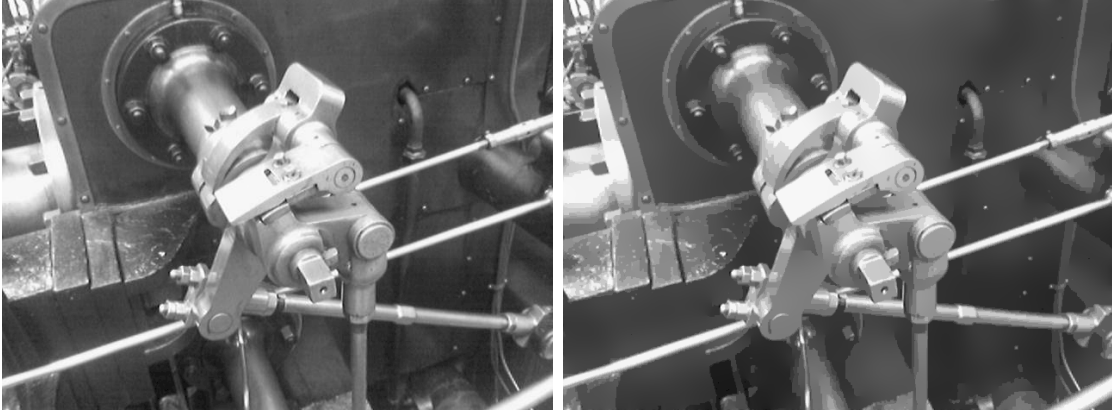# Anisotropic Filtration

Figure 1: Input image (*left* image); filtered image after 1000 iterations (*right* image).

Today's exercise is focused on the implementation of the anisotropic filtering of images. In contrast to the Gaussian blur, anisotropic filtration gives us filtered image that still contains sharp edges. This filtration is based on simple physical phenomenon of spreading an energy from higher concentrations to lower ones. In images, energy concentration can be, for example, the brightness value in each pixel. All the pixels are placed in a grid that forms a mesh network. Neighbour pixels are connected with each other by resistors and their resistance (or conductance) is based on their similarity. The filtration then proceeds in time. In each time step, small amounts of energy flows between pixels. After some predefined number of time steps, the iteration stops.

A model of a pixel neighborhood is depicted in Fig. 2. Conductances between pixels can be computed as follows

$$
\begin{aligned}
c_{N_{i,j}}^t &= g\left(\left\|\nabla_N I_{i,j}^t\right\|\right) \\
c_{S_{i,j}}^t &= g\left(\left\|\nabla_S I_{i,j}^t\right\|\right) \\
c_{E_{i,j}}^t &= g\left(\left\|\nabla_E I_{i,j}^t\right\|\right) \\
c_{W_{i,j}}^t &= g\left(\left\|\nabla_W I_{i,j}^t\right\|\right),
\end{aligned}
\tag{1}
$$

where g is defined as follows

$$
g(\nabla I) = e^{\left(-\frac{|\nabla I|^2}{\sigma^2}\right)}
\tag{2}
$$

and $\nabla_N I_{i,j} = I_{i,j-1} - I_{i,j}$. For other directions, it works the same way (see Fig. 2).

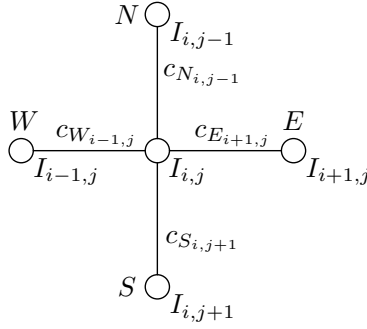In each iteration step, a new value in a pixel is computed using the following formula

Figure 2: A model of a pixel at coordinates i, j with *north* (N), *south* (S), *west* (W), and *east* (E) neighbors.

$$I_{i,j}^{t+1} = I_{i,j}^{t}\left(1 - \lambda\left(c_N + c_S + c_E + c_W\right)_{i,j}^{t}\right) + \lambda\left(c_N \cdot I_N + c_S I_S + c_E I_E + c_W I_W\right)_{i,j}^{t}, \quad (3)$$

where $I_{i,j}^{t+1}$ is a new brightness value at coordinate i, j at time $t + 1$ and $I_{i,j}^{t}$ is an old brightness value at coordinate i, j at time t. Using the Eq. (3) in each pixel, we compute new values at time $t + 1$ based on the values at time t. Do not forget that this is not an in-place operation. You have to compute new values to a new image.

Set $\sigma = 0.015$ and $\lambda = 0.1$ for your experiments.

Hint: Use `double` data type to represent the input and output images.