

Segmentation and Image Analysis - Practical work 4

This work aims at computing the euclidean distance map of a 3D digital object : each point is labeled with its distance to the background. You will implement an algorithm which is based on the spread of the distance information in the 3D volume (Sequential Euclidean Distance).

1. *Data structure :*

Each voxel stores the vector (d_x, d_y, d_z) which links it to the nearest background voxel. To limit the computation time, the squared norm of this vector can also be stored ($d_2 = d_x^2 + d_y^2 + d_z^2$). The background voxels are initialized with the values ($d_x = 0, d_y = 0, d_z = 0, d_2 = 0$). At the beginning, the other voxels contain ($d_x = \infty, d_y = \infty, d_z = \infty, d_2 = \infty$), in other words any value meaning that they have not been processed.

2. *Spread of the distance information :*

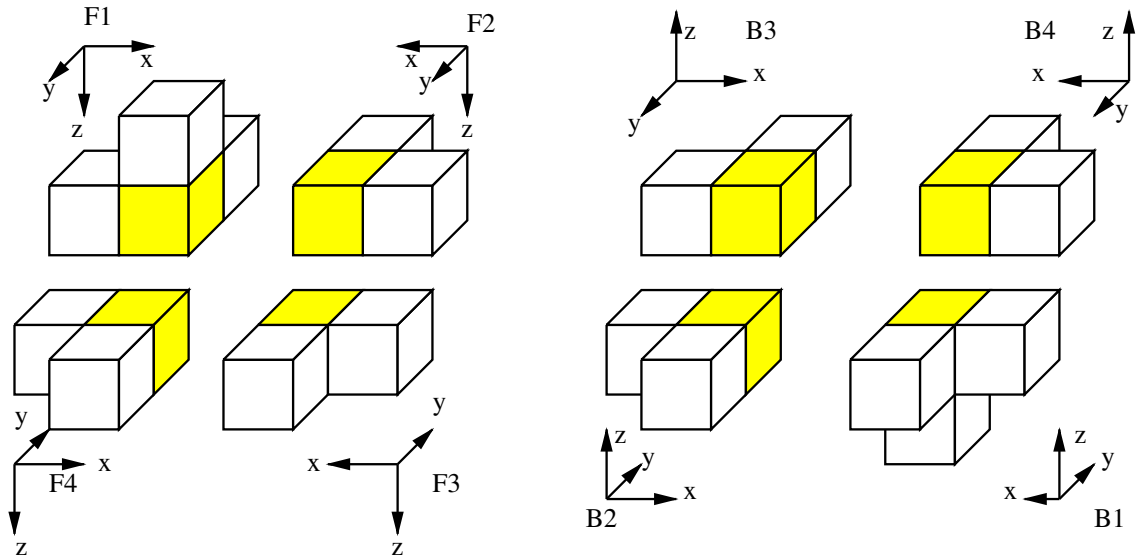
During the volume processing, which is done in several passes, each voxel reads the distance values of its 6-neighbors to update its own nearest background voxel. Let $p = (x, y, z)$ be a voxel of the volume and $q = (x + \delta_x, y + \delta_y, z + \delta_z)$ be a neighbor of p . The distance from p to the nearest background voxel of q can be computed as : $(q.d_x + \delta_x)^2 + (q.d_y + \delta_y)^2 + (q.d_z + \delta_z)^2$. If $p.d_2$ (distance stored at voxel p) is greater than the computed distance, it has to be updated.

3. *Passing through the volume :*

The first principal pass through the volume is done according to the increasing z coordinates (F for forward in the following). For each slice, 4 sub-steps are carried out (F1 to F4 on the figure). The steps F1 and F2 are performed according to increasing ordinates. The step F1 is done on a line according to increasing abscissae : for each point (x, y, z) on the line, 3 neighbors are tested $(x, y, z - 1)$, $(x, y - 1, z)$ and $(x - 1, y, z)$. Then the step F2 is carried out on the same line, according to decreasing abscissae : for each point (x, y, z) on the line, 2 neighbors are tested $(x + 1, y)$ and $(x, y - 1, z)$. In the same way, the steps F3 and F4 correspond to a backward pass followed by a forward pass on each line (according to decreasing ordinates).

The second principal pass through the volume is done according to the decreasing z coordinates (B for backward in the following). For each slice, 4 sub steps are performed (B1 to B4 on the figure).

Be aware that the coordinate axes are fixed in the figure (x to the right, y forward, z down) and the arrows indicate the direction for iterating on the volume.



To do

Some test images are given : `1-reference-point.tif` (small data volume with only one reference point - the isosurfaces of the distance map should be spheres), `2-reference-points.tif` (larger data volume with 2 reference point), `hourglass.tif`, `flower.tif`. You can also use the image `Bat Cochlea Volume` available in the *ImageJ Samples*.

1. Write an `ImageJ` plugin processing a binary 3D image that creates, computes and displays a distance map (the black points will be the reference points). Distinguish the computed distance values and the gray levels chosen for displaying (distance values normalized between 0 and 255). The result can be visualized as a sequence of slices or with the `3D Viewer` plugin. If you visualize an isosurface of the distance map with `3D Viewer`, the points that are considered to be inside the volume defined by the surface are those with a value greater than the threshold given as parameter.
2. How to characterize the points (or some points) of the skeleton of a 3D binary shape from its distance map? Make an argued proposal and implement it. Test on the images `hourglass.tif`, `flower.tif` et `Bat Cochlea Volume`.

You will submit your work by mail to `vialard@labri.fr`. It will include :

- your source code with comments
- the answer to question 2.
- result images