Part I

Feature-based image analysis

EXTRACTING IMAGE FEATURES and analyzing images using feature-based representations is central to many applications. We have chosen three topics that we will explore in this part of the course. First, we will work with scale-space for detecting image features independently of scale, with a specific focus on scale-space blob detection. Next, we will work with feature-based image registration. Finally, we will investigate the use of features as a foundation for pixel classification used for image segmentation.

2 *Scale-space*

Methods from scale-space allow scale invariant detection of image structures like blobs (binary large objects), corners, ridges and edges. Here we do not talk about features of depicted objects, but corners and edges in the image intensities. To visualize this, you can think of a 2D image as a landscape, with pixel intensities corresponding to height measurements at regularly placed positions. In this landscape, an edge is a line where high abruptly changes. A corner will be a height-change point where two (more or less) orthogonal edges meet, and other types of features can be described in the same way.

Scale invariance means that we characterize (make a mathematical description) the same feature shown at different scale in two images. This is very convenient in computer vision where images of the same object are often captured from difference distance, and it is desired to be able to measure a feature independent of its scale. But it also allows us to measure image structures that are different in size for example from microscope images, as we will be working with here.

We will base this chapter on the article of Lindeberg[1] that gives an introduction to scale-space theory. This theory is fundamental for a range of image analysis methods and is extensively used in computer vision. In the exercise you will implement scale-space blob detection and use it for detecting and measuring the size of fibres imaged using X-ray CT.

With scale-space, we will represent image features at all scales at once and detect features based on criterion that is independent of the scale. We will work with the Gaussian scale-space, and the scale is achieved by smoothing the image with a Gaussian filter. In Lindeberg[2] the scale-space representation is defined for a general N-dimensional signal $f : \mathbb{R}^N \rightarrow \mathbb{R}$, but we will use it for a 2D image $I : \mathbb{R}^2 \rightarrow \mathbb{R}$. For a 2D image, the Gaussian scale-space representation is $L : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$, which in practice is a 3D object, with the two spatial image dimensions (x, y) and the scale in the third dimension. Since scale is obtained by smoothing with a Gaussian, the variable determining the degree of smoothing is the variance t. Also the standard deviation $\sigma = \sqrt{t}$ is

[1] Tony Lindeberg. Scale-space: A framework for handling image structures at multiple scales. 1996

[2] Tony Lindeberg. Scale-space: A framework for handling image structures at multiple scales. 1996

used in the article, but here we have simplified the notation and use only the variance t.

The Gaussian scale-space L is defined for N-dimensional signals by

$$L(x;t) = \int_{\xi \in \mathbb{R}^N} f(x - \xi) g(\xi; t) d\xi \tag{2.1}$$

with $g : \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}$ being the N-dimensional Gaussian kernel

$$g(x;t) = \frac{1}{(2\pi t)^{N/2}} e^{-(x_1^2 + \cdots + x_N^2)/(2t)}. \tag{2.2}$$

In practice we will work with the Gaussian scale-space for the discrete 2D images. Therefore, we can write the Gaussian scale-space (ignoring boundary issues) as

$$L(x,y;t) = \sum_{-\gamma}^{\gamma} \sum_{-\delta}^{\delta} I(x - \gamma, y - \delta) g(\delta, \gamma; t) \tag{2.3}$$

ß where $g : \mathbb{R}^2 \times \mathbb{R}_+ \to \mathbb{R}$ is the 2D Gaussian kernel

$$g(x,y;t) = \frac{1}{2\pi t} e^{-(x^2 + y^2)/(2t)}. \tag{2.4}$$

Computing the scale-space is done at a discrete set of steps, and we have the start condition with $t = 0$ defined as $L(x,y;0) = I(x,y)$.

For feature detection, we need to compute the derivatives of a scale-space representation. Note that this is conveniently achieved by convolving an image with a kernel that is a derivative of a Gaussian. Blob detection uses second order derivatives, more precisely the Laplacian $\nabla^2 L = L_{xx} + L_{yy}$ which gives a high response where there is a blob in the image. To detect blobs, we need to find local maxima and minima of the Laplacian. Some local maxima and minima will, however, be very weak and should not be detected as blobs. Therefore, only maxima and minima with absolute Laplacian response higher than a certain threshold should be detected as blobs.

We still need to ensure that we detect blobs across different scales. The image in scale-space representation is increasingly smoothed, and with increasing scale t, pixel intensity values will shift towards the average value of the image. Therefore, the absolute values of derivatives will become smaller when increasing t. For blob detection, this means that the magnitude of the local maxima and minima in the Laplacian od of the scale-space $\nabla^2 L$ will decrease and this smoothing must be compensated. The compensation factors for different features are given in Lindeberg[3] and for the blob feature it is t. This means that the scale normalized Laplacian of the scale space is $t\nabla^2 L$.

[3] Tony Lindeberg. Scale-space: A framework for handling image structures at multiple scales. 1996

2.1 Exercise in scale-space blob detection

In this exercise you will implement scale-space blob detection to detect and measure glass fibres from images of a glass fibre composite. An image example in Figure 2.1 shows a polished surface of a glass fibre composite sample, where individual fibres can be seen. Since these fibres are relatively circular we will model them as circles. This means that we must find their position (center coordinate) and diameter, and for this we will use the scale-space blob detection. After having computed the fibres parameters, we will carry a statistical analysis of the results.

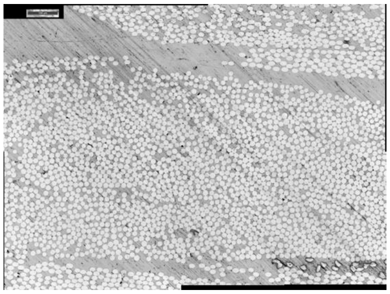

Figure 2.1: Example of fibre image acquired using an optical microscope.

2.1.1 Computing Gaussian and its second order derivative

We will approach this analysis in steps that lead to the final algorithm. First we will use synthetic data to develop and test our algorithm, and after that we will carry out the analysis on the real images.

Since we focus on blob detection, we must have a Gaussian kernel and its second order derivative. Some convolution libraries have already implemented the second order derivative of a Gaussian that you are welcome to use for the exercise. But we will here assume that you need to implement the kernel yourself.

The Gaussian is separable and we can always employ 1D filters, both for the Gaussian, and its derivatives. The 1D Gaussian is given by

$$g(x) = \frac{1}{\sqrt{2\pi t}} e^{\frac{-x^2}{2t}} \ . \tag{2.5}$$

Suggested procedure

1. Derive (analytically) the second order derivative of the Gaussian
$$\frac{\mathrm{d}^2 g}{\mathrm{d}x^2} \ .$$
2. Implement a function that takes the variance t as input and outputs a filter kernel of g and $\mathrm{d}^2 g/\mathrm{d}x^2$. You should use a filter kernel with a size of at least $\pm 3\sqrt{t}$. Why? (*Hint:* Set a variable $r = \lceil 3\sqrt{t} \rceil$, make an array with the integer values $[-r, \dots, r]$ and compute the Gaussian on these values.)
3. Try the filter kernel on the synthetic test image `test_blob_uniform.png` and inspect the result.

2.1.2 Detecting blobs at one scale

Here you will implement a function to detect blobs at a single scale. Blobs can be found as spatial maxima (dark blobs) or minima (bright blobs) of the Laplacian

$$\nabla^2 L = L_{xx} + L_{yy} \ . \tag{2.6}$$

Suggested procedure

1. Compute the Laplacian of the scale-space containing only one scale for the synthetic test image `test_blob_uniform.png`. When using separable 1D kernels to compute L_{xx} and L_{yy}, remember to convolve with second order Gaussian derivative in the one direction, and with the Gaussian smoothing kernel in the other direction. This is because L_{xx} and L_{yy} correspond to derivatives of the image smoothed with 2D kernel.

2. Detect of maxima and minima in the Laplacian. You can make your own function or use `peak_local_max` from `skimage.feature` Detect blobs as maxima or minima that have an absolute value of the Laplacian larger than some threshold.

3. Plot the center coordinates and circles outlining the detected blobs. The radius of the circles should be $\sqrt{2t}$.

4. Try varying t such that the blobs in `test_blob_uniform.png` are exactly outlined.

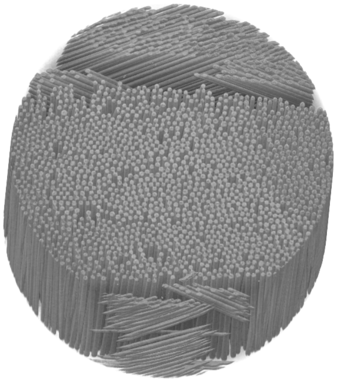

Figure 2.2: Visualization the 3D fibers scanned with the high resolution X-ray CT-scanner.

2.1.3 *Detecting blobs on multiple scales*

Now you should extend the blob detection at a single scale to multiple scales. To find blobs at multiple scales, we must use the scale-space representation. This can conveniently be done by representing the Laplacian of the scale-space $\nabla^2 L$ as a 3D array (volumetric image).

Suggested procedure

1. Decide on a set of scales at which the Laplacian must be computed. A good idea is to make it at equal steps in t. Remember that the radius of the blobs is $\sqrt{2t}$, so you can visually estimate the size of the structures that you want to detect and decide a good range of scales.

2. Compute the scale normalized scale-space Laplacian $t\nabla^2 L$ for the test image `test_blob_uniform.png`. It is very important that you remember to use scale normalization, i.e. that you multiply the Laplacian $\nabla^2 L$ by t. Otherwise, you will not detect the correct scales. See Figure 2.3 for a sketch of an efficient implementation.

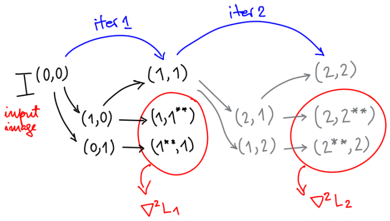

Figure 2.3: The sketch of an efficient implementation of the scale-space representation using 1D kernels. Each tuple indicates the amount of smoothing for the two directions, while the star indicates derivative. Using five 1D convolutions, each iteration gives a new scale.

3. Find coordinates and scales of maxima and minima in this scale-space and plot the detected blobs on top of the image. What are the detected scales and what is the diameter of the blobs? Remember that a pixel value needs to be larger (or smaller) then neighbors in both spatial dimensions and in the scale dimension. (*Hint:* To ensure that extrema is correctly detected in the first and the last scale, you can pad the scale-space representation with zeros.)

4. Detect blobs in the test image `test_blob_varying.png`.

5. Verify that you detected the blobs at the correct scale by showing an image with the detected blobs plotted on top as circles with a diameter of $\sqrt{2t}$.

2.1.4 Detecting blobs in real data

We will now continue with the real images of fibers. The fibre data is obtained using different scanning methods including scanning electron microscopy (`SEM.png`), optical microscopy (`Optical.png`), synchrotron X-ray CT (`CT_synchrotron.png`), and three resolutions of laboratory X-ray CT (`CT_lab` images with high, medium or low resolution. The CT data is a single slice very close to the top of the sample, so we assume the images to be showing the same structures, and this allows us to directly compare the fibers. We will do this comparison in later exercise, but in this we will compute the fiber location and their diameter. In Figure 2.2 you can see a visualization of the fibre data from the high resolution X-ray CT scan.

We start by testing the blob-detection on this real data.

Suggested procedure

1. Run your blob-detection function from above on a cut-out example of one of the images. It is important that you tune your parameters to get the best possible results.

2.1.5 Localizing blobs

It turns out, that it is difficult to detect blobs in the Laplacian scale-space in the fiber image, such that all fibers are found. To overcome this, we will detect the fibers as maxima in a Gaussian smoothed image. Since the fibers are almost the same size, we can use a single scale of the Gaussian to detect the fiber centers.

Suggested procedure

1. Smooth an image of fibers with a Gaussian and visualize the result.

2. Find locations of maxima in this image and plot the positions on top of the original image.

3. Compute the Laplacian scale-space for the image.

4. Find the scale of each fibre as the minimum over scales at the fiber locations.

5. Plot circles according to the found scale on top of the original image.

6. Detect fibers in all six fiber images. Save the locations and diameters.

In the exercise in Week 3, you will work with feature-based image registration, and you be using the results obtained in this exercise. So, in Week 3 it will be possible to continue working on the parts that you did not finish here.