Viability of FREAK for Panoramic Image Construction

Matthew Dans, Joshua Han, Asad Jamil
University of Toronto
Toronto, Canada

ABSTRACT

An integral part of any panoramic image construction (PIC) pipeline is determining and matching keypoints between images in order to seamlessly stitch them together. Currently, a number of robust algorithms are regularly used in PIC for key point detection and matching, including Scale Invariant Feature Transform (SIFT) [7] and Speed-up Robust Feature (SURF) [3] and Binary Robust Invariant Scalable Keypoints (BRISK) [6]. However, there is a continual effort in the field of keypoint descriptors to develop more efficient algorithms for keypoint matching that are faster and can run on devices with lower computational power such as smartphones or drones. This search for more efficient algorithms led to the Fast Retina Key-point (FREAK) algorithm [1], modeled after the human retina, which provides time and memory improvements over previous algorithms in the use case of object detection.

Based on FREAKs promising results when used for object detection, we investigated the use of FREAK descriptors in Panoramic Image construction. With significant improvements on feature descriptor generation and matching time and memory requirements, while maintaining accuracy, the drawback of FREAK is the substantial overhead required to determine the optimal sampling pattern prior to descriptor generation. We found that further optimization to the sampling pattern generation algorithm is required for FREAK to be a viable part of a PIC pipeline and further investigation is required.

INTRODUCTION

Panoramic images are images that typically contain many objects in a high-resolution, using a wide-angle view of some space. They have applications in many fields including art, geography and medicine. Today's cameras cannot capture panoramic images as a single image due to limitations of camera lens and are instead constructed by combining multiple images of a subject taken at different times from varying vantage points. The process of combining images relies on an imaging processing pipeline that uses various algorithms, such as Feature-based registration, homography calculation and image stitching, to determine where and how best to overlay the images and then seamlessly combine them into one single image. As more and more devices are being equipped with cameras there is a push to make these PIC pipelines more efficient so that smaller devices with lower computational power can still create panoramic images in real time. However, advances in the field of panoramic image construction are not the only thing improving these pipelines. Due to the multitude of use cases for many of the processes involved, advances in other fields within computer vision can be used to improve PIC pipelines. This is the case for feature-based registration which has use cases in object detection and facial recognition as well as being a key step in any panoramic image construction pipeline.

Feature based registration is the process of finding points of interest in an image, typically corners, edges or intensity blobs, creating a meaningful mathematical representation of said point and then comparing those representations between images to determine common points between the images in question. In order for these feature descriptions, more commonly called feature vectors, to be effective they must be robust with respect to scale, rotation and intensity changes caused by blurring, exposure, etc. The gold standard for robust feature descriptors and matching accuracy is the SIFT algorithm, however this accuracy comes at the cost of a large 128-dimension feature vector which is fairly computationally heavy to compute [2]. In the past two decades, substation research has been done to improve the efficiency of the algorithms that compute these feature vectors as well as reducing the size of the vectors themselves, all while matching or surpassing SIFTs accuracy. Researchers at the University of British Columbia discovered that the dimensionality of feature vectors could be significantly reduced by storing them in binary strings [4] which lead to a number of new algorithms including Binary robust independent elementary features (BRIEF) [4], ORB and BRISK. These breakthroughs were then coupled with inspiration from the human eye by scientists at Ecole Polytechnique Federale de Lausanne to create FREAK, a feature descriptor and matching algorithm based on the working of the human retina [1]. While FREAK's efficacy has been tested and proven in the object detection domain, as it can be trained on a single image and then used to quickly detect matches in any number of other images, there is little research into its abilities in other tasks involving feature-based

registration. We will be evaluating FREAK's usability within panoramic image construction pipelines as a better alternative to algorithms such as SIFT, SURF, and BRISK, specifically to determine if the overhead of training time is justified by its gain in feature descriptor generation and matching speed.

RELATED WORKS

In 1999, David Lowe from the University of British Columbia published the Scale-Invariant Feature Transform or SIFT as a method for locating, describing and matching points of interest in images [5]. SIFT uses the difference of gaussians and multiple scales to determine keypoints, making them scale invariant, and then orients them using image gradients to make the descriptors rotation invariant. The descriptors themselves are a 128-dimensional vector consisting of a 4x4 array of 8 bin orientation histograms. SIFT remains as the gold standard in key point descriptor accuracy today and is used to benchmark new feature-based registration algorithms. In the late 2000's, Bay Et al. build on SIFT's ideas to propose SURF, a similar algorithm that was able to achieve similar accuracy results use 64-dimensional feature vector as opposed to SIFT's 128-dimensional vector making it faster and more space efficient than SIFT [3].

Both SIFT and SURF rely on large dimensional feature vectors which present limitations in improving time and space efficiently in their respective algorithms. As a solution to this, Calonder et al proved that binary strings, or binary descriptors, could be used in place of multi-dimensional feature vectors [4]. These binary strings are significantly faster to compute and compare as each bit represents independent information about a key point, meaning they can be compared using bitwise operations. Calonder et al. used their new binary descriptor to propose Binary Robust Independent Elementary Features (BRIEF) [4]. BRIEF considers a patch of an image around a point of interest and compares intensity values on randomly selected pairs of points to construct a binary string. The binary feature descriptor that BRIEF produces consists of 2 components, a binary representation of the randomly selected sampling pattern and the binary string generated by comparing the pairs. This new algorithm showed promise as it was fast and accurate, however it was not rotation invariant. Scientists at OpenCV labs developed ORB, or Oriented FAST and Rotated BRIEF, which added an orientation component and improved sampling pattern to the BRIEF algorithm making it rotation invariant [8]. ORB calculates a vector from the keypoint to the image patches intensity center of mass in order to orient the descriptor and uses a learned optimal sampling pattern instead of the random pattern generated in BRIEF. Building further on the improvements proposed in ORB, BRISK was developed by researchers at ETH Zurich using image gradients to determine the keypoints orientation and a set sampling pattern of concentric circles around the image patch used to generate the pairs for intensity comparison [6]. Unlike SIFT and SURF, these algorithms all rely on other algorithms (commonly FAST or AGAST) to locate points of interest in images.

As binary feature descriptors had been proven to be a viable alternative to histogram-based feature vectors in terms of accuracy and perform significantly better with respect to time and space complexity researches turned to human biology for inspiration. In 2012 Alahi et al. out of the Ecole Polytechnique Federale de Lausanne, presented FREAK which used the anatomy of the human retina to create their sampling pattern of varying gaussian filters in a similar algorithm to BRISK [1]. Later, in 2019, Morphological REtinA Keypoint (MREAK) was published by H. Vaghela et al., adding inspirations from the pupil of the human eye dilating to help focus in their implementation of an improved FREAK algorithm [10]. FREAK and MREAK where both proposed and tested in the context of object detection algorithms, noting that they would have use cases in other fields as well.

FREAK (FAST RETINA KEY-POINT)

Overview

The FREAK algorithm builds on the success of ORB and BRISK by improving the sampling pattern, pair selection and matching algorithms in creating their binary feature descriptors taking inspiration from the human eye, more specifically the function of the retina.

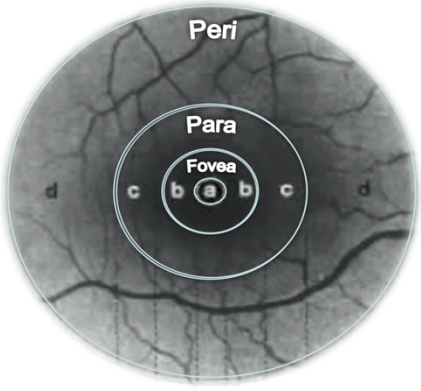

Figure 1. Labeled anatomy of the human retina. [5]

The human retina is composed of retinal ganglion cells arranged in a four-part circular pattern comprised of the foveola, fovea, parafoveal and perifoveal with the foveola being the smallest innermost circle acceding to the perifoveal which is the largest outermost circle (Figure 1). Ganglion cells in these regions encode on/off or binary signals for the brain based on the intensity of light hitting that part of the retina. Signals for the larger perifoveal area are used to determine more coarse changes in intensity consistent with edges or changes from background to foreground and focuses on a general area. Information encoded from the increased smaller retinal zones are used to detect increasingly finer changes in intensity focusing in on detail and specific points as the eye searches an image. FREAK uses this idea of coarse to fine intensity comparisons to create their sampling pattern and optimize their matching algorithm.

Sampling Pattern

Inspired by the coarse-to-fine nature of human retina zones, FREAK's sampling pattern consists of circular gaussian filters of varying standard deviations known as receptive fields. Radii of circles indicate the size of the gaussian kernels. The sampling pattern is created starting with a single small receptive field around the keypoint, followed by 7 overlapping rings, each consisting of 6 receptive fields, making up a total of 43 receptive fields. Each ring, starting from the center moving out, consists of larger circles with a larger standard deviation for the gaussian filter they contain (Figure 2).

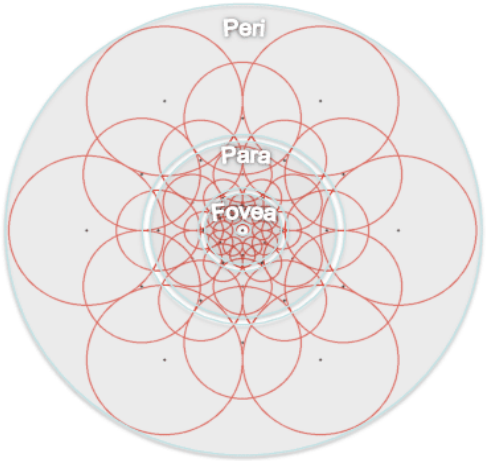

Figure 2. The FREAK receptive fields sampling pattern with retinal zones overlaid for context. [1]

Using the 43 points representing the center of each receptive field, pairs of 2 receptive fields each are created to perform the intensity comparison that make up each bit of the binary descriptor (examples of these pairs shown in figure 3). The larger receptive fields pick up large scale changes in intensity as those regions of the original image are blurred to a higher degree, the inner, smaller receptive fields detecting more subtle changes as there is less blurring applied to those sections of the original image.

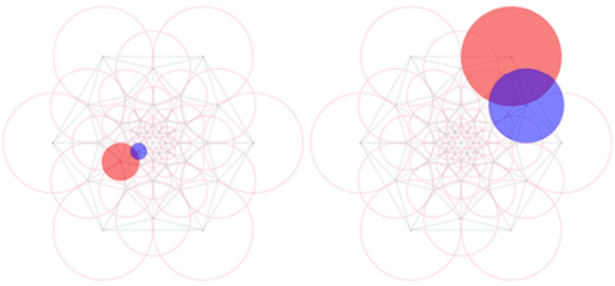

Figure 3. Examples of two different pairs of receptive fields exhibiting the coarse-to-fine structure. [9]

Pair Selection

The FREAK sampling pattern of 43 circles allows for 903 possible intensity comparisons for each keypoint which would result in a 903-bit binary string representation. Each bit of the binary string is computed using the following formula:

$$T(P_a) = \begin{cases} 1 & \text{if } (I(P_a^{r_1}) - I(P_a^{r_2}) > 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $I(P^{r_1}_a)$ and $I(P^{r_2}_a)$ are the gaussian blurred intensities of the receptive fields of a pair.

The developers of the algorithm determined that out of the 903 pairs, only 512 pairs were required to achieve accurate matches. In order to determine which 512 pairs to consider when constructing descriptors, FREAK learns which pairs are optimal by determining which pairs provide the least overlap in information between keypoints in a training image of the intended object to detect. This process is completed in a 4-step algorithm:

1. Large descriptors consisting of the binary representation of all 903 possible pairs are computed and arranged in a matrix M where each row represents a keypoints descriptor and each column represents a possible pair of points from the sampling pattern.

2. The mean of each column is calculated and the column with a mean closest to 0.5 is selected as that column represents the pair with the most variance between key points.

3. All other columns are ordered based on their mean.

4. Pairs with columns that have the lowest correlation to the originally selected column are continually selected adding that pair to the descriptor until 512 pairs are selected.

It was found that pairs that had the most variance most often consisted of two points from the outer layer of the sampling pattern, and only as more and more pairs were added did pairs consisting of points from the inner layers get added. This resulted in feature descriptors typically being ordered in a coarse to fine intensity comparison order which is consistent with how the brain interprets information form the human retina. Examples of learned sampling patterns are illustrated in Figure 4.

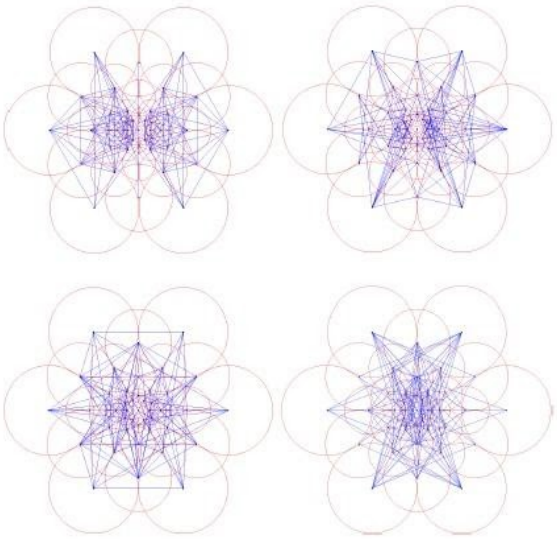

Figure 4. Diagrams of learned 512 pair sampling patterns from FREAK pair selection algorithm. [1]

Orientation

Similar to the method used by ORB, FREAK achieves rotation invariance by determining the dominant orientation for the image patch centered around the keypoint in question by calculating an orientation vector and using it to orient the sampling pattern prior to computing the binary descriptor. However, unlike ORB, FREAK does not use the image patch's center of mass to determine orientation, instead they combine BRISKs approach of estimating the gradient of the image patch using long pairs from their sampling pattern (pairs at each level with the greatest distance between their receptive fields) combined with insights from the human eyes use of saccades when focusing the eye on a target. Saccades are rapid, minute eye movements across a region to gather information to help orient the field of view. FREAK uses this in their orientation implementation by mainly considering pairs whose receptive fields are symmetric about the keypoint (Figure 5).

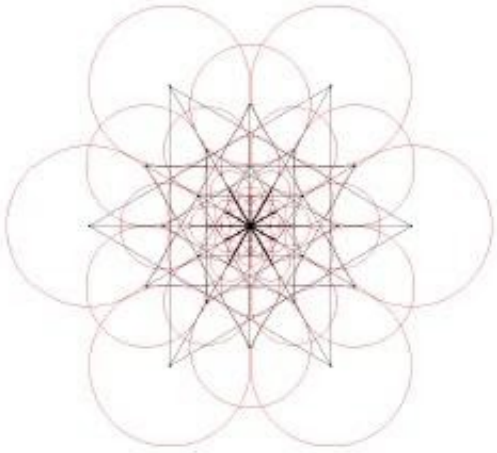

Figure 5. Illustration of pairs used for FREAK orientation [1]

BRISK considers hundreds of pairs to compute their orientation whereas FREAK achieves successful results only using 45 pairs. These 45 pairs are used to compute an orientation vector for the keypoint using the following formula:

$$O = \frac{1}{M} \sum_{P_o \in G} (I(P_o^{r_1}) - I(P_o^{r_2})) \frac{P_o^{r_1} - P_o^{r_2}}{\|P_o^{r_1} - P_o^{r_2}\|},$$

Where M is the number of pairs in the set G, $I(P_{r_o})$ is the smoothed image intensity at point P and P_{r_o} is the vector representation of the point in the pair with respect to the center of the receptive field.

Descriptor Matching

Comparing FREAK feature descriptors to locate matches between images is extremely efficient. Due to the consistent sampling pattern and descriptor length for all keypoint Hamming distance, which can be done using bitwise XOR, can be used to determine distance between points. To improve this even further, the coarse to fine nature of the feature descriptors that the pair selection algorithm generates can be utilized in that comparison algorithm. Similar to how the human eye locates a general region before focusing on smaller intensity changes, the first 16 bytes of the descriptor can be used to perform a coarse comparison

of descriptors which was found to reject more than 90% of potential match candidates. This greatly reduces the amount of computation needed to find matches which is what makes FREAK's matching much more efficient than other algorithms.

IMPLEMENTATION

During the implementation of the FREAK algorithm, we first had to figure out a way to compute the 2D coordinates of the points in the sampling pattern around each keypoint. After understanding the pattern (Figure 2) of linearly increasing radii of circles as we move away from the keypoint, we decided to approximate the coordinates of the sampling pattern by using visual measurements.

Initially, we used OpenCV's SIFT implementation to compute the keypoint coordinates. However, we found out that by default, SIFT creates multiple keypoints for the same points in the image if any peak in SIFT's orientation histogram is above 80% of the highest peak [7]. This results in duplicate keypoint coordinates which are of no significance to us since we are only interested in the computing keypoint locations. Instead, these duplicates increase the total number of keypoints for each image, which in turn increases the time required to compute FREAK's binary descriptors matrix, thus making our algorithm slow. Therefore, we decided to use OpenCV's ORB implementation to compute the keypoint coordinates which return a fixed number of keypoints for each image without any duplicates. We were able to improve timing for our algorithm significantly using this change.

After implementing the FREAK algorithm, we then tested its accuracy of matching keypoints against SIFT and BRISK using visual comparison of matches on a rotated and scaled image. Lastly, we compared FREAK against SIFT and BRISK it as the feature-based registration algorithm in our panoramic image construction pipeline. Here are the steps we perform in the entire pipeline:

1. Use ORB keypoint detector in OpenCV python library to compute the 2D coordinates of keypoints for all the images used to create the panorama.
2. Blur each image with 7 different gaussian kernels. We start with an initial $\boldsymbol{\sigma}$ of gaussian kernel and linearly increase it across each of the 7 levels.
3. Get the sampling pattern by computing the 2D coordinates of all the receptive fields around each keypoint in the image.
4. Compute the dominant orientation angle of each keypoint using methodology explained in the Orientation section. Rotate the 2D coordinates of all keypoint's receptive fields by this angle with respect to the keypoint to achieve rotation invariance. This results in a rotated sampling pattern, i.e new 2D coordinates of all the receptive fields of each keypoint.
5. Starting from the middle image in the panoramic sequence and going towards the right images, for the keypoints of the first 2 images, perform the gaussian blurred intensity comparisons for receptive fields in each pair. This will give us matrix M as explained in the pair selection section.
6. Use OpenCV python brute force matching using hamming distances to match the binary keypoint descriptors in both images.
7. From the matches obtained above, compute the best approximation of the homography matrix between the two images using the RANSAC approach.
8. Use the homography matrix to warp the second image onto the plane of the first image.
9. Stitch the non-overlapping part of the warped image with the first image (i.e blend).
10. Use the stitched image and the next image in the sequence to repeat the same steps starting from step 4 until all images towards the right of the middle image in the sequence are stitched. Do the same for the images towards the left of the middle image in the sequence to create the entire panorama.

RESULTS

We performed two different timing tests to compare our implementation of FREAK with two other commonly used feature detection algorithms: SIFT and BRISK. The tests were performed using Python and we used computer vision algorithms and implementations of SIFT and BRISK from the OpenCV Python libraries. We implemented our own FREAK algorithms by following the research paper by Alahi et al. [1].

In the first test, we made a copy of an image that was scaled by 60% of the original image size and rotated by 90° in the counterclockwise direction. Then, we computed the descriptors for both the original image and the rotated image. Figures 7, 8, and 9 show the drawings of the fifteen best matches using FREAK, SIFT and BRISK descriptors, respectively. The results of the rotation-invariance test show that FREAK has a substantially smaller description time than SIFT and BRISK. However, it does suffer from

an overhead training cost that makes FREAK less desirable for real-time applications. The timing results of one rotation test are shown in Figures 6 and 7. Additionally, the rotation test proves that the methods of Alahi et al. allow the FREAK descriptor to be rotation and scale invariant.

In the second test, we applied each feature descriptor through the panoramic image construction pipeline in order to compare their performances. Figures 13, 14, and 15 show the panoramas constructed from 10 images using FREAK, SIFT and BRISK descriptors, respectively. The timing results in Figures 11 and 12 show how much the training overhead costs affect the performance of FREAK descriptors. The panoramic image construction time for FREAK was over 40 seconds on average. This is because the iterative stitching algorithm requires us to train FREAK descriptors on the new stitched image for each iteration.

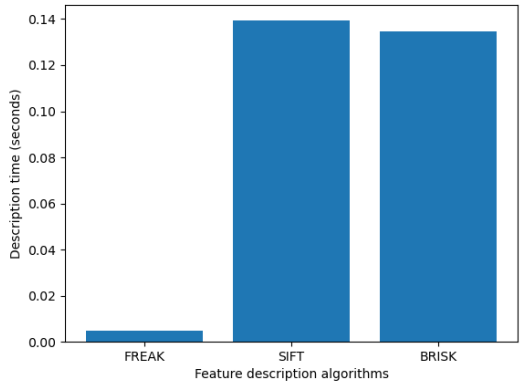

Figure 6. Visual comparison of the description times of the SIFT, BRISK, and FREAK feature descriptors.

```
ROTATION TEST:
FREAK select pairs training time: 4.8166 seconds
FREAK description time: 0.0024 seconds

SIFT description time: 0.1226 seconds

BRISK description time: 0.1317 seconds
```

Figure 7. Timing results of the SIFT, BRISK, and FREAK feature descriptors during the execution of the rotation-invariance test.

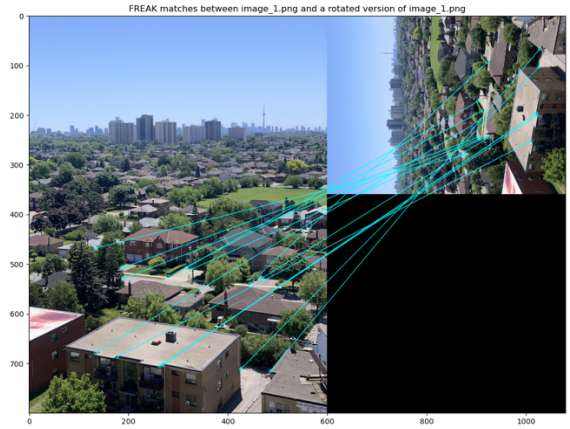

Figure 8. Drawing of the best fifteen matches found using FREAK descriptors between image_1.png and a scaled and rotated version of image_1.png.

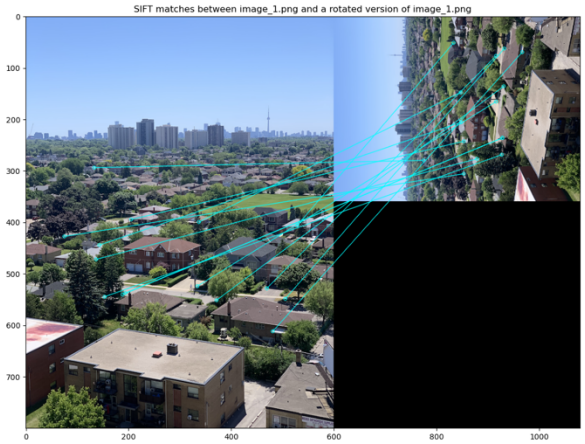

Figure 9. Drawing of the best fifteen matches found using SIFT descriptors between image_1.png and a scaled and rotated version of image_1.png.

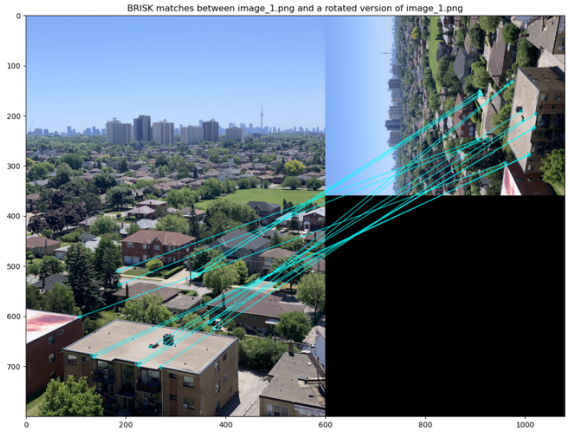

Figure 10. Drawing of the best fifteen matches found using BRISK descriptors between image_1.png and a scaled and rotated version of image_1.png.

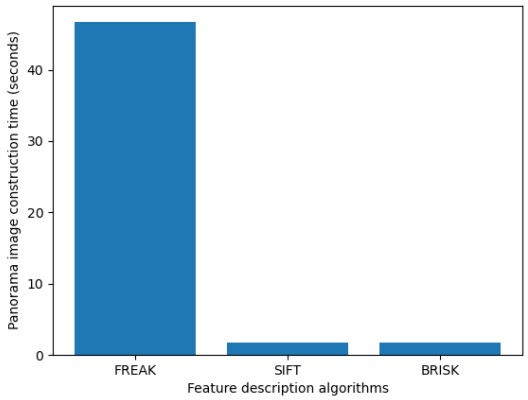

Figure 11. Visual comparison of the panoramic image construction times of the SIFT, BRISK, and FREAK feature descriptors.

```
PANORAMA TEST:
FREAK panoramic image construction time: 46.6814 seconds

SIFT panoramic image construction time: 1.6934 seconds

BRISK panoramic image construction time: 1.6867 seconds
```

Figure 12. Timing results of the SIFT, BRISK, and FREAK feature descriptors during the execution of the panorama image construction test.

Figure 13. Panorama image construction from 10 images of Toronto apartments using FREAK feature descriptors.

Figure 14. Panorama image construction from 10 images of Toronto apartments using SIFT feature descriptors.

Figure 15. Panorama image construction from 10 images of Toronto apartments using BRISK feature descriptors.

CONCLUSION

Based on our findings, the FREAK algorithm as implemented is not sufficient to be used in place of other algorithms (SIFT, BRISK) for feature-based registration in a panoramic image construction pipeline. Although FREAK matched the other algorithms in terms of accuracy and was faster at matching keypoints, the pair selection process involved prior to generating FREAK descriptors made it significantly slower than the other algorithms. FREAK out performs other feature descriptors in the field of object detect due to its ability to pre-train the optimal pair selection for a specific object, which can then be used to quickly generate and compare feature descriptors for that object in any other image. However, this does not translate to PIC as the keypoints in each pair of overlapping images are typically unique to those images, meaning the pair selection algorithm needs to be run for every pair of images adding substantial runtime overhead. As the feature descriptor generating and comparison aspects of FREAK are significantly faster than other descriptors, we believe further investigation should be done to optimize the pair selection aspect of FREAK, specifically for PIC. We propose using machine learning to train the FREAK pair selection using a large data set of sets of images that combine create a panoramic image in order to learn the optimal pair selection pattern for various categories of panoramic images (ie. Cityscape, Mountain-scape, body of water, etc.). These predetermined, categorized, pair selections would eliminate the need to run the pair selection algorithm for each pair of images when constructing a panoramic, significantly speeding up the run time of FREAK to a point where it would potentially outperform algorithms such as SIFT and BRISK.

AUTHORS' CONTRIBUTIONS

Asad

- **Report Sections:** Sampling pattern of FREAK, orientation of FREAK and Implementation.
- **Video Presentation:** Explanation of FREAK algorithm and PIC Pipeline.
- **Code:** Implemented the FREAK algorithm except the rotation invariance part. Compared keypoint matching results with Joshua's implementation and the results were the same. However, my code is not used in the final code of the PIC pipeline.

Joshua

- **Report Sections:** Results
- **Video Presentation:** Showing results and giving conclusions.
- **Code:** Implemented all parts of the FREAK algorithm, including orientation calculation. Conducted tests on FREAK, SIFT, and BRISK descriptors and collected their results. My code is the final code used for both the rotation-invariance test and panoramic image construction test.

Matthew

- **Report Sections:** Abstract, Introduction, Related works, FREAK, Conclusion, References
- **Video Presentation:** Introduction/Related works, video editing
- **Code:** initial implementation of freak (each started implementing freak individually to better understand the algorithm). Not used in the final version.

REFERENCES

1. Alahi, Alexandre & Ortiz, Raphael & Vandergheynst, Pierre. (2012). FREAK: Fast retina keypoint. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 510-517. 10.1109/CVPR.2012.6247715.
2. Ali, S., & Hussain, M. (2012, December). Panoramic image construction using feature based registration methods. In *2012 15th International Multitopic Conference (INMIC)* (pp. 209-214). IEEE.
3. Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, *110*(3), 346-359.
4. Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In *European conference on computer vision* (pp. 778-792). Springer, Berlin, Heidelberg.
5. Hogan, M., and Weddell, J. JA. Histology of the human eye: an atlas and textbook. 1971.
6. Leutenegger, Stefan & Chli, Margarita & Siegwart, Roland. (2011). BRISK: Binary Robust invariant scalable keypoints. Proceedings of the IEEE International Conference on Computer Vision. 2548-2555. 10.1109/ICCV.2011.6126542.
7. Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1150-1157). Ieee.
8. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: An efficient alternative to SIFT or SURF. In *2011 International conference on computer vision* (pp. 2564-2571). Ieee.
9. Suh, B., Choi, S., & Lee, H. A Keypoint Descriptor Inspired by Retinal Computation.
10. Vaghela, H., Oza, M., & Bagul, S. (2019, March). MREAK: Morphological Retina Keypoint Descriptor. In *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)* (pp. 10-15). IEEE.