R *CS11B023*

12th May 2014

Abstract

Video stabilization is an important video enhancement technology which aims at removing annoying shaky motion from videos. The key to this is to find out the motion of the camera and filling out the portions of the frames where data is lost due to the irregular motion of the camera. In this report we will describe a video stabilization algorithm based on Point Feature Matching [1]. Missing pixels in the video are reduced by trimming the stabilized video and then filling the remaining missing part with the neighbouring frame pixels.

I. Introduction

Video enhancement has been steadily gaining in importance with the increasing prevalence of digital visual media. One of the most important enhancements is video stabilization, which is the process for generating a new compensated video sequence where undesirable image motion caused by camera shaking is removed. A major problem of current software video stabilizers is that missing image areas appear in the stabilized video due to the compensation of the motion path. This problem has been handled by either trimming the video to obtain the portion that appears in all frames or constructing image mosaics by accumulating neighboring frames to fill up the missing image areas. In our method we use the former method for filling in missing areas.

II. ALGORITHMIC DESCRIPTION

The approach followed by us to achieve the stabilization involves the following steps:

1. Global motion estimation to compensate camera shake

2. Trimming the video to focus on known pixels

Sai Surendra Reddy - IIT Madras, 3rd year CS&E, *saisurendra157@gmail.com*

Nikhil Rongala -IIT Madras,3rd year CS&E,*nikhil11rongala@gmail.com*

A. Global Motion Estimation

1. Read Frames from a Video File
2. Collect Salient Points from Each Frame
3. Select Correspondences Between Points
4. Estimating Transform from matching Correspondences

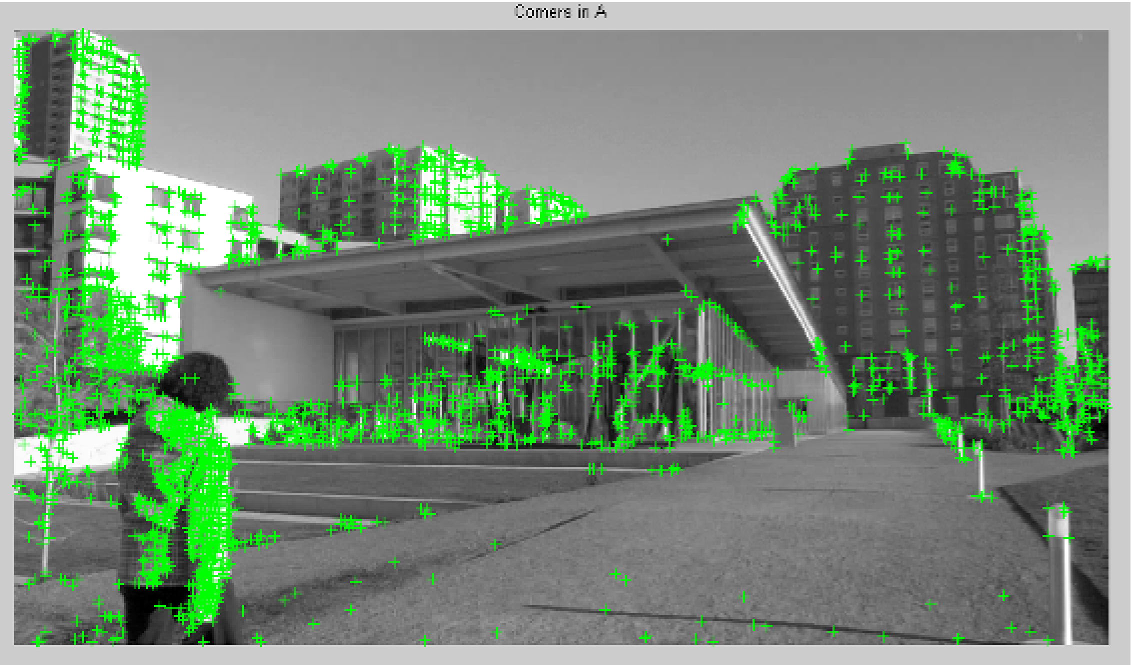
- **Reading frames**:

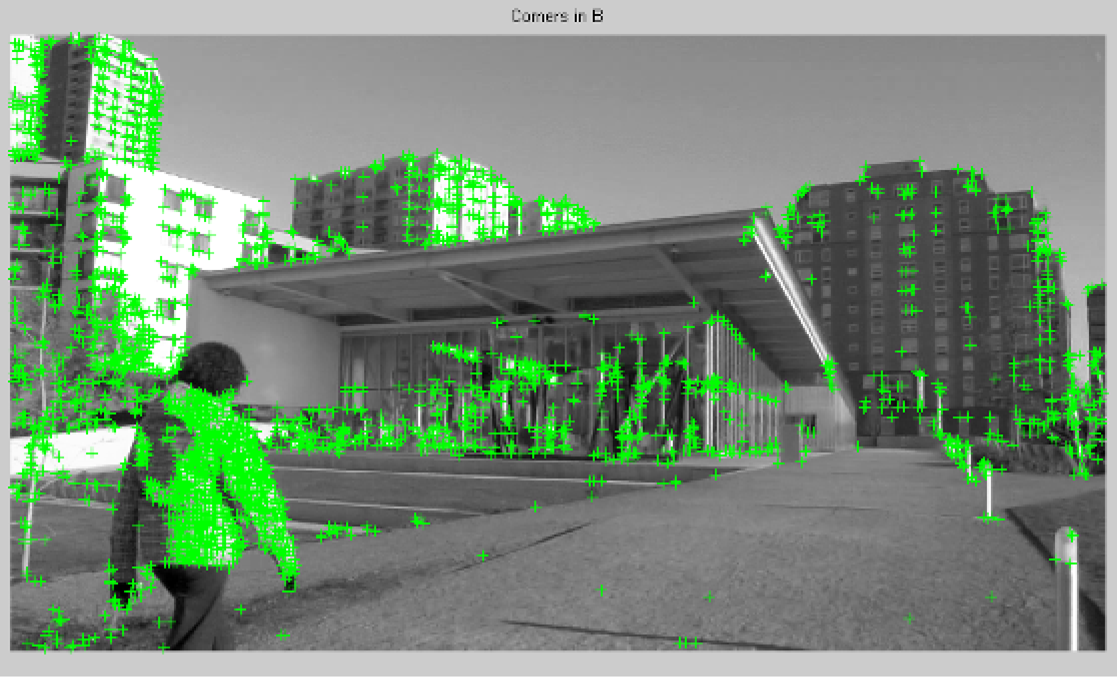
  Every pair of adjacent frames is selected and the rest of the steps are applied to it.

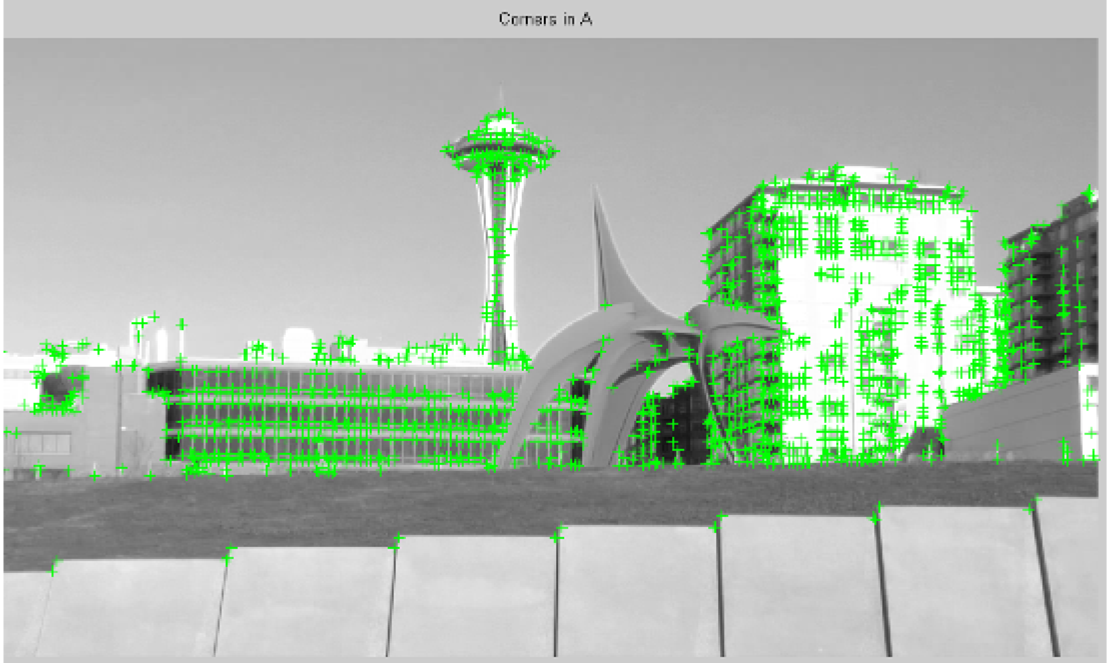
- **Collect Salient points**:

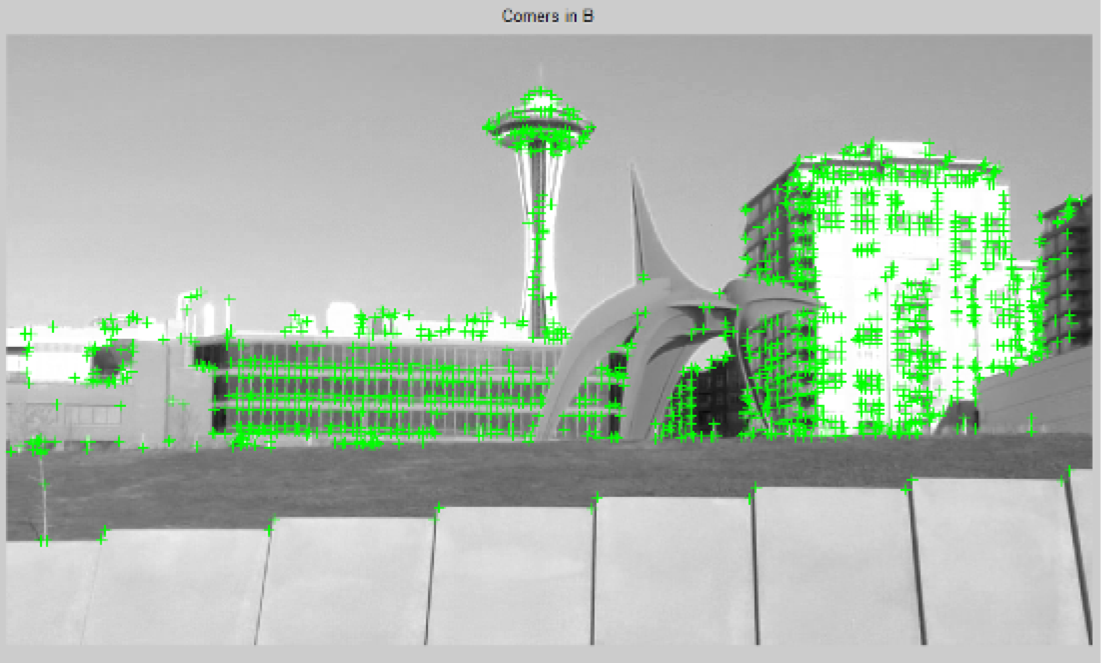
  An affine transform is to be found which accounts for the distortion between the two frames. To construct this transformation, a set of point correspondences are needed, which are derived from salient points of the frames. Salient points are selected in the presently concerned frame using any corner detection algorithm. The Features from Accelerated Segment Test (FAST) algorithm [7] has been used to find feature points of the frame.

Before this is done, it is essential that the extraneous detail and noise be removed for the corner detector algorithm to give a consistent output. So, the 3-dimensional RGB values are converted to simpler 1-dimensional gray-scale values by averaging the intensities and then normalising them.



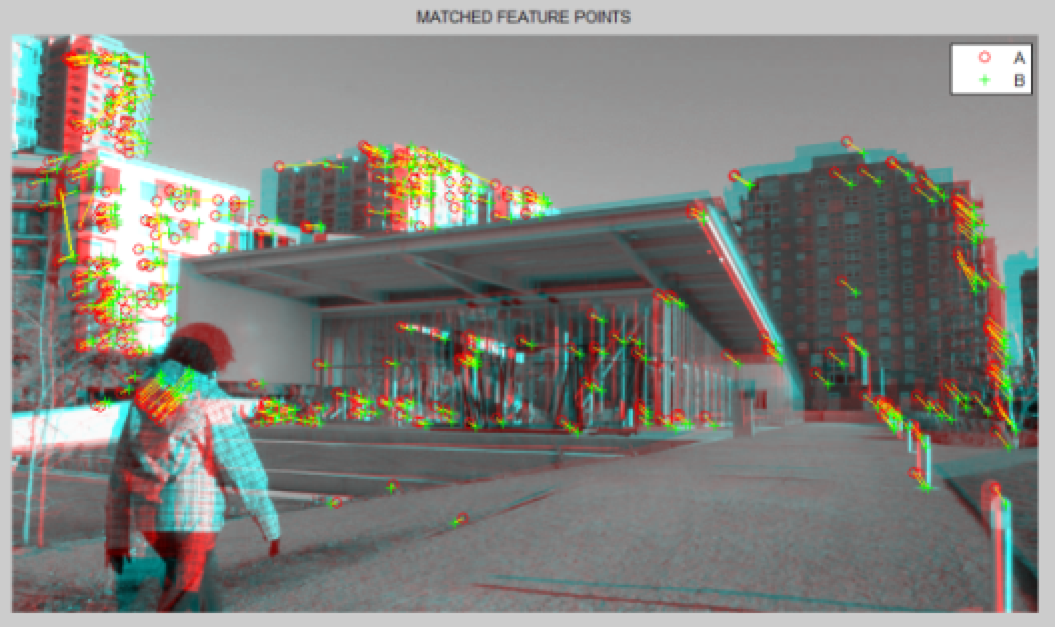


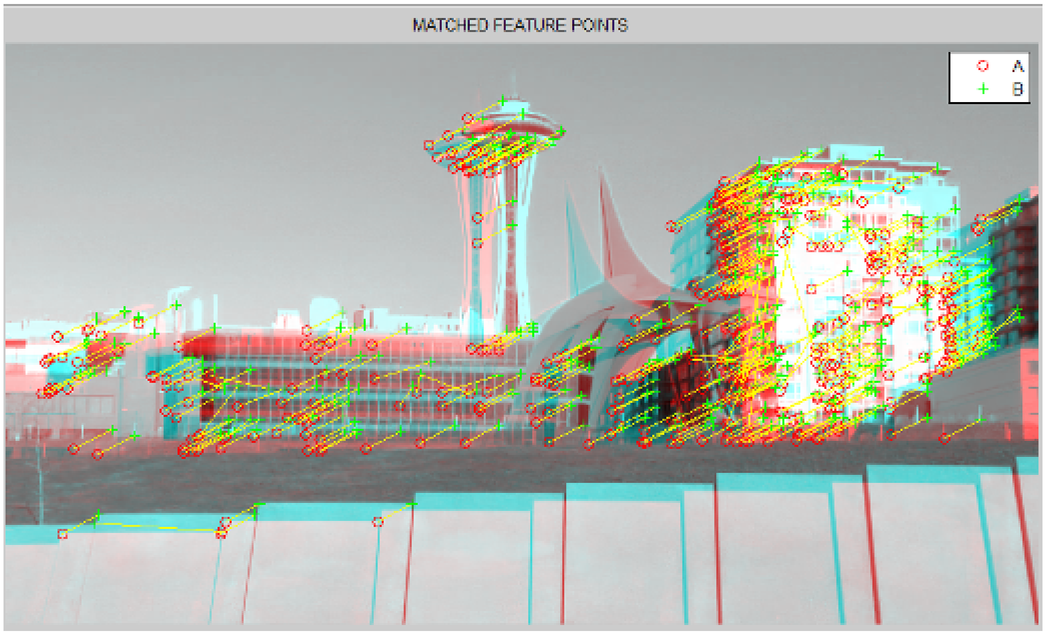

Corners in A


Corners in B

- **Select correspondences**:

  Next we pick correspondences between the points derived above. For each point, we extract a Fast Retina Keypoint (FREAK) [9] descriptor centered around it. The matching cost we use between points is the Hamming distance [10] since FREAK descriptors are binary.



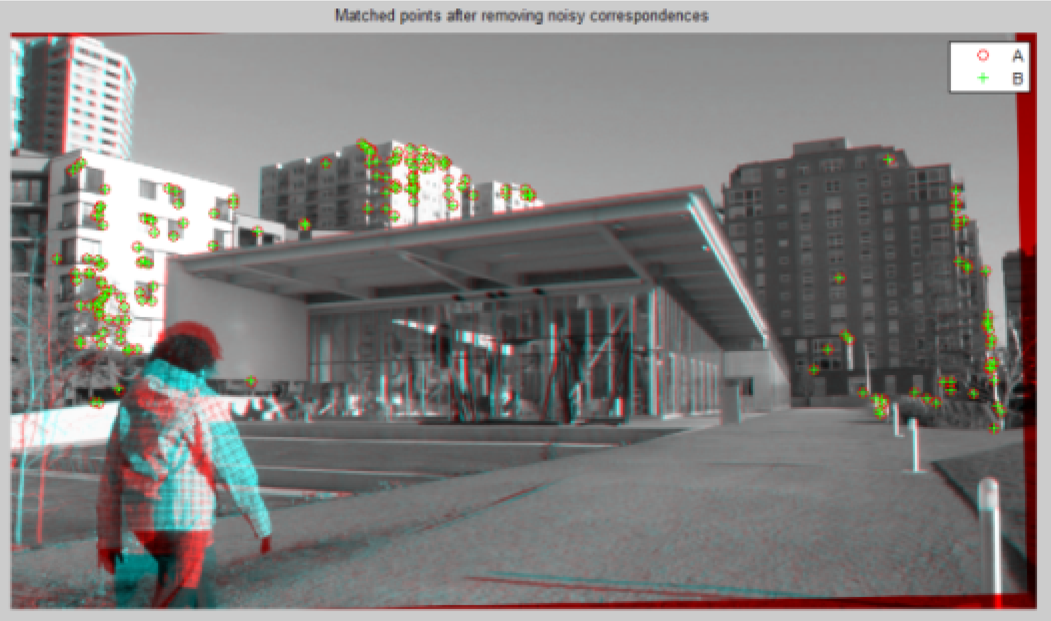


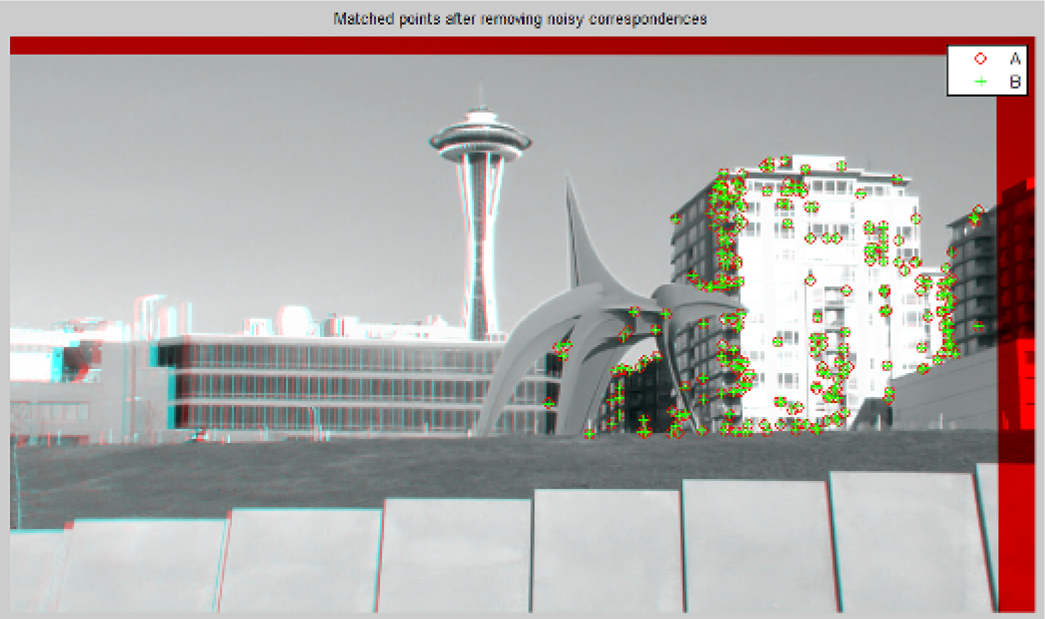
- **Estimating transform**:

  Many of the point correspondences obtained in the previous step are incorrect because of a considerable amount of outliers. So, we implement the MSAC [11] algorithm, which is a variant of RANSAC[12]. This considers only the inlier correspondences and therefore, a fairly good affine transformation is expected. To define the global contextual descriptor, we compute a matrix that represents the relative distribution of the other points in regards to the selected point. it will then derive the affine transform that makes the inliers from the first set of points match most closely with the inliers from the second set. This affine transform will be a 3-by-3 matrix.

$$\begin{pmatrix} a_1 & a_3 & 0 \\ a_2 & a_4 & 0 \\ t_x & t_y & 1 \end{pmatrix}$$

  These parameters of *a* denote the scale, rotation and shearing effects of the transform. The parameters of *t* are translation parameters. This transform is used to warp the images such that their corresponding features will be moved to the same image location.





B. Trimming the video

The video has been stabilized, but some of the frames have missing pixels due to warping of the frames. One method to remove this is trimming of the missing pixels from those frames. Depending on the size of frames we trim some parts from boundaries maintaining the aspect ratio of each frame.

After trimming of the boundaries, remaining missing pixels are removed by filling them with the pixel values from the previous frames.

III. OUTPUT

Given a shaky video our algorithm first estimates the camera motion and transforms the frames such that the camera motion is compensated. This video is passed as an input for algorithm which trims the video to reduce number of missing pixels. We have an option for filling the missing pixels with previous frames and then trimming. The final output will be a video which has no shake.

A. Observation

When the MSAC[11] algorithm is applied on the set of correspondences obtained from the `matchFeatures`[17] function, it gives an output showing the matched features, which are mostly background. This means that the affine transform infact corresponds to the camera motion, assuming that the backgorund is static in the interval of frames considered.

Trimming of the frames and filling missing pixels with previous frames still does not fully give perfect videos. However, video stabilization is achieved by these methods.

IV. Conclusion

We have used **SURF**[6] and **FAST**[7] feature extraction methods. Based on results Surf method works better than Fast feature extraction method. Trimming of video and filling missing pixels with neighbouring frame pixels gave better results compared to the untrimmed frames. Camera motion estimation done using SURF features has low value of rotation parameters compared to the ones done using FAST feature extraction method.

References

[1] Meng, Ling Han, Joseph Geumlek, Holly Chu, and Justin Hoogenstyrd. ”Contextual Point Matching For Video Stabilization.” Southern California Conference on Undergraduate Research, 2012.

[2] Tordoff, B; Murray, DW. ”Guided sampling and consensus for motion estimation.” European Conference on Computer Vision, 2002.

[3] Lee, KY; Chuang, YY; Chen, BY; Ouhyoung, M. ”Video Stabilization using Robust Feature Trajectories.” ICCV 2009.

[4] Litvin, A; Konrad, J; Karl, WC. ”Probabilistic video stabilization using Kalman filtering and mosaicking.” IS & T/SPIE Symposium on Electronic Imaging, Image and Video Communications and Proc., 2003.

[5] Matsushita, Y; Ofek, E; Tang, X; Shum, HY. ”Full-frame Video Stabilization.” Microsoft Research Asia. CVPR 2005.

[6] Bay, Herbert, et al. ”Speeded-up robust features (SURF).” Computer vision and image understanding 110.3 (2008): 346-359.

[7] Rosten, Edward, and Tom Drummond. ”Fusing points and lines for high performance tracking.” ICCV 2005.

[8] Rosten, Edward, and Tom Drummond. ”Machine learning for high-speed corner detection.” ECCV 2006.

[9] Alahi, Alexandre, Raphael Ortiz, and Pierre Vandergheynst. ”Freak: Fast retina keypoint.” Computer Vision and Pattern Recognition (CVPR), 2012

[10] Steane, Andrew M. ”Error correcting codes in quantum theory.” Physical Review Letters 77.5 (1996): 793.

[11] Wang, Hanzi, Daniel Mirota, and Gregory D. Hager. ”A generalized kernel consensus-based robust estimator.” Pattern Analysis and Machine Intelligence, 32.1 (2010): 178-184.

[12] Fischler, Martin A., and Robert C. Bolles. ”Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.” Communications of the ACM 24.6 (1981): 381-395.