

## A Real-Time Image Mosaicing using Single Board Computer

<sup>1</sup>K. Sai Venu Prathap, <sup>2</sup>S.A.K. Jilani and <sup>3</sup>P. Ramana Reddy

<sup>1</sup>Department of ECE, Jawaharlal Nehru Technological University Anantapur (JNTUA),  
Ananthapuramu, India

<sup>2</sup>Department of ECE, Madanapalle Institute of Technology and Science (MITS),  
Madanapalle, Angallu, India

<sup>3</sup>Department of ECE, JNTUA College of Engineering, Ananthapuramu, India

**Abstract:** An image mosaic is a technique of combining partial images that refer to the same scene to obtain a single large image that contains each information of a partial image. In this study, image mosaic is constructed with a single board Intel Compute Stick hardware device. The performance of various algorithms is evaluated in terms of criteria such as execution time, accuracy, Consistency of Corner Numbers (CCN) and a number of detected features with a rotation, horizontal shear, vertical shear transformations and uniform scaling. The image acquisition sensor of single-board hardware captures the remote sensing areas and scenes to construct the mosaic image. The image mosaic is executed in many phases such as feature detection, feature description, geometric computation, image warping and blending. Steerable filters of the 1st derivative is used for an orientation selective. The proposed algorithm is a combination of steerable filter and Features from Accelerated Segment Test (FAST) detector. Experimental results prove that steerable filters with FAST corner detector and Fast Retina Keypoint (FREAK) binary descriptor play vital roles in real-time applications than other detectors and descriptors.

**Key words:** Mosaic, Intel Compute Stick, FAST-Steerable detector, FREAK descriptor, accuracy, recall, CCN

### INTRODUCTION

Image mosaicing combines multiple images of a scene to form a single set image. The Field-of-View (FOV) of human vision is  $200 \times 135^\circ$  and the FOV of compact cameras is  $50 \times 35^\circ$ , resembles that a single shot of camera never produces a vision of human beings, at this situation mosaic technique is used to increase the FOV of a captured scene (Subramanyam, 2013). Image mosaic has been playing a vital role in various fields such as scene stabilization, video mapping (Gracias *et al.*, 2004) and matting, film making, video compression, panorama creation (Shum and Szeliski, 1998) and etc. The majority of researchers proposed many criteria for performance evaluation of feature detectors that have many shortcomings. The mosaic process needs a specialized hardware of more expensive. In this study, image mosaic is constructed with a general rapid prototype Intel Compute Stick. This is a pen drive size device with inbuilt windows operating system, easy to carry and capture the remote sensing images.

A real-time image mosaicking is executed in 6 phases, first phase, initiates with the capturing of partial images of a scene. The second phase, detects the invariant feature

points (Mallick, 2002). Feature extraction is the third phase, SIFT and SURF are the traditional descriptors have efficient feature detection with a more computation time. Still, several issues have remained unsolved how to effectively choose the perfect pairs of an image patch? How to match them? Curiously such issues are solved with a binary descriptor inspired by a human visual system and exact with retina named it as the FREAK but it is only a descriptor not a detector, so, FREAK needs one more detector to identify the corner points. The proposed FAST-steerable detector detects invariant feature points of each input image in selective orientation, FREAK descriptor extracts all the feature vectors of each corner point in an image. Binary descriptors performance is best suited for the real-time applications. Evaluation of binary descriptors has become more interesting in many computer vision applications (Prathap *et al.*, 2016a, b) due to its high efficiency and low memory cost. FREAK encodes all intensity comparisons of an image on a particular pattern at small scale and large scale and gathers information similar to the human visual system.

The fourth phase is the similarity measurement, the similarity between feature points of distinct images is measured with the Euclidean distance in the traditional

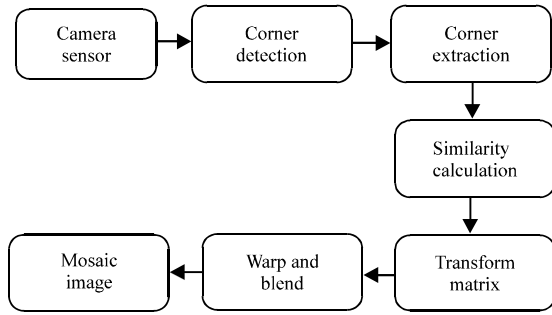


Fig. 1: Image mosaic process

Table 1: Specifications of Intel Compute Stick

Parameters	Specification
Operating system	Windows 10(32 bit)
System memory	2GB inbuilt RAM, DDR3L, 1333 MHz
Flash storage	32 GB microSD
Processor	Intel atom process Z3735F
Graphics	Intel HD graphics
Audio	Intel Audio via. HDMI
Dimensions	103.4*37.6*12.5 mm
Power supply	5V AC-DC power adaptor
Cost	Rs. 8999/-

descriptors. Hamming distance swapped the Euclidean distance in the binary descriptors. Geometric computation transformation is the fifth phase which maps and aligns the feature points of distinct images. Image warping and blending is the last phase, warping corrects the dimension distortions of the image in order to align properly and blending decreases the seams or artifacts of an image. Seams or artifacts degrade the vicinity of a mosaic image.

Our experimental results show that the combination of FAST-steerable detector and FREAK descriptor along with hamming distance computation is well suited for real-time image mosaicing applications. Image registration plays a vital role in image mosaicing. Image registration is a technique of aligning and mapping of pixels of distinct images (Subramanyam, 2012). Figure 1 demonstrate an outline of our framework, starting with camera sensor used to capture images of a scene in real-time. In this study, FAST-steerable corner detector detects robust invariant feature points in a selected orientation and these points are focused on the color image which is known as corner points. An automatic real-time image mosaicing is constructed with a rapid prototype Intel Compute Stick[].

Mosaicing frames of smooth surface form a good vicinity of the mosaic image (Chen, 1995). This device has inbuilt windows operating system and constructs a mosaic image of remote sensing images. The specifications of the device is shown in Fig. 2 and given in Table 1.

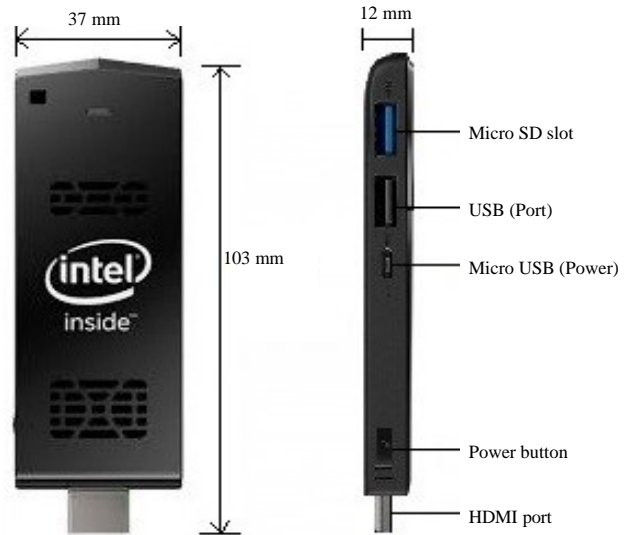


Fig. 2: Intel Compute Stick

## MATERIALS AND METHODS

### Feature detectors

**Harris corner detector:** Harris and Stephens (1988) corner detector is based on the Moravec algorithm (Harris and Stephens, 1988). Corner points have a high intensity than surrounding pixels. It is a window based algorithm. The movement of a window from the corner point in any direction tends to high intensity change. The relation between window, intensity of present pixel and new pixel is given in Eq. 1:

$$E(s, t) = \sum_{u, v} w(u, v) [I(u+s, v+t) - I(u, v)] \quad (1)$$

$$E(s, t) \cong (s, t) M \begin{bmatrix} S \\ t \end{bmatrix} \quad (2)$$

where,  $\det(M) = \lambda_1, \lambda_2$  and  $\text{trace}(M) = \lambda_1 + \lambda_2$ .

**SUSAN corner detector:** Harris corner detector has the worst performance with noisy images (Smith and Brady, 1997). The drawbacks of Harris algorithm were overcome by SUSAN detector. SUSAN synonym stands for “Smallest Univalued Segment Assimilating Nucleus”. No derivatives are utilized find corner points, it works with a mask of center pixel as nucleus:

$$c(\vec{m}) = e^{-\left(\frac{|I(\vec{m}) - I(\vec{m}_0)|}{t}\right)} \quad (3)$$

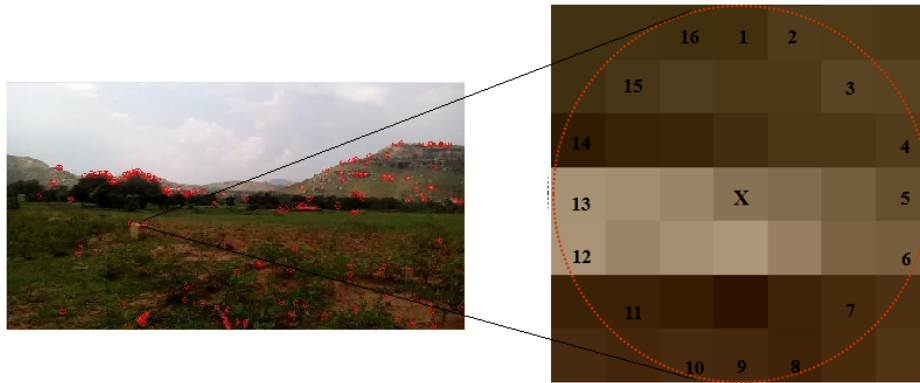


Fig. 3: Pixel X under test surrounded by 16 neighborhood pixels

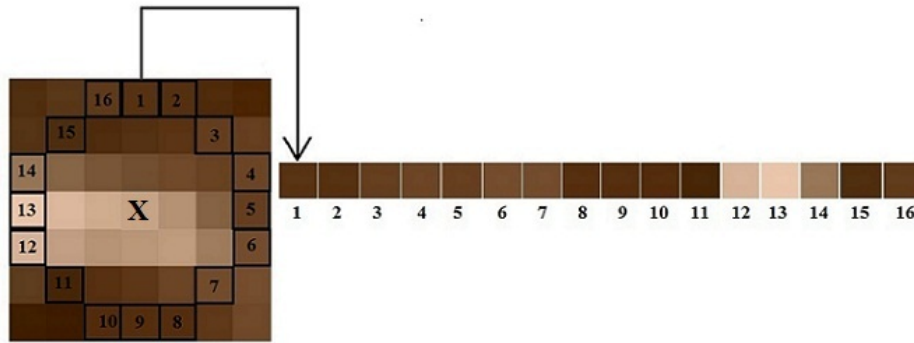


Fig. 4: Pixel X surrounded with 16 pixels stored in vector form

**Features from Accelerated Segment Test (FAST) feature detector:** FAST was introduced by Rosten and Drummond (2006) to identify corner points of an image (Rosten and Drummond, 2006). Corner point detection plays a vital role in various computer vision Gracias *et al.* (2004) applications like object recognition, image matching, tracking, etc. Corner points in an image have a defined position, high local information, robustly detected and they should be ideally detected in all images. The idea behind to develop FAST corner detector was to use interest points in real-time applications like SLAM on a mobile robot.

FAST algorithm validate each pixel of an image in such a way that a pixel is a corner point or not by comparing each pixel with its 16 neighbor pixels that form a circle (Bresenham circle). For every pixel X, store surrounding 16 neighbor pixels as a vector as shown in Fig. 3 and 4. Each one of the 16 pixels should be any one of three states. Darker than intensity of pixel X ( $I_x$ ), similar to, intensity of pixel X or brighter than the intensity of pixel X as shown in Fig. 3 and 4 and mathematical representation is shown in Eq. 4 as:

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

Here,  $S_x$  is the state of pixel X,  $I_x$  is the intensity of pixel X and  $t$  is the threshold value Fig. 4. Representing, ring as a ternary vector and segment test used to classify the vectors. Non-maxima pixel points are suppressed in two ways, initially differentiating pixel intensity values with the center pixel intensity value. If the difference between this two pixel values is greater than threshold value 't' similarly, if the difference between this two pixel values is less than threshold value 't' then the difference of center pixel value and pixel intensity values are considered and entire process can be summarized with mathematical Eq. 5 as:

$$\text{Non-max suppression} = \max \begin{cases} \sum (PV - X) \text{ if } (value - X) > t \\ \sum (X - PV) \text{ if } (value - X) > t \end{cases} \quad (5)$$

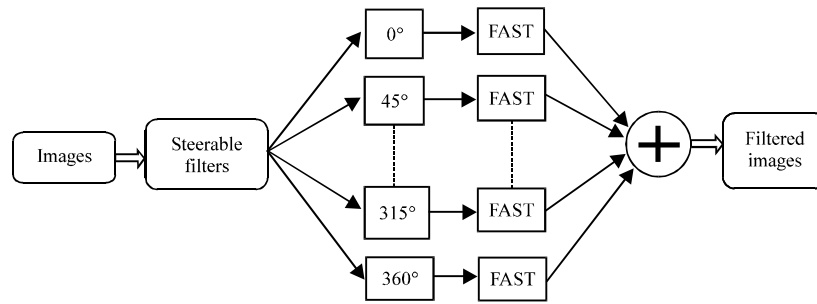


Fig. 5: FAST-steerable corner detector

Where:

X = The center pixel and t is the threshold value for detection

PV = The Pixel Value of N contiguous pixels in the 16 neighborhood pixels

The process can be accelerated by comparing the intensity of pixels  $I_{11}$ ,  $I_5$ ,  $I_9$  and  $I_{13}$  with pixel X intensity  $I_x$ . At least three of four pixels should satisfy threshold criteria that is three pixels should greater or lesser than  $I_x+t$  or  $I_x-t$ , than only corner point exist otherwise, it is not a corner point. Repeat the process for all image pixels.

**Proposed FAST-steerable corner detector:** The proposed algorithm is a combination of steerable filters with the FAST corner detector and it is named as FAST-steerable corner Detector algorithm (FSD). It detects corner points of an image in multiple orientations (Subramanyam, 2013) but to construct a real-time image mosaic only one orientation is taken into account. This is an efficient corner detector with efficient computation time. Figure 5 depicts the combination of steerable filter and FAST detector at multiple orientations. The steerable corner detection for any image at a specified angle is given in Eq. 6:

$$f^\theta(x, y) = \sum_{m=1}^M k_m(\theta) f(x, y) \quad (6)$$

Where:

$f^\theta(x, y)$  = Rotated version of  $f(x, y)$  by an arbitrary angle of  $\theta$

$k_m(\theta)$  = The interpolation functions

M = The number of basis functions

## RESULTS AND DISCUSSION

### Feature descriptors

**FREAK; Fast Retina Keypoint:** SIFT Prathap *et al.* (2016a, b) and Mallick (2002) and SURF (Prathap *et al.*, 2016a, b) are traditional descriptors with efficient corner point detection rate but due to the lack of computation

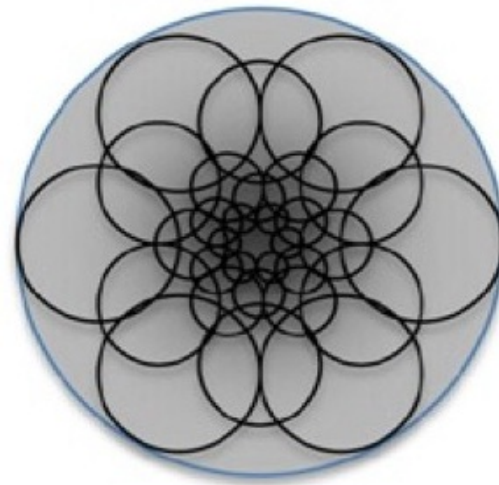


Fig. 6: FREAK sampling pattern inspired from human retina

time, they are far away from the real-time applications. This study proposes FREAK descriptor (Alahi *et al.*, 2012) introduced by Alahi *et al.* (2012). It is a unique corner point descriptor propelled by the human visual system and exactly the retina, it is a biologically inspired descriptor. A binary string is computed effectively by comparing the intensities of an image over a sampling pattern. A genuine challenge is to identify the corner points with stable, compact and robust representations invariant to noise, rotation, scale and affine transformations (Subramanyam, 2012). FREAK is a compact and real-time descriptor. FREAK descriptor estimates forty-three weighted Gaussian image patches around the interesting point, however, Gaussian patterns are inspired by a retinal pattern. Near key point the density of points is high as shown in Fig. 6, each circle symbolizes a receptive field and smoothen with the specific Gaussian kernel. Experimental outcomes proved that the FREAK descriptor is more rapid in computational time with lower memory load. The descriptors of each corner point form a binary string F:

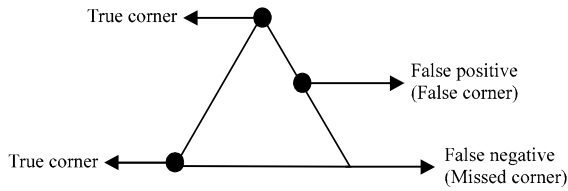


Fig. 7: Classification of the true corner point

$$F = \sum_{0 \leq a < N} 2^a T(P_a) \quad (7)$$

Where:

$P_a$  = Pair of receptive fields

$N$  = Desired descriptor size

$$T(P_a) = \begin{cases} 1 & \text{if } (I(P_a^{r1}) - I(P_a^{r2})) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Here,  $I(P_a^{r1})$  and  $(P_a^{r2})$  are the smoothed receptive fields. Hamming distance calculates the similarity metric when features of input images are binary objects (Bostanci, 2014). The binary descriptors of two feature points  $P$  and  $Q$  is  $B(P)$  and  $B(Q)$  than the hamming distance is:

$$\text{Ham}(P, Q) = \sum_{i=1}^n B_i(P) \phi B_i(Q) \quad (8)$$

The hamming distance performance time is known as matching cost that compares two binary vector strings with Boolean XOR operation to deliver match metric. Hamming distance is desired for matching binary descriptors such as BRISK, BRIEF (Calonder *et al.*, 2010), LBP (Ojala and Pietikainen, 1999), FREAK and ORB (Rublee *et al.*, 2011).

**Performance evaluation:** There are several feature detector and descriptor algorithms, to know the best real-time performer is a challenging task. Various standard images are tested to evaluate the best performance of feature detectors. Performance evaluation of Harris, SUSAN, FAST, proposed FAST-Steerable is compared with respect to multiple criteria for the synthetic image in Fig. 7 and 8. Execution time, a number of features detected, recall, accuracy and Consistency of Corner Number-CCN are enough to evaluate the performance of detectors and descriptors. As the accuracy, recall and CCN are near to 100% refers their robustness (Mokhtarian and Mohanna, 2006). The true corner classification is shown in Fig. 7. Table 2 refers to true corners, false positive and false negative parameters.

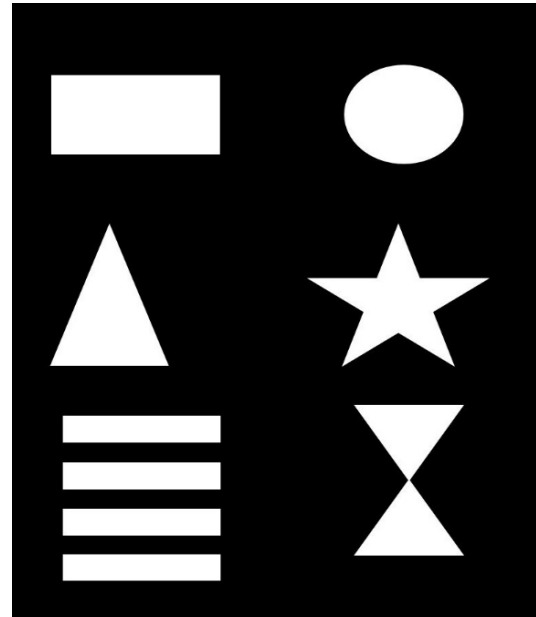


Fig. 8: Synthetic image

Table 2: Classification of true corners

Algorithms	Ground truth corners	Detected corners	True corners	False positive	False negative
Harris	39	55	33	22	6
SUSAN	39	64	19	45	20
FAST	39	33	23	10	10
Proposed	39	39	34	5	5

Accuracy is governed by ground truth measurements values. Ground truth values are of majority human judgment values:

$$\text{Accuracy} = 100 * \left( \frac{\frac{N_T}{N_o} - \frac{N_T}{N_g}}{2} \right) \quad (9)$$

Where:

$N_T$  = Number of true corners

$N_o$  = Total Number of corners detected

$N_g$  = Total Number of ground truth corners

Recall is defined as the ratio of true corners and the sum of true corners and false positive:

$$\text{Recall} = \frac{\text{No. of true corners}}{\text{No. of true corners} + \text{No. of false positive}} \quad (10)$$

The Consistency of Corner Numbers (CCN) denotes the number of corners that are invariant to the combination of rotation, shearing, uniform scaling or non-uniform scaling:

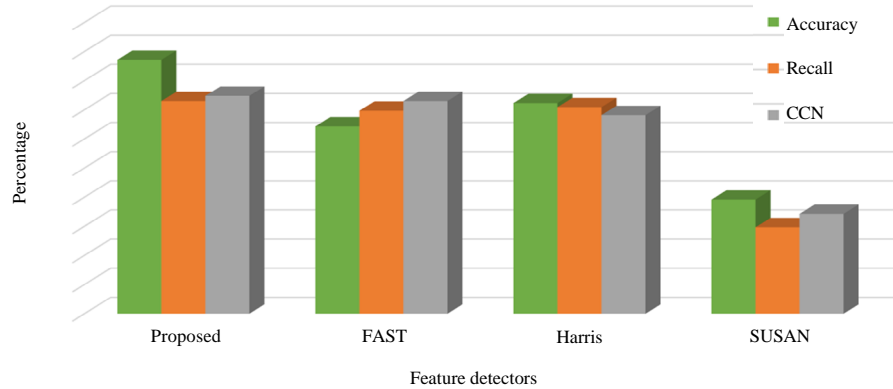


Fig. 9: Comparison of proposed method with other methods interms of accuracy, recall and CCN

Table 3: Accuracy, recall, CCN of synthetic image

Algorithm	Accuracy (%)	Recall (%)	CCN (%)
Proposed	87.17	73.00	74.80
FAST	64.33	69.69	73.01
Harris	72.33	70.90	68.17
SUSAN	39.20	29.68	34.28

Table 4: Computation time of comer detection with stick and laptop

Algorithm	Execution time (sec)	
	Intel compute stick	Laptop
SUSAN	4.8661	2.5946
Harris	0.5145	0.3299
FAST	0.2831	0.1028
Proposed	0.3819	0.2665

$$CCN = 100 * 1.1^{-|N_t - N_o|} \quad (11)$$

Where:

$N_t$  = The Number of comer points of transformed image

$N_o$  = The Number of corner points of the original image

The proposed FAST-steerable detector is 4.04 times faster than Harris detector, 190 times faster than SUSAN algorithm as shown in Table 4. Though FAST detector is computationally faster than FAST-steerable detector due to poor accuracy robustness is compromised. Similarly, FREAK descriptor is 11.17 times faster than SURF descriptor and 30.63 times faster than SIFT descriptor as depicted in Table 2 and 3. Therefore, the combination of the FAST-steerable.

The proposed methods has highest accuracy, recall and CCN than the FAST, Harris and SUSAN methods. The proposed method has 1.355 times more accuracy than the FAST detector, 1.205 times more accuracy than the Harris detector and 2.22 times more accuracy than the SUSAN detector. Similarly, the proposed detector has best recall and CCN criterias than the other methods. It is shown in Table 3 and illustrated in Fig. 9.

Table 4 shows the corner detection time of each algorithm with both the intel compute stick and laptop. Among all detectors FAST has quick execution time but

Table 5: FAST detector frame mosaicing time of a video (proposed algorithm)

Total frames of video	Frame frequency	Total frames used for mosaicing	Mosaicing time (sec)	
			Intel Compute Stick	Laptop
627	1(all)	627	179.200	104.500
627	2	313	82.810	61.520
627	5	125	48.930	24.400
627	10	62	28.910	12.810
627	20	31	13.030	8.569
627	30	20	8.101	5.236

Table 6: FAST detector frame mosaicing time of a video

Total frames of video	Frame frequency	Total frames used for mosaicing	Mosaicing time (sec)	
			Intel Compute Stick	Laptop
627	1(all)	627	194.60	122.600
627	2	313	133.90	65.440
627	5	125	51.28	26.250
627	10	62	27.69	15.130
627	20	31	18.24	11.410
627	30	20	13.47	9.253

it has poor accuracy, recall and CCN parameters. The proposed method is 1.34 times faster than the Harris and 12.77 times faster than the SUSAN with the Intel Compute Stick. The proposed method is 1.23 times faster than the Harris and 9.96 times faster than the SUSAN with the laptop. Table 4 conclude that the proposed method has quick computation time than the than the Harris and SUSAN methods and Table 3 indicates that the proposed methods is robust than the FAST, Harris and SUSAN algorithms.

Therefore, the frames of a video is mosaiced with both the proposed method and FAST detector. Table 5 and 6 illustrates the frame mosaic time of a video constructed by both intel compute stick and laptop with FAST detector and proposed algorithm, respectively. Initially mosaic image is obtained with the all frames of a video with frame frequency of 1. As the frame frequency increases from 1-30, the number of frames and execution



Fig. 10: Mosaic image constructed with only 20 frames out of 627 frames of a video with Intel Compute Stick

time for mosaicing decreases. Figure 10 is a resultant of aerial view mosaic image constructed with only 20 frames out of 627 frames of video.

### CONCLUSION

This study proposes an experimental real-time image mosaic using the proposed FAST-steerable corner detector executed by the Intel Compute Stick. The obtain result are compared with the laptop. The low weight and area of stick has high mobility and easy to utilize in any real-time remote sensing areas like drone footages. The Table 2 indicated the robustness of a proposed algorithm in terms of true corners classification. Though the FAST detector has quick computation time but lack in terms of accuracy, recall and CCP criteria's. Experimental results prove that the combination of FAST-steerable detector and FREAK descriptor is suitable for real-time applications than Harris, SUSAN, FAST detectors. The proposed FAST-steerable detector accuracy is close to 90%, CCN of average 74.80% and recall of 73.00% with efficient execution time. The real-time corner detection and mosaic image of a video can be constructed with the proposed method and is best suited for the real-time applications to increase the Filed-of-View. The whole work shows that real-time image mosaic is robust and well executed with the combination of FAST-steerable, FREAK algorithms by the high mobility intel compute stick.

### REFERENCES

Alahi, A., R. Ortiz and P. Vandergheynst, 2012. FREAK: Fast retina keypoint. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR'12), June 16-21, 2012, IEEE, Providence, Rhode Island, USA., ISBN:978-1-4673-1226-4, pp: 510-517.

Bostanci, E., 2014. Is hamming distance only way for matching binary image feature descriptors?. *Electron. Lett.*, 50: 806-808.

Calonder, M., V. Lepetit, C. Strecha and P. Fua, 2010. Brief: Binary Robust Independent Elementary Features. In: *Computer Vision-ECCV*, Daniilidis, K., P. Maragos and N. Paragios (Eds.). Springer, Berlin, Germany, ISBN: 978-3-642-15560-4, pp: 778-792.

Chen, S.E., 1995. Quicktime VR: An image-based approach to virtual environment navigation. Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, August 06-11, 1995, ACM, New York, USA., pp: 29-38.

Gracias, N., J.P. Costeira and J.S. Victor, 2004. Linear global mosaics for underwater surveying. Proceedings of the 5th IFAC-EURON Symposium on Intelligent Autonomous Vehicles Vol. 1, July 5-7, 2004, Instituto Superior Tecnico, Lisboa, Portugal, pp: 1-6.

Harris, C. and M. Stephens, 1988. A combined corner and edge detector. Proceedings of the 4th Alvey Vision Conference, Volume 15, August 31-September 2, 1988, Manchester, UK., pp: 147-151.

Mallick, S.P., 2002. Feature based image mosaicking. Master Thesis, University of California, Oakland, California.

Mokhtarian, F. and F. Mohanna, 2006. Performance evaluation of corner detectors using consistency and accuracy measures. *Comput. Vision Image Understanding*, 102: 81-94.

Ojala, T. and M. Pietikainen, 1999. Unsupervised texture segmentation using feature distributions. *Pattern Recogn.*, 32: 477-486.

- Prathap, K.S.V., S.A.K. Jilani and P.R. Reddy, 2016a. A critical review on image Mosaicing. Proceedings of the International Conference on Computer Communication and Informatics (ICCCI'16), January 7-9, 2016, IEEE, Coimbatore, India, ISBN: 978-1-4673-6679-3, pp: 1-8.
- Prathap, K.S.V., S.A.K. Jilani and P.R. Reddy, 2016b. A real-time image mosaicing using scale invariant feature transform. Indian J. Sci. Technol., Vol. 9, 10.17485/ijst/2016/v9i12/88175
- Rosten, E. and T. Drummond, 2006. Machine Learning for High-Speed Corner Detection. In: Computer Vision-ECCV, Leonardis, A., H. Bischof and A. Pinz (Eds.). Springer, Berlin, Germany, ISBN: 978-3-540-33832-1, pp: 430-443.
- Rublee, E., V. Rabaud, K. Konolige and G. Bradski, 2011. ORB: An efficient alternative to SIFT or SURF. IEEE Proceeding of the International Conference on Computer Vision (ICCV), November 6-13, 2011, Barcelona, pp: 2564-2571.
- Shum, H.Y. and R. Szeliski, 1998. Construction and refinement of panoramic mosaics with global and local alignment. Proceedings of the 6th International Conference on Computer Vision, January 7, 1998, IEEE, Bombay, India, pp: 953-956.
- Smith, S.M. and J.M. Brady, 1997. SUSAN-A new approach to low level image processing. Int. J. Comput. Vision, 23: 45-78.
- Subramanyam, M.V., 2012. Automatic feature based image registration using SIFT algorithm. Proceedings of the 3rd International Conference on Computing Communication and Networking Technologies (ICCCNT'12), July 26-28, 2012, IEEE, Coimbatore, India, pp: 1-5.
- Subramanyam, M.V., 2013. Feature based image mosaic using steerable filters and Harris corner detector. Intl. J. Image Graphics Signal Process., 5: 9-15.