

A Performance Evaluation of a Real-time Image Mosaicing with a Rapid Prototype Hardware

^{*1}K. Sai Venu Prathap, ²S. A. K. Jilani, ³P. Ramana Reddy

¹ Research Scholar, E.C.E Department, JNTUA, Ananthapuramu, ² Supervisor & Professor, E.C.E Department, MITS. Co-supervisor & Professor, JNTUA CE, Ananthapuramu

Email: [*venu9459@gmail.com](mailto:venu9459@gmail.com), drsakjilani@mits.ac.in, prjntu@gmail.com

Received: 06th October 2017 Accepted: 14th November 2017, Published: 31st December 2017

Abstract

The Field-Of-View(FOV) of compact cameras are very less compared to the FOV of humans. A single shot of capturing compact camera sensor does not capture the scenes as the human's vision. Mosaicing is a procedure of uniting the partial parts of a scene to form a large image that represents the whole scene. The main aim of mosaicing is to increase the FOV and to trace the path of any moving objects. In this paper, a real-time mosaicing is implemented with the various algorithms and compared with the proposed algorithm. The proposed method is a combination of FAST detector and Steerable filters. All the algorithms are executed with the multiple rapid prototype hardware's and compared their performance in terms of many parameters like corner detection rate, image mosaicing time, video mosaicing execution time. Geekbench mark test of having various parameters is conducted on three hardware's. Experimental results proved that the intel compute stick and Lenovo Ideacentre 300 hardware have high mobility to capture the remote areas of a scene or low altitude images to construct the mosaic image.

Keywords: Mosaic image, feature points, Intel Compute stick, Lenovo Ideacentre 300, FAST-Steerable filters.

Introduction:

There are some situations where the scene is beyond the scope of the camera. Compact cameras have low FOV compare to the human visual system because the Field-Of-View(FOV) of compact cameras are around the 35x55 degrees where as human being has 135x180 degrees of FOV. In this scenarios mosaic procedure is applied to construct the scene with more FOV and the image obtain the by this process is called the mosaic image. Image mosaic is a procedure of uniting partial images of a scene to produce large mosaic image [1]. Feature detectors detects all the keypoints of an image and descriptors extract the data around each corner point [2]. Feature based mosaic algorithms have some common phases they are capturing the real-time scene, interest points detections, extracting the information around the corner points, comparing the feature matching using the hamming distance, mapping the matched corner points by using geometric measurement and warping and blending. Mosaicing is constructed with the SUSAN, Harris, KLT, FAST and proposed methods. All algorithms are compared in terms of accuracy, recall and true cornerness of synthetic image.

Farzin M. and Farahnaz M. computed the performance of various algorithms in terms of accuracy [3]. So many authors introduced many algorithms each have their own advantage and disadvantages. Harris, SUSAN, KLT, Tomsai and Shi and FAST algorithms are familiar feature detectors.

Among above mentioned detectors, only FAST algorithm is best suited for real-time applications like Simultaneous Localization And Mapping [4]. Harris detector is an efficient method, but it has more computation time compare to the FAST method. SUSAN if purely applicable for non-real-time scenarios because it takes much time for corner detection than any other algorithms. In this paper a real-time image mosaicing and video mosaicing is implemented with the existing methods and as well as with the proposed method with three hardware. The proposed algorithm is a combination of FAST detector and steerable filters. The FAST detector detects the feature points and steerable filters are introduced to steer the image in any orientation. Steerable filters main advantage is selective orientation.

A real-time mosaicing of each algorithm is executed with the three hardware devices and their performance is compared in terms of corner detection rate, image mosaicing computation time and video mosaicing time. The hardware performance is measures with the GeekBench mark tool. Though the performance of laptop is high but the mobility (Dimensions and weight) is poor when compared with the intel compute stick and Lenovo Ideacentre Stick300 on board computer devices. This stick is used to capture the remote images and construct the mosaic image in real-time. This compute sticks can be stick to any moving vehicle like drone and capture the remote images. So, need of specialized hardware required for computer vision applications [5].

Phases of Mosaicing:

Feature based methods [6] have common steps for the construction of mosaic image, they are depicted in below figure.1. Initial phase stars with the capturing of frames and each selective frame under goes feature detection and extraction. The similarity between two frames is measured by the hamming distance technique [7]. Transformation is used to map the selected consecutive frames one by one. Warping corrects the dimensional distortions of geometry.

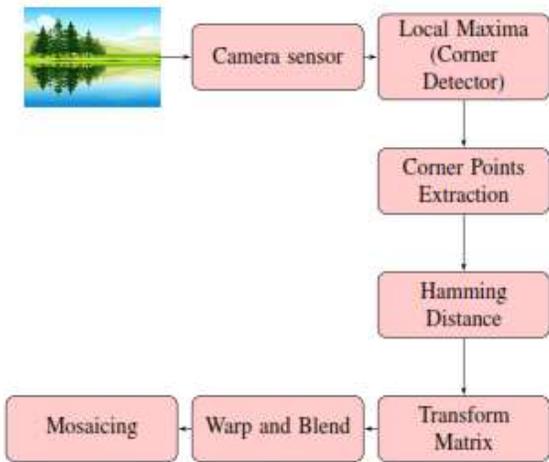


Fig.1:

Workflow of real-time image mosaicing

Keypoint Detectors:

SUSAN: SUSAN synonym stands for “Smallest Univalve Segment Assimilating Nucleus”. No derivatives are utilized to find corner points [8], it works with a mask of center pixel as nucleus.

$$c(\vec{m}) = e^{-\left(\frac{I(m) - I(m_0)}{t}\right)^6} \quad (1)$$

Here $I(m_0)$ is the intensity of origin pixels and $I(m)$ is the intensity of other pixels.

KLT Detector: KLT Stands for “Kanade-Lucas-Tomasi”. KLT detector computes the derivative of an images [9]. It is an imitating algorithms of Harris detector. A corner point has two high variation orthogonal directions. It varifies every pixel of an image with a local structure as given below

$$M_{Loc} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2)$$

Harris: Harris detector is based on the Moravec algorithm [10]. Harris detected proposed corner point detector in 1988 depending up on the intensity variation of the pixels of an image [11]. It is window based algorithms

$$E(p,q) = \sum_{s,t} W(s,t) [I(s+p, t+q) - I(s,t)]^2 \quad (3)$$

where $E(p, q)$ = Intensity difference between shifted pixel and original pixel.

$W(s, t)$ = Desired window at (p, q)

$I(s, t)$ = original frame intensity

FAST: FAST stands for “Features from Accelerated Segment Test”. It was proposed in 2006 by Rosten and Tom [4]. It considers only the pixels on a circle of fixed radius.It

compares the intensity of pixels on a circle of fixed radius with the center pixel. A pixel is a recognized as a corner, if the intensity of 12 contiguous pixels out of 16 pixels should be greater or lesser than the center pixel as given in below equation.4.

$$S_X = \begin{cases} \text{Darker} & I_X \leq I_X - t \\ \text{Similar} & I_X - t < I_X < I_X + t \\ \text{Lighter} & I_X + t < I_X \end{cases} \quad (4)$$

where S_X is the pixel state, I_X is the intensity of pixel X and t is the threshold value.

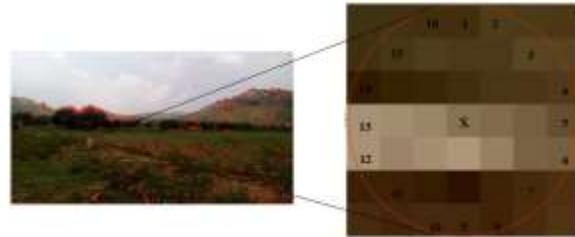


Fig.2: Center pixel X surrounded by 16 pixels on an equal circle of radius.

Proposed Algorithm: It is combination of FAST detector [12] and Steerable filters [13], [14]. Though FAST detector has quick computation time but poor in accuracy, true cornerness and recall criteria’s. The main advantage of steerable filters is that an image can be steer at any used defined orientation that is orientation selective.

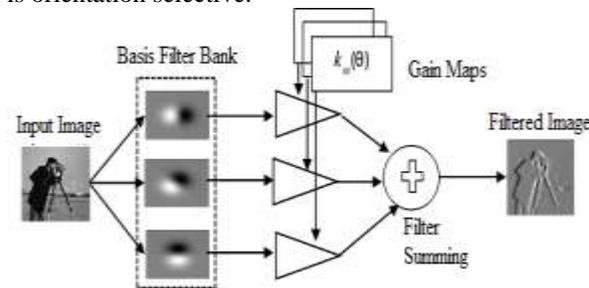


Fig.3: Block Diagram of Proposed algorithm

The above figure.3 depicts the various blocks of steerable filter like basis filters bank, Gain maps. For real-time applications only one basis filter is chosen for quick access. After obtaining filtered image, FAST detector is applied for feature detection.

Hardware Description:

The Compute sticks are thin and light compare to the laptop having inbuilt wi-fi and Bluetooth with 32GB extended memory with USB port. This sticks has high mobility in the form of their dimensions and weight compare to the laptop. Though desktop or laptop have high efficient performance and computation time but due to poor mobility they are strictly confined to laboratory verification. The Table.1 describes the specifications of laptop, Intel Compute stick and Lenovo Ideacentre 300 hardware devices. The Figure.4 illustrates the dimensions of compute sticks

and laptop used in the construction of mosaic image in real-time. As the dimensions increases the mobility decreases and vice versa. Efficient mobility devices are used to trace the path of moving object [15] and to capture remote images.



Fig. 4: Intel Compute Stick, Lenovo Ideacentre Stick 300 and laptop with their dimensions

Performance:

Initially performance of the detectors like Harris, KLT, SUSAN, FAST and FAST-Steerable are evaluated in terms of accuracy, recall and true cornerness. A synthetic test image shown in figure.5 is evaluated with all the parameters mentioned above. This synthetic image has 35 ground corner points(Ng). Ground truth corner points are the majority human judgement corner points. Each algorithm is implemented on the test image and noted the total detected corner points(Na) and exactly detected the corner points. An algorithm performance is high as accuracy [3], true cornerness and recall are close to 100 percentage. Matched corner points are exactly corner points. False positive are not the corner points but detected as corners by an algorithm and false negative are corner points which was not detected by the method. Table.2 and 3 illustrates the ground truth corners, total detected corners, matched corners, false positive points, false negative points, accuracy, recall, true cornerness of above mentioned algorithms for a synthetic image.

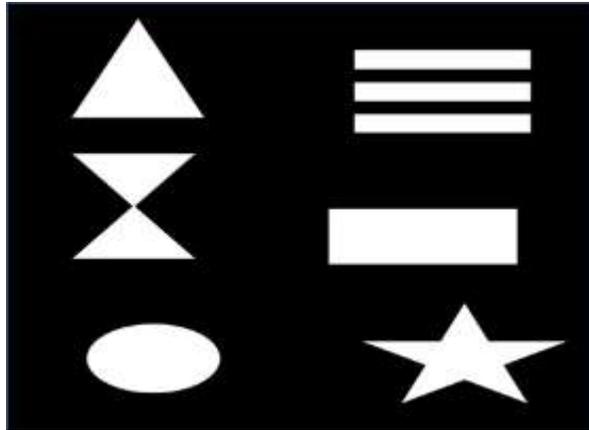


Fig.5: Synthetic Image

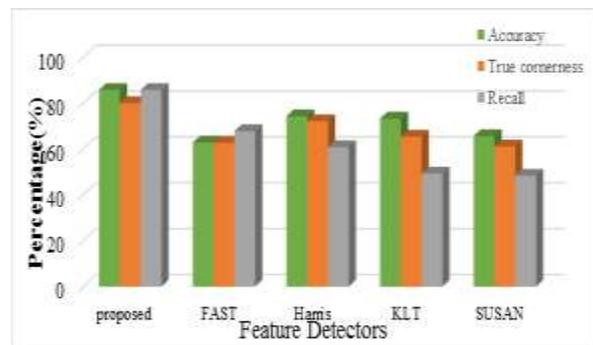


Fig.6: Accuracy, True Cornerness and Recall of all detectors obtained with the synthetic image.

The figure.6. illustrate that, proposed method has highest accuracy, true cornerness and recall than the remaining methods. The Table.4 depicts the corner detection rate of various algorithms. Though FAST has quick computation time but poor in accuracy, recall and true cornerness. The proposed method corner detection rate is 10.4 times faster than SUSAN, 5.24 times faster than KLT and 1.44 times faster than Harris methods with the laptop. The Intel Compute Stick and Lenovo Ideacentre 300 sticks has almost equal performance. Table.5 and 6 describes the total frames in the video and frame frequency. If all frames in a video are considered to mosaicing than the computation time increases a lot. As the frame

Table 1: Specifications of Intel Compute Stick, Lenovo Ideacentre stick and Laptop

| Parameters | Intel Compute Stick | Lenovo Ideacentre Stick 300 | Laptop |
|-----------------|---------------------|-----------------------------|--------------|
| Processor | Quad Core Intel | Quad Core Intel | Intel i3 |
| Clock Frequency | 1.333GHz | 1.333Ghz | 2130MHz |
| System Memory | 2GB | 2GB | 3GB |
| Storage Memory | 32 GB eMMC | 322 GB eMMC | 320 GB |
| Size | 103x37x12mm | 101x38x15mm | 378x262x31mm |
| Power | 5V, 2A | 5V,2A | 19.5V, 4.62A |
| Weight | 53.52g | 63.95g | 5.8 pounds |

Table 2: Corner Classification of various detectors for the synthetic test image.

| Algorithm | Ground Truth Corner (Ng) | Total detected Corners (Na) | Matched Corners (Nm) | False positive | False negative | Execution time |
|-----------|--------------------------|-----------------------------|----------------------|----------------|----------------|----------------|
| Harris | 35 | 51 | 31 | 20 | 6 | 0.407 sec |
| SUSAN | 35 | 60 | 29 | 31 | 20 | 2.788 sec |
| KLT | 35 | 69 | 34 | 35 | 2 | 1.611 sec |
| FAST | 35 | 32 | 21 | 11 | 13 | 0.108 sec |
| Proposed | 35 | 35 | 30 | 5 | 5 | |

Table 3: Multiple criteria of corner detectors for synthetic image.

| Algorithm | Ground Truth Corner (Ng) | Total detected Corners (Na) | atched Corners (Nm) | Accuracy | Recall | True Cornerness |
|-----------|--------------------------|-----------------------------|---------------------|----------|--------|-----------------|
| Harris | 35 | 51 | 31 | 74.25% | 60.78% | 72.09% |
| SUSAN | 35 | 60 | 29 | 65.59% | 48.33% | 61.05% |
| KLT | 35 | 69 | 34 | 73.20% | 49.27% | 65.38% |
| FAST | 35 | 32 | 21 | 62.81% | 67.74% | 62.68% |
| Proposed | 35 | 35 | 30 | 85.71% | 85.71% | 80.00% |

Table 4: Corner Detection Rate

| Algorithms | Computation time of Detectors (seconds) | | |
|------------|---|---------------------|--------|
| | Lenovo Ideacentre Stick 300 | Intel Compute Stick | Laptop |
| SUSAN | 4.40 | 4.81 | 2.60 |
| KLT | 2.07 | 2.23 | 1.31 |
| Harris | 0.41 | 0.45 | 0.36 |
| FAST | 0.23 | 0.22 | 0.11 |
| Proposed | 0.37 | 0.39 | 0.25 |

frequency increases from 1 to 30th than the mosaic time also efficiently changes. For the real-time applications the frame frequency is high leads to quick mosaicing computation time as depicted in Table.5. The mosaic image is constructed with proposed method by using only 14 frames out of 409 frames as shown in figure.10, it has more Fied-of-View [16]. The execution time of mosaic with the laptop is 7.25 seconds and with the intel compute stick is 12.55 seconds and with the Lenovo Ideacentre stick 300 computation time is 11.60

seconds only. Similarity is measured with the hamming distance calculation [17]. RanSac is applied to map the frames of a video [19]. Seamless mosaic image can be obtaining with the blending technique [18].

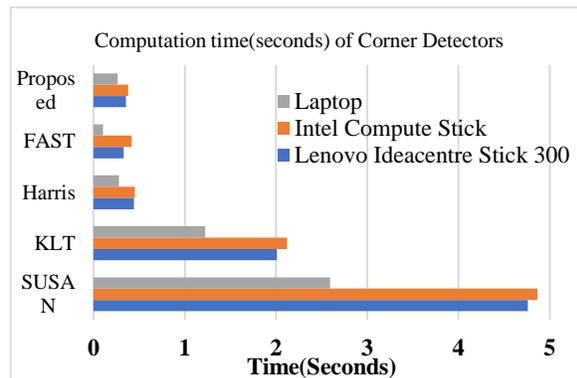


Fig.7: Execution time of various detectors with distinct hardware.

Table 5: Mosaicing the frames of a video with the FAST detector

| Total video frames | Frame Frequency | Frames used for mosaic | Computation time of Hardwares(seconds) | | |
|--------------------|------------------|------------------------|--|---------------------|--------|
| | | | Lenovo Ideacentre Stick 300 | Intel Compute Stick | Laptop |
| 409 | 1(all) | 409 | 158.6 | 168.6 | 101.2 |
| 409 | 2 nd | 205 | 76.23 | 80.55 | 58.75 |
| 409 | 5 th | 82 | 44.67 | 46.40 | 21.94 |
| 409 | 10 th | 41 | 22.64 | 26.81 | 12.01 |
| 409 | 20 th | 21 | 12.11 | 12.69 | 7.508 |
| 409 | 30 th | 14 | 7.581 | 7.888 | 4.89 |

Table 6: Mosaicing the frames of a video with the proposed detector

| Total video frames | Frame Frequency | Frames used for mosaic | Computation time of Hardwares(seconds) | | |
|--------------------|------------------|------------------------|--|---------------------|--------|
| | | | Lenovo Ideacentre Stick 300 | Intel Compute Stick | Laptop |
| 409 | 1(all) | 409 | 176 | 190.1 | 118.6 |
| 409 | 2 nd | 205 | 128.8 | 131.5 | 62.81 |
| 409 | 5 th | 82 | 47.59 | 49.21 | 23.91 |
| 409 | 10 th | 41 | 23.31 | 26.29 | 12.30 |
| 409 | 20 th | 21 | 16.40 | 17.90 | 9.310 |
| 409 | 30 th | 14 | 11.60 | 12.55 | 7.251 |

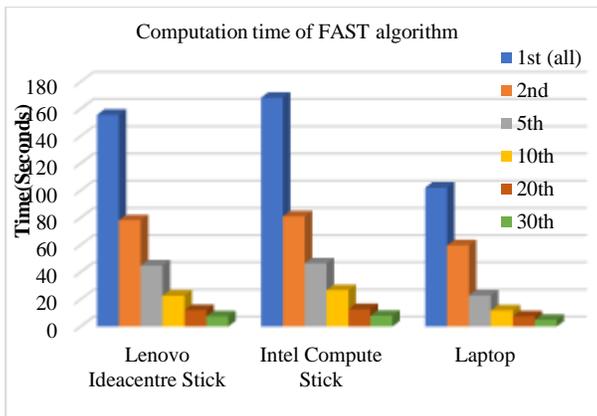


Fig.8: Computation time of FAST corner Detector

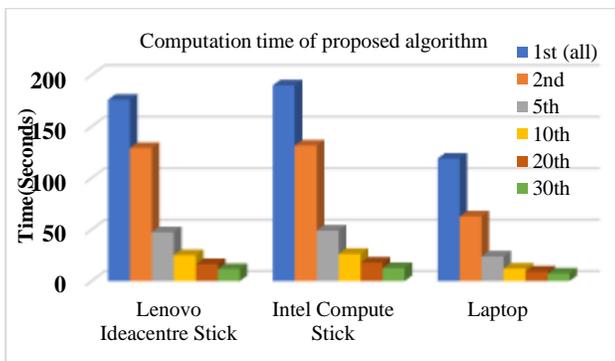


Fig.9: Computation time of proposed FSD corner Detector



Fig.10: Mosaic image constructed with 14 frames out of 409 frames of a real-time video with proposed method.

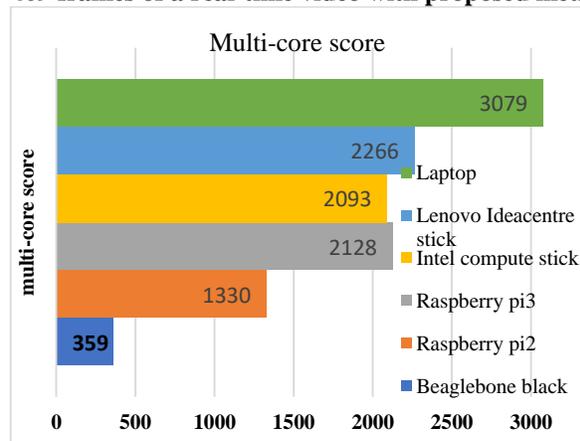


Fig.11: GeekBench mark score of Laptop, Lenovo stick, Intel stick, Raspberry pi3, Raspberry pi2 and Beagle bone black

GeekBench mark test is an application used to calculate the performance of electronics devices in terms of various categories. Figure.11 depicts the results of Geek bench test of Laptop and compute sticks in various parameters like memory latency, Dijkstra, camera and so on.

Conclusion:

In this paper, an innovative method was implemented with the general-purpose hardware devices. The main advantage of this approach is no need of specialized hardware for the mosaicing of remote images. The proposed method was 1.44 times faster than Harris corner detector, 5.24 times faster than KLT method and 10.4 times quicker than SUSAN technique. Corner detection rate is calculated with all the detectors and hardwares. Frames of a video are used to mosaic and obtain mosaic image with the three hardware devices by the FAST and proposed methods. Geek bench test is conducted for the three hardware devices to calculate the performance. Experimental results proved that proposed method is best suited for the real-time image mosaicing with an efficient accuracy, recall and true cornerness parameters and with extended field-of-view.

References:

1. Mahesh, Subramanyam M.V. Automatic feature based image registration using SIFT algorithm. ICCNT-12, Coimbatore. July 2012.
2. Katari.S.V.P., Jilani, Ramana R.P., A Critical Review on Image mosaicing. International Conference on Computer Communication and Informatics, Jan.07-09, 2016, Coimbatore, INDIA.
3. Farzin M. and F. Mohanna, Performance evaluation of corner detectors using consistency and accuracy measures Computer Vision and Image Understanding Volume-102, pages 81-94, 2006.
4. E. Rosten and T. Drummond, Machine learning for high speed corner detection. in 9th European Conference on Computer Vision, vol. 1, 2006, pp.430-443.
5. David A. Forsyth, Jean Ponce., Computer vision: A Modern Approach. Prentice Hall, August 2002.
6. Mallick and S. Prakash, Feature based image mosaicking, Department of Electrical and Computer Engineering, University of California, San Diego, 2002.
7. K.S.V. Prathap, Jilani S.A.K and P.R.Reddy A Real-Time Image Mosaicing Using FAST Detector and FREAK Descriptor IEEE WISPNET 2017 conference.PP.2444-2449. DOI.978-1-5090-44429/17/\$31.00©2017.
8. S. M. Smith and J. M. Brady (May 1997). SUSAN-a new approach to low level image processing. International Journal of Computer Vision. 23 (1): 45-78.DOI:10.1023/A:1007963824710.
9. Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
10. H. Moravec, Obstacle avoidance and navigation in the real world by a seeing robot rover, Tech Report CMURI-TR-3 Carnegie-Mellon University, Robotics Institute, 1980.
11. Harris and Stephens, A combined corner and edge detector, in Proceedings of the 4th Alvey Vision Conference, pp. 147-151, 1988.
12. Nigar and Mahesh, Image mosaic using fast corner detection, Intern. Journal of Advanced Research in Electronics and Communication Engineering, vol. 1, no. 6, 2012.
13. Mahesh, M.V. Subramanyam. Feature Based Image Mosaic Using Steerable Filters and Harris Corner Detector I.J. Image, Graphics and Signal Processing, 2013, 6, 9-15 DOI: 10.5815/ijigsp.2013.06.02
14. Mahesh, M.V. Subramanyam. Invariant Corner Detection Using Steerable Filters and Harris Algorithm. Signal & Image Processing: An International Journal (SIPIJ) Vol.3, No.5, October 2012.
15. J. Davis, Mosaics of scenes with moving objects, in Proc. CVPR, 1998.
16. R. Szeliski and H. Y. Shum, Creating full view panoramic image mosaic and environment maps, in Proc. SIGGRAPH'97, 1997.
17. E. Bostanci, Is hamming distance only way for matching binary image feature descriptors? Electronics Letters, vol. 50, no. 11, pp. 806-808, May 2014.
18. Assaf Zomet, Anat Levin, Shmuel Peleg, and Yair Weiss, Seamless image stitching by minimizing false edges, IEEE Transactions on Image Processing, vol. 15, no. 4, Apr. 2006.
19. M. A. Fischler and R. C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, C. of the ACM, Ed. Communications of the ACM, June 1981, vol. 24, no. 6.