Pseudo-code for Boundary Extraction for 3D binary image by Distance Transformation Algorithm

Developed by Astushi Imiya & Ken Tatara

Boundary Extraction by Distance Transformation

INPUT : $N \times N \times N$ binary image $\{P(i,j,k)\}$.
/$P(i,j,k) = 0$ is a background voxel and $P(i,j,k) = 1$ is an object voxel /

OUTPUT : $N \times N \times N$ binary image $B = B(i,j,k)$.
/B is a boundary of image./

/move image/
1:procedure **TRANCELATE-ON-ORTHOGONAL-DIRECTIONS**
2: **begin**
3: input $P(i,j,k)$
4: **for** $i = 0,1,2,...,N, j = 0,1,2,...,N, k = 0,1,2,...,N$ **do**
5: **if** $P(i,j,k) = 1$
6: **then** $P_m(i,j,k) = P(i + move_x, j + move_y, k + move_z)$
$/move_x = -1,0,1, move_y = -1,0,1, move_z = -1,0,1/$
7: output $\{P_m(i,j,k) = 1\}$
8: **end**

/extract inner and outer boundary/
1:procedure **BOUNDARY-EXTRACTION**
2: **begin**
3: input $P(i,j,k)$
4: input $P_m(i,j,k)$
5: **for** $i = 0,1,2,...,N, j = 0,1,2,...,N, k = 0,1,2,...,N$ **do**
6: $B_{inner} = \bigcup\{P \backslash (P \cap P_m)\}$ /B_{inner} is INNER Boundary/
7: $B_{outer} = \bigcup\{(P \cap P_m) \backslash P\}$ /B_{outer} is OUTER Boundary/
8: output $\{B_{inner}\}$
9: output $\{B_{outer}\}$
10: **end**

/boundary extraction/

1:procedure **BOUNDARY-EXTRACTION**

2: **begin**

3: **begin**

4: **TRANCELATE-ON-ORTHOGONAL-DIRECTIONS**

5: **end**

6: **begin**

7: **BOUNDARY-EXTRACTION**

8: **end**

11: output $B = B_{inner} \cup B_{outer}$

12: **end**