u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

Lab1. Report

1.1 SIFT

We filled the missing code in *find_extremas.m*, the function loops through each pixel and determines if it is an extremum or not. We define a neighborhood of the pixel with the help of a defined radius and find local maxima and minima. Then we compare if the pixel's value is maxima or minima, thus setting the boolean variables *is_maxima* and *is_minima* accordingly.

Results:

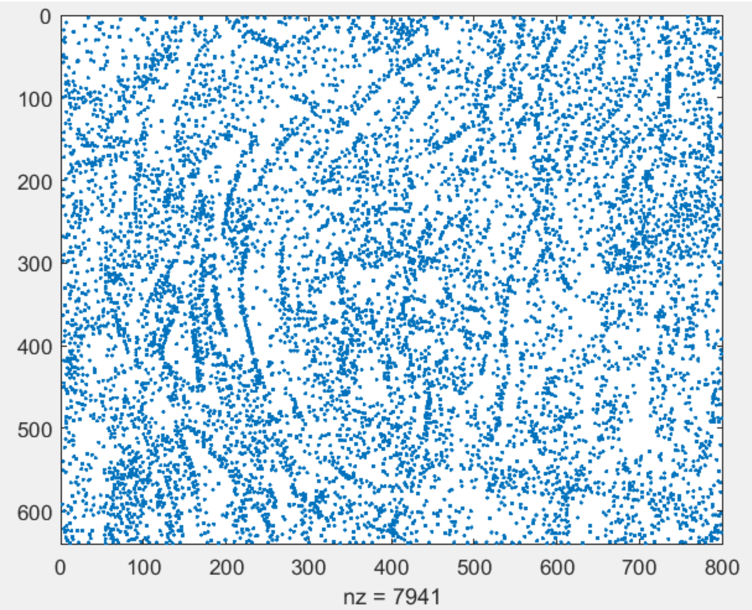


Figure 1: Plotting of keypoints for img3.jpg

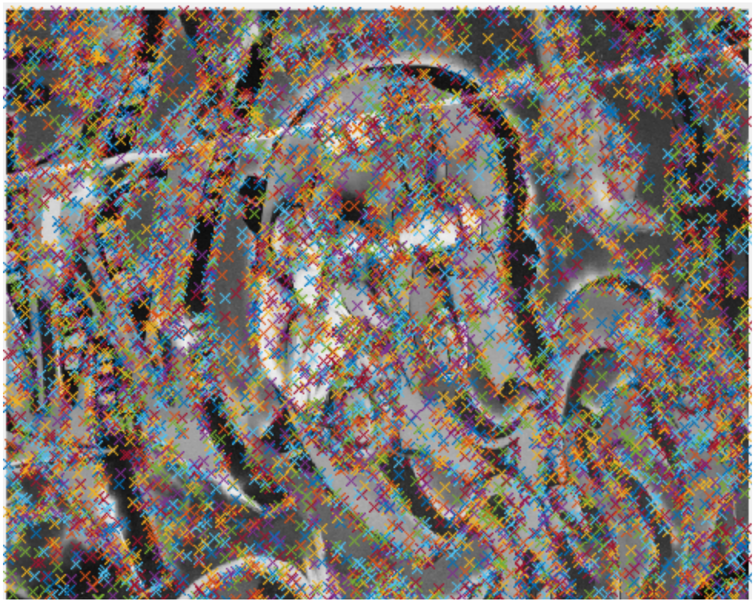


Figure 2: Keypoints marked on img3.jpg

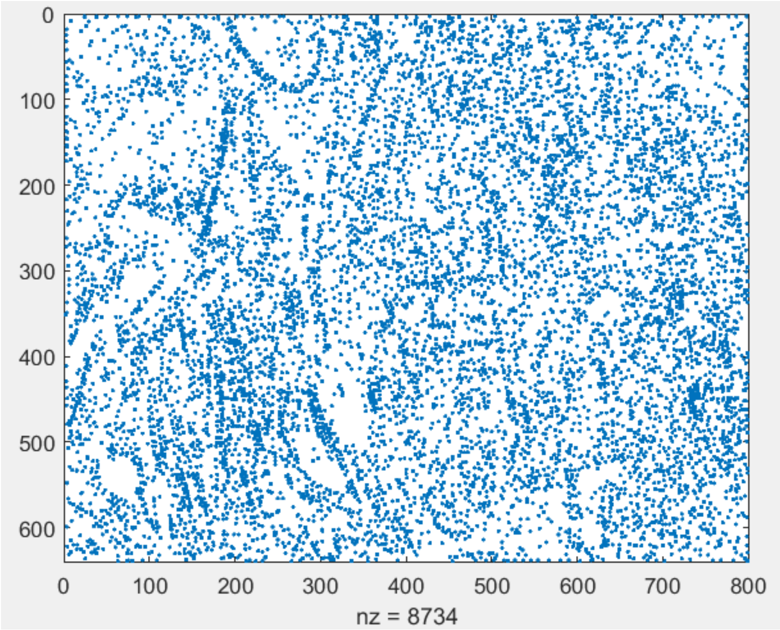


Figure 3: Plotting of keypoints for img5.jpg

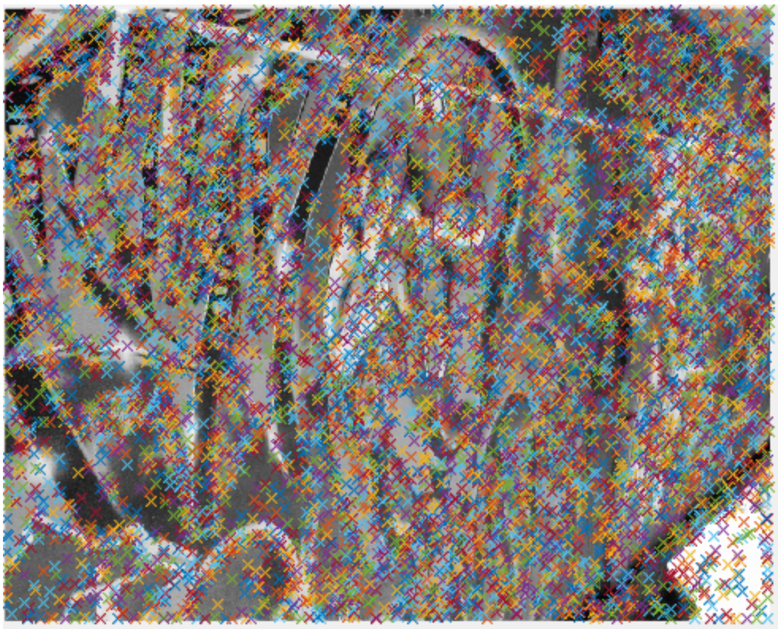


Figure 4: Keypoints marked on img5.jpg

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

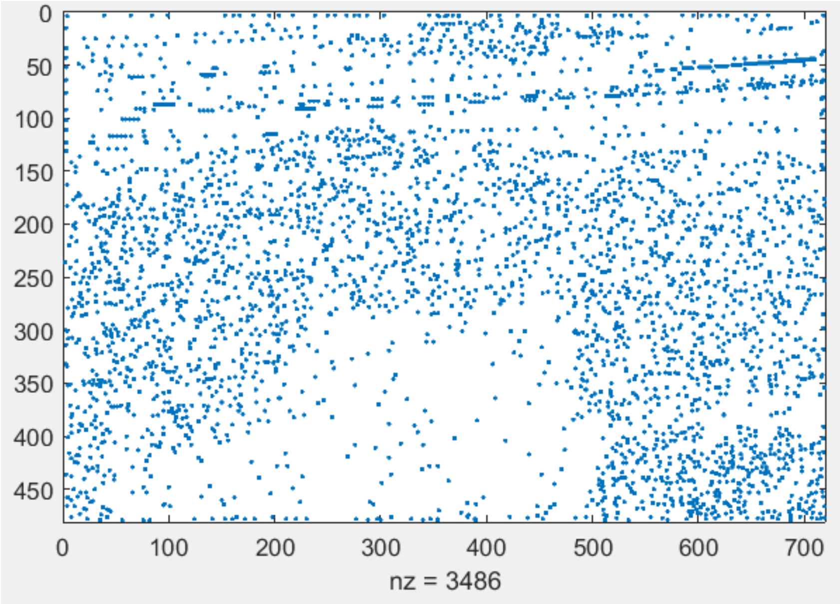


Figure 5: Plotting of keypoints for landscape-a.jpg

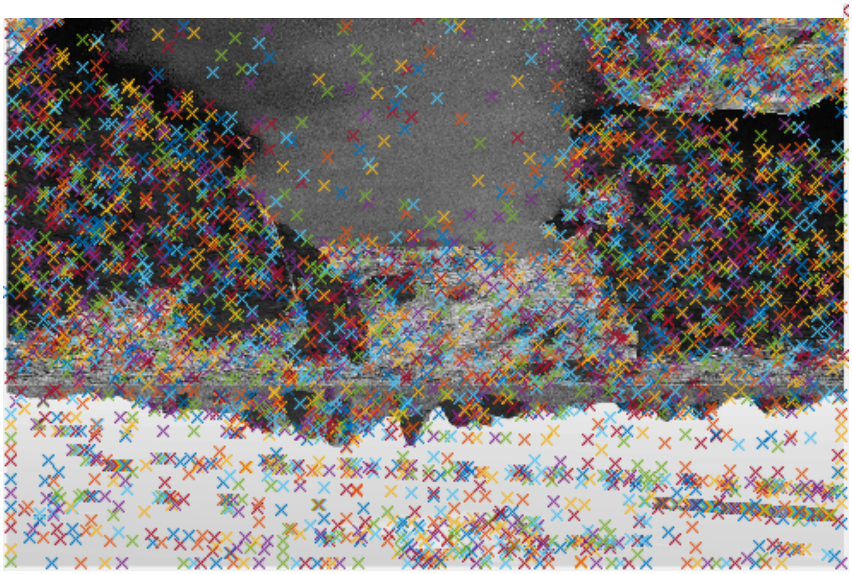


Figure 6: Keypoints marked on landscape-a.jpg

Results on textureless areas

To analyse the performance of SIFT on images with textureless objects, we run it on two images from T-LESS dataset (found online).

Figure 6: Keypoints marked on textureless1.jpg

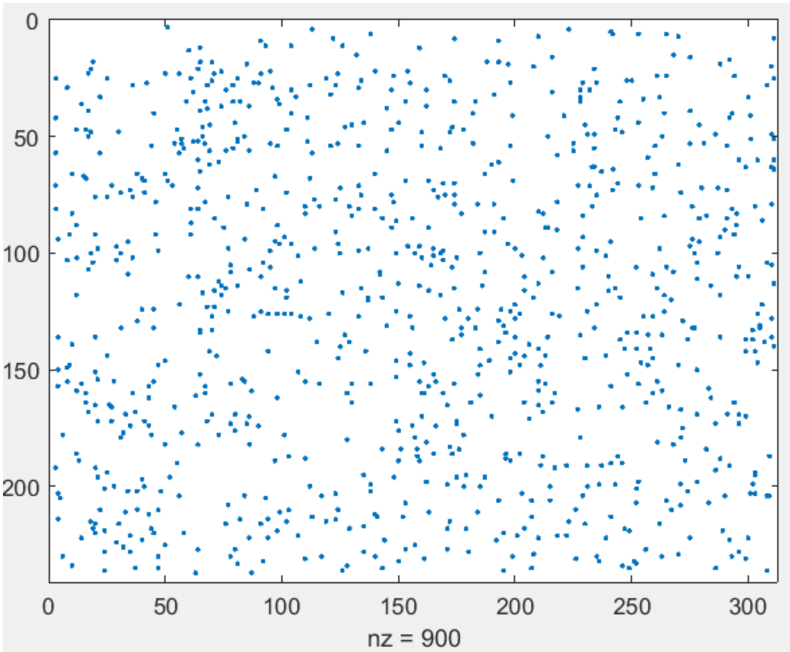


Figure 7: Plotting of keypoints for textureless1.jpg

Figure 9: Keypoints marked on textureless2.jpg

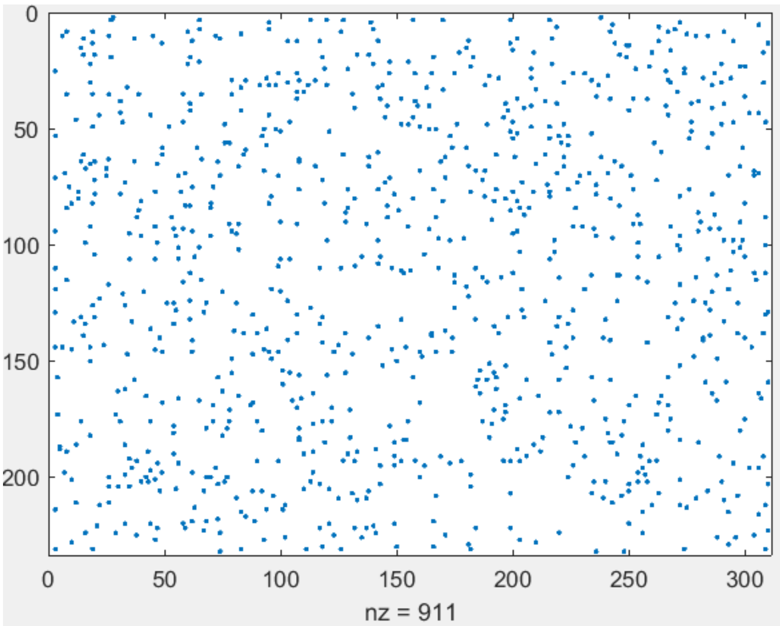


Figure 8: Plotting of keypoints for textureless2.jpg

Analysis of the results

We can observe that SIFT performs relatively well on graffiti and landscape examples from the dataset. The results still contain outliers in the less “texture-dense” areas. As for images with textureless objects, SIFT struggles to detect keypoints well. We can conclude that SIFT is highly dependent on texture presence in the images (we need texture-rich regions for keypoint detection), and it does not deal with edges and shapes well.

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

Task 1.2:

The features FAST,SIFT, SURF,KAZE,BRISK,ORB,Harris,MSER have provided function by MATLAB. These functions have specific arguments. In this part we are going to tune the parameters with different values and compare them according to processing times and number of detected features by providing obtained images. Here is the default values of FAST,SIFT,SURF,KAZE,BRISK,ORB,Harris,MSER algorithms.

FAST TUNNING:

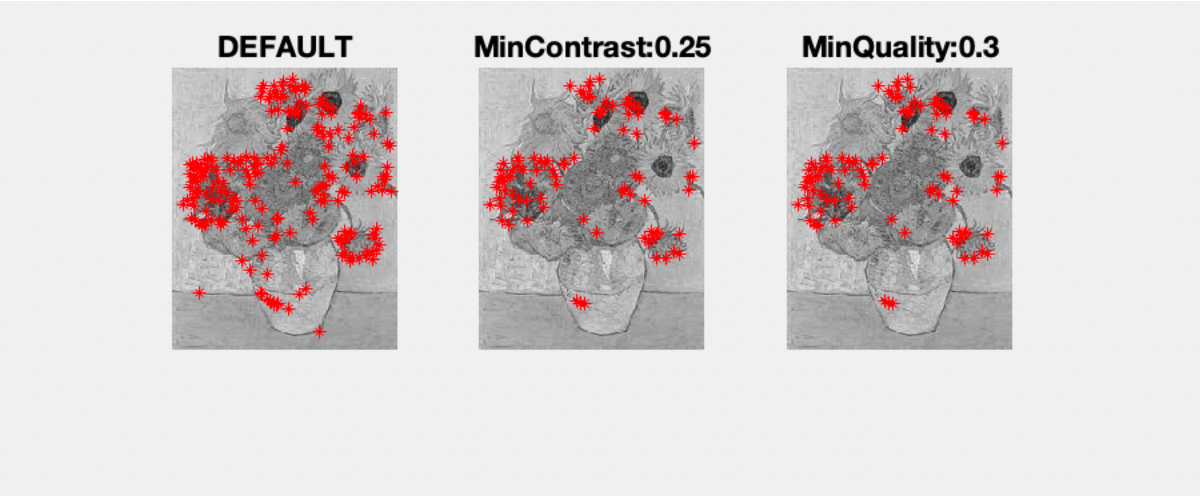
MinQuality: A scalar Q, 0 <= Q <= 1, specifying the minimum accepted quality of corners as a fraction of the maximum corner metric value in the image. Larger values of Q can be used to remove erroneous corners.

Default: 0.1

MinContrast: A scalar T, 0 < T < 1, specifying the minimum intensity difference between a corner and its surrounding region, as a fraction of the maximum value of the image class. Increasing the value of T reduces the number of detected corners.

Default: 0.2

I applied the fast feature detection algorithm with different parameter values. I also extracted the computational time as second. Control value is default.



FASTDEFAULT -> 0.0004119051775 second

FASTMinQuality:0.3 -> 0.00035038083905 second

FASTMinContrast:0.25 -> 0.00035538939 second

According to results, Default value gives better result but with more computational time.

SIFT TUNNING:

Contrast Threshold: A non-negative scalar, 0 <= ContrastThreshold <= 1,

which specifies a contrast threshold for selecting

the strongest features. The contrast threshold is

used to filter out weak features in low-contrast regions. Increasing the threshold decreases number of returned features.

Default: 0.0133

Edge Threshold: A non-negative scalar, EdgeThreshold >= 1, which specifies an edge threshold for filtering out unstable edge-like features that are susceptible to noise. Larger the EdgeThreshold, the less features are filtered out.

Default: 10.0

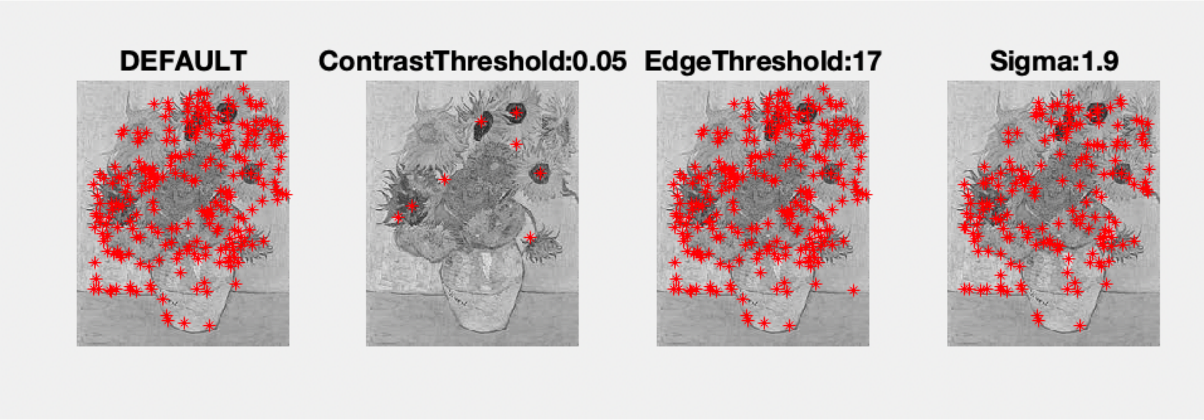
NumLayersInOctav: Integer scalar, NumLayersInOctave >= 1, which specifies the number of layers in each octave. The number of octaves is computed automatically from the image resolution. Increase this value to detect larger features.

Default: 3

Sigma: The sigma of the Gaussian applied to the input image at the zeroth octave. Typical range is between 1 and 2. If the image is blurry then selecting a lower sigma value is recommended.

Default: 1.6

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva



SIFTDEFAULT -> 0.010098659630000 second

Contrast Threshold:0.05 -> 0.008405034630000 second

Edge Threshold :17 -> 0.009235658630000 second

Sigma :1.9 -> 0.009916576630000 second

According to second figure increasing the parameter Edge Threshold to 17 provides better feature extraction. Increasing sigma to 1.9 gives also better result. In addition, each tunning has less computation time for the default parameters. However increasing Contrast Threshold to 0.05 gives worse result for the feature extraction but with less computation time.

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

SURF TUNNING:

MetricThreshold: A non-negative scalar which specifies a threshold for selecting the strongest features. Decrease it to return more blobs.

Default: 1000.0

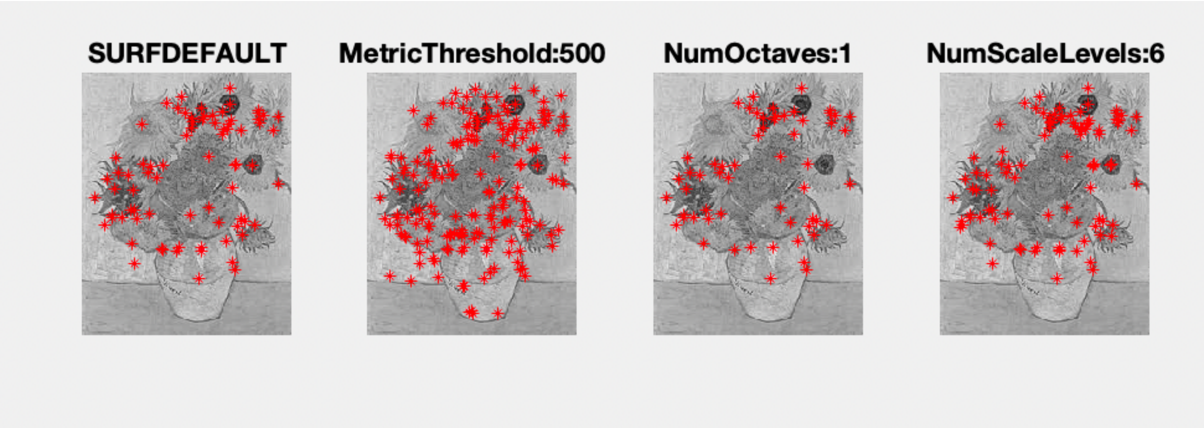
NumOctaves: Integer scalar, NumOctaves >= 1. Number of octaves to use. Increase this value to detect larger blobs. Recommended values are between 1 and 4.

Default: 3

NumScaleLevels: Integer scalar, NumScaleLevels >= 3. Number of scale levels to compute per octave. Increase this number to detect more blobs at finer scale

increments. Recommended values are between 3 and 6.

Default: 4



SURFDEFAULT -> 0.002332386174500 second

MetricThreshold:500 -> 0.002365053174500 second

NumOctaves:1 -> 0.001734428174500 second

NumScaleLevels:6 -> 0.002394345174500 second

According to Surf feature extraction with different parameter, the best feature extraction is obtained with decreasing the Metric Threshold value from 1000 to 500. In addition, computational time did not change radically. However, maximizing the Number Scale Levels to 6 does not change feature extraction compared to default control computation. However minimizing the number of octaves to 1 has worse result then default image, but it has less computational time.

KAZE TUNNING:

Diffusion: A string or character array specifying the method to be used for computing the conductivity based on first order derivatives of a layer in scale space. Possible values are:

'region' - This option promotes wider regions over smaller ones.
'sharpedge' - This option promotes high-contrast edges.
'edge' - This option promotes smoothing on both sides of an edge stronger than smoothing across it.

Default: 'region'

Threshold: Double scalar, Threshold >= 0. Increase this value to exclude less significant local extrema.

Default: 0.0001

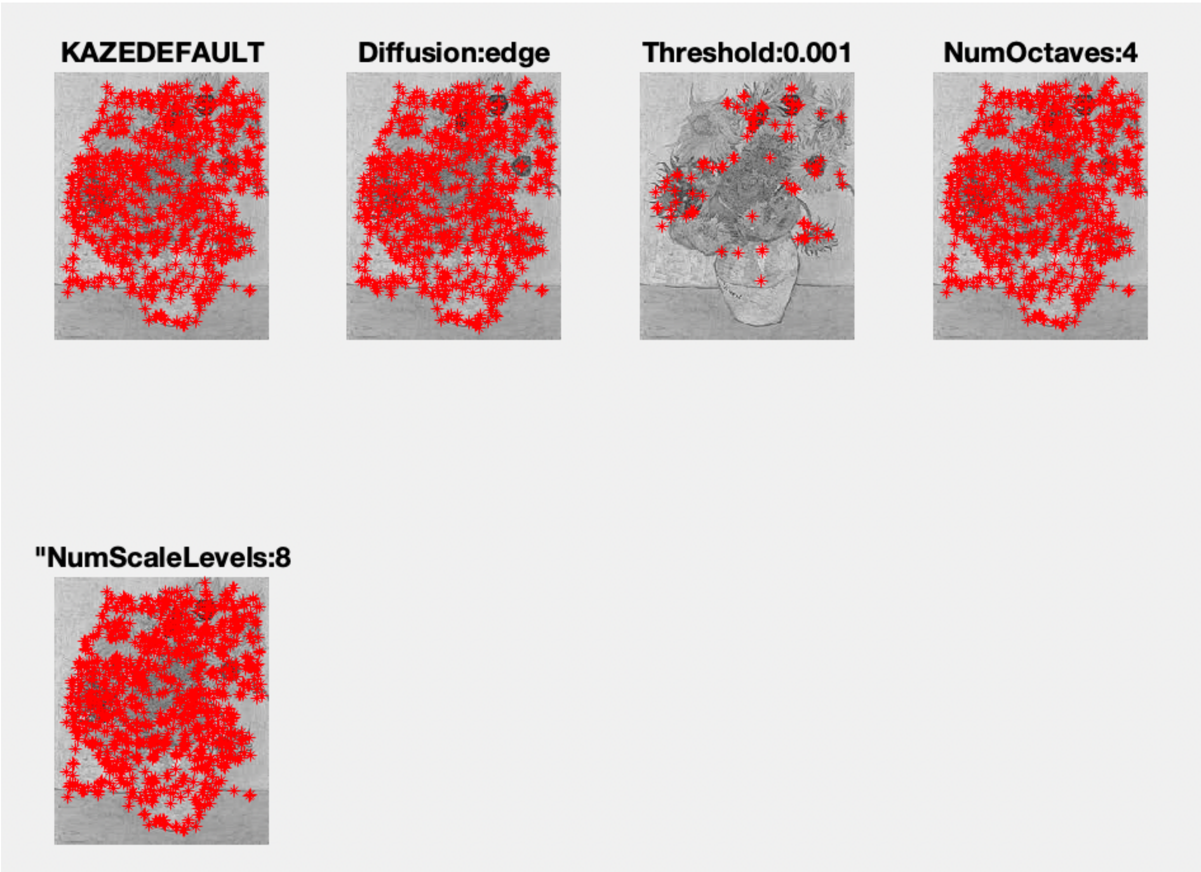
NumOctaves: Integer scalar, NumOctaves >= 1. Increase this value to detect larger features. Recommended values are between 1 and 4. Setting NumOctaves to one disables multi-scale detection and performs the detection at the scale of input image I.

Default: 3

NumScaleLevels: Integer scalar, NumScaleLevels >= 3 and <= 10.

Increase this value to achieve smoother scale changes, along with getting more intermediate scales between octaves. Recommended values are between 1 and 4.

Default: 4



Kaze default: 0.012932706830500 second

Diffusion(edge) 0.015413748830500 second

Threshold 0.001 0.010488748830500 second

NumOctaves 4 0.017002040830500 second

NumScale Levels 8 0.017764748830500 second

According to results tuning to the parameter Diffusion into the “edge”, increasing number of octaves and increasing number of scale levels into the 8 do not change radically the feature extraction results on the image. Whereas increasing Threshold to 0.001 decrease the efficiency of the algorithm on this image but it increases the computation time compare to default parameter on the Kaze extraction.

BRISK TUNNING:

MinContrast: A scalar T, $0 < T < 1$, specifying the minimum intensity difference between a corner and its surrounding region, as a fraction of the maximum value of the image class. Increasing the value of T reduces the number of detected corners.

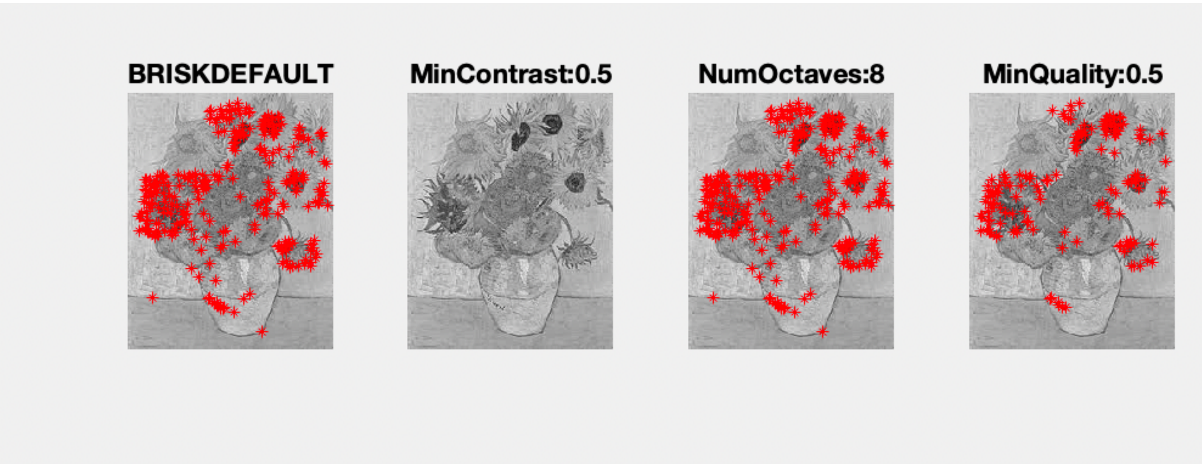
Default: 0.2

NumOctaves: Integer scalar, NumOctaves >= 0. Increase this value to detect larger features. Recommended values are between 1 and 4. Setting NumOctaves to zero disables multi-scale detection and performs the detection at the scale of I.

Default: 4

MinQuality: A scalar Q, 0 <= Q <= 1, specifying the minimum accepted quality of corners as a fraction of the maximum corner metric value in the image. Larger values of Q can be used to remove erroneous corners.

Default: 0.1



Brisk Default: 0.071723591422500 second

MinContrast_0.5: 0.070826465422500 second

NumOctaves_8: 0.072221341422500 second

MinQuality_0.5: 0.072067340422500 second

According to result which you can see above's figure, default configuration of Brisk algorithm is the best option to select for the feature extraction. Second best is the result which we obtained from increasing number of octaves to the 8. However, if the minimum contrast parameter is increased 0.5, the result is unapplicable for this image. The computational time are very close to each other.

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

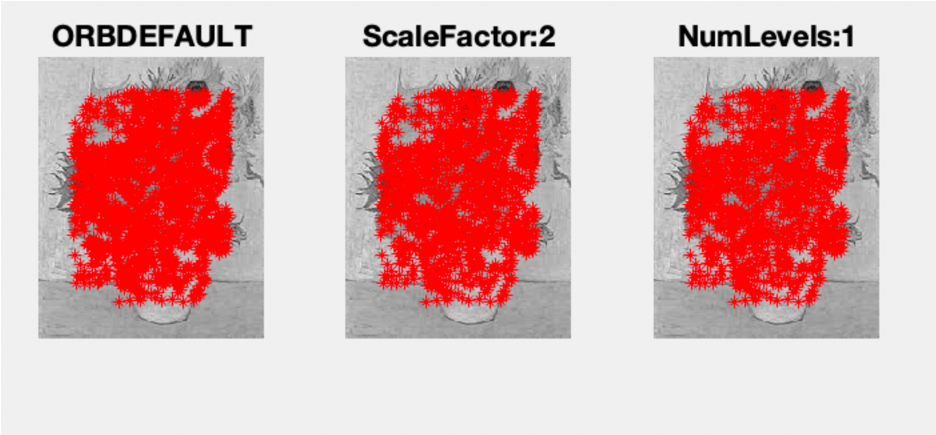
ORB TUNNING:

ScaleFactor: A scalar greater than 1, specifying the pyramid decimation ratio. Higher values reduce the number of pyramid levels and speeds up computation. However, it degrades the feature matching scores. Lower values (slightly over 1) increase the number of pyramid levels and improves the feature matching scores at the cost of computation speed.

Default: 1.2

NumLevels: An integer scalar greater than or equal to 1, specifying the number of pyramid levels. Increase this value to detect keypoints at more pyramid levels. Along with 'ScaleFactor', this value controls the number of pyramid levels on which the interest points are evaluated.

Default: 8



ORB Default: 0.002600370835000 second

NumLevels_1: 0.001098204835000 second

Scale Factor_2: 0.001738143600 second

According to results, the ORB feature detection is not suitable for the feature extraction of the given image. Nevertheless, minimizing NumLevels into the 1 gives better feature detection with less computational time.

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva

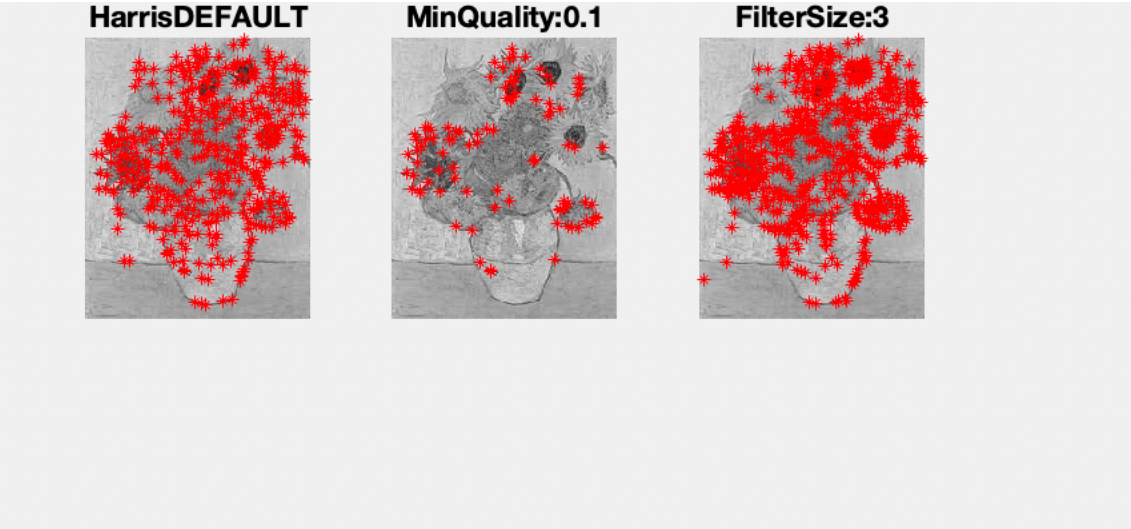
HARRIS TUNNING:

MinQuality: A scalar Q, 0 <= Q <= 1, specifying the minimum accepted quality of corners as a fraction of the maximum corner metric value in the image. Larger values of Q can be used to remove erroneous corners.

Default: 0.01

FilterSize: An odd integer, S >= 3, specifying a Gaussian filter which is used to smooth the gradient of the image. The size of the filter is S-by-S and the standard deviation of the filter is (S/3).

Default: 5



Harris Default: 0.007311155518000 second

Min Quality:0.1 0.006786197518000 second

FilterSize:3 0.005892489518000 second

According to results Harris feature detection algorithm is a suitable algorithm for the feature extraction from the given image. Increasement of Min Quality gives rise to worse feature detection but with less computational time. Furthermore, decreasing Filter Size into the 3 pose a worse detection, but with less computational time. In general, it can be expressed that default configuration of the Harris provides the best result on the given image but with more computational time.

MSER TUNNING:

regions = detectMSERFeatures(I) returns an MSERRegions object, regions,

containing region pixel lists and other information about MSER features

detected in a 2-D grayscale image I. detectMSERFeatures uses Maximally

Stable Extremal Regions (MSER) algorithm to find regions.

[..., cc] = detectMSERFeatures(I) optionally returns MSER regions in a

connected component structure. This output is useful for measuring

region properties using the regionprops function. The connected

component structure, cc, contains four fields:

Connectivity   Connectivity of the MSER regions (default is 8)

ImageSize   Size of I.

NumObjects   Number of MSER regions in I.

PixelIdxList   1-by-NumObjects cell array where the kth element

in the cell array is a vector containing the linear

indices of the pixels in the kth MSER region.

regions = detectMSERFeatures(I,Name,Value) specifies additional

name-value pair arguments described below:

ThresholdDelta   Scalar value, 0 < ThresholdDelta <= 100, expressed

as a percentage of the input data type range. This

value specifies the step size between intensity

threshold levels used in selecting extremal regions

while testing for their stability. Decrease this

value to return more regions. Typical values range

from 0.8 to 4.

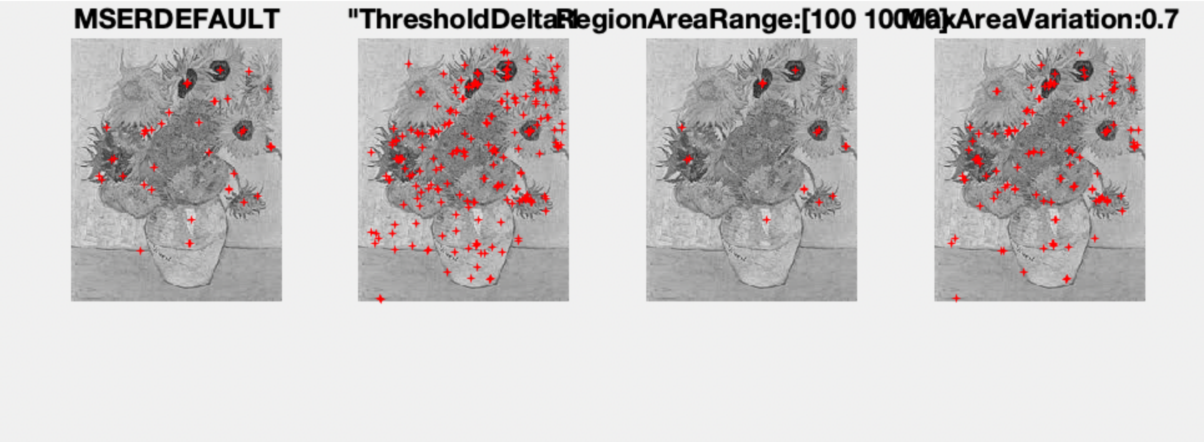
Default: 2

RegionAreaRange: Two-element vector, [minArea maxArea], which specifies the size of the regions in pixels. This value allows the selection of regions containing pixels between minArea and maxArea, inclusive.

Default: [30 14000]

MaxAreaVariation: Positive scalar. Increase this value to return a greater number of regions at the cost of their stability. Stable regions are very similar in size over varying intensity thresholds. Typical values range from 0.1 to 1.0.

Default: 0.25



MSER Default: 0.014326410998500 second

Threshold Delta:1	0.037047534998500 second
Region Area Range [100 10000]	0.008494951998500 second
MaxAreaVariation:0.7	0.024830785998500 second

According to results increasing Area Variation from the default value 0.25 to 0.7 provides better result for the feature detection whereas the algorithm is nearly two times slower than MSER default value. In addition, narrowing the range into the [100 10000] provides worse feature detection result, with less computation time. Decreasing Threshold Delta into the 1 gives the points which are not maxima and minimums.

General Comparison:

In this part we compare the feature detection algorithms (with default parameters).

FAST: 6.283265790000001e-04 second

SIFT: 0.010267702579000 second

SURF: 0.002068243579000 second

KAZE: 0.013094993579000 second

BRISK: 0.071504410579000 second

ORB: 0.002618160579000 second

HARRIS. :0.006113327579000 second

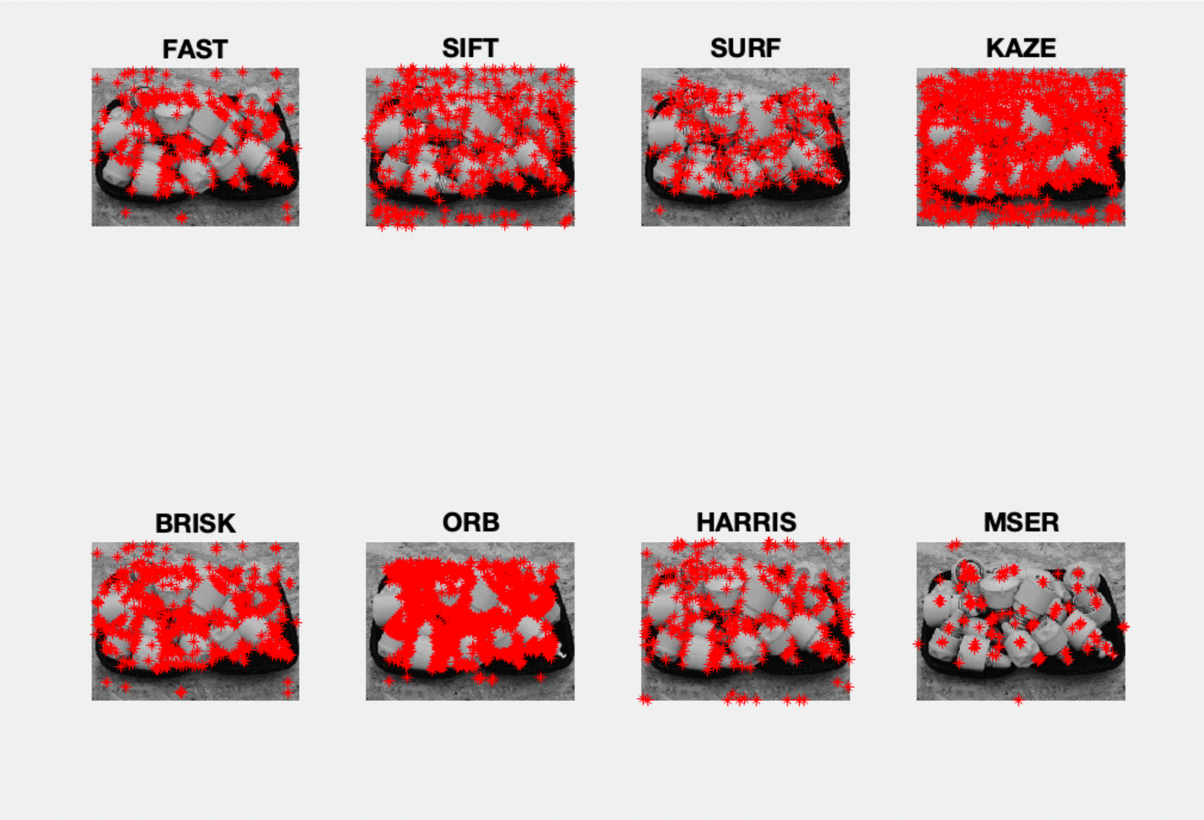
MSER: 0.012639326579000 second

According to results, FAST requires less computation time compared to other algorithms. HARRIS provides the best feature detection algorithm for this task. Whereas MSER could not detect whole the key points, but ORB detect unnecessary point as a key point. For this task HARRIS is preferable compared to other algorithms in terms of quality of obtained key points and computational time efficiency.

Textureless Area:

In this part we will compare the algorithms on the texture less area and check which are scale independent by zooming a part of an image. Here is the original textureless image and scaled version.

u195839 Umut Ekin Gezer
u204256 Ekaterina Erofeeva



FAST:5.756235855000000e-04 second

SIFT:0.012416915585500 second

SURF:0.003266414585500 second

KAZE:0.017940998585500 second

BRISK:0.072579873585500 second

ORB:0.002467165585500 second

HARRIS:0.007712582585500 second

MSER:0.038050498585500 second

According to results, the feature key points are well detectable by HARRIS algorithm.

Now we will check whether these algorithms have the same results with different scale of the given image.

According to result, whole feature detection algorithms are scale independent. In addition Harris and MSER provides again the best results, event scale is changed f