

Efficient 3D stereo vision stabilization for multi-camera viewpoints

Sharif Shah Newaj Bhuiyan, Othman O. Khalifa

Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia

Article Info

Article history:

Received Jan 29, 2019

Revised May 12, 2019

Accepted May 28, 2019

Keywords:

Feature extraction

Image processing

Multi-camera stabilization

Similarity transformation

Video rendering

ABSTRACT

In this paper, an algorithm is developed in 3D Stereo vision to improve image stabilization process for multi-camera viewpoints. Finding accurate unique matching key-points using Harris Laplace corner detection method for different photometric changes and geometric transformation in images. Then improved the connectivity of correct matching pairs by minimizing the global error using spanning tree algorithm. Tree algorithm helps to stabilize randomly positioned camera viewpoints in linear order. The unique matching key-points will be calculated only once with our method. Then calculated planar transformation will be applied for real time video rendering. The proposed algorithm can process more than 200 camera viewpoints within two seconds.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Othman O. Khalifa,

Department of Electrical and Computer Engineering,

International Islamic University Malaysia,

Jalan Gombak, 53100 Kuala Lumpur, Malaysia.

Email: khalifa@iium.edu.my

1. INTRODUCTION

3D image or view requires at least two stereo images taken by two different viewpoints or multiple viewpoint of cameras. It could be at the exact same time or different interval of time. It is also called freezing time if at the same time. In this paper, our conducted experimentation involves more than two (2) stereo images. We have conducted our research very extensively up to 300 synthesis viewpoints using optical flow but 200 stereo images are real image from Canon DSLR. Here we added the various result of maximum 200 stereo images. The various number of image sets are 200, 120, 60, 55, 48, 24, 20, 12, 9, 7, 5, 4.

Viewpoint stabilization is one of the important phase in the development of multi-camera digital world. Single camera can also have different viewpoints if changes in time. It is a challenging step but not new in image and video software development. Human vision system (HVS) also has two viewpoints left eye and right eye for stereoscopic 3D depth. Viewpoint stabilization is a field of image registration process. There are lots of research on image registration using Scale-Invariant Feature Transform (SIFT), Random Sample Consensus (RANSAC) [1, 2], Speeded Up Robust Features (SURF) [3], Harris Corner Detection, Optical Flow (Lucas-Kanade method) algorithm using motion vector. But all the existing process are not very robust from practical points. Here we will show how our method can work in very random viewpoint like changes in focal length, orientation from different angles.

There are many real life applications including stereo image rectification, bullet time effect stabilization for 3D replay, 3D scene reconstruction [4], 3D preview for sports, object tracking from different angle [5-6], multiple robotic vision adjustment if human interaction is not possible, multi-camera viewing group programs in same time etc. Again 3D means very exciting topic for development and research in various fields. Using Harris-Laplace detector algorithm in pyramid level can give very accurate matching points in various lighting conditions or intensity differences, different focal length among different cameras

and rotational changes [7, 8]. To make the whole stabilization process 100% accurate, faster and minimize the stabilization error globally, we apply the spanning tree [9-11].

This paper presents a robust automatic multi-camera viewpoint stabilization method for 3D view. In this research, we are only focusing on multi-camera stabilization [12-15] at the same time (freezing time) with region of interest (ROI) in center. Shuaicheng, et al [16] compared between the traditional 2D stabilization (a single global camera path) and bundled camera paths stabilization where they generated comparable results to 3D methods while keeping merits of 2D methods. Though our new proposed method can be applied for other purposes of image and video stabilization system. The rest of this paper is organized as follows. Section 2 presents the basic idea of Harris Corner Detection, Laplace Pyramid scaling and Minimum Spanning Tree, Section 3 presents the related works, Section 4 presents the details of the proposed scheme, the Section 5 is the experimental results and the last is conclusion.

2. DEFINITION OF HARRIS CORNER DETECTION, LAPLACE MULTI-SCALE AND MINIMUM SPANNING TREE

Harris corner detector algorithm requires two scale parameters. One is differentiation scale and another integration scale. Differentiation scale requires for smoothing before applying image derivatives. Integration scale requires to control Gaussian window size for derivative responses. But in general Harris corner detection method is sensitive to photometric/contrast change, scale change and viewpoint change.

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \tag{1}$$

Here, $w(x,y)$ is the window function (square). We will replace it by Gaussian window (non-square).

To compute very small shift of change then we need to move to the differentiation. By expanding $E(u,v)$ to Taylor series and taking 2nd order, we get the following matrix form.

$$E(u, v) = [u, v] M \begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

Here, M is an intensity matrix of an image. It comes from image derivatives for change of intensity values. Matrix size is 2x2. Following is the details matrix for M .

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3}$$

For corner detection, we need to calculate Response value by following formula. Then find local maxima R map. You can use threshold value as well.

$$R = \text{determinant}(M) - k (\text{trace } M)^2 \tag{4}$$

This R value is not invariant to scale changes. So we must apply multi-scale Harris detector method for accurate function to find local key points which will be rotation and intensity invariant. To solve the contrast, scale and viewpoint issues, LOG (Laplacian-of-Gaussian) has been used to find characteristic scale. It will detect same interest points for images. This is the algorithm to solve our stabilization process faster. It is also very robust process against viewpoint & intensity changes.

Minimum Spanning Tree (MST) is an undirected connected graph where we can find a tree or relation with minimum number of edges. Edges can be weighted with different cost function or default is always 1. MST can be solved with very fast algorithm, it is called Union-Find disjoint set algorithm. Time complexity of this algorithm in worst case is $O(n)$, where n is the number of nodes. Also time complexity can be improved to $O(\log N)$ by rank or height. Figure 1 shows an example of calculating Minimum spanning tree with red color from an undirected graph.

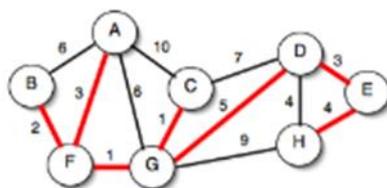


Figure 1. Calculating spanning tree

3. RELATED WORKS

In [5] Tomas Svoboda et al, they proposed a multi-camera calibration system. Here they used 1-point bright point for calibration which is easier to track in dark area. Reject any inconsistent points by RANSAC method. Calculate projective depth then filling missing points to perform scaling measurement. Using rank 4 factorization, it tries to find the Euclidean matrix structures to align with 3D world coordinate system. It is not a robust method for larger number of camera viewpoint system like 100 or 300 camera viewpoints. Also it depends on the outlier and inlier of RANSAC algorithm. If it does not find right points then it will not find the proper 3D coordinate system to align all the viewpoint images properly. Still need to wait for longer time. Their experiment shows full calibration time for 16 cameras were 60-90 minutes, which is not feasible for real time applications. Figure 2 shows a four cameras position C^1, C^2, C^3, C^4 have been shown with four corner rectangles for view P^1, P^2, P^3, P^4 . Then drawn four lines from object X_j to each camera position to explain the 3D projection matrix.

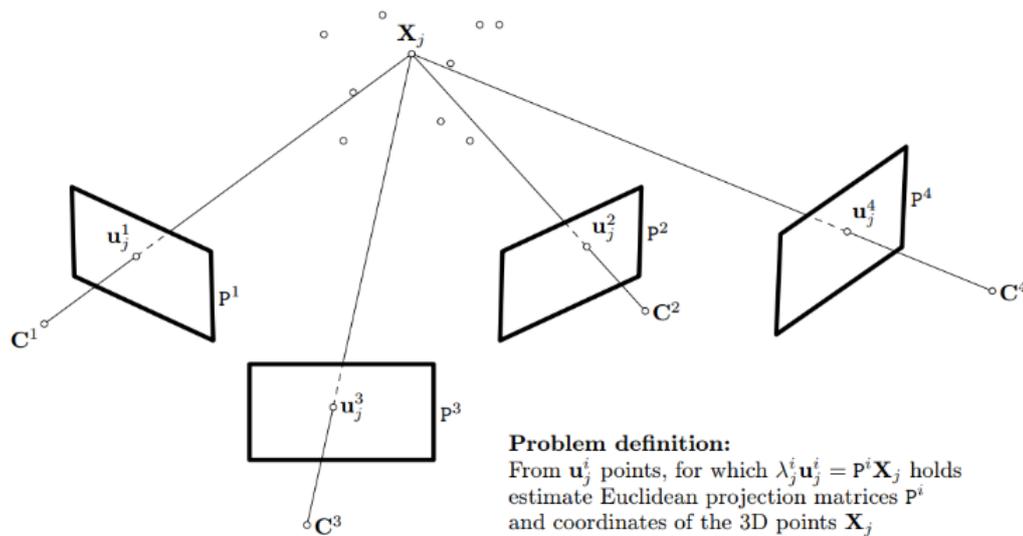


Figure 2. Stero vision [5]

In F. Perazzi et al. [6], they calculated motion vector by calculating Homography matrix between images then warped with minimum errors. They also introduced graph relation among all the individual images as Vertex and find edges. They called it maximum weighted graph matching. It runs in few iteration until it becomes one single image or component. But this process is very slow and not accurate all the time because of the optical flow implementation. It can also produce visual artifacts issues consistently.

In Harpreet S. Sawhney et al. [8], they didn't use matching feature point algorithm. They used four reference points from each image then did local translation and projective mapping. Finally applied Minimum spanning tree for global bundle optimization to minimize errors because of linear chain of frames. The problem formulation seems very unreliable as their matching process is Guess Work. They do normalized correlation matching with block by block. Then, they do sum of squared differences (SSD) for Laplacian pyramid. Most importantly, it is not very accurate and fast method which can be used for very high resolution images as well as huge number of cameras. Also can't be applied for automatic processing. In our proposed method, we have improved all the mentioned aspects like analyzing the key-points faster, reduced the time complexity, increased the stabilization efficiency and made the implementation simple and consistent.

4. PROPOSED METHOD

For our proposed method, we used the Harris corner detection with LOG for four (4) bright unique points for optimum output then apply the graph based MST algorithm to optimize the error through bundle adjustment. Then, calculated the planar similarity transformation (scale, rotation, translation) for unique matching key-points with reference points for alignment data and save the results for next stage. Then we

applied the planar similarity data matrix to the final image sets until the camera perspective or position is changed. We used image interpolation method in second stage for planar similarity matrix. It worked very fast and consistently for multi-camera viewpoints stabilization for 3D stereo view. If we increase the minimum image pair's correspondences from 4 to 8/10 then it takes little extra processing time with slight different image output but stabilization remains accurate. It is just the similarity transformation for images become little different. Our proposed stabilization process for multi-camera is below in block diagram. First we analyzed the matching key-points. Then we applied for the scene images.

Figure 3 shows the block diagram of the proposed method, first ROI is detected then applied the Harris Laplace detector to find four consistent key points to calculate the similarity transformation at the end. After Harris Laplace detector has been used, we have constructed graph image mosaic then applied the MST to find the best accurate closest four key points among other images.

During the alignment, we try to the minimizing the global matching error. Analyze will be the only first time for unstructured/un-stabilized multi-camera array. Once calculated the planar similarity transformation data, then it is very fast process with parallel threading in multi-core processing. We can improve the processing for video encoding if runs in GPU level. We always consider first image as reference image for similarity transformation and stabilize all other images based on the first image. We used FFmpeg to generate 3D video of freezing moment using aligned images. We can also apply for multiple video files, but will be longer process.

Figure 4 shows the image warping interpolation wher the image pixel is transformed for un-aligned images based on the calculated similarity transformation with previous pair or base image. Transformation and interpolation process will reconstruct the new aligned image. There will be small area of black region for it. Figure 5 shows a diagrams that showing three different structure of multiple camera array positions. It helps to understand how the algorithm to design and perform better in all environments.

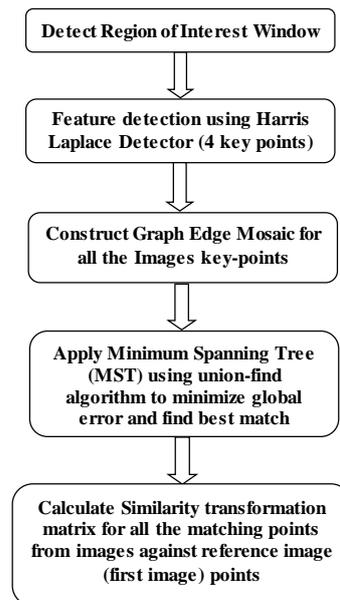


Figure 3. Proposed method

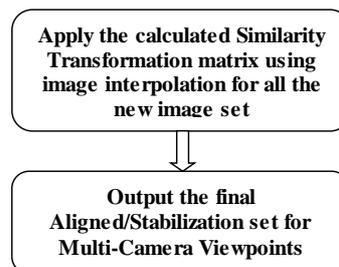


Figure 4. Image warping interpolation

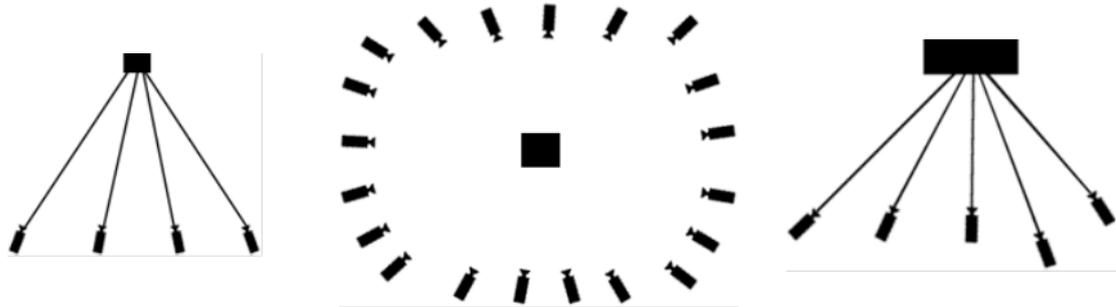


Figure 5. randomly unstructured camera array setup example(s)

5. EXPERIMENTAL RESULTS

We performed various experiments with our own multiple data sets for our method's accuracy. For all of our image sets, we used Canon DSLR cameras. We collected images from various sources to test our methods. Figure 6 shows a Non-stabilized (aligned) 4 images set, while Figure 7 shows a stabilized (aligned) 4 images set with black area.

Figure 8 shows Spanning Tree graph edge and Figure 9 shows the Edge for matching points. All image set of 4 Canon DSLR. Resolution is 2592x1728, however, the time took for finding matching key-points & alignment is 0.2 ms total in normal computer.

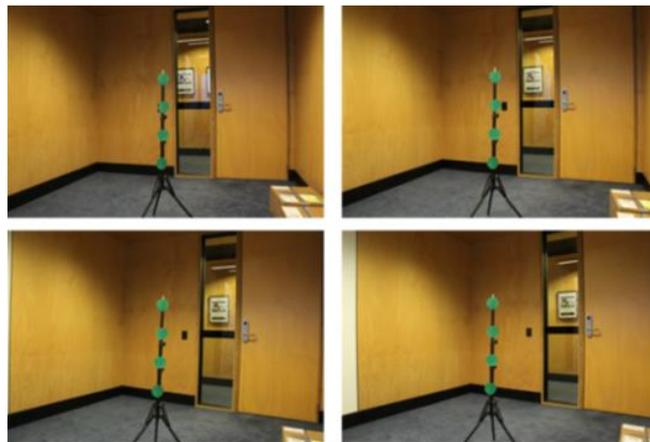


Figure 6. Non-stabilized (aligned) 4 images set

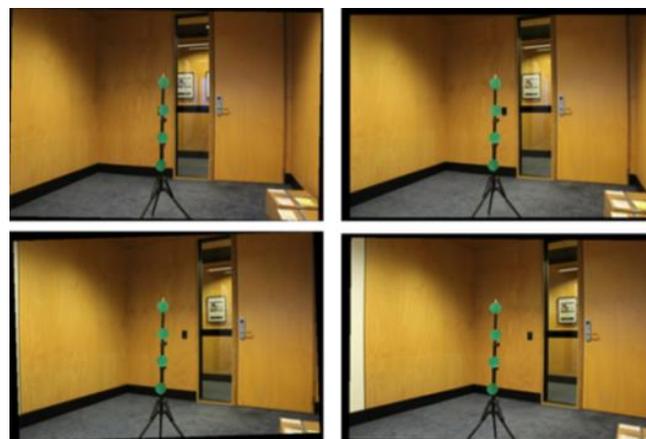


Figure 7. Stabilized (aligned) 4 images set with black area

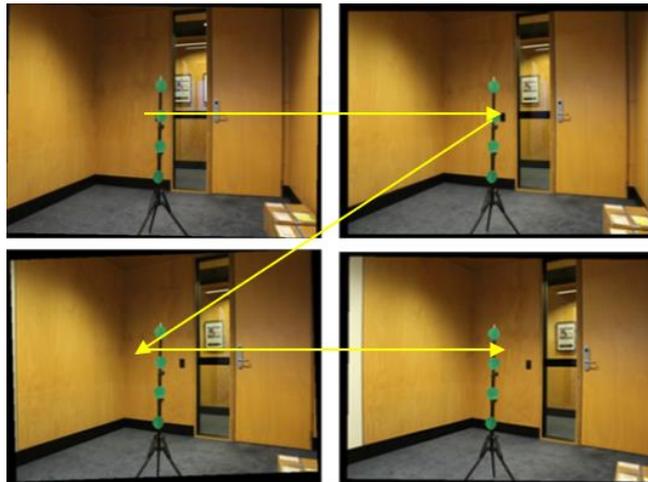


Figure 8. Spanning Tree graph edge



Figure 9. Edge for matching points

In Figure 10, an example of randomly unstructured camera array in linear order patterns. For this pattern, MST works very fast and accurate to minimize the whole stabilization error. Middle line is the base line which is first camera. It is the reference image here. From the qualitative experimental results, we can see that using MST process, we can always get accurate stabilization 3D view. Our proposed method is very clean and anyone can easily follow the whole process to implement it in their own development.

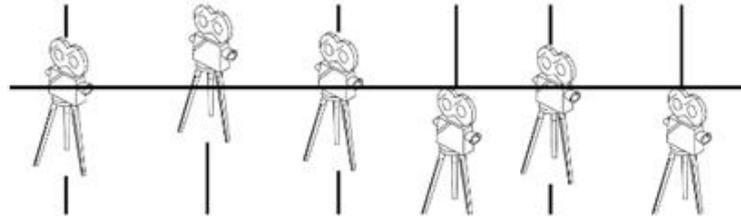


Figure 10. Vertical and horizontal camera position

Table 1 shows the quantitative result of the proposed method as shown in Figure 3 for cameras 24, 48, 60, 120, 200. Used image resolution are 2592x1728 and 5184x3456. For large camera array (number of images), higher memory is important. Otherwise processing will be slower. Also, Solid State Drive (SSD) is important for faster disk operation.

Table 1. Key-points finding with MST method

Image Resolution	2592x1728	2592x1728	2592x1728	2592x1728	5184x3456	5184x3456
Number of Cameras (images)	24	60	120	200	48	120
Connectivity	Able to connect fully (24)	Able to connect fully (60)	Able to connect fully (120)	Able to connect fully (200)	Able to connect fully (48)	Able to connect fully (120)
Computer	Core i7, 8GB	Core i7, 32GB	Core i7, 32GB	Core i7, 32 GB	Core i7, 16GB	Core i7, 32 GB
Average Time	0.52 sec	0.93 sec	1.36 sec	1.91 sec	1.19 sec	1.43 sec

After calculating the similarity transformation, we do not need to find those matching key-points again until cameras' position (perspective) is changed. So, it can save almost half of our time by not calculating matching transformation points again.

Table 2 shows the quantitative result of the proposed method as shown in Figure 4 for cameras 24, 48, 60, 120, 200. Used image resolution are 2592x1728 and 5184x3456. This phase is only for applying transformation matrix data to original image for new aligned image reconstruction. Because similarity transformation matrix is calculated only once at the key-point detection and analyzing phase.

Table 2. Image warping with planar similarity transformation

Image Resolution	2592x1728	2592x1728	2592x1728	2592x1728	5184x3456	5184x3456
Number of Cameras (images)	24	60	120	200	48	120
Computer	Core i7, 8GB	Core i7, 32GB	Core i7, 32GB	Core i7, 32GB	Core i7, 16GB	Core i7, 32 GB
Average Time	0.81 sec	1.11 sec	1.29 sec	2.05 sec	1.09 sec	1.73 ec

6. CONCLUSION

This proposed method works very fast, accurate and shows consistent speed for Multi-camera stabilization. It then creates 3D effect from images via rendering. It can be used for Video Stabilization directly and run in GPU for fast processing time. By using MST algorithm, it helps to minimize the global error for matching key-points very fast and accurately. Then just apply the analyzed transformation data for

subsequent multi-camera image sets instantly. Our current data shows that it works very well for randomly placed camera, also video output is very smooth and accurate. It can be improved further by reducing the original images resolution then analyze the unique key-points in smaller image sets for faster processing. In future, we are hoping to apply it for any 3D view reconstruction with faster processing and accurate stabilization. Then we don't have worry about manual work for aligning all the cameras in larger production.

REFERENCES

- [1] M safy, Guangming Shi, Zhenfeng Li and Ahmed Saleh Amein, "Improved Harris corner detector algorithm for image co-registration", *International Conference on Computer Science and Network Technology (ICCNT)*, 2013.
- [2] Matthew Brown & David G. Lowe, "Automatic Panoramic Image Stitching using Invariant Features, Department of Computer Science", University of British Columbia, 2006.
- [3] Nan Geng, Dongjian He and Yanshuang Song, "Camera Image Mosaicing Based on an Optimized SURF Algorithm", *International Journal of Electrical and Computer Engineering (IJECE)*, 2012, 10(8): 2183-2193.
- [4] Vladimir Kolmogorov, Ramin Zabih, Steven Gortler, "Generalized Multi-camera Scene Reconstruction Using Graph Cuts", *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2003.
- [5] R. Guerschouche, F. Coldefy and T. Zaharia, "Accurate Camera Calibration Algorithm Using a Robust Estimation of the Perspective Projection Matrix", *Proceedings Volume 6315, Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications IX: 63150D*, San Diego, California, United States. 2006.
- [6] Tomas Svoboda, "A Software for Complete Calibration of Multicamera Systems", Czech Technical University, Faculty of Electrical Engineering, 2005.
- [7] F. Perazzi and et al, "Panoramic Video from Unstructured Camera Arrays", *Computer Graphics Forum*, 2015.
- [8] Chin-Sheng Chen, Kang-Yi Peng, Chien-Liang Huang and Chun-Wei Yeh, "Corner-Based Image Alignment using Pyramid Structure with Gradient Vector Similarity", *Journal of Signal and Information Processing*, 2013.
- [9] Harpreet S. Sawhney, Steve Hsu and R. Kumar, "Robust Video Mosaicing through Topology Inference and Local to Global Alignment", *European Conference on Computer Vision (ECCV)*, 1998.
- [10] S. S. N. Bhuiyan and O. O. Khalifa, "Robust Automatic Multi-Camera Viewpoint Stabilization using Harris Laplace corner detection and Spanning Tree," *2018 7th International Conference on Computer and Communication Engineering (ICCCCE)*, Kuala Lumpur, 2018, pp. 1-5.
- [11] Andrew Richardson, Johannes Strom and Edwin Olson, "AprilCal: Assisted and repeatable camera calibration", University of Michigan, 2013.
- [12] R.A. Setyawan, R. Sunoko, M.A. Choiron, P.M. Rahardjo, "Implementation of Stereo Vision Semi-Global Block Matching Methods for Distance Measurement," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, 12(2), pp. 585-591, 2018.
- [13] Yasutaka Furukawa, Jean Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment", *International Journal of Computer Vision*, 2009.
- [14] C. Strecha, W. von Hansen, L. Van Gool, P. Fua and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1-8.
- [15] C. Banz, S. Hesselbarth, H. Flatt, H. Blume and P. Pirsch, "Real-time stereo vision system using semi-global matching disparity estimation: Architecture and FPGA-implementation," *2010 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, Samos, 2010, pp. 93-101.
- [16] L. Shuaicheng, L. Yuan, Ping Tan, Jian Sun, "Bundled Camera Paths for Video Stabilization", *ACM Transactions on Graphics (TOG)*, 2013.