

Fast local thickness

Vedrana Andersen Dahl and Anders Bjorholm Dahl

Abstract—We propose a fast algorithm for computation of local thickness in 3D and 2D. Our fast algorithm computer local thickness very similar to conventional algorithm, in just a fraction of time. Our algorithm is implemented in python and is freely available as a pip-installable module.

I. BACKGROUND

Contemporary high-resolution scanners typically produce volumes containing 2048^3 voxels, or even 4096^3 voxels. With volumes containing billions of voxels even the simplest processing methods may be computationally demanding. This is especially pronounced when using a kernel which also grows cubically in size; And processing large images often requires using large kernels.

For this reason, efficient 3D analysis requires re-visiting processing algorithms. In this paper, we consider the computation of local thickness. As we show, time complexity of conventional local thickness algorithm is such that if it takes 1 second to process a volume of size 512^3 , processing a 2048^3 volume takes 4.5 hours and a 4096^3 volume takes 24 days to process.

II. INTRODUCTION

For an object in 3D, the local thickness in any point of the object is defined as the radius of the largest sphere which fits inside the object and contains the point, see Fig. 1. Despite this simple definition, computing local thickness using a direct implementation of the definition is computationally demanding, and for large volumetric data computationally infeasible.

While there are a few efficient alternatives to the conventional approach [4][5], we propose an algorithm which is even faster, and only slightly reduces the quality of computed local thickness.

Our algorithm utilizes two approximations for a more efficient computation. First, the dilation with the sphere of a large radius is approximated with the consecutive dilations with the sphere of radius one. This results in a dilation not being perfectly spherical, but polygonal (polyhedral). Second (and optional), we use scaled approach, where the local thickness is computed for a downscalled image, then upscaled, and corrected.

III. RELATED WORK

BoneJ [4] simplifies the search set to the distance ridge. This leaves small discrepancy at the surface, which requires cleanup. The result is practically identical to the result of the conventional algorithm.

Porespy [5] use a range of dilate values, and morphological operations are implemented to utilize fft-based filtering. The

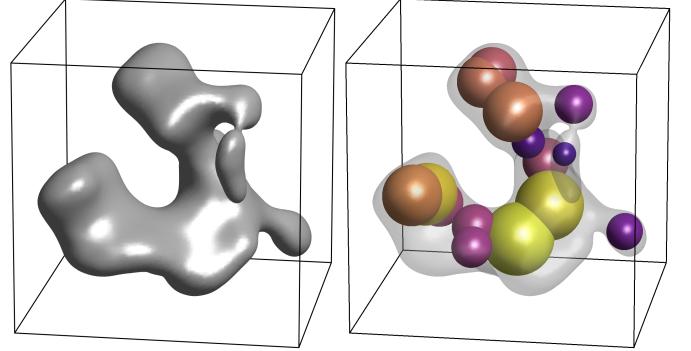


Fig. 1. *Left:* An example of an object in 3D which has local thickness. *Right:* A few spheres of different radius which can be fitted inside the object, depending on its local thickness. Every point inside, say, the orange sphere has the local thickness which is equal or larger than the radius of the sphere.

result is less smooth, but comparable to the result of the conventional algorithm.

VAND: Other related papers: [1] 3700 citations, [2] 589 citations, [3] 245 citations, [7] 181 citations, [6] 1895 citations, [8] 11 citations, [9] 85 citations, [10] 583 citations.

IV. CONVENTIONAL ALGORITHM FOR LOCAL THICKNESS

For an object in 3D, the local thickness in any point of the object is defined as the radius of the largest sphere which fits inside the object and contains the point.

In other words, for an object $A \subset \mathbf{R}^3$ and a point $\mathbf{x} \in A$, local thickness

$$t(\mathbf{x}) = \max\{r \in \mathbf{R} : \mathbf{x} \in S(\mathbf{c}, r) \subseteq A, \mathbf{c} \in \mathbf{R}^3\}, \quad (1)$$

where

$$S(\mathbf{c}, r) = \{\mathbf{p} \in \mathbf{R}^3 : \|\mathbf{c} - \mathbf{p}\|_2 \leq r\}. \quad (2)$$

Conventional algorithm for local thickness involves distance field and dilation with spherical structuring elements. How the definition of local thickness (1) may be expressed in terms of distance field and dilation, is sketched below.

Distance field is defined as

$$d(\mathbf{x}) = \max\{r \in \mathbf{R} : S(\mathbf{x}, r) \subseteq A\}. \quad (3)$$

For a point $\mathbf{x} \in S(\mathbf{c}, d(\mathbf{c}))$ we know that $t(\mathbf{x}) \geq d(\mathbf{c})$. So we have

$$t = \max\{d(\mathbf{c}) I_{S(\mathbf{c}, d(\mathbf{c}))}, \mathbf{c} \in A\} \quad (4)$$

where $I_S : \mathbf{R}^3 \rightarrow \mathbf{R}$ is an indicator function of $S \subset \mathbf{R}^3$ and local thickness t is extended to \mathbf{R}^3 , having value 0 for points outside A .

Since

$$I_{S(\mathbf{c}, d(\mathbf{c}))} = I_{\mathbf{c}} \oplus S_{d(\mathbf{c})}, \quad (5)$$

where $S_{d(\mathbf{c})}$ is a sphere of radius $d(\mathbf{c})$ centered at origin, we have

$$t = \max\{d(\mathbf{c}) I_{\mathbf{c}} \oplus S_{d(\mathbf{c})}, \mathbf{c} \in A\}. \quad (6)$$

If we now consider points where d has value larger than r by defining

$$d_r(\mathbf{x}) = \begin{cases} d(\mathbf{x}) & d(\mathbf{x}) \geq r \\ 0 & \text{elsewhere} \end{cases} \quad (7)$$

we know that

$$t \geq d_r \oplus S_r, \text{ for every } r \in \mathbf{R}, \quad (8)$$

so we have

$$t = \max\{d_r \oplus S_r, r \in \mathbf{R}\}. \quad (9)$$

This is an operative approach to local thickness leading to Alg. 1, conventional local thickness algorithm. It is conducted on discretized distance field with $h \in \{1, \dots, \text{floor}(\max(d))\}$. In the algorithm, we use square brackets to denote boolean (logical) indexing into the array, so $a[b]$ means: values of a where b has values True. Furthermore, we expect the distance transform to be computed, so algorithm takes distance field as input.

Algorithm 1 Conventional local thickness

```

1: function CONVENTIONAL(df)
2:   out  $\leftarrow$  df
3:   for r = 1 to floor(max(df)) do
4:     df[df < r]  $\leftarrow$  0
5:     temp  $\leftarrow$  df  $\oplus$  sphere(r)
6:     change  $\leftarrow$  temp > out
7:     out[change]  $\leftarrow$  temp[change]
8:   return out

```

A. Variants of the conventional algorithm

In Alg. 1 line 5 may check whether temp is larger than r, or larger than 0. The algorithm may also be formulated by starting with the largest r. In that case dilation needs to operate on more values of df in each iteration so line 4 should be modified.

B. Time complexity of the conventional algorithm

Consider a 3D object occupying a cube as in Fig. 1. Denote the side length of the cube as X and the largest local thickness of the object as S . If the structure is imaged such that length X is divided in x pixels, the thickness of the object in pixels is $s = \frac{S}{X}x$. The conventional computation of local thickness includes a loop

```

for r = 1 to s do
  df  $\oplus$  sphere(r)

```

Computation time complexity of dilation is linear in respect to the number of voxels to be processed, in this case x^3 , and the size of the structuring element, here r^3 . That is, dilation takes $\mathcal{O}(x^3r^3)$ per iteration. On the other hand, the number

volume size x^3	512^3	1024^3	2048^3	4096^3
processing time $\mathcal{O}(x^7)$	1 sec	2 min	4.5 h	24 days

TABLE I
THE INCREASE IN PROCESSING TIME, ASSUMING THAT IT TAKES 1 SEC TO PROCESS THE SMALLEST VOLUME.

of iterations s grows linearly with x . The time complexity of the loop is therefore

$$\mathcal{O}(x^3) \sum_{r=1}^{\mathcal{O}(x)} \mathcal{O}(r^3) = \mathcal{O}(x^3)\mathcal{O}(x^4) = \mathcal{O}(x^7).$$

This leads to rapid increase in processing time as volume size grows, see Tab. I.

V. FAST LOCAL THICKNESS

In the conventional algorithm, the dilation always operates on the (subset of) values from the distance field, and the radius of the spherical structuring element is increasing in each iteration. However, by utilizing the property of the distance field, and the commutativity of dilation, it is possible to formulate algorithms which use the sphere of radius 1 in every iteration. To obtain the result as with the conventional algorithm, the dilation operates on the result of the previous iteration. This way, instead of dilating with a sphere of radius n , we use n consecutive dilations with a sphere of radius 1. See Alg. 2 for one variant of this approach.

In a continuous case, or in a 1D setting ([VAND: Add figure with 1D example](#)), the conventional and the fast algorithm yield the same result. The problem with this approach is that in discrete setting in 2D and 3D, due to the shape of the discrete 1-sphere, a dilation with a large radius cannot be achieved by many dilations with a 1-sphere.

Still, by carefully choosing the structural elements when computing dilation, we can achieve a result where consecutive dilation yields a polyhedron (polygon in 2D) which approximates a sphere.

Therefore, we formulate the dilation with a 1-sphere as a weighted sum of dilations with simple structuring elements.

Algorithm 2 Fast local thickness

```

1: function FAST(df)
2:   out  $\leftarrow$  df
3:   for r = 0 to floor(max(df)) do ▷ Use DilateOne
4:     temp  $\leftarrow$  out  $\oplus$  sphere(1)
5:     change  $\leftarrow$  out > r
6:     out[change]  $\leftarrow$  temp[change]
7:   return out

```

Algorithm 3 Dilate with sphere(1)

```

1: function DILATEONE(v)
2:   selemi = selems for 2D or 3D
3:   wi = weights for 2D or 3D
4:   out  $\leftarrow$   $\sum_i w_i v \oplus$  selemi
5:   return out

```

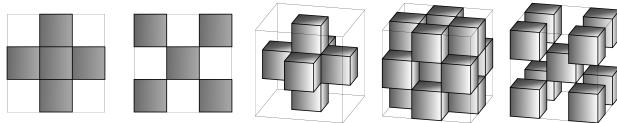


Fig. 2. Structuring elements for fast local thickness in 2D and 3D.

A. Structuring elements

The structuring elements used for our algorithm have kernel size length of 3, so smallest possible kernel. In 2D, the structuring elements are a discrete disc and annuli of growing radii. In 3D, it is spheres and spherical shells. We make sure that the central pixel (voxel) is always set to one. Algorithms are 4 and 5. Here $\text{sphere}(r)$ is structuring element with ones in all pixels with distance to kernel center being smaller or equal to r . **VAND: Maybe rather in text?** The resulting structuring elements may be seen in the Fig. 2.

The weights w_i determine the shape of the non-binary discrete approximation of the circle. We weight the structuring elements by the inverse of their largest radius. So the element involving $\text{sphere}(\sqrt{3})$ would be weighted proportional to $1/\sqrt{3}$. Since we normalize the weights to sum to 1, we arrive to expressions in 4 and 5. Our results show that this weighting yields a good approximation. Still, it is worth noticing that this weighing is slightly arbitrary. It turns out that weights affect the shape of the resulting approximation in nontrivial way, and the quality of the approximation is difficult to assess.

VAND: Add the result of consecutive dilation in 2D and 3D.

Algorithm 4 Selems and weights for 2D

- 1: $\text{selem}_1 = \text{disk}(1)$
 - 2: $\text{selem}_2 = \text{disk}(\sqrt{2}) - \text{disk}(1) + \text{disk}(0)$
 - 3: $w_1 = \frac{\sqrt{2}}{1+\sqrt{2}}$
 - 4: $w_2 = \frac{1}{1+\sqrt{2}}$
-

Algorithm 5 Selems and weights for 3D

- 1: $\text{selem}_1 = \text{sphere}(1)$
 - 2: $\text{selem}_2 = \text{sphere}(\sqrt{2}) - \text{sphere}(1) + \text{sphere}(0)$
 - 3: $\text{selem}_3 = \text{sphere}(\sqrt{3}) - \text{sphere}(\sqrt{2}) + \text{sphere}(0)$
 - 4: $w_1 = \frac{\sqrt{6}}{\sqrt{6}+\sqrt{3}+\sqrt{2}}$
 - 5: $w_2 = \frac{\sqrt{3}}{\sqrt{6}+\sqrt{3}+\sqrt{2}}$
 - 6: $w_3 = \frac{\sqrt{2}}{\sqrt{6}+\sqrt{3}+\sqrt{2}}$
-

B. Variants of the fast algorithm

In Alg. 2 dilation in line 4 operates on out . It would however be enough to consider only values of out which are larger than r , as only those values may be changed. Similar to the conventional algorithm the fast algorithm may also be formulated such that it starts with the largest r .

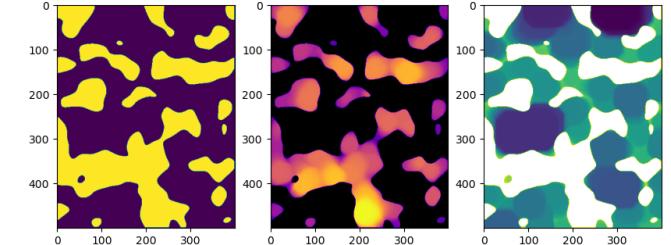
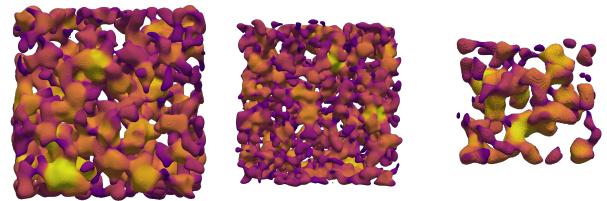


Fig. 3. A placeholder for some results.

C. Time complexity of our algorithm

The computational complexity of our method is constant for the size of the structuring element 1^3 , that is $\mathcal{O}(x^3 1^3)$ per iteration, therefore we obtain

$$\mathcal{O}(x^3) \sum_{i=1}^{\mathcal{O}(x)} 1^3 = \mathcal{O}(x^3) \mathcal{O}(x) = \mathcal{O}(x^4).$$

D. Local thickness with scaling

In continuous setting, shrinking the object with a factor, say 2, would result in local thickness being reduced everywhere by the same factor.

This can be utilized to speed up the computation, with any algorithm. **VAND: Describe.**

We add cleanup.

E. Code

<https://github.com/vedranaa/local-thickness>

VI. RESULTS

See Fig. 3 and notebooks.

VAND: Copy results from notebooks: 2D notebook 3D notebook

VII. CONCLUSION

VAND: The conclusion goes here. May also be copied from notebooks.

REFERENCES

- [1] Mary L Bouxsein, Stephen K Boyd, Blaine A Christiansen, Robert E Gulberg, Karl J Jepsen, and Ralph Müller. Guidelines for assessment of bone microstructure in rodents using micro-computed tomography. *Journal of bone and mineral research*, 25(7):1468–1486, 2010.
- [2] Helen R Buie, Graeme M Campbell, R Joshua Klinck, Joshua A MacNeil, and Steven K Boyd. Automatic segmentation of cortical and trabecular compartments based on a dual threshold technique for in vivo micro-ct bone analysis. *Bone*, 41(4):505–515, 2007.

- [3] David ML Cooper, Andrei L Turinsky, Christoph Wilhelm Sensen, and Benedikt Hallgrímsson. Quantitative 3d analysis of the canal network in cortical bone by micro-computed tomography. *The Anatomical Record Part B: The New Anatomist: An Official Publication of the American Association of Anatomists*, 274(1):169–179, 2003.
- [4] Robert Dougherty and Karl-Heinz Kunzelmann. Computing local thickness of 3d structures with imagej. *Microscopy and Microanalysis*, 13(S02):1678–1679, 2007.
- [5] Jeff T Gostick, Zohaib A Khan, Thomas G Tranter, Matthew DR Kok, Mehrez Agnaou, Mohammadamin Sadeghi, and Rhodri Jervis. Porespy: A python toolkit for quantitative analysis of porous media images. *Journal of Open Source Software*, 4(37):1296, 2019.
- [6] Tor Hildebrand and Peter Rüegsegger. A new method for the model-independent assessment of thickness in three-dimensional images. *Journal of microscopy*, 185(1):67–75, 1997.
- [7] Delphine Maret, Norbert Telmon, Ove A Peters, B Lepage, J Treil, JM Inglèse, A Peyre, Jean-Luc Kahn, and Michel Sixou. Effect of voxel size on the accuracy of 3d reconstructions with cone beam ct. *Dentomaxillofacial Radiology*, 41(8):649–655, 2012.
- [8] D Müter, HO Sørensen, Jette Oddershede, KN Dalby, and SLS Stipp. Microstructure and micromechanics of the heart urchin test from x-ray tomography. *Acta Biomaterialia*, 23:21–26, 2015.
- [9] Punam K Saha, Robin Strand, and Gunilla Borgefors. Digital topology and geometry in medical imaging: a survey. *IEEE transactions on medical imaging*, 34(9):1940–1964, 2015.
- [10] Naoya Taniguchi, Shunsuke Fujibayashi, Mitsuru Takemoto, Kiyoyuki Sasaki, Bungo Otsuki, Takashi Nakamura, Tomiharu Matsushita, Tadashi Kokubo, and Shuichi Matsuda. Effect of pore size on bone ingrowth into porous titanium implants fabricated by additive manufacturing: an in vivo experiment. *Materials Science and Engineering: C*, 59:690–701, 2016.