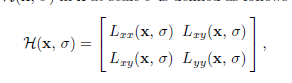
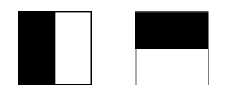
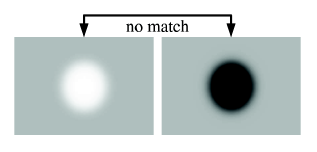
1. First of all, as we all know SURF is a feature point (rather known as interest points) detection algorithm
2. It uses a very basic Hessian Matrix approximation. Hessian matrix is a square matrix of the second order partial derivatives of a function
3. Then it makes use of Integral images. Using integral images, it takes only 3 additions to calculate the sum of the intensities over any rectangular area. So it is faster.
4. Suppose there’s a rectangular region formed between the point *x* and the origin. Then the sum of the intensities of all the pixels in that region can be given by . Where
5. So the computation of the intensities takes a very less time using the integral images as compared to calculating each pixel at a time
6. Now, the detector used is based on Hessian Matrix because it performs well in accuracy.
7. Wherever the determinant of the matrix is maximum, we detect blob like structures. SURF depends on the determinant of the Hessian matrix for both scale and location info.
8. Now we calculate the hessian matrix as:

where x=(x,y) and 6 is the scale

Lxx is the convolution(multiplication) of the Gaussian 2nd order derivative and similarly, Lxy is

1. Though Gaussians are optimal for analysis, they have to be discretised and cropped.
2. Though this is a preferable way using the Gaussians, they lead to a loss in repeatability under image rotations, around odd multiples of . The repeatability is max around the multiples of . Nevertheless, the detector performs well and is fast, so no harm using it.
3. Using the integral images and the fast computation of the Gaussians, the computation time is independent of the size and is cost efficient as well.
4. The relative weight of the response(output) of the filter is used to balance the expression for Hessian determinant. It is a constant =0.9 as changing it didn’t have an impact on the results.
5. Now in the actual SURF algorithm, there are 3 steps:
   1. Fixing an orientation based on the info obtained from the circular region around the interest point
   2. Extracting the SURF descriptors from it
   3. Matching between two images
6. For orientation, SURF uses responses in horizontal and vertical direction for a neighbourhood of size 6s, where s is the scale with which the interest point is detected
7. The horizontal and vertical filters used for calculating the responses. For the dark parts, weight is -1 whereas it is 1 for the bright parts.
8. The dominant responses are calculated by summing all the responses in a sliding window of size 600.
9. Using the integral images, they can be calculated very easily.
10. But for many applications, orientation is not necessary. So we can disable the orientation flag so as to speed up the process. This method is called as U-SURF (upright)
11. Then the second part in SURF, the feature description.
12. SURF uses wavelet responses in h&v directions. A neighbourhood of size 20s \* 20s is taken around the interest point. It is then divided into 4X4 subregions.
13. The responses are calculated for each of the subregion, known as dx, dy. Also |dx| and |dy| are calculated and then all four are summed up for each region. This results into a vector of descriptor size 64 dimensions.
14. Lower the dimension, higher the speed, but less distinctiveness of features. So an additional descriptor of 128 size can also be used.
15. Then comes the last stage, the matching of interest points.
16. For fast indexing of interest points, size of Laplacian is used. Typically, the interest points are blob like structures.
17. Use of Laplacian gives no overhead as it is already calculated during computation. The sign of the Laplacian distinguishes the bright and the dark blob.
18. In this stage, we only compare the features if they have the same type of contrast.
19. Brighter blob v/s Dark blob
20. This is done for every interest point in the image.
21. In short, SURF adds a lot of features to improve the speed.
22. Analysis shows that it is almost 3 times fast as compared to SIFT.
23. SURF is good at handling images with blur or rotation or other transformation, but not with a viewpoint change or an illumination change.