Scale-Space Blob Detectors

David Štych
Aleksandra Jamróz

October 7, 2022

1 LoG filter kernel size

Sigma is a gaussian parameter defining variance of the gaussian. In case of our task, it defines how wide the filter will spread over the image. Kernel size is dependent from chosen sigma value. Radius of kernel is in direct proportion to sigma. The bigger values, the larger blobs in the image we can detect, because the filter id wider. Meanwhile smaller sigmas allow us to find tiny blobs. That's why in order to detect blobs of all sizes, we create range of sigmas that we iterate on later.

2 Speed comparison

Down-sampling the image was significantly faster than changing the filter size.

Input image	Time [ms] changing filter size	Time[ms] down-sampling the image
sunflowers.jpg	247.194	104.404
fruits.jpg	359.280	136.804
water_texture.jpg	302.661	139.446

Down-sampling the image takes some processing time. However, after it is done, all subsequent convolution operations are simplified and require less time to complete. We noted over 2x improvement of results after using down-sampling algorithm instead of the changing filter size one.

3 Scale-space selection

Our parameters are:

$$\sigma_0 = \frac{R_{min}}{\sqrt{2}} = \frac{2}{\sqrt{2}}$$

$$s = [1.2, 1.6] \quad \textit{We tuned the parameter to obtain desired results.}$$

$$k = 0, 1, 2...9$$

We chose σ_0 value depending on the smallest predicted blob radius, marked as R_{min}. Next sigmas are calculated according to the given equation:

$$\sigma_k = \sigma_0 s^k$$

4 Down-sampling image - scale

We selected σ_0 the same way $\sigma_0 = \frac{R_{min}}{\sqrt{2}}$. The image is then scalled down with factor s^k, while keeping σ constant $\sigma = \sigma_0$. It allows us to keep the same blob sizes as we got in filter-changing algorithm.

In the first iteration, the scale is not changed s^0.

5 Down-sampling image - normalization

As σ increases, derivatives are attenuated. We need to normalize the filter response by multiplying it by σ^2 to keep good algorithm results.

However, when down-sampling the image, we keep σ constant. Therefore, it is not necessary to normalize the filter response.
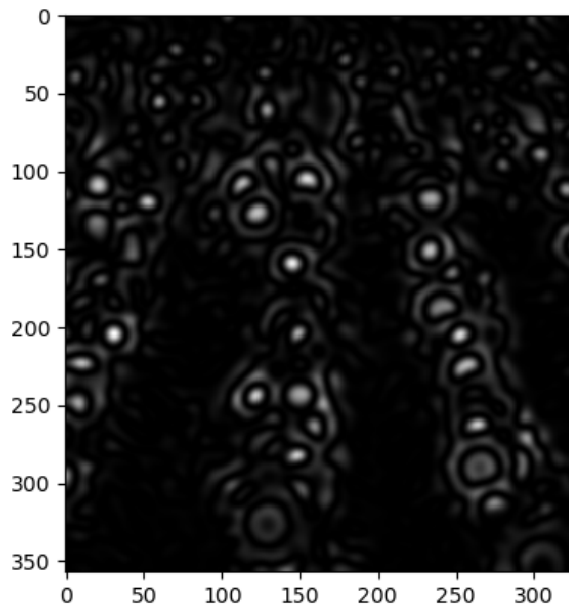
6 Non-maximum suppresion function

Figure 1: Filter response

Figure 1 shows the filter response. The goal of the non-maximum suppression function is to find the local maxima of the filter response. As we see on the image, we try to separate white regions from black background. Some blobs are filtered using thresholds. Overlapping blobs are removed as well.

7 Other images

By default, the LoG filter looks for dark blobs.

For image *sunflowers.jpg* we squared the filter response. It is because we are looking for dark as well as light blobs. Therefore, squaring the filter response makes it positive everywhere and we can find dark/light blobs based on the amplitude.

```
filtered = ndimage.gaussian_laplace(image, sigma=s)**2
```

For image *water_texture.jpg* we kept the original filter (no changes in sign or squaring the response). It is because we are looking for dark blobs only.

```
filtered = ndimage.gaussian_laplace(image, sigma=s)
```

For image *fruits.jpg* we multiplied the filter response by -1. It is because we are looking for light blobs only.

```
filtered = -ndimage.gaussian_laplace(image, sigma=s)
```
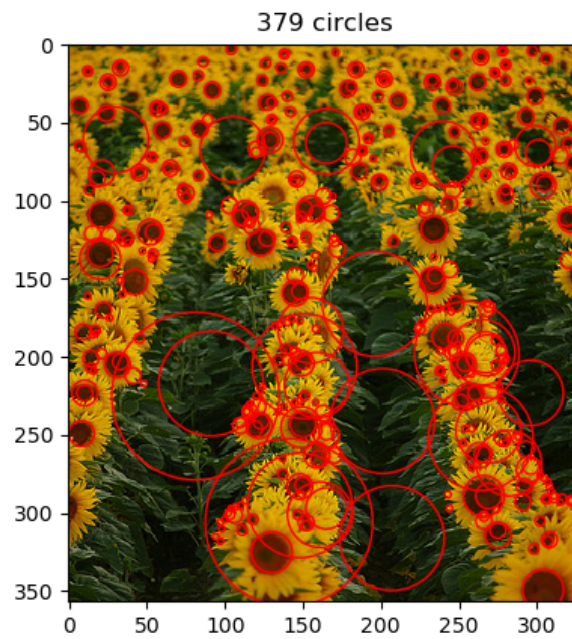
8 Sample images

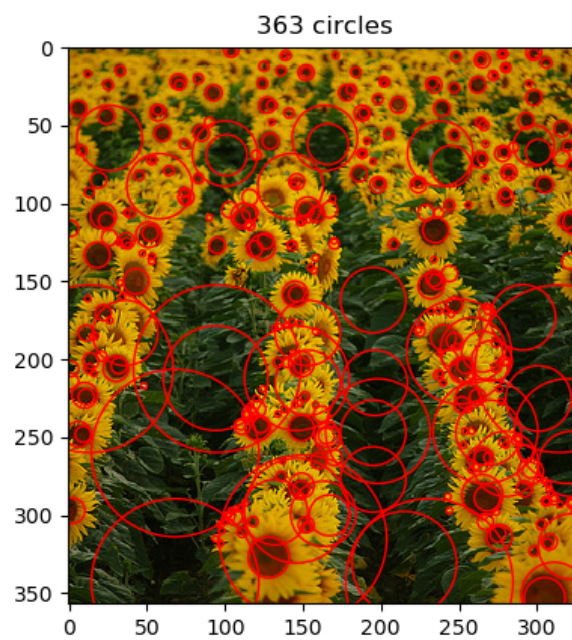Figure 2: *sunflowers.jpg* processed by changing filter size

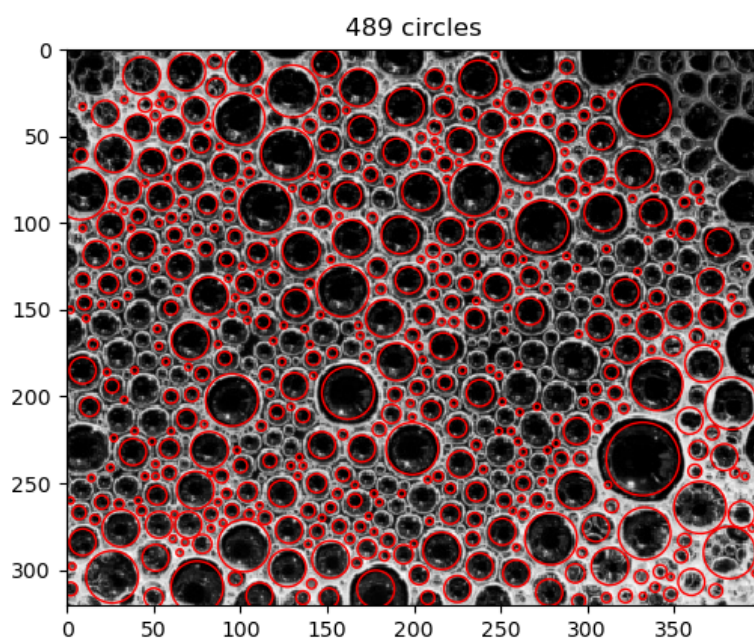Figure 3: *sunflowers.jpg* processed by changing image size

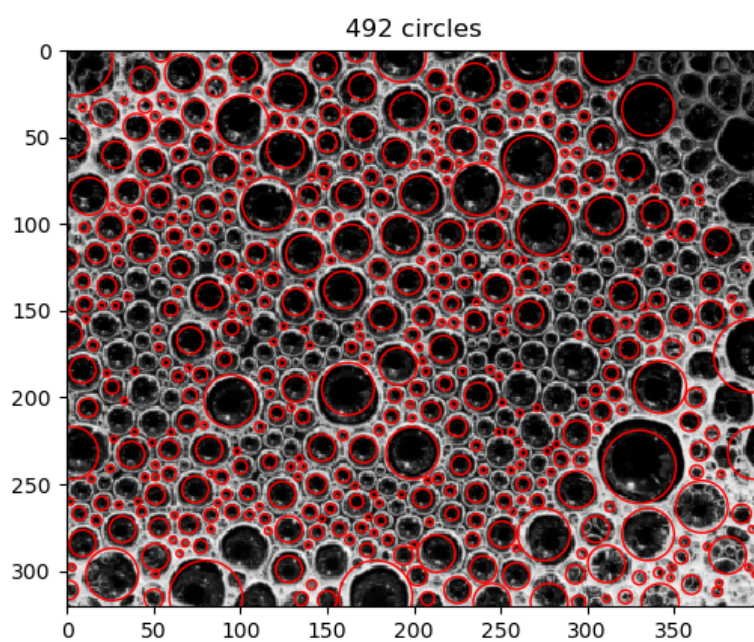Figure 4: *water_texture.jpg* processed by changing filter size

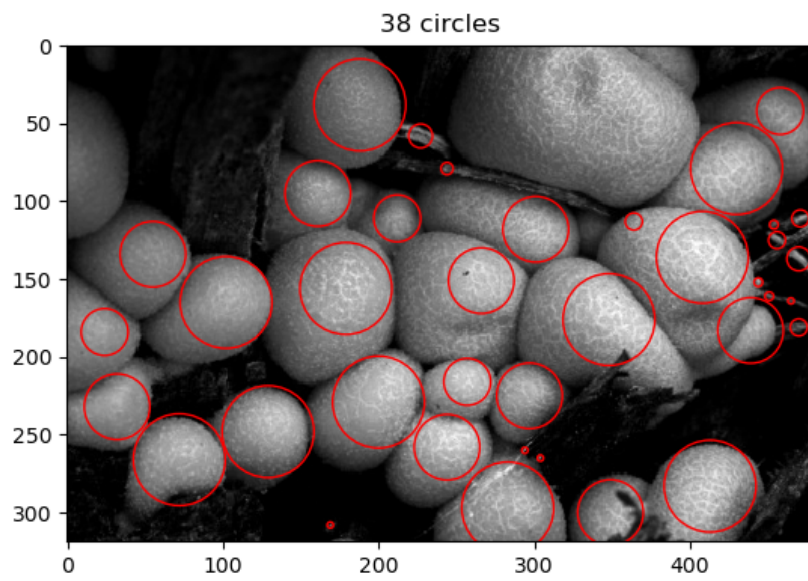Figure 5: *water_texture.jpg* processed by changing image size
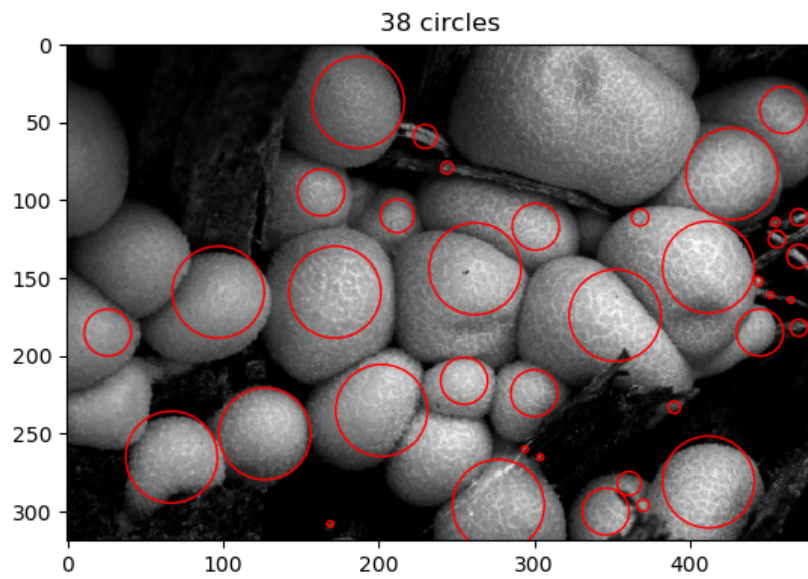
Figure 6: *fruits.jpg* processed by changing filter size

Figure 7: *fruits.jpg* processed by changing image size