

A Comparison of SIFT, SURF and ORB on OpenCV



[Mikhail Kennerley](#)

.

[Follow](#)

7 min read

.

May 21, 2021

6



SIFT Key-Points and Key-Point Matching

Feature extraction is an important part of many image processing methods with use cases ranging from image panorama stitching and robotics. The ideal feature extraction method would be robust to changes in illumination, rotation, scale, noise and other transformations while being fast enough to be of use in real-time scenarios. We know that this is not currently the case, every method currently used has its advantages and

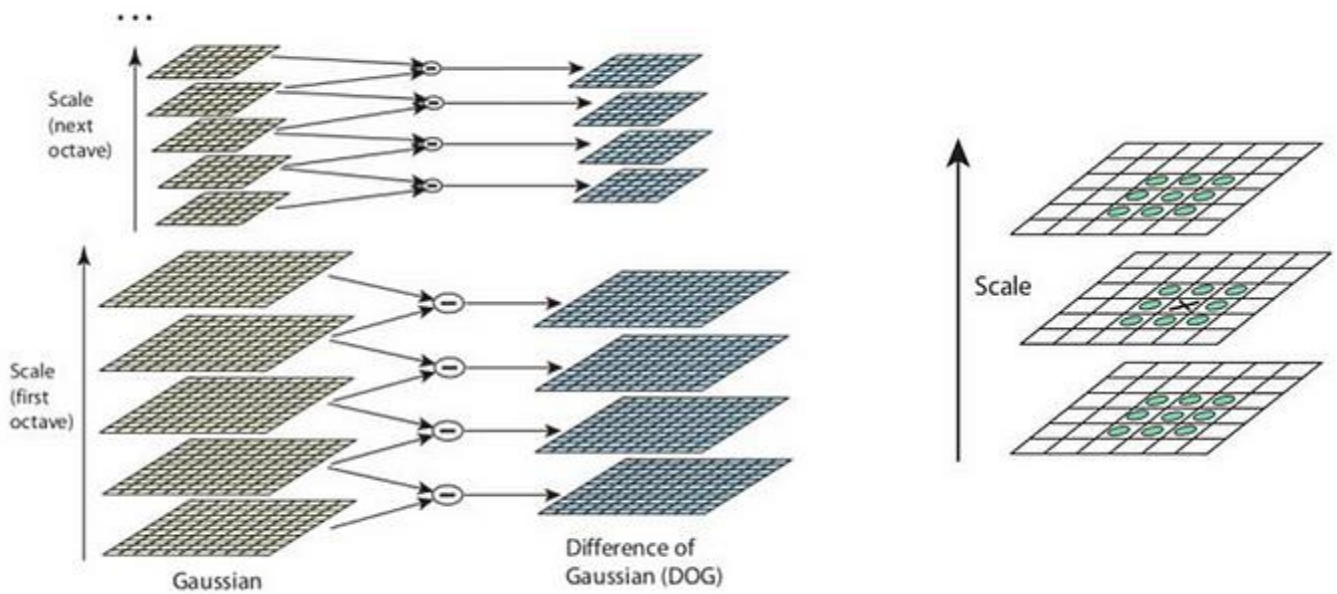
drawbacks, and today we will explore three popular feature extraction methods that are available on OpenCV.

The three methods we will explore today are **Scale-Invariant Feature Transform (SIFT)**, **Speeded Up Robust Feature (SURF)** and **Orientated FAST and Robust BRIEF (ORB)**.

Scale-Invariant Feature Transform (SIFT)

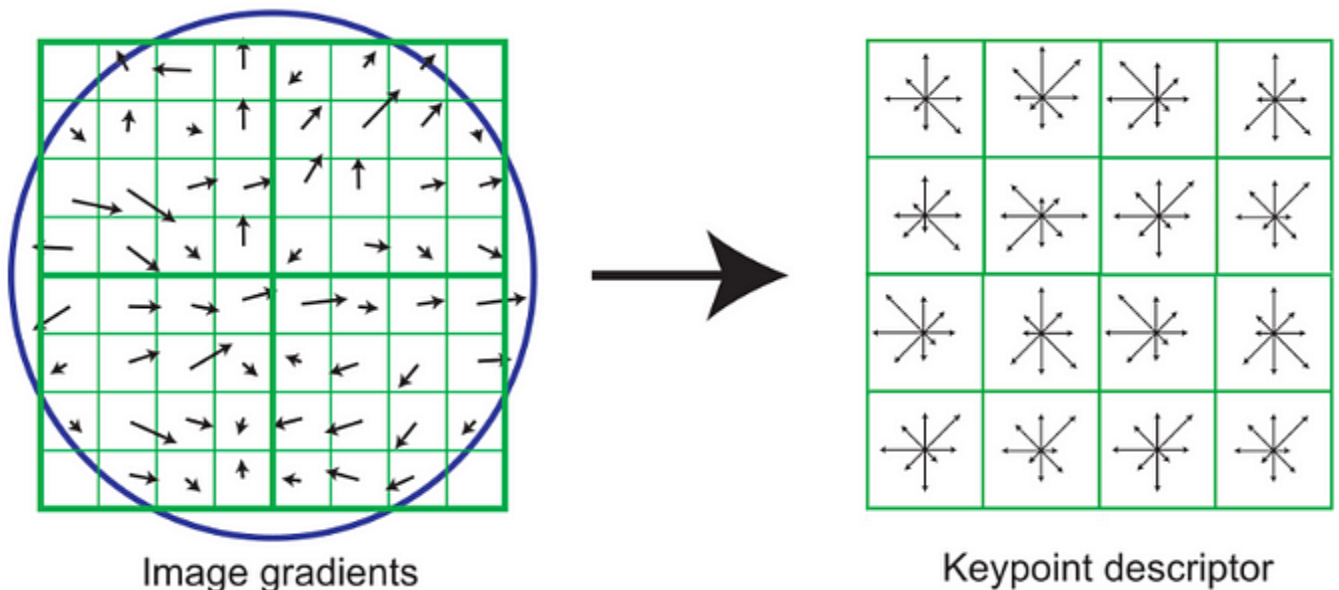
SIFT was created in 2004 by D. Lowe in the University of British Columbia to solve the problem of scale variance for feature extraction. SIFT can be broken down into two parts: **key-point detection** and **key-point descriptor extraction**.

The key-point detection works by approximating Laplacian of Gaussian, which solves the scale variance problem but is expensive to compute, with **Difference of Gaussian (DoG)**. The DoG is stack and searched for **local extrema** in a $3 \times 3 \times 3$ neighborhood to be identified as key-points.



An **orientation** also is assigned to each key-point. This is done by extracting the neighborhood around the key-point and creating a **orientation histogram**, the peak of the histogram is used as the orientation, though any other peak above 80% is also considered for the calculation.

To generate the descriptor , a **16x16 neighborhood** around the key-point is taken and divided into 4x4 cells. An **orientation histogram** is calculated for each cell and the combined histograms are concatenated into a **128 dimension feature descriptor**.

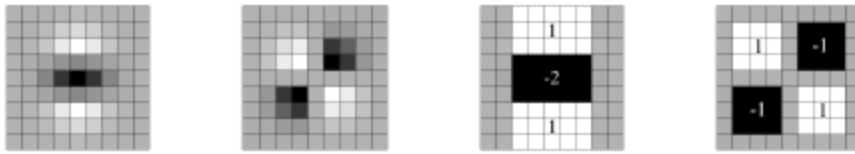


SIFT — Descriptor Generation

Speeded-Up Robust Feature (SURF)

SURF was created as an improvement on SIFT in 2006, aimed at **increasing the speed** of the algorithm.

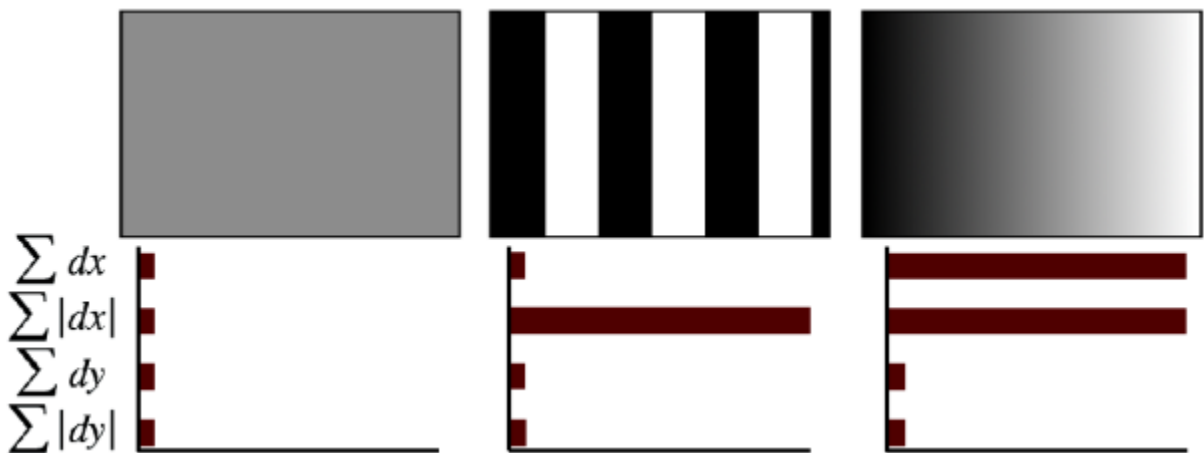
Rather than using Difference of Gaussian to approximate LoG, SURF utilises **Box Filters**. The benefit of this is that box filters can be **easily calculated** and calculations for different scales can be done simultaneously.



Two examples of Gaussian Second Order Partial Derivatives (left) and Corresponding Box filters (right)

To account for orientation, the **Haar-wavelet** responses in both the x and y direction are calculated in a $6s$ neighbourhood around the key-point at a sampling step of s with s being proportional to scale. The sum of the responses in a **sliding scanning area** is used to determine the orientation.

To extract the features of the key-point, a $20s \times 20s$ neighborhood is extracted and divided into 4×4 cells. X and y wavelet responses are extracted from each cell and responses from each cell are concatenated to form a **64 dimension feature descriptor**.

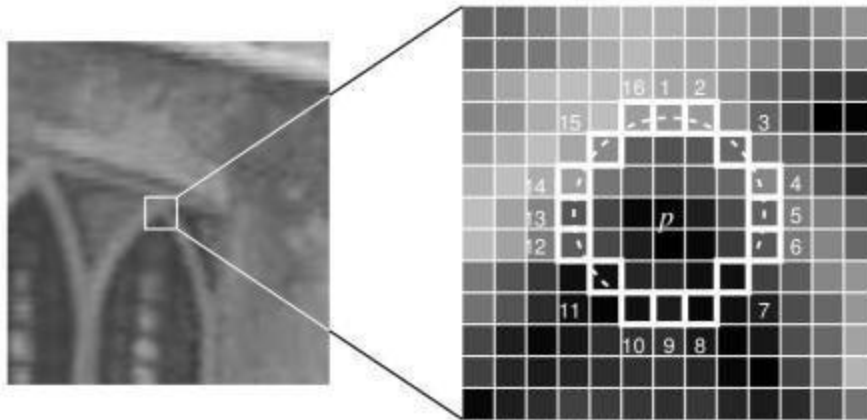


SURF — Descriptor Generation

Orientated FAST and Robust BRIEF (ORB)

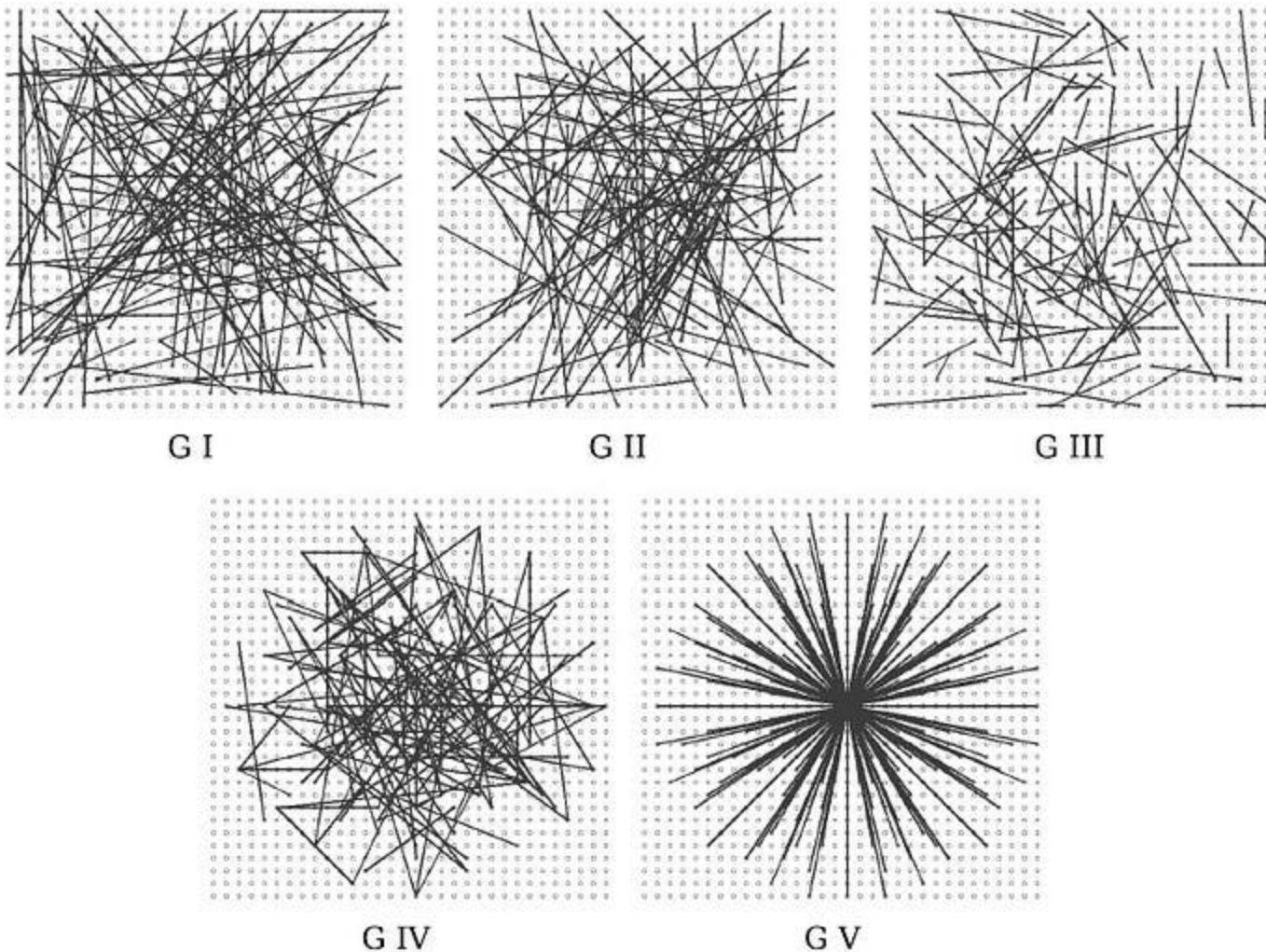
ORB, which as the name suggests is the combination of two algorithms **FAST** and **BRIEF** and was created as an alternative to both SIFT and SURF in 2011.

FAST or Features from Accelerated Segment Test is used as the key-point detector. It works by selecting pixels in a radius around a key-point candidate and checks if there are **n continuous pixels** that are all brighter or darker than the candidate pixel. This is sped up by only comparing a subset of these pixels before testing the whole range. One thing to note is that **FAST does not compute orientation**, to solve this the authors of ORB uses the **intensity weighted centroid** of the key-point patch and the direction of this centroid with reference to the key-point is used as the orientation.



ORB — FAST key-point detection

BRIEF or Binary Robust Independent Elementary Features is used as the key-point descriptor. As BRIEF performs poorly with rotation, the computed orientation of the key-points are used to **steer the orientation** of the key-point patch before extracting the descriptor. Once this is done, a series of **binary tests** are computed comparing a **pattern of pixels** in the patch. The output of the binary tests are concatenated and used as the feature descriptor.



ORB — An example of Binary Test Patterns for BRIEF

Comparison in Python OpenCV

Evaluation

We will be comparing the three feature extraction methods on:

- Speed to compute ~300 key-points
- Total number of key-points detected
- Total percentage of matched key-points to a transformed image

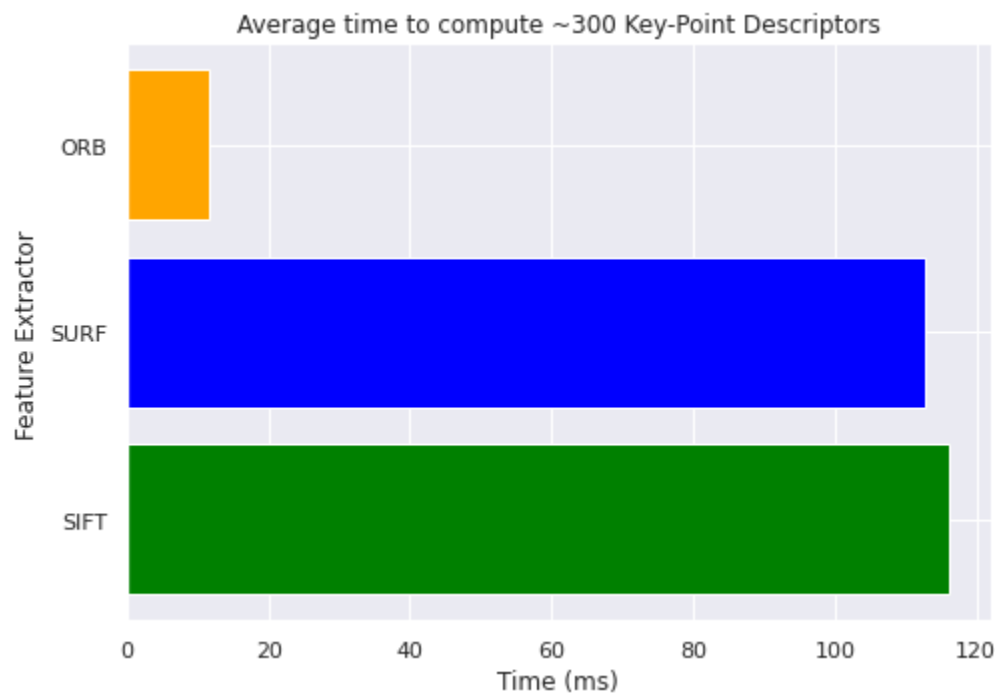
- Average drift of top 500 matched key-points

We will be computing this across 5 images which are shown below and transforming the image luminosity and rotation.



Speed to Compute ~300 Key-Points

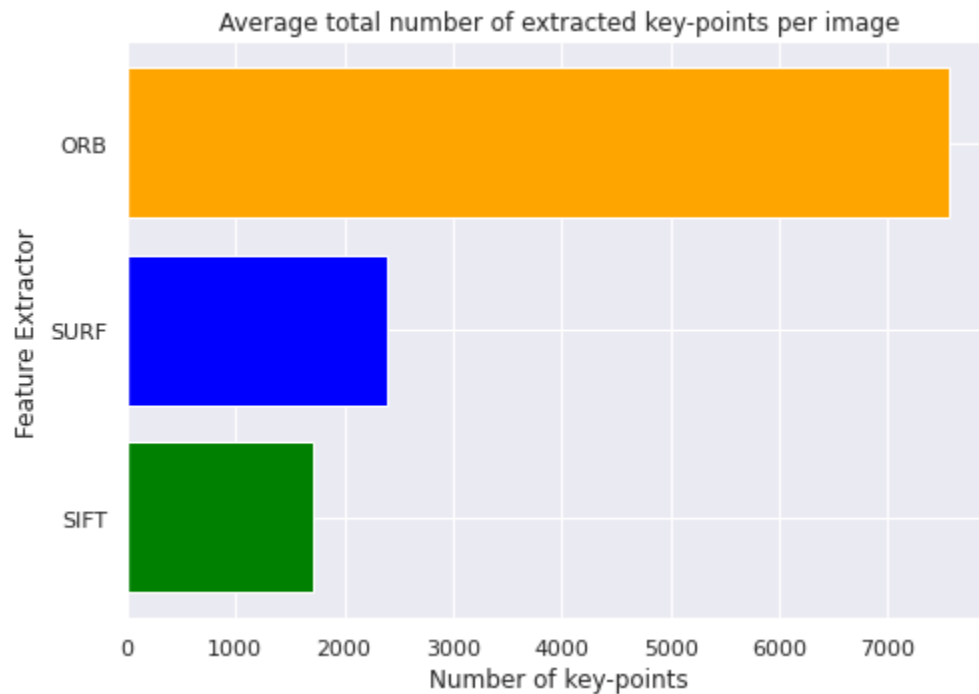
SIFT and ORB allows us to set a limit of the maximum amount of key-points, to do so for SURF we had to adjust its variable to get as estimated average output of 300 key-points (305.4).



From the table above we can see that SURF does indeed perform faster than SIFT though by a small amount (112.8 ms vs 116.2 ms respectively), while ORB is computed an order of magnitude faster than either at 11.5 ms.

Total Number of Key-Points Detected

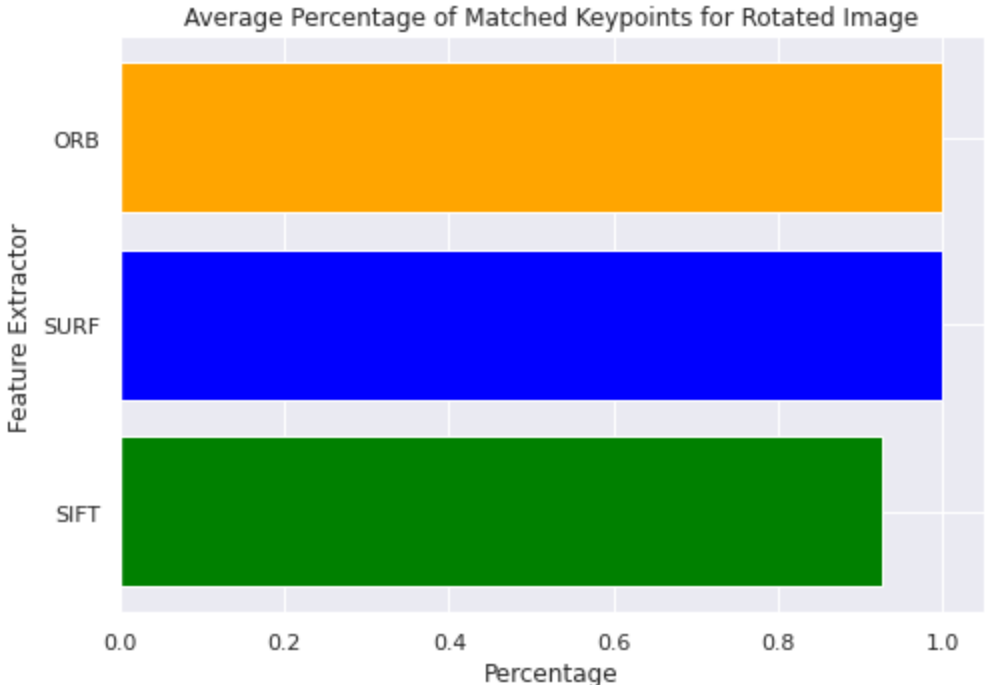
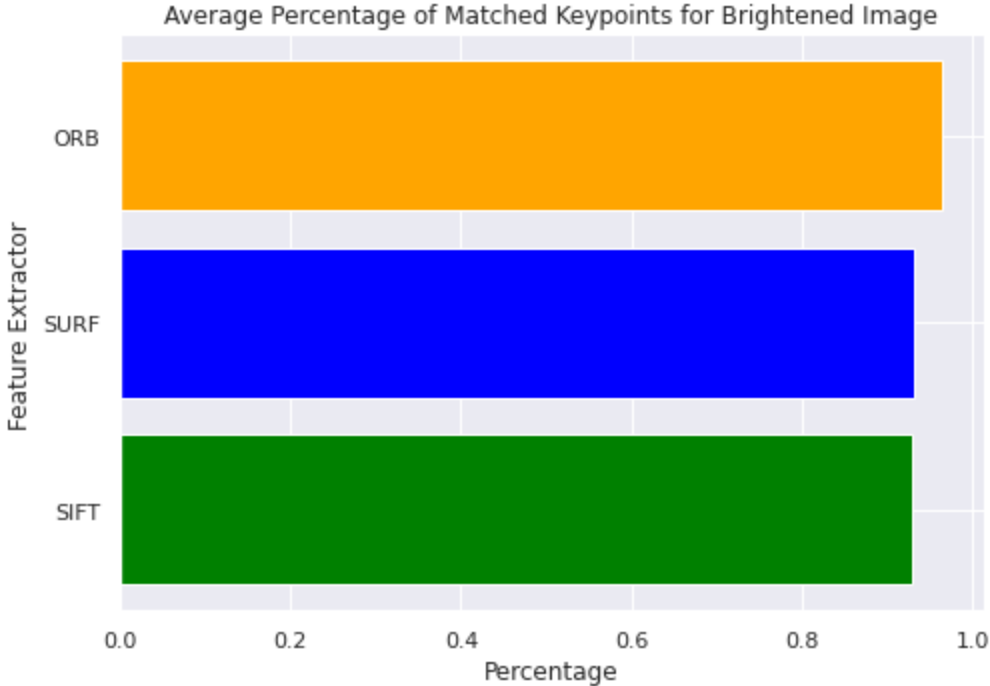
For this test we allowed SIFT and SURF to be run at their default settings. For ORB we set the upper-limit at 10,000 as by default it stops at 500 detections.



Here we can see that ORB is able to extract the most number of key-points per image, nearly triple that of SURF. This is likely due to ORB not requiring the key-point to be the absolute local maxima but the maxima/minima of an n-continuous series.

Percentage of Matched Key-Points

For this test we transform the image in two ways, the first being a slight increase in brightness and the other is to rotate the image 180 degrees. We then compare the number of matched key-points to the total number of key-points in the original image.

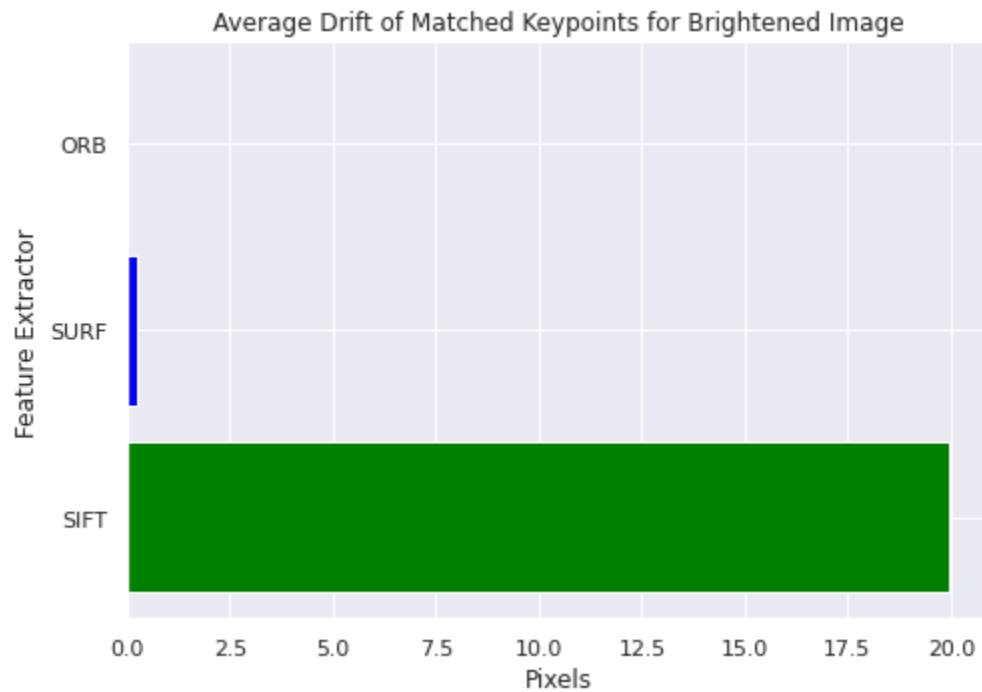


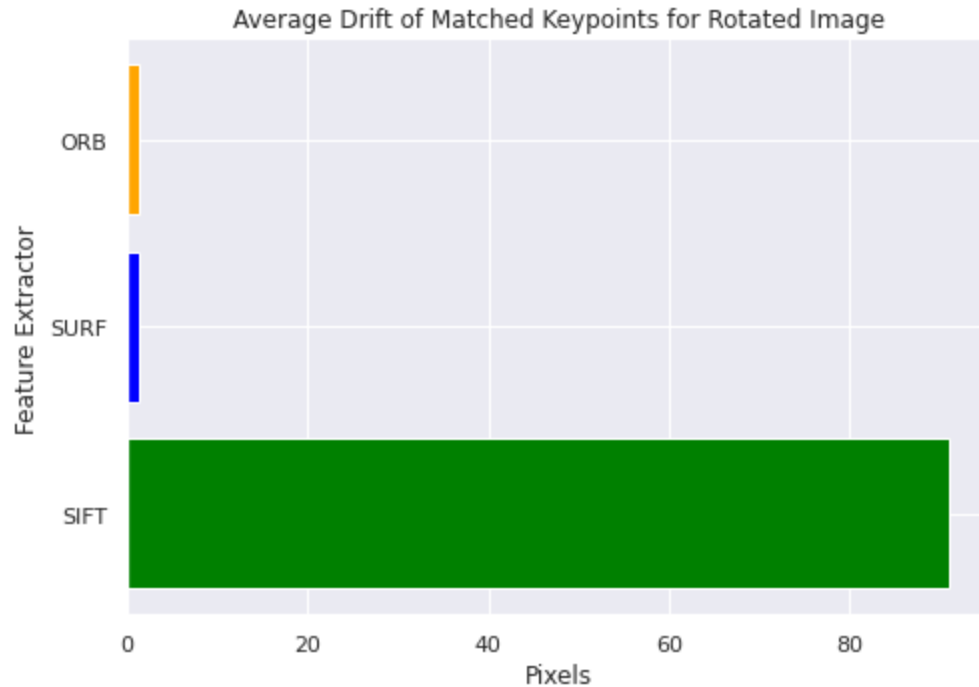
For the brightened image, we can see that the performance of SIFT and SURF are similar with a difference of less than 1% and ORB edging out in performance at 96% matched key-points.

For the rotated image, ORB and SURF both matched 100% of the key-points. SIFT was able to match ~93% of its key-points which is similar to its performance for the brightened image.

Drift of Matched Key-Points

Only the top 500 key-points are used in this test. The coordinates of the transformed images are compared to its relevant pre-computed pair.





We can see that SIFT performs poorly in these tests compared to SURF and ORB, with an average drift of 20 and 91 pixels for brightened and rotated images. SURF performs well, closely matching ORB for the rotated image and with a drift of less than 1 pixel for the brightened image. ORB has 0 drift in the brightened image and less than 2 pixels drift in the rotated image.

The performance across the methods are better with illumination change compared to a rotation.

Summary

Based on the tests, ORB has the best performance across the board with fast computation and is shown to be robust to illumination and rotational change. There may be other situations where SIFT or SURF would be needed, but for most use-cases it would seem that ORB is the best feature extraction method of the three.

This comparison was done on Google Colab which can be accessed here:

<https://colab.research.google.com/drive/1IFC-9QxRzTKoxjdTECoby1B3CamNWFDY?usp=sharing>

Images used in the comparison can be accessed here:

<https://drive.google.com/drive/folders/1v9U8sVAaexx2lrOlQIopP8NrKcRuS7KQ?usp=sharing>