



# A Micro-computed Tomography Database and Reference Implementation for Parallel Computations of Trabecular Thickness and Spacing

ISSUES IN RESEARCH  
SOFTWARE

THI-NGOC-THU NGUYEN

ANDREAS HÖFTER

KEVIN LEONARDIC

STEFANIE ROSENHAIN

FABIAN KIESSLING

WANNIDA SAE-TANG

UWE NAUMANN

FELIX GREMSE

*\*Author affiliations can be found in the back matter of this article*

][ubiquity press

## ABSTRACT

We provide a well-defined accurate brute force implementation without approximations as a reference to support future development of correct and fast algorithms to measure the trabecular thickness and spacing. Using artificial ellipsoid examples, the systematic error is shown to be related to the voxelization, which can be reduced by upsampling. Furthermore, we collected a database of 40 three-dimensional micro-computed tomography images of trabecular bone regions of a sheep femur. With the coverage of a broad range of trabecular thickness and spacing, the database is suitable for assessing the correctness of algorithms for the measurements of trabecular thickness and spacing. The bone data is shared publicly in the online repository 'Figshare' and the brute force code is freely available in Code Ocean. Scientists are encouraged to implement faster algorithms, e.g., graphic processing unit-accelerated, with accurate or controlled approximate behaviour to compute trabecular thickness and spacing. Especially, the provided data can be re-employed together with the reference algorithm to evaluate their accuracies.

## CORRESPONDING AUTHOR:

### Thi-Ngoc-Thu Nguyen

Student, The Sirindhorn International Thai-German Graduate School of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand; Intern, Gremse-IT GmbH, Aachen, Germany

[thuthi.n-sse2017@tggs.kmutnb.ac.th](mailto:thuthi.n-sse2017@tggs.kmutnb.ac.th)

## KEYWORDS:

Trabecular thickness; Trabecular spacing; Three-dimensional; Volume-based thickness; Brute force

## TO CITE THIS ARTICLE:

Nguyen TNT, Höfter A, Leonardic K, Rosenhain S, Kiessler F, Sae-Tang W, Naumann U, Gremse F 2022 A Micro-computed Tomography Database and Reference Implementation for Parallel Computations of Trabecular Thickness and Spacing. *Journal of Open Research Software*, 10: 4. DOI: <https://doi.org/10.5334/jors.360>

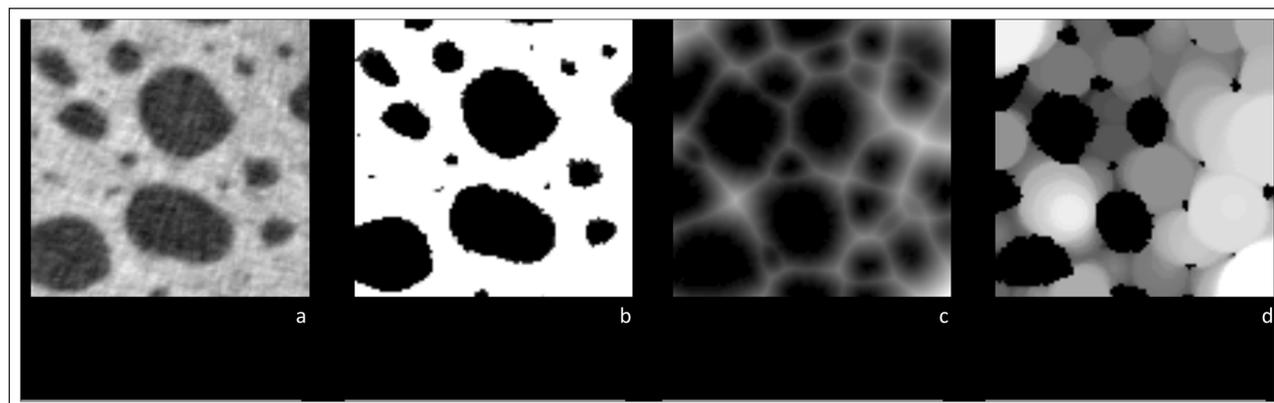
**(1) OVERVIEW**  
**INTRODUCTION**

Trabecular bone is a region of bone with a spongy, honeycomb-like structure. Analysis of trabecular bone provides important information to accurately characterize bone tissue, helping in researching bone diseases and investigating the progress of bone formation. Bone density, another bone characteristic, reveals the amount of mineral per volume of bone, but does not provide information directly about bone morphology. In contrast, trabecular thickness (Tb.Th) and trabecular spacing (Tb.Sp), two important parameters of trabecular structure, which are the average thickness of bone and non-bone, respectively, give more value information about bone structure to assess the effectiveness of bone forming agents [1–5]. Previous studies used diverse methods to estimate trabecular thickness, resulting in different, potentially non-reliable results. Standardization of trabecular thickness measurement is of paramount importance, consequently, trabecular thickness should be computed by a well-defined procedure.

The oldest method to calculate the average thickness of thin structures is histomorphometry. Thereby the thickness is calculated indirectly as a half of the area divided by its perimeter with an assumption based on the two-dimensional model [3]. This method can be incorrect, for example, when obtained points and intersections for the area and measurements of the perimeter are not representative for the object. Cruz-Orive proposed a surface-based analysis of trabecular thickness with a model assumption, in which the trabecular thickness at any point is measured by the distance from that point to the opposite surface [6]. However, this surface-based definition tends to induce inaccuracy problems for arbitrary and non-ideal structures. To assess the thickness of any two-dimensional object of any shape, Garrahan et al. introduced a solution to estimate the local thickness by fitting the maximal circles to every point inside the structure [7]. A similar method for three-dimensional structure analysis using the definition of the volume-based thickness was developed by Hildebrand

and R uegsegger and applied in the well-known open source software BoneJ [4, 5, 8]. In that way, the local thickness at one point is computed as the diameter of the greatest sphere that fits within the structure and contains the point. This algorithm accepts a binary mask input. Accordingly, the Euclidean distance of each point in the foreground to the closest background one is measured and shown as intensity at each point in the distance map [6] (Figure 1). From that, the local thickness map of the structure is produced, of which intensity at a point is the value of the local thickness at the respective point in the object. In turn, the average trabecular thickness is defined as arithmetic mean value of the local thicknesses at all points in the structure. The algorithm based on the definition of Hildebrand and R uegsegger shows good performance for extraction of the quantitative values with high resolution and informative inputs such as micro-computed tomography ( $\mu$ CT). In comparison with other software such as CT Analyser and Scanco, BoneJ provides similar results of trabecular thickness measurements with variations of up to 20% [4]. A reason for these differences could occur due to different approximations required by the algorithms to derive the results in acceptable computation times. Therefore, the need for a simple, well-defined and accurate reference algorithm without approximations to calculate trabecular thickness and spacing is apparent.

In this work, we present a database of  $\mu$ CT bone images of a sheep femur as well as a simple reference brute force implementation for accurate trabecular thickness and spacing computations, which is derived from the volume-based thickness concept of Hildebrand and R uegsegger. The algorithm is designed to be suitable for massively parallel computing because voxel values can be computed independently from each other. The accuracy of the reference implementation without approximations is validated with artificial datasets of ellipsoids. Furthermore, we assess how the correctness of trabecular thickness measurements depends on the voxel size and show that sub-voxel processing or upsampling processing can be used to increase the accuracy at the



**Figure 1** Example of a trabecular bone with a scale bar of 2mm and the respective measurements. **a)** Trabecular bone **b)** Binary mask **c)** Distance map **d)** Local thickness map.

expense of computation time. The reference algorithm is applied to the introduced  $\mu$ CT database covering a wide range of trabecular thickness and spacing. The results are correlated with those from the current de-facto reference implementation, BoneJ, over a large distribution of trabecular thickness and spacing, showing strong agreement. However, the computation of the brute force method has time complexity  $O(n^6)$  for cubic volumes with the size of each dimension of  $n$  and is therefore prohibitive for high resolution  $\mu$ CT datasets. We encourage scientists to develop fast parallel graphic processing unit accelerated (GPU-accelerated) algorithms to compute trabecular bone thickness and spacing, either accurately or approximately with well-understood error bounds. The accuracy of these methods can be assessed by using the provided datasets and the reference brute force implementation. The intermediate data from the brute force method are also provided to enable verification of their intermediate steps.

### IMPLEMENTATION AND ARCHITECTURE $\mu$ CT datasets

40 datasets were cropped from a  $\mu$ CT scan of a femur of a 60-month-old female sheep, which was from a study about an animal model for aortic aneurysms [9]. The scan is  $1,600 \times 1,600 \times 800$  equilateral voxels, which was acquired by a  $\mu$ CT (MILabs B. V., Utrecht, The Netherlands) with a voltage of 55 kV and a current of 0.17 mA with 1,440 projections. Three-dimensional (3D) bone images were cut at varying positions of the  $\mu$ CT scan by using the software Imalytics Preclinical (Gremse-IT GmbH, Aachen, Germany) to get different rectangular forms, which have distinct trabecular thickness and spacing ([Figure 2](#)).

Our introduced database of bone images consists of 40 three-dimensional bone images in the nifti file format (.nii) with the image size of  $100 \times 100 \times 100$  voxels, voxel size of  $20 \times 20 \times 20 \mu\text{m}^3$ . Along with each bone image, the intermediate results, which are computed by both the brute force method and BoneJ, are also available.

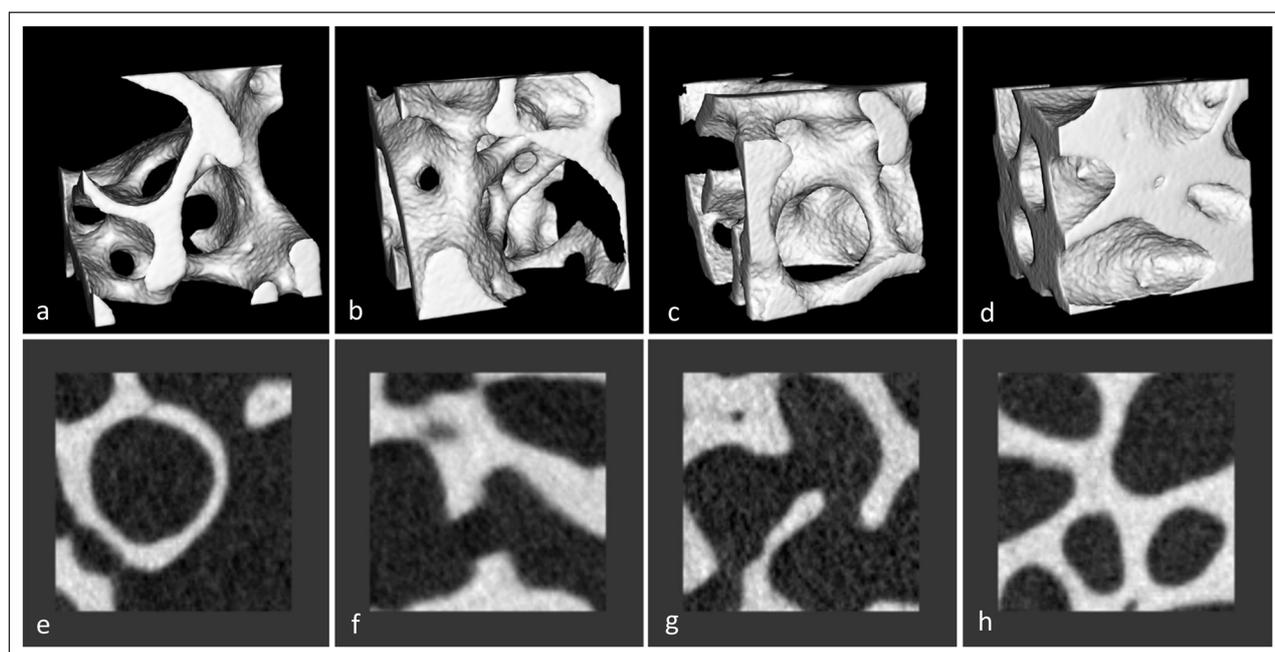
### Bone Segmentations

Since BoneJ and the brute force method require a binary mask as input, the bone images were segmented by using a single thresholding segmentation. The thresholding value was set as 2,000 HU (Hounsfield Units) and applied on the HU-calibrated  $\mu$ CT scans. The resulting binary mask includes trabecular bone as the foreground and the none-bone part as the background and is also attached in the dataset as stated in the section  $\mu$ CT dataset.

### Brute force algorithm

To analyse the effect of voxelization, particularly for thin structures, we upsampled the input data using trilinear interpolation by a factor of two, meaning the voxels size is reduced by two. This upsampling was performed before applying the threshold to generate the binary mask. The thickness and spacing computations were performed with and without upsampling.

Moreover, the original bone images were downsampled with a factor of 0.5 and then upsampled with a factor of two. The thicknesses of the downsampled bone images and the downsampled-upsampled bone images are computed with the brute force method and compared together with the thicknesses of original images to examine the enhancement of accuracy.



**Figure 2** Examples of bone images with different forms and trabecular thickness. 3D rendering (upper panel) and 2D axial slice view (lower panel), respectively.

**Thickness measurements**

The original volume-based trabecular thickness definition can be expressed by the formula 1 and **Figure 3**. According to the definition, the distance map value of a voxel describes the radius of the largest sphere contained in the foreground and centered at the voxel. From the distance map, the local thickness map is computed which describes the diameter of the largest sphere contained in the foreground including the voxel. The local thickness at point  $\bar{p}$  is equal to twice the highest intensity in the distance map, point  $\bar{q}$ , that is bigger than the Euclidean distance between those two points as the following equations 2 and 3:

$$LTM(\bar{p}) = 2 \max_{\bar{q} \in \Omega} (r | \bar{p} \in \text{sph}(\bar{q}, r) \subseteq \Omega) \quad (1)$$

$$DM(\bar{p}) = \min_{\bar{x} \notin \Omega} (|\bar{p} - \bar{x}|) \quad (2)$$

$$LTM(\bar{p}) = 2 \max_{\bar{q} \in \Omega} (DM(\bar{q}) | DM(\bar{q}) + \epsilon > |\bar{p} - \bar{q}|) \quad (3)$$

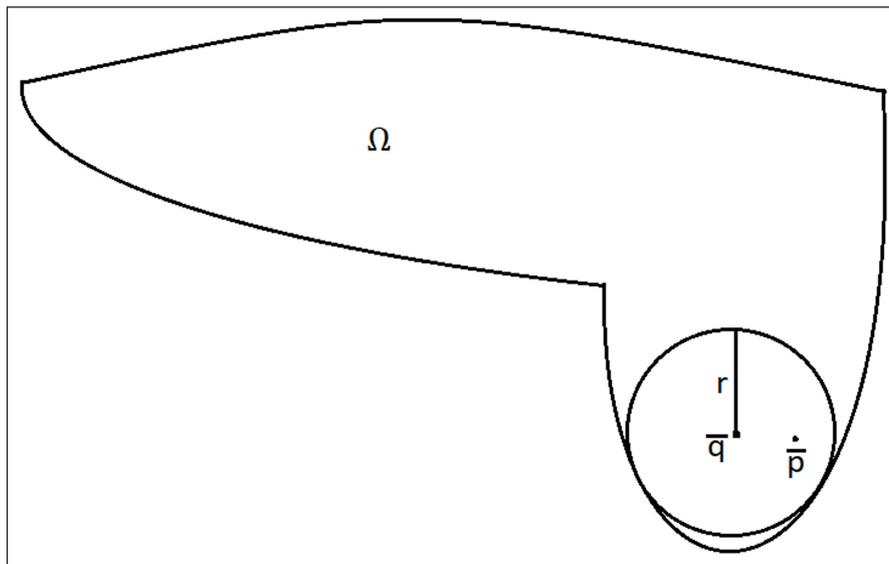
where  $\bar{p}$ ,  $\bar{q}$  are points inside the object  $\Omega$ ,  $r$  is the radius of the sphere  $\text{sph}(\bar{q}, r)$  with the center at  $\bar{q}$ ,  $\bar{x}$  is a point outside the object,  $DM(\bar{p})$  is the distance map

at the point  $\bar{p}$ ,  $LTM(\bar{p})$  is the local thickness at the point  $\bar{p}$ ,  $\Omega \in \mathbb{R}^3$  is the set of all points in the object,  $|\bar{p} - \bar{x}|$  is the Euclidean distance between  $\bar{p}$  and  $\bar{x}$ ,  $|\bar{p} - \bar{q}|$  is the Euclidean distance between  $\bar{p}$  and  $\bar{q}$ , and  $\epsilon$  is a defined error, an important factor to account for border voxels, which are close to the surfaces of spheres. The value  $\epsilon$  is selected experimentally as 0.5, a good value to include accurately border voxels without requiring a clean-up step as in BoneJ.

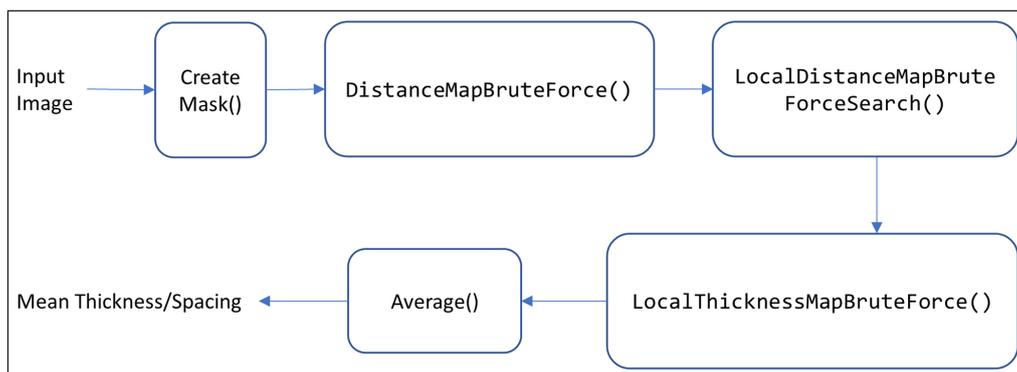
The brute force implementation runs over all other voxels in parallel to search for the maximum containing sphere, according to the equation 3. Therefore, it provides well-defined results without approximations at the expense of computation time and can be used as a reference algorithm.

**Implementation**

The algorithm was implemented in C++ with the standard library and Open Multi-Processing statements. The code contains 64 lines of code for the local thickness map calculation, 66 lines of code for distance map calculation with five primary functions and 1,900 lines of code for helper functions (**Figure 4**). Main functions are:



**Figure 3** A surface of the object  $\Omega$ . Measurement of volume-based thickness at the point  $\bar{p}$  inside a surface of the object  $\Omega$ .



**Figure 4** Workflows of the brute force algorithm.

*CreateMask()* creates a binary mask of the input image.

*DistanceMapBruteForce()* calculates the distance map

*LocalDistanceMapBruteForceSearch()* searches for the biggest sphere that contains a give voxel

*LocalThicknessMapBruteForce()* calculates the local thickness map

*Average()* calculates the mean value of the local thickness map

### Generation of artificial datasets

For validation of the brute force algorithm, we assumed a fixed ellipsoid with diameters 100 x 100 x 300 mm<sup>3</sup>. A set of 500 3D binary images are created by generating voxelized binary masks of 500 images of an ideal ellipsoid in a cube, defined by an ellipsoid formula, sampling the fixed ellipsoid. The binary datasets are composed of 50 subsets differing in voxel sizes, in which 10 images of each subset share the same voxel size, but different relative positions of the ellipsoid in the cube. The whole of datasets varies namely in grid sizes of Z dimension from 15 to 309 voxels with increments between subsets of six voxels, leading to decreasing voxel sizes, from 40 mm to 1.942 mm. The binary images are inputted to the brute force method and BoneJ to produce the local thickness map of each ellipsoid cube image. The mean thickness of a subset of ellipsoid cube images with a certain voxel size is correspondingly evaluated and considered as the mean thickness of the ellipsoid with that voxel size with averaged error in trabecular thickness of the ellipsoid due to voxelization. Then, these results are compared to the thickness of the respective ideal ellipsoid of the same voxel size, as an exact value, which is computed by the closed formula by Hildebrand and R uegsegger.

### QUALITY CONTROL

The algorithm was tested with 500 ellipsoid cube images covering a wide range of voxel sizes from 1.942 mm to 40 mm. Furthermore, the code was also tested with the

whole provided database of 40 scanned bone images with a broad range of trabecular thickness and spacing.

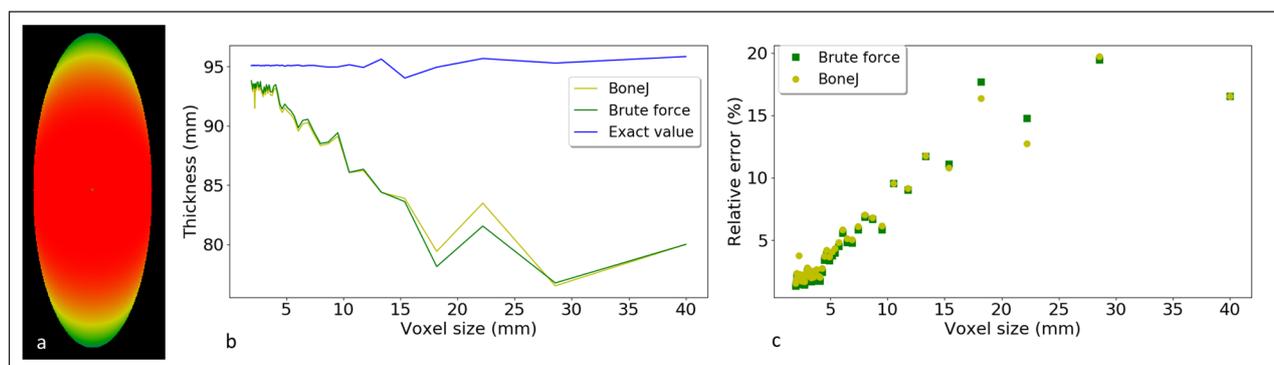
### Artificial ellipsoid cube datasets

To validate the brute force algorithm, we used artificially generated images with different voxel sizes of a fixed ellipsoid with diameters 100 x 100 x 300 mm<sup>3</sup>. The trabecular thickness inside the respective ideal ellipsoid is computed with a closed formula of Hildebrand and R uegsegger and compared with the results from the brute force algorithm and BoneJ. Finally, the trabecular thickness and spacing of the 40 bone samples were calculated using brute force and BoneJ and correlated to examine their accuracies, supporting that the brute force can be used to validate the future methods for trabecular thickness measurements.

Since the local thickness map computation involves voxelization, a discretization error in the range of the voxel size is expected in both the distance map and the local thickness map. In **Figure 5** the error in the mean thickness of ellipsoids appears systematic due to the voxelization. Generally, voxelization may result in a large relative error if large voxel sizes relative to the structure sizes are used. **Figures 5b** and **5c** show that the discretization error becomes worse for bigger voxel sizes. **Figure 5c** illustrates that the brute force method and BoneJ involve large relative errors up to 20% for big voxel sizes. Both the brute force algorithm and BoneJ require voxel sizes under 5.4 mm and 4.4 mm or 111 voxels and 135 voxels in the Z dimension, respectively, to reach relative errors below 4%. The correctness of the brute force and BoneJ improves for the smaller voxel sizes, suggesting the usage of the upsampling technique with these methods for accuracy enhancement in future applications.

### Scanned bone datasets

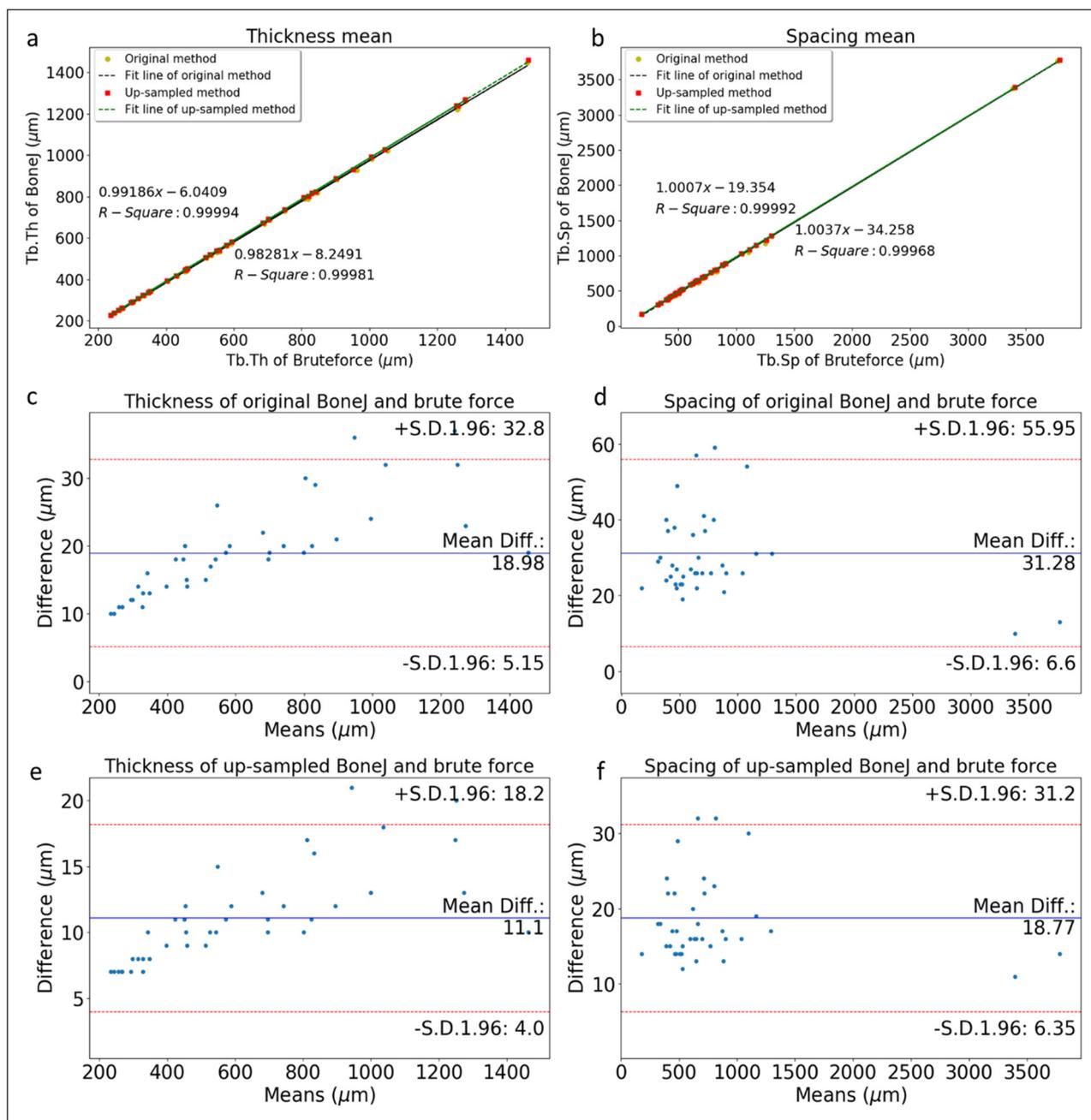
Trabecular thickness and spacing were computed for the bone images using the brute-force method and BoneJ. Furthermore, the intermediate and final results were



**Figure 5 Accuracy of thickness computations on ellipsoid. a)** Slice through the analytically derived local thickness map of the ellipsoid with diameters 100 x 100 x 300 mm<sup>3</sup>. The red color has a higher intensity, corresponding to a bigger local thickness. **b)** Mean local thickness as a function of voxel size in comparison with the exact values, which were mathematically calculated per voxel. Coarse voxels cause an underestimation of the thickness. **c)** The relative error to the exact value increases with the voxel size. BoneJ and the brute force algorithm have similar relative errors which increase with the growth of the voxel size.

analyzed for original and upsampled datasets. With the same binary mask input, the brute force method and BoneJ produced identical distance maps but the local thickness maps differed slightly, possibly owing to a tradeoff in BoneJ favoring computation time over measurement accuracy. The 40 bone images showed a broad distribution of values for trabecular thickness (236  $\mu\text{m}$  to 1,469  $\mu\text{m}$ ) and spacing (326  $\mu\text{m}$  to 3,790  $\mu\text{m}$ ) as shown in [Figure 6](#), emphasizing that the provided dataset is well-suited to examine the accuracies of trabecular thickness and spacing computations. [Figure 6a](#) shows a strong correlation for the trabecular thickness of the original brute force method and BoneJ as well as

for the upsampled brute force method and BoneJ with correlation coefficients of 0.98 and 0.99, respectively. The Bland-Altman plots in [Figure 6c](#) suggest that the results of the brute force method are always slightly higher than those of BoneJ, by a difference on the order of the voxel size. Their difference tends to increase for thicker structures, is in the range of the voxel size and can be attributed to the compromise in BoneJ between the computation time and the thickness correctness. The trends are shared for the upsampled methods in [Figure 6e](#) with smaller mean difference 11.1  $\mu\text{m}$ . For trabecular spacing, the mean difference is higher without an apparent systematic trend ([Figure 6d, 6f](#)).



**Figure 6 Comparison between brute force and BoneJ for 40 bone datasets.** Strong correlation between brute force and BoneJ appears for trabecular thickness (a) and trabecular spacing (b), with and without upsampling. Bland-Altman plots for trabecular thickness (c) and trabecular spacing (d) show systematically lower values for BoneJ compared to brute force, on the order of the voxel size. Bland-Altman plots for upsampled results (e, f) show stronger agreement between BoneJ and brute force.

To confirm the effect of voxelization, the thicknesses of the downsampled bone images with the voxel size of 40 μm and the downsampled-upsampled bone images with the voxel size of 20 μm are compared to those of the original images. **Figure 7** demonstrates that the relative deviations of the thicknesses of the downsampled images from the thicknesses of original images are mostly bigger than those of the downsampled-upsampled images. In other words, with upsampling, the thicknesses of the downsampled-upsampled images are closer to those of the original images, which is in strong agreement with **Figure 5c**. The upsampling enhances certainly the accuracy of thickness measurement.

## (2) AVAILABILITY OPERATING SYSTEM

The code may be compiled on any platform and C++ compiler but was only tested with Windows 7 and 10 with Visual Studio 2019.

## PROGRAMMING LANGUAGE

C++

## ADDITIONAL SYSTEM REQUIREMENTS

CMmake Version 3.1 or newer is required to generate the project files, for example, for Visual Studio 2019, which can then be used to compile the code. Our code only uses C++ with the standard library and OpenMP statements.

## DEPENDENCIES

N/A

## LIST OF CONTRIBUTORS

N/A

## SOFTWARE LOCATION

Archive

**Name:** Figshare

**Persistent identifier**

**For software:** <https://doi.org/10.6084/m9.figshare.17209319>

**For bone dataset:** <https://doi.org/10.6084/m9.figshare.9864677>

**Licence:** MIT licence

**Publisher:** Thi-Ngoc-Thu Nguyen, Felix Gremse

**Version published:** 3.0

**Date published:** 15/09/21

Code repository

**Name:** Code Ocean

**Identifier:** <https://doi.org/10.24433/CO.3957929.v3>

**Licence:** MIT licence

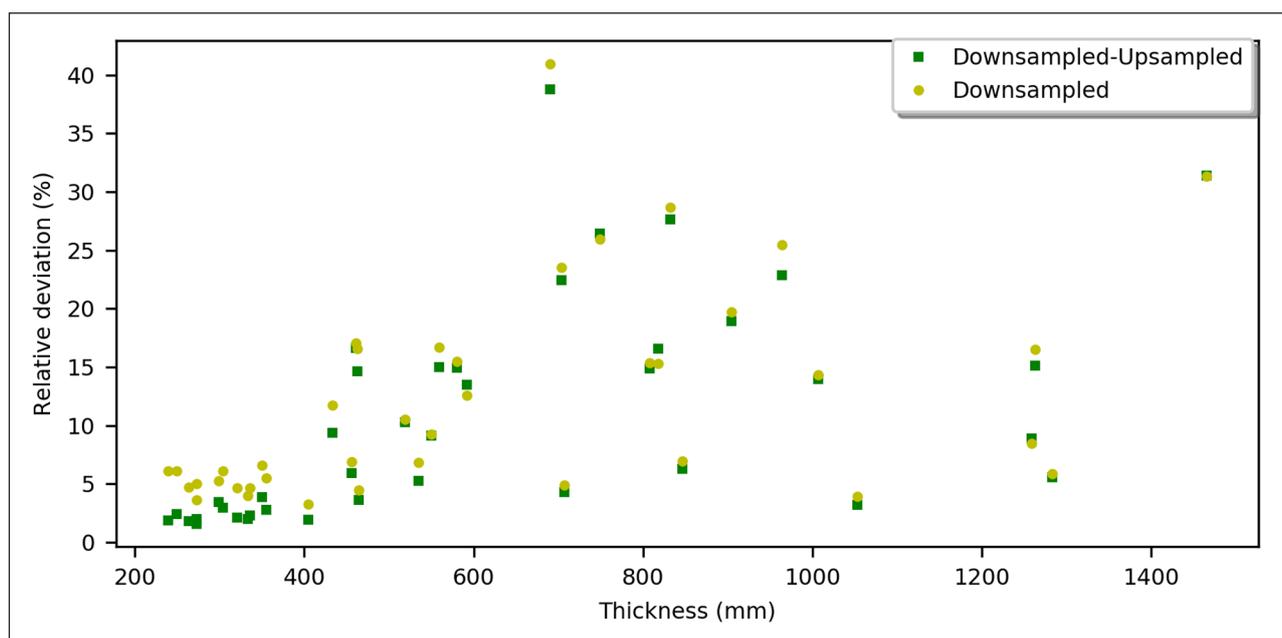
**Date published:** 15/09/21

## LANGUAGE

English

## (3) REUSE POTENTIAL

We implemented a simple brute force algorithm of 130 lines of code with the helper functions of 1,900 lines of code to compute trabecular thickness and spacing according to the concept of Hildebrand and Rüegsegger. Its accuracy was evaluated by the comparison of



**Figure 7** Relative deviation of thicknesses of downsampled bone images and downsampled-upsampled bone images from those of the original images by the brute force method.

the outputs with those of BoneJ on artificial ellipsoid datasets of different resolution and on bone datasets. BoneJ with a certain approximation and a complicated algorithm of more than 1,500 lines of code along with the helper functions of thousands of lines of code and the straightforward brute force method without approximations produce very similar results. The both methods show systematic underestimation of trabecular thickness which increases with the voxel sizes. This phenomenon can be attributed to the similar binarization steps inside brute force and BoneJ. By upsampling the ellipsoid images using trilinear interpolation, utilizing information on the subvoxel level, binary images are generated at higher resolution, resulting in more accurate values for trabecular thickness, indicating that upsampling can reduce the systematic underestimation caused by the binarization. This is also completely in agreement with the experiment of downsampled bone images and downsampled-upsampled bone images.

Using 40 cubic datasets of different bone regions of a sheep femur with a broad range of trabecular thickness and spacing, we show that brute force and BoneJ strongly correlate in their estimations of trabecular thickness and spacing. Their results indicate differences on the order of the voxel size which can be attributed to implementation details. Upsampling reduces the mean difference between the two algorithms, supporting the idea that upsampling can increase the accuracy by performing the binarization on a finer grid. This is illustrated with the probability  $p$ -value  $< 0.001$  in Supplementary File S1 against the null hypothesis of mean difference between two methods. The paired two-sample  $t$ -test demonstrates that the mean of the differences between thicknesses of bone images by the upsampled brute force method and BoneJ is smaller than those by the original brute force method and BoneJ. The thickness measurements by the upsampled methods reach closer to the exact values.

Unfortunately, upsampling strongly increases the computation time to generate trabecular thickness and spacing. Particularly, the brute force method with the time complexity  $O(n^6)$  becomes extremely time-consuming for the large input images. Faster methods could be implemented using GPU-acceleration or by performing a hill-climbing search to find the largest enclosing sphere for each voxel. A pyramid approach could be used to avoid local maxima during hill-climbing. While we intend to provide a simple reference algorithm without approximation for such performance optimizations, our analysis also shows that BoneJ with a certain approximation performs well for both artificial and real datasets, delivering values close to our brute force algorithm, supporting its wide-spread use.

Researchers are encouraged to download the databases for reuse from Figshare. Dataset usage

is meant for evaluation of the accuracy of methods to compute trabecular thickness and spacing. The trabecular thickness results of future methods can be compared with those of the brute force method over the whole dataset to get the distribution of their difference over a range of data. The intermediate data from the dataset such as binary masks, distance maps and local thickness maps of the brute force method can also be used to verify intermediate steps of the future methods. Moreover, the researchers can test the brute force algorithm with images beyond the provided database.

## ADDITIONAL FILE

The additional file for this article can be found as follows:

- **Supplementary File S1.** Paired two-sample  $t$ -test for comparison of the means of the differences of thicknesses of the bone images calculated by the original brute force method and BoneJ and those by the upsampled brute force method and BoneJ.  $P(T \leq t)$  two-tail is the important probability  $p$ -Value. DOI: <https://doi.org/10.5334/jors.360.s1>

## ACKNOWLEDGEMENTS

This work was supported by the European Union (FP7), the German Federal State of North Rhine Westphalia (EFRE), the German Research Foundation (DFG; GR 5027/2-1) and the German Academic Exchange Service.

## COMPETING INTERESTS

Felix Gremse is the owner of Gremse-IT GmbH and works for MILabs B.V. as consultant. The other authors declare no competing interests.

## AUTHOR CONTRIBUTIONS

Thi-Ngoc-Thu: Nguyen Producing the data, developing the software, executing the experiments and writing the manuscript.

Andreas Höfter: Developing the software.

Kevin Leonardic: Developing the software.

Stefanie Rosenhain: Reviewing the manuscript.

Fabian Kiessling: Reviewing the manuscript.

Wannida Sae-Tang: Reviewing the manuscript.

Uwe Naumann: Reviewing the manuscript.

Felix Gremse: Scanning the samples and supervising the study.

## AUTHOR AFFILIATIONS

**Thi-Ngoc-Thu Nguyen**  [orcid.org/0000-0003-2860-8571](https://orcid.org/0000-0003-2860-8571)

Student, The Sirindhorn International Thai-German Graduate School of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand; Intern, Gremse-IT GmbH, Aachen, Germany

**Andreas Höfter**

Developer, Gremse-IT GmbH, Aachen, Germany

**Kevin Leonardic**

Developer, Gremse-IT GmbH, Aachen, Germany

**Stefanie Rosenhain**  [orcid.org/0000-0003-2471-7551](https://orcid.org/0000-0003-2471-7551)

Researcher, Gremse-IT GmbH, Aachen, Germany; Researcher, Institute for Experimental Molecular Imaging, University Hospital Aachen, Aachen, Germany

**Fabian Kiessling**  [orcid.org/0000-0002-7341-0399](https://orcid.org/0000-0002-7341-0399)

Professor, Institute for Experimental Molecular Imaging, University Hospital Aachen, Aachen, Germany; Fraunhofer MEVIS, Institute for Medical Image Computing, Aachen, Germany; Comprehensive Diagnostics Center Aachen, Aachen, Germany;

**Wannida Sae-Tang**  [orcid.org/0000-0001-7834-2252](https://orcid.org/0000-0001-7834-2252)

Assistant Professor, The Sirindhorn International Thai-German Graduate School of Engineering, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand

**Uwe Naumann**  [orcid.org/0000-0002-7518-5922](https://orcid.org/0000-0002-7518-5922)

Professor, Software and Tools for Computational Engineering, RWTH Aachen University, Aachen, Germany

**Felix Gremse**  [orcid.org/0000-0002-6522-9004](https://orcid.org/0000-0002-6522-9004)

Director, Gremse-IT GmbH, Aachen, Germany; Researcher, Institute for Experimental Molecular Imaging, University Hospital Aachen, Aachen, Germany; Software and Tools for Computational Engineering, RWTH Aachen University, Aachen, Germany

## REFERENCES

1. **Tanaka Y, Nakayamada S, Okada Y.** Osteoblasts and osteoclasts in bone remodeling and inflammation. *Curr. Drug Targets Inflamm. Allergy.* 2005; 4: 325–328. DOI: <https://doi.org/10.2174/1568010054022015>
2. **Gremse F, Stärk M, Ehling J, Menzel JR, Lammers T, Kiessling F.** Imalytics Preclinical: Interactive Analysis of Biomedical Volume Data. *Theranostics.* 2016; 6(3): 328–341. DOI: <https://doi.org/10.7150/thno.13624>
3. **Parfitt AM, Mathews CHE, Villanueva AR, Kleerekoper M, Frame B, Rao DS.** Relationships between Surface, Volume, and Thickness of Iliac Trabecular Bone in Aging and in Osteoporosis. *J. Clin. Invest.* 1983; 72: 1396–1409. DOI: <https://doi.org/10.1172/JCI111096>
4. **Doube M, Klosowski MM, Arganda-Carreras I, Cordelières FP, Dougherty RP, Jackson JS, et al.** BoneJ: Free and extensible bone image analysis in ImageJ. *Bone.* 2010; 47(6): 1076–1079. DOI: <https://doi.org/10.1016/j.bone.2010.08.023>
5. **Hildebrand T, Rüegsegger P.** A new method for the model-independent assessment of thickness in three-dimensional images. *J. Microsc.* 1997; 185: 67–75. DOI: <https://doi.org/10.1046/j.1365-2818.1997.1340694.x>
6. **Cruz-Orive LM.** Estimation of Sheet Thickness Distributions from Linear and Plane Sections. *Biom. J.* 1979; 21: 717–730. DOI: <https://doi.org/10.1002/bimj.4710210803>
7. **Garrahan NJ, Mellish RWE, Vedi S, Compston JE.** Measurement of Mean Trabecular Plate Thickness by a New Computerized Method. *Bone.* 1987; 8(4): 227–230. DOI: [https://doi.org/10.1016/8756-3282\(87\)90169-4](https://doi.org/10.1016/8756-3282(87)90169-4)
8. **Hildebrand T, Laib A, Müller R, Dequeker J, Rüegsegger P.** Direct Three-Dimensional Morphometric Analysis of Human Cancellous Bone: Microstructural Data from Spine, Femur, Iliac Crest, and Calcaneus. *J. Bone Miner. Res.* 1999; 14(7): 1167–1174. DOI: <https://doi.org/10.1359/jbmr.1999.14.7.1167>
9. **Saha PK, Wehrli FW.** Measurement of Trabecular Bone Thickness in the Limited Resolution Regime of In Vivo MRI by Fuzzy Distance Transform. *IEEE Trans. Med. Imaging.* 2004; 23(1): 53–62. DOI: <https://doi.org/10.1109/TMI.2003.819925>

### TO CITE THIS ARTICLE:

Nguyen TNT, Höfter A, Leonardic K, Rosenhain S, Kiessling F, Sae-Tang W, Naumann U, Gremse F 2022 A Micro-computed Tomography Database and Reference Implementation for Parallel Computations of Trabecular Thickness and Spacing. *Journal of Open Research Software*, 10: 4. DOI: <https://doi.org/10.5334/jors.360>

Submitted: 04 December 2020 Accepted: 20 February 2022 Published: 16 March 2022

### COPYRIGHT:

© 2022 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press.