2019 SAUPEC/RobMech/PRASA Conference
Bloemfontein, South Africa, January 28-30, 2019

CREAK descriptor evaluation for visual odometry

Dehan Smulders*[1], Kenneth Uren*[2], George van Schoor*[3], Corné van Daalen**[4] and Japie Engelbrecht**[5]
*School of Computer and Electronic Engineering, North-West University, Potchefstroom, South Africa
**School of Electrical, Electronic and Computer Engineering, Stellenbosch University, Stellenbosch, South Africa
[1] dehansmulders@gmail.com, [2] Kenny.Uren@nwu.ac.za, [3] George.vanSchoor@nwu.ac.za, [4] cvdaalen@sun.ac.za, [5] jengelbr@sun.ac.za

***Abstract*—Each year, latest state of the art technologies and algorithms arise that claim and prove to out-shine their predecessors. One such algorithm is the Colour-based Retina Key-point (CREAK) descriptor, which is based on the FAST Retina Key-point (FREAK) descriptor with the included functionality of considering colour information in its Key-point description. This paper explores the implementation of CREAK in a "real-time" visual odometry application by means of a comparative study of the more well-known FREAK algorithm. Although FREAK achieved more accurate odometry when key-points were abundant, this proved to be too computationally expensive. CREAK on the other hand outperformed FREAK when key-points are scarce due to its lack of false-positive matches.**

Keywords—FREAK, CREAK, visual odometry, feature descriptor, evaluation

I. INTRODUCTION

In the modern age of automation, odometry is the prevalent field of interest. Finding a computationally efficient method to determine odometry that is accurate as well as cost effective is no simple task. Visual odometry is the method of determining location by using visual image input. In most cases, visual odometry is required by unmanned robotic vehicles to determine their current location, and therefore visual odometry is required to be in real-time. We define "real-time" as a rate no slower than 10 frames per second (FPS), since according to the KITTI Vision Benchmark Suite [1] on which the odometry is performed, the camera setup is set to capture an image every 100 ms or at a rate of 10 FPS.

In visual odometry, the input image needs to be processed to determine detected key-points. These key-points then need to be described and then matched to the same key-points in a later point in time. Thereafter, the visual odometry is determined. This paper will focus on the description aspect of the visual odometry problem. The Fast Retina Key-point (FREAK) descriptor was first introduced by Alahi *et al.* in [2], which is inspired by the human retina. This descriptor is a binary descriptor, meaning that the hamming distance can be calculated to determine matches, which is highly efficient computationally.

The FREAK descriptor is highly respected in the image processing community due to its high descriptiveness and low computational complexity. More recently, Chen *et al.* [3] proposed the Colour-based Retina Key-point (CREAK) descriptor, that, similar to FREAK, is modelled after the human retina. However, CREAK takes the colour-space information into account when computing the descriptor, resulting in higher descriptive power.

The outline of this paper is as follows: The next section will discuss the related work performed in the field of visual odometry and key-point detectors and descriptors, followed by section III describing the experimental design used for the evaluation of the visual odometry by using FREAK and CREAK. In section IV some results will be given and discussed. Finally, the paper will conclude in section V.

II. RELATED WORK

A. Key-point Detectors

A key-point is a point in an image that can be detected and reliably re-detected by key-point detector software. Many detectors have been created throughout the years, each implementing their own variation for determining key-points from an image. The first approach for detectors include corner detection methods, that function by comparing the neighbouring pixel intensities around a pixel of interest to detect 'corners' within the image scene.

Another more complex approach for detectors were designed that makes use of the responses of certain filters to determine possible key-points. The Features from Accelerated Segment Test (FAST) detector [4] is a highly robust and efficient detector that makes use of the former. In FAST, a pixel of interest p is considered within the image. Thereafter, a circle of 16 pixels surrounding the pixel of interested is compared against a threshold intensity value. Should a continuous number N of surrounding pixels with their respective intensities, be within the threshold value, the pixel of interest will be labelled a key-point.

More recently, Rublee *et al*. in [6] proposed the Oriented FAST and Rotated BRIEF (ORB) detector and descriptor. ORB identified that the FAST detector does not provide a measure for cornerness, and after determining FAST key-points, a Harris corner measure in employed as in [7] to place the FAST key-points in order. Thereafter ORB picks the top N key-points. Once key-points have been detected using a detector algorithm such as FAST or ORB, they are not yet useful for the visual odometry application. For them to serve their purpose as landmarks in visual odometry, each key-point needs to be described such as to recognize them when seen in a later point in time. Such algorithms are known as descriptors.

Feature based descriptors can be categorized into two main groups, float-point based and binary descriptors. Float-point based descriptors such as the Scale Invariant Feature Transform (SIFT) [8] and Speeded Up Robust Features (SURF) [9] use a vector of histograms to describe the key-point. Therefore, the Euclidian distance between the key-points need to be calculated in the matching process.

The binary descriptor algorithms such as ORB [6], FREAK [2], and CREAK [3] makes use of a bit-set array as a descriptor. This allows for the hamming distance to be calculated between key-points in the matching process, which is significantly faster to perform.

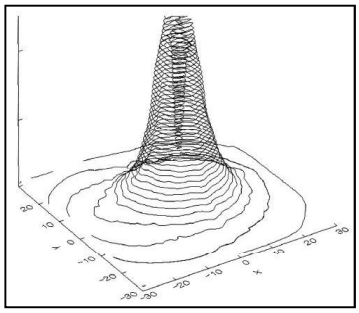

Fig. 1. Density of Ganglion Cells [2]

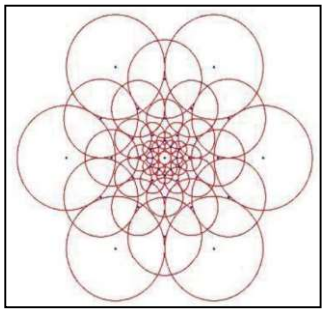

Fig. 2. FREAK Kernel Pattern [2]

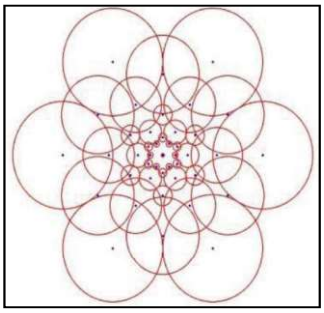

Fig. 3. CREAK Kernel Pattern [3]

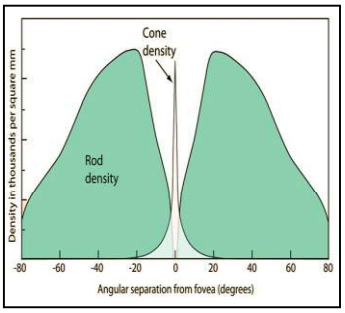

Fig. 4. Density of Cone Cells [5]

B. FREAK Descriptor

FREAK is a binary descriptor modelled after the human retina. The retina comprises of multiple concentric areas known as kernels, that each has a concentration of ganglion cells that gets denser the closer they are to the foveal (Centre of the eye) as shown in Fig. 1. The region where light influences the response of a ganglion cell is the receptive field, which increases in size with radial distance from the foveal. The special distribution of ganglion cells are distributed into four main regions, each detecting exponentially less resolution and image detail proportional to the distance from the foveal. This decrease in resolution can be interpreted as body resource optimisation.

Re-creating the retinal analogy digitally, photoreceptors determine the light intensity of pixels within an image, then the ganglion cells compare the average pixel intensities from specific areas to one another to determine the binary string descriptor. This process is described below:

- To replicate the distribution of ganglion cells, the sampling pattern seen in Fig. 2, is created around the key-point point of interest with each kernel area representing a retinal receptive field.
- Each kernel is smoothed to be less sensitive to noise and the average pixel intensity of each kernel is determined.
- The intensities of each kernel is compared to each other, and bit set depending on the difference in intensities.

Since there are 43 kernels as shown in Fig. 2, leaving a total of 903 possible matching pairs, not all pairs need to be compared as not all convey useful information. Therefore, an approach similar to ORB [6] was taken to learn the best 512 matching pairs for the descriptor from training data. The binary descriptor F is constructed by thresholding the difference between pairs of receptive fields with their corresponding Gaussian kernel. In other words, F is a binary string formed by a sequence of one-bit Difference of Gaussians (DoG):

$$F = \sum_{0 \le a < N} 2^a T(P_a) \tag{1}$$

where P_a is a pair of receptive fields, N is the desired size of the descriptor, and $T(P_a)$ is:

$$T(P_a) = \begin{cases} 1 & \text{if } (I(P_a^{r1}) - I(P_a^{r2}) > 0, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Since 512 matching pairs are considered, the descriptor will be 512 bits (or 64 bytes) in size.

C. CREAK Descriptor

CREAK is also a binary descriptor modelled after the human retina and is built upon the foundations laid by FREAK. Due to the nature of modern binary descriptors such as ORB [6] and FREAK [2], only the grey intensity values of images are considered. This negates a significant amount of descriptive information that can be achieved by considering the blue, red and green colour values. According to Chen *et al.* in [3], by taking the colour-space information into account will significantly aid in creating a descriptor that is more robust and discriminative.

As in FREAK, the CREAK descriptor functions with a deliberate sampling pattern, designed to mimic the functionality of the human retina and the spatial distribution of ganglion cells in receptive fields. However, the CREAK descriptor had made alterations to the design of the sampling pattern, as shown in Fig. 3.

Due to the overlapping that occurs in the innermost kernels, spatial redundancy increases and therefore promotes better discriminative power in the CREAK descriptor. Because colour information is being considered, the density of photoreceptive cone cells (which determine colour information) is higher than rods (which determine light intensity) near the foveal as shown in Fig. 4. The kernels near the centre are then decreased further to preserve colour information and to better resemble the retina model.

The CREAK descriptor algorithm is based on the foundation laid by FREAK, and from the article describing CREAK in [3], the general operational process of CREAK remains the same as in FREAK, however the colour image is split into its three components, to reveal three separate grey images, each representing the colour-spaces for blue, green and red intensities. The average kernel intensities are then determined for each of the three colour-spaces, and the kernel comparisons are performed for each.

The binary descriptor D is then constructed by thresholding the difference between pairs of receptive fields with their corresponding Gaussian kernel, but repeated once for each of the three colour-spaces. In other words, D is a binary string formed by a concatenation of sequences of one-bit Difference of Gaussians (DoG):

$$D = \sum_{i=0}^{\frac{N}{3}-1} [C_i] \tag{3}$$

and C_i is defined as:

$$C_i = 2^i T(B_i) + 2^{i+\frac{N}{3}} T(G_i) + 2^{i+\frac{2N}{3}} T(R_i), \quad (4)$$

where *B, G, R* represent colour channels blue, green, and red, while B_i, G_i, *and* R_i are colour test pairs of receptive fields with their respective corresponding channels. The form of $T(B_i)$, $T(G_i)$ and $T(R_i)$ are each functionally the same as $T(P_a)$ in FREAK, described by (2).

The complete binary descriptor D of size N is formed by concatenating three $N/3$ binary test results with Chen *et al.* in [3] claiming that $N = 192$ is the optimal descriptor size, and enlarging the size yields no apparent improvement. Therefore 64 bits are reserved for each colour-space.

D. Visual Odometry

Complex solutions to visual odometry known as Simultaneous Localization and Mapping (SLAM) [10], exist and are widely used in the robotics and automation community. Some of these SLAM solutions such as [11], [12] use complex algorithms to determine the 3D coordinates of each key-point, then stores the key-point, its 3D relative coordinates, as well as description in a database. Then from recognizing key-points from the database later in time, and from a different perspective, the SLAM algorithm can then calculate the new current position of the vehicle.

Less complex solutions exist that can take several matched key-points from two images taken from different points in time and determine the camera's current change in translation and rotation. This is done by using the Essential matrix discussed in [13]. Once the Essential matrix is known, R, t and the camera matrices can be recovered from it, where R is the rotation and t is the translation matrices. The accuracy of the rotation and translation values are solely dependent on the quality of matched key-points. Therefore, if the pixel locations of matched key-points are without error, the resulting odometry obtained will be without error as well.

The Essential matrix approach only takes two image frames into consideration. Therefore, no knowledge of prior detected key-points exist. This also implies that if an error occurs at any point in time, the error will be taken forward without any hope of restoration, unless loop closure methods are introduced. Nistér in [13] proposed that the Random Sampling Consensus (RANSAC) outlier detector be used when computing the Essential matrix.

RANSAC is an iterative algorithm that randomly samples five points from the set of matched key-points, determines the Essential matrix, then checks if all other points are inliers with the model. After a fixed number of iterations, the Essential matrix with which the maximum number of matched key-points agree, is used.

III. Experimental Design

For this study, the CREAK descriptor will be evaluated against the FREAK descriptor and will be compared specifically in a visual odometry implementation. The Microsoft Visual Studio 2017 environment will be used to perform all necessary tasks in C++. Chen in [3] claimed that using the ORB detector for both FREAK as well as CREAK yielded unbiased results, thus the ORB detector is used.

Before the descriptors are implemented in visual odometry, they first need to be re-created from their respective articles. The OpenCV library includes the FREAK descriptor, thus it can be used to verify the re-created FREAK algorithm. Although the OpenCV library includes a FREAK function, it is necessary to re-create the FREAK descriptor algorithm because the CREAK descriptor that is to be created, will be built upon the foundation laid by FREAK. Therefore, by re-creating the FREAK descriptor, the performance of CREAK can be more accurately determined, as any bias from lack of optimization can be negated.

The operational flow of the odometry evaluation application as is shown in Fig. 5:

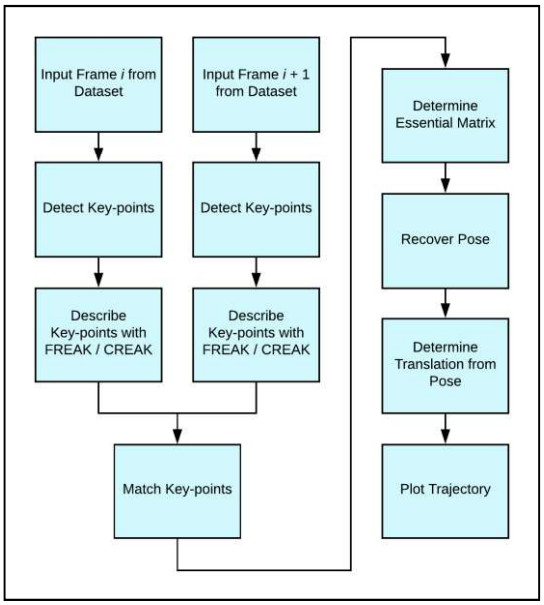

Fig. 5. Operational Flow for Odometry Evaluation Application

As described in Fig 5, the application iterates through all the frames from the KITTI Vision Benchmark Suite [1] dataset, considering two sequential frames at a time. Key-points are extracted for each individual frame, and then described using the FREAK or CREAK descriptor as desired. The described key-points are then matched to identify correlating key-points on each frame, and then used to determine the Essential matrix. From this, the pose is recovered, and the translation determined. The results are then plotted as the trajectory.

IV. Results

To compare the descriptor results, the images had been taken from the University of Oxford Visual Geometry Dataset [14]. Each dataset consists of a scene, and five variations thereof. Each scene has a different type of transformation occurring, such as viewpoint change, rotation, zoom, blur and luminance. Fig. 9 to Fig. 16 show the most difficult, successful result of each dataset. TABLE I shows the results of FREAK and CREAK on the datasets. The time in *ms* for the descriptor to perform description as well as the number of matches achieved (in some cases where

applicable, incorrect matches are indicated with a forward slash in the form *#matches/#correct_matches*).

The compared image scenes from the dataset take the first scene as reference and iterate through the varying difficulties (the sixth scene will always have the most extreme transformation and will thus be most difficult) and is shown as *reference_image|compared_image*.

TABLE I. FREAK VS CREAK MATCHING DATASETS

	FREAK		CREAK	
	Time(ms)	Matches	Time(ms)	Matches
Graffiti (Viewpoint)				
1|2	4	150	10	127
1|3	5	55	11	49
1|4	4	16	10	8
1|5	4	4/0	11	8/7
1|6	4	5/0	9	0
Bark (Zoom/Rotation)				
1|2	3	18	8	17
1|3	2	2	8	2
1|4	3	5/2	6	2
1|5	3	0	7	0
1|6	3	0	6	0
Bikes (Blur)				
1|2	7	285	15	260
1|3	6	250	15	219
1|4	7	197	15	177
1|5	6	167/166	14	135
1|6	6	111/110	12	90
Leuven (Luminance)				
1|2	4	82	7	74
1|3	3	53	7	51
1|4	3	46/45	7	39/38
1|5	4	37	6	33
1|6	3	32	8	29

As shown by TABLE I, the FREAK descriptor was significantly faster than CREAK, however this was expected as Chen *et al.* in [3] also found that the description times were almost twice that of FREAK.

Comparing the matching results of the FREAK and CREAK descriptors in TABLE I, we can see that the results are very similar. FREAK does have slightly more matches in every scenario, however CREAK appears to be more consistent, achieving less false positive matches. In two cases, FREAK claimed to have 4 and 5 matches, of which all were incorrect. This is detrimental in a visual odometry implementation, as any false positive match can have huge effects on the odometry error. This was put to test in a visual odometry implementation, where a crude and minimalistic visual odometry algorithm was set up.

The visual material was downloaded from the KITTI Vision Benchmark Suite [1] and a simple C++ application

Fig. 6. Detected and Tracked Key-points using the CREAK descriptor

was created to display the results. Fig. 6 displays the application's visual feed as input, with all matched key-points drawn over the image. The cyan lines are connecting lines that match key-points to their location in the previous frame. Once again, the ORB detector was used in three of the experiments that were performed. The ORB detector was set to generate 500, 1000 and 2000 key-points respectively in each experiment. The odometry results obtained for 2000 frames can be found in Fig. 7, where the cyan path represents the ground truth, the magenta representing FREAK and the red representing CREAK.

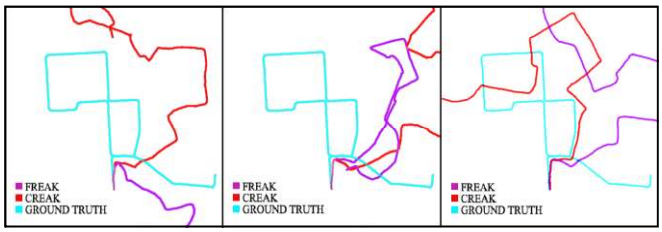

Fig. 7. Odometry with ORB detecting 500, 1000 and 2000 Key-points

Thereafter, the FAST detector was used that generated roughly 5000 key-points per frame. This took significantly longer in processing time, as shown in TABLE II. The results are displayed in Fig. 8, again with cyan representing the ground truth, magenta representing FREAK and red representing CREAK.

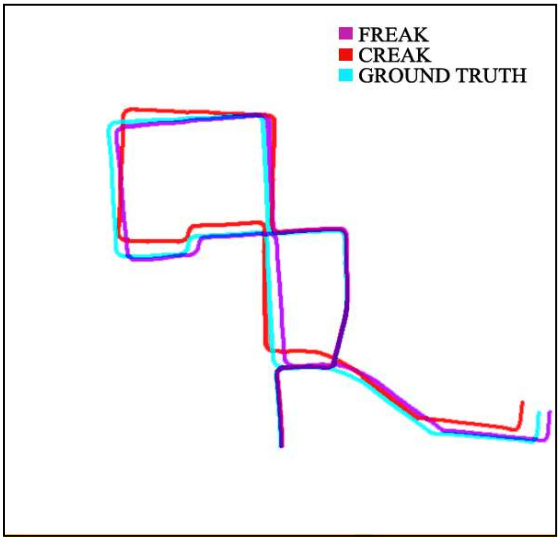

Fig. 8. Odometry with FAST detecting roughly 5000 Key-points

Fig. 9. Graffiti 1|5 Viewpoint FREAK results (4 Matches, 0 Correct)

Fig. 11. Graffiti 1|5 Viewpoint CREAK results (8 Matches, 7 Correct)

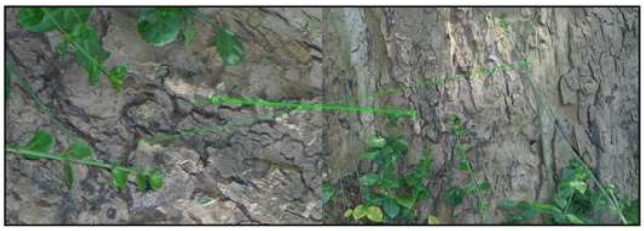

Fig. 10. Bark 1|4 Rotation/Zoom FREAK results (5 Matches, 2 Correct)

Fig. 12. Bark 1|4 Rotation/Zoom CREAK results (2 Matches, 2 Correct)

Fig. 13. Bikes 1|6 Blur FREAK results (167 Matches, 166 Correct)

Fig. 15. Bikes 1|6 Blur CREAK results (135 Matches, 135 Correct)

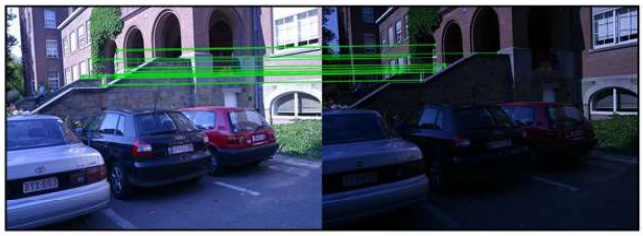

Fig. 14. Leuven 1|6 Luminance FREAK results (32 Matches, 32 Correct)

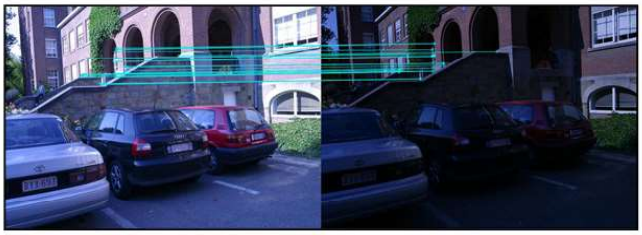

Fig. 16. Leuven 1|6 Luminance CREAK results (29 Matches, 29 Correct)

From Fig. 7 we can conclude that even though the visual odometry algorithm itself is nowhere near ideal, for the purposes of evaluating the CREAK descriptor in a visual odometry environment, it is sufficient. In each case where the ORB detector had been used, CREAK is clearly superior to FREAK in terms of descriptive power. The odometry path created by using CREAK bears closer resemblance to the ground truth, compared to FREAK in each case of 500, 1000 and 2000 key-points. Performance for more than 2000 key-points were not explored, because as TABLE II shows, more key-points would take longer to compute, and therefore would not be classified as "real-time" performance.

In Fig. 8, where the FAST detector had been used, FREAK does in fact yield better results when compared to the results of CREAK, however this is theorized to be due to the fact that using the FREAK descriptor always results in a higher number of matches, although not always correct matches, as shown in TABLE I. Due to the sheer volume of key-points generated by the FAST detector, and the 5-point relative pose method of [13] in the Essential matrix, there were enough correct matches to nullify the incorrect matches, and therefore FREAK allowed the Essential matrix to more accurately model the odometry.

TABLE II. TIME TO PROCESS 2000 FRAMES

	FREAK		CREAK	
Detector	Time(s)	FPS	Time(s)	FPS
ORB 500	121.61	16.44	143.15	13.97
ORB 1000	135.66	14.74	158.26	12.64
ORB 2000	157.36	12.70	194.06	10.31
FAST	410.31	4.87	583.58	3.43

V. CONCLUSION

The CREAK descriptor had been implemented in a crude visual odometry algorithm, and results compared to the more well-known FREAK descriptor. Results showed that FREAK was more accurate in terms of odometry when ample key-points were supplied. However, the computation time was much slower than 10 FPS, and could thus not classify as "real-time". In a more realistic application, where less key-points are supplied, CREAK outperforms FREAK by a notable margin, therefore we can conclude that in a visual odometry environment specifically, CREAK is not only

viable, but superior to FREAK as a rival binary descriptor, as it demonstrated more reliable descriptive power, especially in a "real-time" application.

Although not pursued in this study, CREAK is theorised to be advantageous in a SLAM environment, as its smaller bit size requires less storage when a large database of descriptors had been acquired, therefore future work will evaluate the CREAK descriptor in a "real-time" SLAM application.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of South Africa under the following grant number: TP2011073100016. Note, however, that the grant holder acknowledges that the opinions, findings, conclusions and recommendations expressed in any publication generated by the NRF supported research project are that of the author. The NRF accepts no liability in this regard.

REFERENCES

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 510–517, 2012.

[3] Y. Chen, C. Chan, and W. Tsai, "CREAK : Color-based Retina Keypoint Descriptor," *Proc. Int. Conf. Image Process. Comput. Vision, Pattern Recognit.*, pp. 252–258, 2016.

[4] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 430–443, 2006.

[5] R. Nave, "HyperPhysics Light and Vision." [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/vision/rodcone.html#c3. [Accessed: 20-Oct-2018].

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011.

[7] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Procedings Alvey Vis. Conf. 1988*, pp. 147–151, 1988.

[8] G. Lowe, "SIFT - The Scale Invariant Feature Transform," *Int. J.*, vol. 2, pp. 91–110, 2004.

[9] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.

[10] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): part I The Essential Algorithms," *Robot. Autom. Mag.*, vol. 2, pp. 99–110, 2006.

[11] D. J. Mankowitz and E. Rivlin, "CFORB: Circular FREAK-ORB Visual Odometry," 2015.

[12] N. Karlsson, E. Di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2005, vol. 2005, pp. 24–29.

[13] D. Nistér, "An Efficient Solution to the Five-Point Relative Pose Problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, 2004.

[14] "University of OXFORD: Visual Geometry Group," 2004. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/data-aff.html.*