Laplace of Gaussian

Targets of Exercise:

General

- Derivation of the LoG filter
- Implementation of LoG Filter
- Finding an alternative implementation

Solution for task 9.1a

Task: *Derivation of the Laplacian of Gaussian filter.*

The function is:

$$w(r) = -e^{-\frac{r^2}{2\sigma^2}} = -e^{-\frac{x^2+y^2}{2\sigma^2}} \text{ with } \sigma \text{as the standard deviation}$$

$$\nabla^2 w(r) = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}$$

$$\frac{\partial w}{\partial x} = -e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \frac{x}{\sigma^2}$$

in order to calculate $\frac{\partial^2 w}{\partial x^2}$, we apply the chain rule $(f \cdot g)' = f' \cdot g + f \cdot g'$

$$\begin{aligned} f &= \frac{x}{\sigma^2} \\ g &= e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 w}{\partial x^2} &= \frac{1}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{x}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \left(-\frac{x}{\sigma^2}\right) \\ &= \frac{1}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{x^2}{\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

the same:

$$\frac{\partial^2 w}{\partial y^2} = \frac{1}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{y^2}{\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}}$$

hence

$$\begin{aligned} \nabla^2 w(r) &= \frac{1}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{x^2}{\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{1}{\sigma^2} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{y^2}{\sigma^4} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \\ &= e^{-\frac{x^2+y^2}{2\sigma^2}} \left(\frac{1}{\sigma^2} - \frac{x^2}{\sigma^4} + \frac{1}{\sigma^2} - \frac{y^2}{\sigma^4}\right) \\ &= e^{-\frac{x^2+y^2}{2\sigma^2}} \left(\frac{2}{\sigma^2} - \frac{x^2}{\sigma^4} - \frac{y^2}{\sigma^4}\right) \\ &= e^{-\frac{x^2+y^2}{2\sigma^2}} \left(\frac{2\sigma^2 - (x^2+y^2)}{\sigma^4}\right) \end{aligned}$$

Solution for task 9.1b

Task: *Implement a function that generates and stores the following LoG filter.*

The derivation of the filter in the 2D space is done for x and y. The following funktion should be implemented.

$$\nabla^2 w(x,y) = \frac{2\sigma^2-(x^2+y^2)}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Definition of interfaces:

```
// Creates a Laplacian of Gaussion (LoG) filter from the
// mask <mask> with a standard deviation <sigma>
void maskLoG (GrayImage& mask, float sigma);
```

Implementation:

```
//
// (8.1b)
//
void maskLoG (GrayImage& mask, float sigma)
{
    int     width   = mask.getWidth();
    int     height  = mask.getHeight();
    float*  data    = mask.getData();

    int     x0      = width / 2;
    int     y0      = height / 2;
    float   s2      = sigma * sigma;
    float   s4      = sigma * sigma * sigma * sigma;

    for (int y=0; y<height; y++)
    {
        for (int x=0; x<width; x++)
        {
            int x1    = x-x0;
            int y1    = y-y0;
            float exp = (float)(x1*x1 + y1*y1);

            data[y*width+x] =  ((2*s2 - exp)/s4) * expf(-exp/(2*s2));
        }
    }
}
```

Solution for task 9.1c

Task: *Implement the alternative function which is same as the LoG filter but consists of multiple filters. In this case, make use of the given filters i.e. Laplace (*laplace.mask*) and Gauss (*gauss.mask*).*

An alternative implementation is the sequential execution of the Gaussian and Laplacian filter. Therfore the image will be filtered first with the Gaussian filter and then, the resulting image will be filtered with the Laplacian filter. This method is slower than the LoG filter, because a complete image has to be filtered twice, but it delivers the same results.

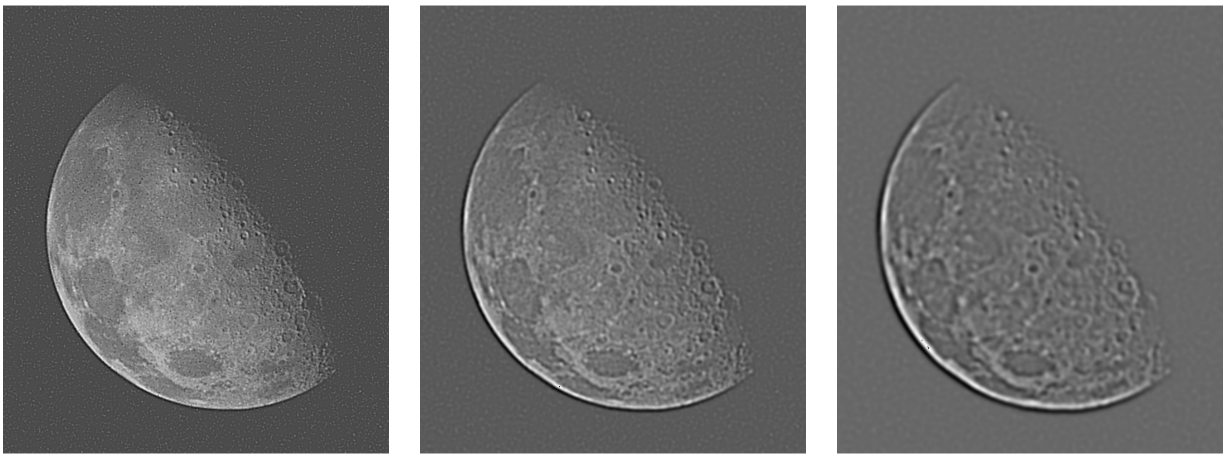
Alternative Implementation:

```
GrayImage laplace;
GrayImage img;

string filename;
cout << "Dateiname des Gauss Filters: ";
cin >> filename;
loadFilterMask(filename, gauss);

cout << "Dateiname des Laplace Filters: ";
cin >> filename;
loadFilterMask(filename, laplace);

cout << "Dateiname des Bildes: ";
cin >> filename;
img.load(filename);

GrayImage result_gauss(img.getWidth(), img.getHeight());
GrayImage result_laplace(img.getWidth(), img.getHeight());

filter(img, result_gauss, gauss);
filter(result_gauss, result_laplace, laplace);

scale(result_laplace);

result_laplace.show();
result_laplace.save();
```

Solution for task 9.1d

Task: *Test the function with of the given image mond_noise.bmp.*

Fig.1 shows the image with different filter sizes s and σ.



(a) (b) (c)

Abbildung 1: Rsults of filtering with different filter sizes, (a) $s = 5\times5$ with $\sigma = 1$, (b) $s = 11\times11$ with $\sigma = 2$, (c) $s = 21 \times 21$ with $\sigma = 3$