



This MICCAI paper is the Open Access version, provided by the MICCAI Society. It is identical to the accepted version, except for the format and this watermark; the final published version is available on SpringerLink.

HDilemma: Are Open-Source Hausdorff Distance Implementations Equivalent?

Gašper Podobnik^(✉) and Tomaž Vrtovec

Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
gasper.podobnik@fe.uni-lj.si

Abstract. Quantitative performance metrics play a pivotal role in medical imaging by offering critical insights into method performance and facilitating objective method comparison. Recently, platforms providing recommendations for metrics selection as well as resources for evaluating methods through computational challenges and online benchmarking have emerged, with an inherent assumption that metrics implementations are consistent across studies and equivalent throughout the community. In this study, we question this assumption by reviewing five different open-source implementations for computing the Hausdorff distance (HD), a boundary-based metric commonly used for assessing the performance of semantic segmentation. Despite sharing a single generally accepted mathematical definition, our experiments reveal notable systematic differences in the HD and its 95th percentile variant across implementations when applied to clinical segmentations with varying voxel sizes, which fundamentally impacts and constrains the ability to objectively compare results across different studies. Our findings should encourage the medical imaging community towards standardizing the implementation of the HD computation, so as to foster objective, reproducible and consistent comparisons when reporting performance results.

Keywords: Segmentation · Performance metrics · Hausdorff distance · Evaluation · Benchmarking · Open-source implementation

1 Introduction

Performance metrics play a central role in assessing the capabilities of methods across various disciplines. While qualitative metrics offer a detailed insight into individual cases, quantitative metrics provide an objective, computationally efficient, transparent and relatively fast mean for evaluating method performance on a broader scale [12]. As such, quantitative metrics are indispensable for comparing different methods, ranking challenge submissions and ensuring statistical reproducibility, and therefore their engineering requires a meticulous design, proper validation and general consensus within the community of interest. Recently, the medical imaging community has placed increasing emphasis on quantitative metrics selection that was encouraged by the exemplary efforts of the MICCAI *Special Interest Group for Challenges*¹. Their work underscores the

¹ <http://www.miccai.org/special-interest-groups/challenges/>

importance of selecting appropriate metrics tailored to specific tasks, moreover, they provide researchers with clear recommendations to efficiently select appropriate metrics and align evaluation strategies with investigation objectives. In addition, platforms such as *Hugging Face*² and *Papers with Code*³ have emerged as valuable resources for evaluating methods through computational challenges and online benchmarking, with an inherent assumption that metrics implementations across studies are consistent and equivalent throughout the community.

In this study, we question this assumption by reviewing multiple open-source implementations of the Hausdorff distance (HD) [7], a quantitative metric measuring the degree of mismatch between two sets [1] that is commonly used alongside the Dice similarity coefficient (DSC) to assess the performance of semantic segmentation [26]. However, while the DSC is straightforward to implement, the HD presents challenges due to variations in its practical implementation despite sharing a single generally accepted mathematical definition. For two observed sets A and B , the HD measures the largest among all distances of a point in A to the closest point in B , and vice versa (i.e. the *max-min* bidirectional distance):

$$HD_p = \max([D_{AB}]_p, [D_{BA}]_p), \quad (1)$$

where $[D_{AB}]_p = K_{a \in A}^p(\min_{b \in B} \|a - b\|_2)$ is the one-sided (asymmetrical) p -th percentile distance between A and B , computed from the set of closest distances of points $a \in A$ to B that are commonly obtained by the Euclidean norm ($\|\cdot\|_2$). The symmetrical HD_p is then the maximal distance among all such distances between A and B , and between B and A (1). Besides the 100th percentile variant (HD_{100} or simply HD), the 95th percentile variant (HD_{95}) is most often used because it is less sensitive to outliers in the form of noise and artifacts [1,19].

Related Work. Although introduced already in 1914 as part of the mathematical set theory [7], the HD was first proposed for digital image comparison in 1993 [8], and has been since used for object detection and matching [23,24,21] as well as for improving deep learning segmentation models [11,13], but more importantly, it has gained a widespread adoption for assessing segmentation performance [26,2]. As its calculation can be, in comparison to two-dimensional (2D) images, challenging especially for three-dimensional (3D) images where segmentations correspond to large-scale point sets, several computationally efficient algorithms were proposed [8,25,20]. Recently, the *Metrics Reloaded* initiative⁴ [16,19] identified the most common pitfalls associated with metrics, including boundary-based metrics such as the HD. While they offer several desirable properties, such as boundary-awareness, impartiality towards over-/under-segmentation and capability of measuring distances even in the absence of segmentation overlap, it is also important to acknowledge their limitations, such as inclination towards overlooking holes in segmentations and susceptibility to outliers. Because of its general acceptance, several open-source implementations are

² <https://huggingface.co/>

³ <https://paperswithcode.com/>

⁴ <https://metrics-reloaded.dkfz.de/>

available for the HD computation [17,16,3,28,9], and are widely used throughout the medical imaging community for assessing segmentation performance.

Motivation. While the HD computation is based on a single and generally accepted mathematical definition (1), its practical implementation is far from trivial [8]. To determine the HD between two binary segmentations in the image space represented by a discrete regular grid, it is first necessary to perform boundary extraction that is followed by distance computation, which may not be straightforward for the percentile variants. Particularly for 3D images, where the observed sets A and B (1) are represented by segmentation surfaces [25,20,10], calculating the distances alone is insufficient, as the areas of surface elements (i.e. surfels) must be also computed for accurate percentile distance estimation. Considering that several open-source implementations for the HD computation are available [17,16,3,28,9], our aim is to evaluate the extent of these caveats that may result in over-/under-estimation of the segmentation performance.

Contributions. Given different open-source implementations for the HD computation, the main contributions of our study are: (i) we dissect the computation process to clearly illustrate the differences among implementations, (ii) we introduce a mesh-based reference computation that adheres to the mathematical definition, (iii) we formulate a comprehensive procedure for evaluating implementations, (iv) we provide experimental evidence demonstrating notable systematic and statistically significant differences among implementations on clinical data, and (v) we provide recommendations for the research community related to the appropriate usage of open-source implementations for the HD computation.

2 Methods

For a better understanding of the HD computation caveats, we first introduce the proposed evaluation procedure and the mesh-based reference, and conclude with a description of the open-source implementations and their differences.

Evaluation Procedure. To simulate a practical scenario, the input to our evaluation procedure are two raw 3D binary segmentation masks A_0 and B_0 with arbitrary voxel size on which we apply meshing⁵ followed by a gentle Laplacian smoothing to eliminate potential artifacts and generate corresponding meshes \mathcal{M}_{A_0} and \mathcal{M}_{B_0} . Because voxel size impacts the HD computation, we perform voxelization⁶ to a selected target voxel size to convert meshes to 3D binary segmentation masks A and B that are used to compute the HD (Fig. 1).

⁵ Meshing is based on the marching cubes algorithm [14] that takes a 3D mask A and generates a 3D mesh \mathcal{M}_A consisting of vertices and surfels (i.e. triangular faces).

⁶ Voxelization (3D analogue of rasterization) is based on `vtkPolyDataToImageStencil` from the Visualization Toolkit (VTK) [22] that allows completely bijective transforms between the mesh and image space, i.e. voxelization of \mathcal{M}_A , obtained from A by meshing without smoothing, results again in A .

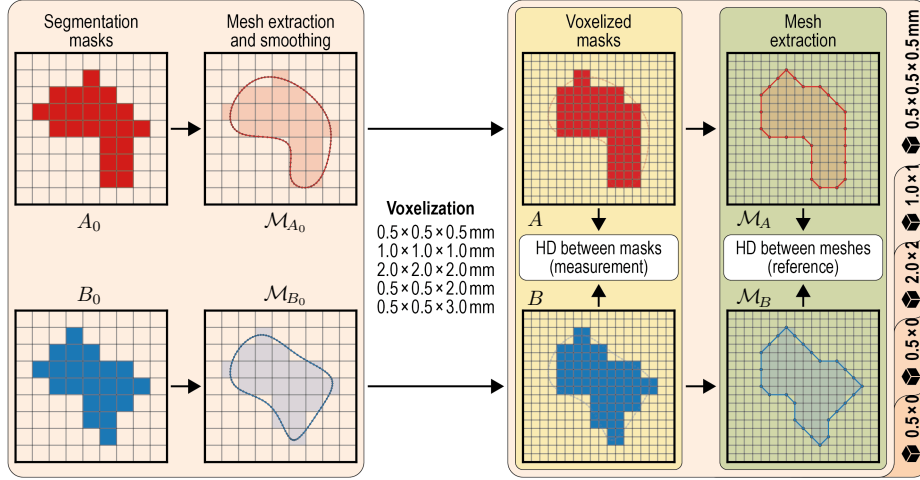


Fig. 1. A schematic illustration of the procedure for evaluating the Hausdorff distance (HD) computation for different voxel sizes.

Mesh-Based Reference. To evaluate an implementation for the HD computation, a highly accurate and precise reference is required, while its computational efficiency is of secondary concern. With a well-defined surface and corresponding surface normals, the mesh space represents an elegant and efficient solution for computing distances between the extracted surfaces. For 3D masks A and B , we first perform meshing without smoothing to obtain corresponding 3D meshes \mathcal{M}_A and \mathcal{M}_B (Fig. 1). To minimize the discretization error [4], we upsample both meshes by dividing each surfel into four smaller sub-surfels (i.e. without making changes to the mesh volume) and compute the corresponding surfel centroids. Centroids of \mathcal{M}_A serve as query points for calculating the set of distances⁷ D_{AB} to \mathcal{M}_B , and vice versa the set of distances D_{BA} (Fig. 2). While the HD is simply computed as $HD = HD_{100} = \max(D_{AB}, D_{BA})$, computing the p -th percentile HD as $\max([D_{AB}]_p, [D_{BA}]_p)$ would be incorrect, because surfel areas need to be taken into account. We therefore first compute the sets of areas S_A and S_B of all surfels in \mathcal{M}_A and \mathcal{M}_B , then we sort distances in D_{AB} and D_{BA} in ascending order and apply the same order to S_A and S_B , and finally we compute indices N_p and M_p of distances in D_{AB} and D_{BA} that correspond to cumulative sums of areas in S_A and S_B equalizing the p -th percentile of their total areas, respectively (Fig. 2). The correct p -th percentile HD is obtained by taking distances $d_{AN_p} \in D_{AB}$ and $d_{BM_p} \in D_{BA}$, and calculating $HD_p = \max(d_{AN_p}, d_{BM_p})$.

Open-Source HD Implementations. We focus on five different open-source implementations that provide the HD and its 95th percentile variant computation for 3D binary segmentations (updated to latest versions in February

⁷ Based on `vtkImplicitPolyDataDistance` in the VTK [22].

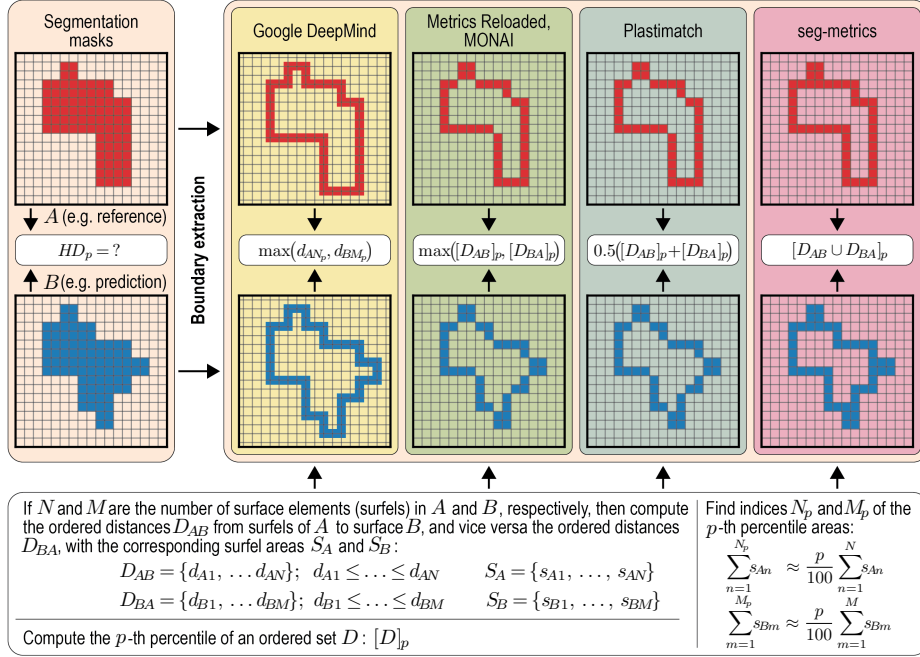


Fig. 2. A schematic illustration showing the differences among five open-source implementations for the computation of the p -th percentile Hausdorff distance (HD_p).

2024): (i) *Google DeepMind*⁸ [17], (ii) *Metrics Reloaded*⁹ [16], (iii) *MONAI*¹⁰ [3], (iv) *Plastimatch*¹¹ [28] and (v) *seg-metrics*¹² [9]. All implementations are available through the *GitHub* developer platform as Python packages, except for *Plastimatch*, which is available through the *GitLab* developer platform as a C++ toolbox and also utilized within *SlicerRT*, a radiation therapy research extension of the open-source software *3D Slicer* [5]. While some implementations allow to modify specific parameters, e.g. the method for boundary extraction, we limit our evaluation procedure to the default parameter values except for the voxel size. All implementations use the Euclidean distance transform [6] to calculate the distances between query points on the extracted boundary and the opposite segmentation mask, resulting in sets of distances D_{AB} and D_{BA} . However, the implementations differ in two fundamental operations when computing the HD in the image space, i.e. in boundary extraction and percentile calculation (Fig. 2). For boundary extraction, the five implementations share three different approaches: (i) *Google DeepMind* efficiently shifts the image grid by a half of the voxel size

⁸ <https://github.com/google-deepmind/surface-distance>, v0.1, sha1-hash: ee651c8

⁹ <https://github.com/Project-MONAI/MetricsReloaded>, v0.1.0, sha1-hash: b3a3715

¹⁰ <https://github.com/Project-MONAI/MONAI>, v1.3.0, sha1-hash: 865972f

¹¹ <https://gitlab.com/plastimatch/plastimatch>, v1.9.4, sha1-hash: 581c7692

¹² https://github.com/Jingnan-Jia/segmentation_metrics, v1.6.1, sha1-hash: fed1852

and extracts a densely connected boundary, (ii) *Metrics Reloaded*, *MONAI* and *Plastimatch* perform binary erosion with a square-connectivity structuring element, while (iii) *seg-metrics* uses a full-connectivity structuring element. For the p -th percentile calculation, the five implementations share four different approaches: (i) *Google DeepMind* is the only implementation that combines distances and surfel areas into $HD_p = \max(d_{AN_p}, d_{BM_p})$, and therefore follows our mesh-based reference approach, (ii) *Metrics Reloaded* and *MONAI* calculate the maximum of both asymmetrical distances corresponding to the p -th percentile in each set as $HD_p = \max([D_{AB}]_p, [D_{BA}]_p)$, (iii) *Plastimatch* calculates the average of both asymmetrical distances corresponding to the p -th percentile in each set as $HD_p = 0.5([D_{AB}]_p + [D_{BA}]_p)$, while (iv) *seg-metrics* calculates the p -th percentile of the union of both distance sets as $HD_p = [D_{AB \cup BA}]_p$.

3 Experiments and Results

To provide insights into the differences of implementations in practice, we set up experiments for comparing 3D segmentations of anatomical structures as one of the most common fields of application of the HD computation.

Data and Experiments. To emulate a workflow from practice, we resorted to data from radiotherapy planning [18], where the HD and especially its 95th percentile variant are, besides the DSC, reference metrics for assessing the organ-at-risk (OAR) (auto-)segmentation [27,15]. For this purpose, we devised 60 computed tomography and magnetic resonance images with corresponding 3D segmentations of up to 30 OARs provided by two clinical experts. We then applied meshing and voxelization on the resulting 1510 segmentation masks for three isotropic voxel sizes of $0.5 \times 0.5 \times 0.5$, $1.0 \times 1.0 \times 1.0$ and $2.0 \times 2.0 \times 2.0$ mm³, and two anisotropic voxel sizes of $0.5 \times 0.5 \times 2.0$ and $0.5 \times 0.5 \times 3.0$ mm³. Each open-source implementation was evaluated by the proposed procedure using each voxel size, and the differences against the proposed mesh-based reference were recorded for the HD (ΔHD_{100}) and its 95th percentile variant (ΔHD_{95}).

Results. In contrast to the DSC, which is a relative metrics bounded between 0 and 1, the HD values are absolute (usually reported in mm) with only a lower bound ($HD = 0$ for two identical surfaces). As comparing different implementations for individual OARs would be challenging due to the disparity of the results, we focus only on the differences among implementations in terms of their deviations from the mesh-based reference, with positive and negative values indicating the HD over- and under-estimation, respectively. Under-estimation of the HD is particularly problematic since it can provide over-optimistic results that may potentially lead to wrong conclusions. The obtained ΔHD_{100} and ΔHD_{95} are for all five open-source implementations reported in Table 1 and shown as box plots in Fig. 3. Statistically significant differences (p -values < 0.0001) were observed using paired t -tests with the Bonferroni correction for every ΔHD_{95} implementation pair except between *Metrics Reloaded* and *MONAI*.

Table 1. Mean \pm standard deviation differences in the 100th and 95th percentile Hausdorff distance (ΔHD_{100} and ΔHD_{95}) against the mesh-based reference.

	Voxel size (mm ³)	Google DeepMind	Metrics Reloaded	MONAI	Plastimatch	seg-metrics
ΔHD_{100}	0.5×0.5×0.5	0.016±0.044	0.023±0.031	0.023±0.031	0.023±0.031	0.021±0.047
	1.0×1.0×1.0	0.039±0.092	0.053±0.069	0.053±0.069	0.054±0.069	0.051±0.083
	2.0×2.0×2.0	0.101±0.193	0.127±0.146	0.127±0.146	0.128±0.147	0.125±0.152
	0.5×0.5×2.0	0.038±0.073	0.052±0.058	0.052±0.058	0.053±0.059	0.049±0.071
	0.5×0.5×3.0	0.038±0.079	0.058±0.088	0.058±0.088	0.059±0.089	0.056±0.098
ΔHD_{95}	0.5×0.5×0.5	-0.104±0.074	0.011±0.181	0.011±0.181	-1.975±5.826	-1.057±1.692
	1.0×1.0×1.0	-0.173±0.169	0.022±0.293	0.022±0.293	-1.981±5.825	-1.132±1.789
	2.0×2.0×2.0	-0.216±0.432	0.125±0.539	0.125±0.539	-1.906±5.846	-0.995±2.160
	0.5×0.5×2.0	-0.072±0.198	0.360±0.826	0.360±0.826	-1.736±5.646	-0.808±1.629
	0.5×0.5×3.0	-0.024±0.290	0.402±1.018	0.402±1.018	-1.747±5.619	-0.716±1.634

4 Discussion

Our study evaluated five different open-source implementations of the HD and its 95th percentile variant computation, and our experiments revealed that differences across implementations exist that should not be overlooked.

HD Computation Performance. The analysis of ΔHD_{95} revealed notable disparities among implementations, particularly for *Plastimatch* and *seg-metrics* with many under-estimation outliers. While variations in boundary extraction exist among implementations, the primary source of outliers stems from the differences in percentile calculation, which is evidenced by the absence of such deviations for ΔHD_{100} . In fact, *Plastimatch* computes the average of $[D_{AB}]_p$ and $[D_{BA}]_p$ that results in substantial under-estimation when one distance outweighs the other, while *seg-metrics* determines the percentile of the D_{AB} and D_{BA} union that again results in under-estimation when one set has smaller values

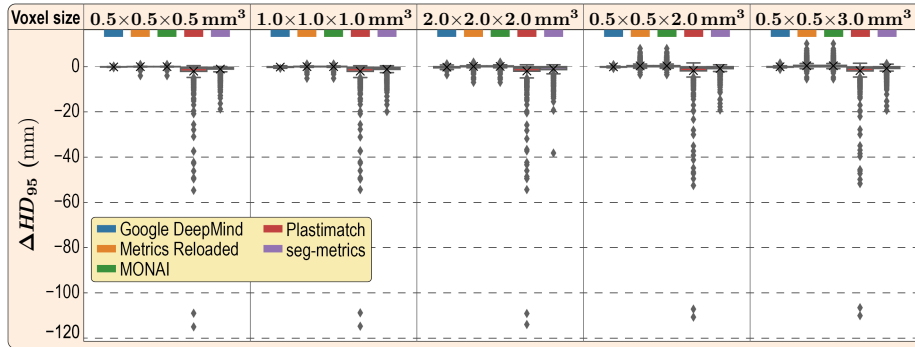


Fig. 3. Box plots of the differences in the 95th percentile Hausdorff distance (ΔHD_{95}) against the mesh-based reference.

that skew the percentile to a lower value. In contrast, *Google DeepMind*, *Metrics Reloaded*, and *MONAI* exhibit considerably less dispersion in ΔHD_{95} . In particular, *Metrics Reloaded* and *MONAI* showcase nearly identical implementations, evident from their analogous statistics (Table 1) and box plots (Fig. 3) that align with their shared boundary extraction and percentile calculation approaches, while *Google DeepMind* is the only implementation that incorporates surfel areas into percentile computation. For isotropic voxel sizes, the discrepancies among these three implementations are minimal, however, variations become more pronounced for anisotropic voxel sizes that correspond to a wider distribution of surfel areas, exerting a greater influence on percentile calculation. Based on this analysis, *Google DeepMind*, *Metrics Reloaded*, and *MONAI* emerge as considerably more accurate than *Plastimatch* and *seg-metrics*. Despite *Google DeepMind* exhibits a higher mean difference for certain voxel sizes when compared to *Metrics Reloaded* or *MONAI*, its superior performance with anisotropic voxel sizes and consistently low standard deviations (Table 1) establish it as the most accurate and robust open-source implementation for the HD computation. Notably, differences among implementations are less pronounced for HD_{100} , where percentile calculation can be simply replaced with the *max* function¹³. We can therefore provide a clear and straightforward answer to the question in the title of this paper: *no, because implementation matters!*

Usage Recommendations. Basing on the performed evaluation and obtained experimental results, we recommend the following for the usage of the HD computation: (i) if possible, use the same implementation that was used in the study under comparison, (ii) list and cite explicitly the implementation used, (iii) observe potential performance over-/under-estimation by using different implementations, and (iv) use *Google DeepMind* that overall results in most accurate HD measurements. We encourage researchers to retrospectively discern potential discrepancies in their results, and stimulate the convergence towards a unified implementation and consistent performance evaluation reporting.

5 Conclusions

Our findings should encourage the medical imaging community towards standardizing the HD implementations, so as to foster objective, reproducible and consistent performance comparisons. Last but not least, we would like to acknowledge the authors of the open-source implementations [17,16,3,28,9] for their valuable contributions; our study should not be regarded as a critique, but rather as a constructive step towards proper metrics implementation and usage.

Acknowledgments. This study was supported by the Slovenian Research and Innovation Agency (ARIS) under projects No. J2-4453, J2-50067 and P2-0232, and by the European Union Horizon project ARTILLERY under grant agreement No. 101080983.

¹³ This is indeed the case for *Plastimatch* that does not implement HD_{100} as a 100-percentile but uses a separate calculation pipeline based on *max* of all distances.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alt, H., Guibas, L.J.: Discrete geometric shapes: matching, interpolation, and approximation. In: Handbook of Computational Geometry, chap. 3, pp. 121–153. Elsevier (2000). <https://doi.org/10.1016/B978-0-444-82537-7.X5000-1>
2. Aydin, O.U., Taha, A.A., Hilbert, A., et al.: On the usage of average Hausdorff distance for segmentation performance assessment: hidden error when used for ranking. *Eur. Radiol. Exp.* **5**, 4 (2021). <https://doi.org/10.1186/s41747-020-00200-2>
3. Cardoso, M.J., Li, W., Brown, R., et al.: MONAI: an open-source framework for deep learning in healthcare. arXiv:2211.02701 (2022). <https://doi.org/10.48550/arXiv.2211.02701>
4. Duprez, M., Bordas, S.P.A., Bucki, M., et al.: Quantifying discretization errors for soft tissue simulation in computer assisted surgery: a preliminary study. *Appl. Math. Model.* **77**, 709–723 (2020). <https://doi.org/10.1016/j.apm.2019.07.055>
5. Fedorov, A., Beichel, R., Kalpathy-Cramer, J., et al.: 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magn. Reson. Imaging* **30**, 1323–1341 (2012). <https://doi.org/10.1016/j.mri.2012.05.001>
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Distance transforms of sampled functions. *Theory Comput.* **8**, 415–428 (2012). <https://doi.org/10.4086/toc.2012.v008a019>
7. Hausdorff, F.: Grundzüge der Mengenlehre [Basics of Set Theory]. Leipzig Viet, Leipzig, Germany (1914), <https://archive.org/details/grundzgedermen00hausuoft/>
8. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 850–863 (1993). <https://doi.org/10.1109/34.232073>
9. Jia, J., Staring, M., Stoel, B.C.: seg-metrics: a Python package to compute segmentation metrics. medRxiv (2024). <https://doi.org/10.1101/2024.02.22.24303215>
10. Jungeblut, P., Kleist, L., Miltzow, T.: The complexity of the Hausdorff distance. *Discrete Comput. Geom.* **71**, 177–213 (2024). <https://doi.org/10.1007/s00454-023-00562-5>
11. Karimi, D., Salcudean, S.E.: Reducing the Hausdorff distance in medical image segmentation with convolutional neural networks. *IEEE Trans. Med. Imaging* **39**, 499–513 (2020). <https://doi.org/10.1109/TMI.2019.2930068>
12. Klette, R., Rosenfeld, A.: Metrics. In: Digital Geometry: Geometric Methods for Digital Picture Analysis, chap. 3, pp. 77–116. Elsevier (2004). <https://doi.org/10.1016/B978-1-55860-861-0.X5000-7>
13. Li, W., Liang, Z., Ma, P., Wang, R., Cui, X., Chen, P.: Hausdorff GAN: improving GAN generation quality with Hausdorff metric. *IEEE Trans. Cybern.* **52**, 10407–10419 (2022). <https://doi.org/10.1109/TCYB.2021.3062396>
14. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3D surface construction algorithm. In: 14th Conference on Computer Graphics and Interactive Techniques - SIGGRAPH'87. pp. 163–169. ACM (1987). <https://doi.org/10.1145/37401.37422>
15. Mackay, K., Bernstein, D., Glocker, B., Kamnitsas, K., Taylor, A.: A review of the metrics used to assess auto-contouring systems in radiotherapy. *Clin. Oncol.* **35**, 354–69 (2023). <https://doi.org/10.1016/j.clon.2023.01.016>

16. Maier-Hein, L., Reinke, A., Godau, P., et al.: Metrics reloaded: recommendations for image analysis validation. arXiv:2206.01653v7 (2023). <https://doi.org/10.48550/arXiv.2206.01653>
17. Nikolov, S., Blackwell, S., Zverovitch, A., et al.: Clinically applicable segmentation of head and neck anatomy for radiotherapy: deep learning algorithm development and validation study. *J. Med. Internet Res.* **23**, e26151 (2021). <https://doi.org/10.2196/26151>
18. Podobnik, G., Ibragimov, B., Strojjan, P., Peterlin, P., Vrtovec, T.: vOARiability: Interobserver and intermodality variability analysis in OAR contouring from head and neck CT and MR images. *Med. Phys.* **51**, 2175–2186 (2024). <https://doi.org/10.1002/mp.16924>
19. Reinke, A., Tizabi, M.D., Sudre, C.H., et al.: Common limitations of image processing metrics: a picture story. arXiv:2104.05642v8 (2023). <https://doi.org/10.48550/arXiv.2104.05642>
20. Ryu, J., Kamata, S.: An efficient computational algorithm for Hausdorff distance based on points-ruling-out and systematic random sampling. *Pattern Recognit.* **114**, 107857 (2021). <https://doi.org/10.1016/j.patcog.2021.107857>
21. Sangineto, E.: Pose and expression independent facial landmark localization using Dense-SURF and the Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 624–638 (2013). <https://doi.org/10.1109/TPAMI.2012.87>
22. Schroeder, W., Martin, K., Lorensen, B.: *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, 4 edn. (2006), <https://isbndb.com/book/9781930934191>
23. Sim, D.G., Kwon, O.K., Park, R.H.: Object matching algorithms using robust Hausdorff distance measures. *IEEE Trans. Image Process.* **8**, 425–429 (1999). <https://doi.org/10.1109/83.748897>
24. Sim, D.G., Park, R.H.: Two-dimensional object alignment based on the robust oriented Hausdorff similarity measure. *IEEE Trans. Image Process.* **10**, 475–483 (2001). <https://doi.org/10.1109/83.908541>
25. Taha, A.A., Hanbury, A.: An efficient algorithm for calculating the exact Hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 2153–2163 (2015). <https://doi.org/10.1109/TPAMI.2015.2408351>
26. Taha, A.A., Hanbury, A.: Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Med. Imaging* **15**, 29 (2015). <https://doi.org/10.1186/s12880-015-0068-x>
27. Vrtovec, T., Močnik, D., Strojjan, P., Pernuš, F., Ibragimov, B.: Auto-segmentation of organs at risk for head and neck radiotherapy planning: from atlas-based to deep learning methods. *Med. Phys.* **47**, e929–50 (2020). <https://doi.org/10.1002/mp.14320>
28. Zaffino, P., Raudaschl, P., Fritscher, K., Sharp, G.C., Spadea, M.F.: Technical note: plastimatch mabs, an open source tool for automatic image segmentation. *Med. Phys.* **43**, 5155 (2016). <https://doi.org/10.1118/1.4961121>