

Article

A Deep Learning Approach to Assist in Pottery Reconstruction from Its Sherds

Matheus Ferreira Coelho Pinho ^{1,*},[†], Guilherme Lucio Abelha Mota ^{1,†} and Gilson Alexandre Ostwald Pedro da Costa ^{1,†}

Post-Graduation Program in Computational Sciences and Mathematical Modeling, Institute of Mathematics and Statistics, Rio de Janeiro State University, São Francisco Xavier, 524, Rio de Janeiro 20550-900, Brazil; guimota@ime.uerj.br (G.L.A.M.); gilson.costa@ime.uerj.br (G.A.O.P.d.C.)

* Correspondence: matheus.coelho@pos.ime.uerj.br

† These authors contributed equally to this work.

Abstract: Pottery is one of the most common and abundant types of human remains found in archaeological contexts. The analysis of archaeological pottery involves the reconstruction of pottery vessels from their sherds, which represents a laborious and repetitive task. In this work, we investigate a deep learning-based approach to make that process more efficient, accurate, and fast. In that regard, given a sherd's digital point cloud in a standard, so-called canonical position, the proposed method predicts the geometric transformation, which moves the sherd to its expected normalized position relative to the vessel's coordinate system. Among the main components of the proposed method, a pair of deep 1D convolutional neural networks trained to predict the 3D Euclidean transformation parameters stands out. Herein, rotation and translation components are treated as independent problems, so while the first network is dedicated to predicting translation moments, the other infers the rotation parameters. In practical applications, once a vessel's shape is identified, the networks can be trained to predict the target transformation parameter values. Thus, given a 3D model of a complete vessel, it may be virtually broken down countless times for the production of sufficient data to meet deep neural network training demands. In addition to overcoming the scarcity of real sherd data, given a virtual sherd in its original position, that procedure provides paired canonical and normalized point clouds, as well as the target Euclidean transformation. The herein proposed 1D convolutional neural network architecture, the so-called PotNet, was inspired by the PointNet architecture. While PointNet was motivated by 3D point cloud classification and segmentation applications, PotNet was designed to perform non-linear regressions. The method is able to provide an initial estimate for the correct position of a sherd, reducing the complexity of the problem of fitting candidate pairs of sherds, which could be then carried out by a classical adjustment method like ICP, for instance. Experiments using three distinct real vessels were carried out, and the reported results suggest that the proposed method can be successfully used for aiding pottery reconstruction.



Academic Editors: Piotr Kuroczyński and Fabrizio Ivan Apollonio

Received: 13 February 2025

Revised: 12 March 2025

Accepted: 31 March 2025

Published: 8 May 2025

Citation: Pinho, M.F.C.; Mota, G.L.A.; da Costa, G.A.O.P. A Deep Learning Approach to Assist in Pottery Reconstruction from Its Sherds. *Heritage* **2025**, *8*, 167.

<https://doi.org/10.3390/heritage8050167>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: cultural heritage; pottery vessels; pottery sherds; deep learning; artificial neural networks; 3D models

1. Introduction

The analysis of pottery in Archaeology allows for the extraction of information related to the dating of a site and provides strong evidence of the technologies employed for its production. The recognition of pottery types can pinpoint the archaeological site in time

and determine its spatial distribution, highlighting the cultural tradition of that society and the sociocultural aspects involved in its use.

Despite the highly informative potential that pottery sherds bring to the archaeological context, they also pose a significant challenge during the post-excavation phase since they are found in large quantities and consume significant laboratory space. Indeed, the vast number of sherd remains means that vessel reconstruction is a meticulous and time-intensive endeavor, often straining the time and resources required for their proper analysis and interpretation.

In addition to being time-consuming, the identification and classification of pottery sherds require prior knowledge and experience. In the process of restoring archaeological pottery, sherds that are supposed to belong to the same vessel are grouped together based on the classification of pottery traditions. Sherds are compared one by one, with the aim of fitting them together until the vessel is reconstructed. In the case of archaeological pottery, missing sherds are common, further complicating the fitting process. Another challenging factor arises when there is a large number of sherds belonging to the same pottery tradition: the greater the number of sherds, the more time it will take to classify them into a pottery type and then proceed with the fitting of the sherds and reconstruction of the vessel.

In this work, we propose a method to aid the reconstruction of pottery vessels from their sherds. The method is founded on 3D models and deep 1D convolutional neural networks. More specifically, a 3D model of a sherd in the form of a point cloud is provided as input to the method. The outcome is a transformation matrix that positions the sherd into the vessel's coordinate system. The proposed approach involves training artificial neural networks for any specific vessel model. In the literature, this class of approach is regarded as "sherds orientation". Our method exploits two networks to predict the respective Euclidean transformation matrix. The first infers translation moments, whereas the other predicts the rotation parameters. For complete vessel reconstruction, it would be necessary to repeat the prediction for all its sherds.

The proposed network architecture, the so-called PotNet, is inspired by the 3D POCO Net [1] and PointNet [2] models. However, the mathematical modeling developed in this research reduces the degrees of freedom of the problem, enabling the definition of a less complex design than other networks, like PointNet and POCO Net, for instance. Thus, in addition to changing the layers on the top of its head to perform regression, the PotNet design eliminates the PointNet's feature transform layers, for instance. This alone makes training and inference processes less costly than those provided by their counterparts. The data used for training PotNet consist of a set of virtual sherds' point clouds. These virtual sherds were generated by breaking apart the 3D model of a vessel a number of times in a digital environment. The use of synthetic data, assisted by the virtual fragmentation of a vessel's 3D model, enables abundant amounts of training data generation. Such a training scheme addresses the issue of limited real-world training data, which is common in many machine learning applications, including those dedicated to Archaeology. Although the proposed method prescribes training with synthetic data, in this work, we evaluated it in predicting the original positions of real sherds produced by physically shattering real vessels.

The literature review conducted for this article, encompassing scientific publications available up to this manuscript preparation, reveals that the proposed method represents a novel application of deep learning in the field of Archaeology. As a matter of fact, during the course of an archaeological excavation, a set of sherds originating from a given vessel of interest should be identified, collected, gathered, and cataloged. Afterward, the type and approximate shape of that vessel should be estimated based on evidence, previously existing catalogs, and the respective archaeological context. That information is used to

bring about an approximate vessel 3D model in the form of a point cloud. Then, given the approximate vessel 3D model, the herein proposed neural networks are trained by using the previously mentioned virtual fragmentation procedure. Therefore, the usage of an approximate vessel's 3D model as input for the herein-presented experiments reinforces the practical applicability of the proposed procedure since the approach does not require very detailed 3D models to be trained. On the other hand, the usage of these networks presupposes the availability of 3D models of the real sherds, which may be obtained using, for instance, a manual 3D scanner either in the field or during post-excavation analysis. Thus, given the 3D models of sherds (also in the form of point clouds), one can use the herein proposed model to perform sherds orientation. Afterward, orientations of a group of sherds are used to support manual or computer-assisted sherd fitting [3–5], even in cases where there is an absence of parts of the vessels. In addition, this procedure may reduce the complexity of vessel reconstruction since the number of sherds to be simultaneously considered during sherds fitting drops to a subset of the entire amount collected after the approximate absolute positions of sherds are obtained. This proposal also provides a good clue for ICP-based matching. In addition to its potential to accelerate the reconstruction process, carrying out reconstruction in a virtual environment avoids wearing sherds out, favoring their conservation. This method highlights the importance of 3D digitization protocols in archaeological data processing based on the assumption that digitized artifacts can bring new possibilities for storing, modeling, and interpreting data, as well as making archaeological data available.

The remainder of this document is structured as follows. Section 2 provides a literature review describing a number of computational applications in Archaeology dealing with pottery sherds assembly. Section 3 describes the materials, including the physical vessels and the respective 3D models. Section 4 presents the proposed method in detail. The experimental procedure is described in Section 5. The obtained results and discussion are shown in Section 6. Finally, Section 7 presents the conclusions and points out directions for future works.

2. Related Works

This section briefly describes works that deal with the assembly of pottery sherds. As a starting point, we refer to a couple of literature surveys, i.e., [6,7].

In the literature, there are a number of sherd-matching approaches regarding 3D mosaicing [8–19]. Most of these approaches rely on traditional methods to perform sherd positioning and assembling. The main limitation of such methods, which are usually based on least squares reduction, is requiring an initial good-enough guess of the sherd's position to perform the registration, ensuring the appropriate optimization process convergence. However, possible initial position guesses for an arbitrary sherd, without any contextual information, may be virtually infinite. Thus, providing it is a complex, ill-posed problem. Our decision to employ neural networks may significantly reduce the inference search space, making the problem more computationally feasible.

Papaioannou et al. [8] present a semi-automatic method for assisting in the reconstruction of archaeological findings from 3D scanned sherds. The method carried out one-by-one sherd matching by minimizing the matching error for all pairwise candidate facets. Cooper et al. [9] use a bottom-up maximum likelihood procedure for pottery assembly. Andrews and Laidlaw [10] determine pairwise match proposals, which are probabilistically evaluated by a series of independent feature similarity modules. Alternatively, given a candidate sherd, Kampel and Sablatnig [11] match it with a reference sherd by providing an Euclidean transformation, connecting both sherds' coordinates systems. Marie and Qasrawi [12] use the shadow moiré technique to obtain the 3D model; then, the profiles and edges of the

sherds are virtually matched based on their 3D models. Given two sherds, S and T, the approach of Huang et al. [13] seeks common features on fractured surfaces in order to obtain an aligning transformation able to bring S close to T. Palmas et al. [14] propose a flexible, interactive system for vessel reconstruction that relies on sherd relation constraints delivered by the experience of an expert. The system employs a global energy minimization strategy to search for a suitable assembly. Zheng et al. [15] perform contour curve matching by estimating Euclidean transformations, with the most likely rigid transformations being selected. Finally, a global refinement alignment improves assembly accuracy. Stamatopoulos and Anagnostopoulos [16] propose an approach that takes into consideration profile information inside the core of a sherd instead of its surface. The authors argue that such an approach works even when sherds are missing, and it is neither affected by the presence of external wear and damages nor by geometrical shape and color degradation. Cohen et al. [17] exploit vessel surface markings and vessel generic models constructed by archaeologists. Sherds alignment considers a set of weighted discrete moments computed from convex hulls of the markings on a sherd's surface and reference points on their borders. Sakpere [18] extracted keypoints from sherd point clouds using Principal Component Analysis. Then, pairwise correspondences are established by employing adapted ColorSHOT descriptors, and the Iterative Closest Point algorithm is used for refining sherd alignment. Eslami et al. [19] model sherd border curve features. After Canny edge detection, sherds are fit together using wavelet transform approximation coefficients.

A few machine learning-based approaches have been proposed thus far to assist in cataloging and restoring archaeological artifacts from sherds. Some notable examples are [4,20–22].

Anichini et al. [21] present the ArchAIDE app, a framework for pottery classification based on deep neural networks aimed at assisting archaeological cataloging in the excavation field. The approach consists of two primary pot-type classification methods: shape-based and decoration-based. Shape-based identification relies on the 2D outline of a sherd's fracture profile. Using a deep learning model, the so-called OutlineNet (inspired by PointNet architecture) encodes shape features for pot-type recognition. To mitigate data limitations, synthetic training samples are generated by computationally fragmenting 3D vessel reconstructions. The decoration-based classification employs an image classification approach based on a convolutional neural network. The decoration-based approach deploys a transfer learning approach using a pre-trained ResNet-50 model to analyze patterns and colors on sherd surfaces. When compared to the approach proposed here, ArchAIDE has only two major similarities to ours: it generates training data by virtually breaking 3D pot models, and it uses deep learning to fulfill its aims. However, the aim of ArchAIDE is pot-type classification, while our method is designed for sherd orientation. Another important difference is that our approach deals with sherds using 3D point clouds, whereas ArchAIDE uses a 2D projection of a sherd's fracture profile during classification, although both designs are inspired by the PointNet architecture.

Kashihara [20] proposes a system based on computational intelligence to assist in the restoration process. The Real-Coded Genetic Algorithm (RCGA) finds the fitness function from image similarity between the target and correct patterns in plane images taken at multiple camera angles.

Rasheed and Nordin [4] present a methodology for automatically classifying and reconstructing broken pottery sherds using neural networks, consisting of two main phases: first, sherds are classified into groups based on color and texture features; once classified, the sherds undergo a reconstruction process that involves extracting geometric features from 3D models and applying a neural network to determine optimal matches between sherd pairs. The method employs Singular Value Decomposition (SVD) for the alignment,

rotation, and transformation of sherds to achieve accurate reconstruction, even in cases with missing pieces; however, the system needs an initial guess for sherd positioning and pair-candidate matching.

To the best of our knowledge, Kim et al. [22] present the initiative that comes closest to our work. The work employs a Dynamic Graph Convolutional Neural Network (DGCNN) with skip-dense connections for both classifying the origin of sherds among groups of pottery types and for predicting its location along a vessel's revolution axis. The sherd's position information delivered by that work is, however, limited in relation to the outcome of the method proposed in the current work. While Kim et al. [22] only infer a sherd's translation along the vertical axis, the proposed method predicts multiple parameters, from which a complete Euclidean transformation matrix is derived, except for the angle of the revolution axes. Such a value was not considered in our work since, for any sherd located in the body of revolving, solid-shaped vessels, that angle is undetermined. Additionally, while only synthetic data were used to test the method proposed by Kim et al. [22], in this work, we evaluate the proposed method using real sherds produced by the fragmentation of real vessels.

3. Materials

The objects employed in this research are shown in Figure 1. The figure shows three real-world pottery vessels named LV, MV, and SV, which are acronyms for Large Vessel, Medium Vessel, and Small Vessel. Regarding shattering, the vessels were simply dropped from an approximate height of 1.5 m. Such a process does not allow for controlling the resultant sherd sizes and may cause loss of vessel mass due to tiny fragments; this provides great potential to increase reconstruction uncertainty. The resulting sherd sizes range from small chips to a maximum length of 10 cm. To avoid excessive uncertainty, only sherds larger than 3 cm were considered in this research, which amounts to 57, 20, and 21 physical sherds for LV, MV, and SV, respectively.

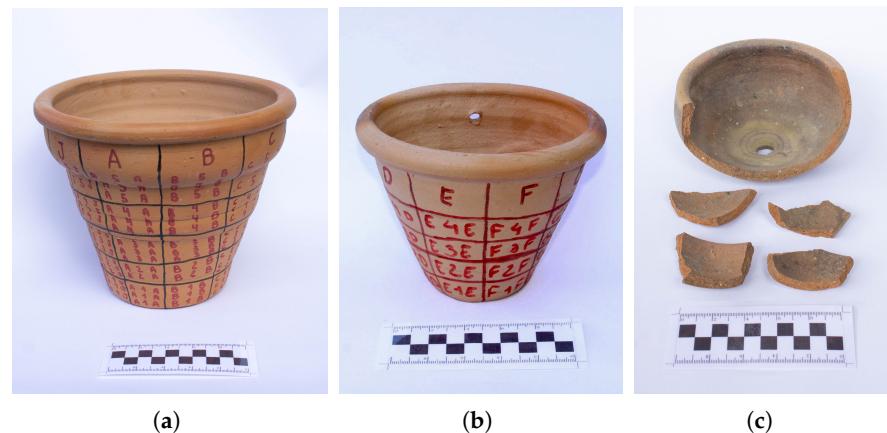


Figure 1. Photographs of the vessels: (a) LV, acronym for Large Vessel; (b) MV, for Medium Vessel; and (c) SV, for Small Vessel.

After shattering, the resulting sherds were digitized using a structured-light stereoscopic 3D scanner, the Virtuo Vivo™ intraoral scanner [23,24]. Despite having been designed for dental use, after a short time of operational training, it proved to be precise enough for generating 3D models of sherds. To simplify sherd identification hereinafter, they were named by concatenating the vessel acronym to its scanning order, e.g., SV21.

Once digitized, the corresponding sherds were manually matched together using the Blender 3D mesh modeling software [25]. Once the reference for matching real-world sherds is known, the process of individually fitting each candidate pair of sherds is replicated in a

virtual environment. This continues until all fragments are correctly assembled, allowing for the approximate shape of the vessel to be recovered. The reassembled vessels are shown in Figure 2, where the readers can easily notice the sherds' boundaries.

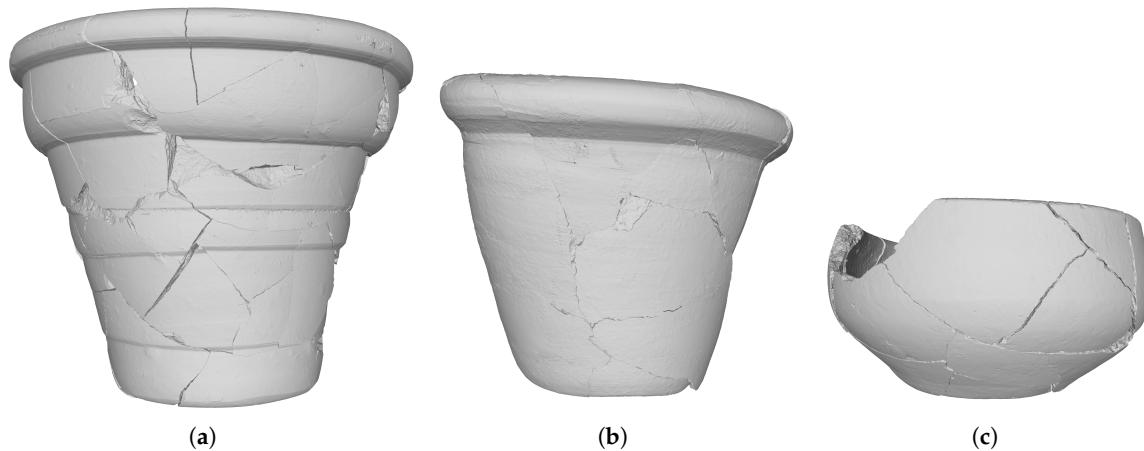


Figure 2. Virtually reassembled vessels: (a) LV; (b) MV; and (c) SV.

To virtually restore those vessels, it was necessary to stitch the sherds together in a single 3D model for each vessel. Considering the high precision of the scanned sherds' meshes, making that procedure feasible required applying a decimation process to every mesh. In this work, the Quadric Edge Collapse Decimation filter [26] available in the Meshlab API [27] was employed. The restored vessels can be seen in Figure 3. Those 3D models were the input for generating the synthetic data used during training as a surrogate to overcome the lack of training data.

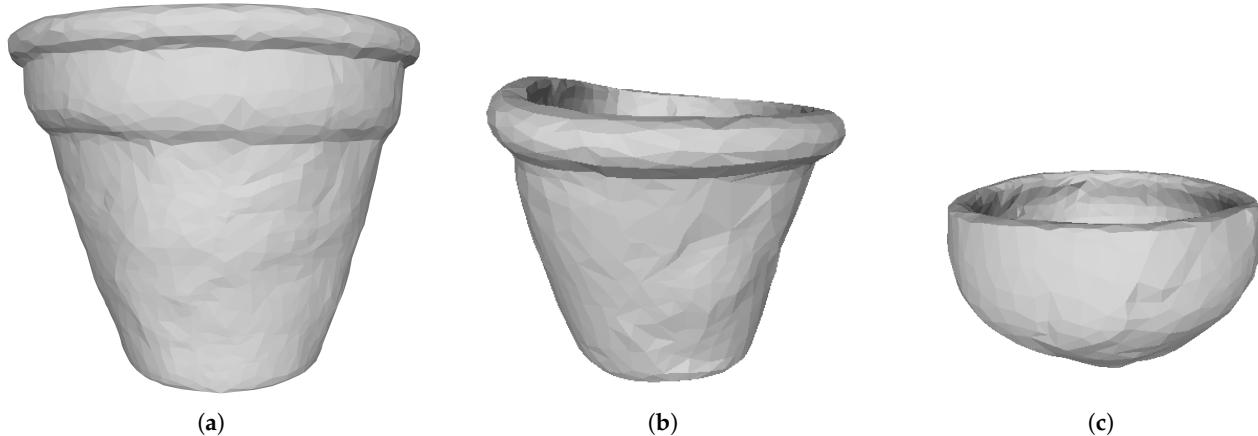


Figure 3. The 3D models of the restored vessels: (a) LV; (b) MV; and (c) SV.

4. Proposed Method

Figure 4 depicts an overview of the herein-proposed sherd orientation method. By and large, given a point cloud \mathbf{X} relative to a specific sherd expressed in its canonical coordinate system, referred to in Figure 4 as Canonical Cloud, the method produces \mathbf{X}' , a version of such a sherd cloud expressed in terms of the vessel (normalized) coordinate system. Mathematically, this process consists of applying a determined Euclidean 3D transformation in Affine space in the form of a linear operator \mathbf{T} , where $\mathbf{X}' = \mathbf{X} \cdot \mathbf{T}$, while \mathbf{X} and \mathbf{X}' are $p \times 4$ matrices, where p stands for the number of points in every cloud. Thus, every column in those matrices denotes a 3D point of the cloud expressed in homogeneous coordinates (consisting of concatenating a value of one after the respective 3D coordinates).

The coefficients of the transformation matrix were obtained through a two-branch neural network. Together, both branches predict the values of the parameter vector $\hat{\mathbf{y}}$. Based exclusively on $\hat{\mathbf{y}}$, a deterministic mathematical procedure brings about the transformation matrix \mathbf{T} .

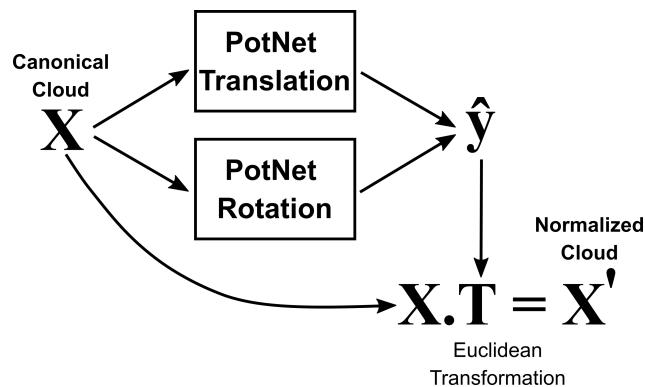


Figure 4. Method overview.

As can be noticed in Figure 4, the core of the proposed method consists of a dual-branch neural network architecture, with both predicting an outcome based on exactly the same input \mathbf{X} . The branches of the neural network use the same special-purpose backbone, the herein proposed PotNet backbone, which was trained through independent training procedures.

There is a subtle difference between the heads of each PotNet branch. While the translation branch produces a two-dimensional vector output relative to the y and z axes translation offsets, the rotation branch produces a vector of six dimensions. Given that 6D vector, the Gram-Schmidt process brings about the rotation matrix coefficients [28].

In the following sections, the most significant components of the proposed method are presented in detail.

4.1. Problem Modeling

First, we assume pottery vessels have axial symmetry in relation to the y axis, meaning virtually infinite valid orientations for a given sherd being produced by simply rotating it by an arbitrary angle around that axis. Thus, by definition, orientation prediction is an ill-conditioned problem. We note, however, that this does not prevent considering components such as the handles or shoulders of a vessel. An additional source of ill-conditioning is the individual sherd coordinate system. Thus, in the absence of a natural standard for the sherd systems, it is imperative to establish one.

To overcome such ill-conditioning problems, we propose simplifying it on both sides. On the one hand, we reduce the vessel system by one degree of freedom by eliminating the angle relative to the rotation axis. This brings about the so-called normalized system¹, which consists of bringing the sherd's centroid to the yz plane. In this work, we approximate the centroid through the arithmetic mean of the points on the original sherd's point cloud. On the other hand, we defined an absolute, unique inner system for each sherd. Such a system, which only depends on the sherd's shape, was named the “canonical system”.

4.1.1. Normalized Cloud

Normalized clouds are obtained by eliminating the rotation angle around the vessel's revolution axis, meaning moving the sherds' centroid to the yz plane. This creates a standard position for every sherd relative to the vessel's system.

Bringing the centroid to the desired position is made easier by defining a spherical coordinate system (see Figure 5). Given (x, y, z) , the Cartesian coordinates of the cen-

troid C , Equation (1) presents the rotation matrix that brings such a point cloud to its normalized position.

$$\mathbf{R}_\varphi = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (1)$$

where $\varphi = \tan^{-1}(x/z)$. Thus, given $\mathbf{x}_o = \{x_o, y_o, z_o\}$, a vector that expresses a given sherd's cloud point in terms of the physical vessel system, \mathbf{x}' —the vector of normalized coordinates for that point—is obtained by multiplying \mathbf{x}_o by \mathbf{R}_φ .

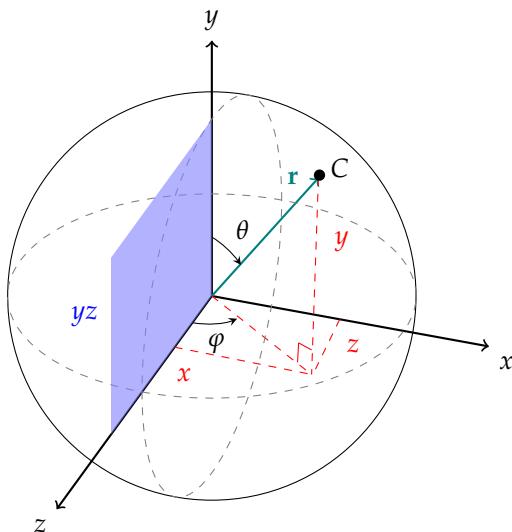


Figure 5. Spherical coordinate system, where C is the center of mass of the sherd's point cloud in the original vessel system.

4.1.2. Canonical Cloud

Canonical cloud stands for the form in which sherds are presented at the input of our approach. Its respective coordinate system helps control the degrees of freedom of the sherd orientation process, creating a deterministic and exclusive canonical model for any given sherd. The origin of the canonical system coincides with the centroid of the sherd, while the vector space aligns with the directions of the minimum-volume, bounding, rectangular cuboid.

Bringing a point cloud from its arbitrary sherd coordinate system to its canonical form involves three steps. Firstly, the sherd's minimal bounding rectangular cuboid must be obtained. Then, the cloud points are translated, bringing the sherd's centroid to the origin of the coordinate system. Finally, Singular Value Decomposition (SVD) [29] provides the matrix that projects the cloud into the canonical coordinate system.

Computing the minimal volume enclosing the bounding cuboid is performed using a faster alternative to the O'Rourke [30] approach: the Jylänki [31] algorithm implemented in the Trimesh API [32].

SVD basically consists of the factorization of a rectangular matrix \mathbf{A} into the product of three other matrices: an $m \times m$ orthogonal matrix \mathbf{U} , the columns of which are the left singular values of \mathbf{A} ; a rectangular $m \times n$ diagonal matrix Σ containing the singular values of \mathbf{A} in descending order; and the transpose of an $n \times n$ orthogonal matrix \mathbf{V} , the columns of which are the right singular values of \mathbf{A} in the following form:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \quad (2)$$

In this work, \mathbf{A} is an 8-by-3 matrix containing the coordinates of the vertices of the minimum-volume, bounding, rectangular cuboid with the centroid at the origin. Σ has the same dimensions as \mathbf{A} , while \mathbf{U} and \mathbf{V} have orders of 8 and 3, respectively. \mathbf{AV} multiplication projects the cuboid vertices into the canonical space.

As a visual example, in Figure 6, the reddish object refers to an arbitrary sherd in its original position relative to the vessel, the bluish object represents the sherd in the normalized position, and the greenish object corresponds to the sherd in the canonical position. At this point, it is important to observe that no knowledge about a sherd's original position in the vessel is needed to place it in its unique canonical position.

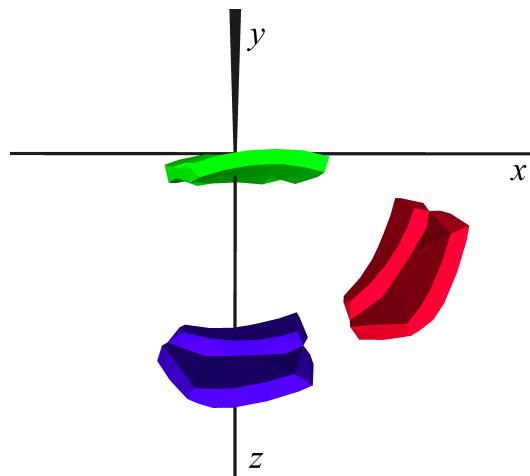


Figure 6. The same sherd in its original position relative to the vessel's coordinate system (in red) in the normalized position (in blue) and in the canonical position (in green).

4.1.3. Target Euclidean Transformation

Given the normalized and canonical point clouds of a sherd, it is necessary to obtain the transformation that relates both clouds. The canonical-to-normalized position transformation is the one used to create the training target \mathbf{y} , which will later be compared with PotNet prediction $\hat{\mathbf{y}}$.

In this work, the Kabsch algorithm [33,34] is used to compute the rotation components of the training targets \mathbf{y} . The Kabsch algorithm is a method to calculate the optimal rotation matrix that minimizes the root mean squared deviation between two paired sets of points. The translation coefficients are computed as the distance between the centroids of the normalized and canonical point clouds.

In this work, we propose the use of a six-dimensional vector for modeling the rotation matrix. The full 3-by-3 rotation matrix may be recovered using the cross-product restricted to the orthogonality constraint.

4.2. PotNet Model

In this work, we introduce PotNet, a neural network model dedicated to non-linear regression, the design of which was inspired by the first network of 3D POCO Net [1] and PointNet [2] without T-net modules. The differences between their architectures reside in the initial layers, activation functions, and output layer size. When receiving as input a point cloud in the form of a matrix, the rows of which represent the points and columns the coordinates, PotNet predicts the transformation parameters associated with that point cloud.

By and large, PotNet's architecture is composed of a number of single-dimensional convolution layers, followed by a global uni-dimensional max pooling layer, aiming at bringing about a number of features for the input point cloud. Such features pass through a

multilayer perceptron network built on top of the convolutional layers. Most of the PotNet layers are followed by ReLU activation functions, the only exception being the output layer, which is coupled to a linear activation function. In this work, we experimented with five convolutional layers, containing, respectively, 64, 64, 64, 128, and 1024 filters. The multilayer perceptron hidden layers, which employ batch normalization, encompass 512 and 256 neurons, while the output layer contains N neurons. The number of neurons in the output layer can be 2 or 6, depending on if it is a translation or a rotation network branch. The proposed PotNet network architecture is summarized in Table 1.

Table 1. PotNet architecture summary: M corresponds to the number of points in the point cloud, and N denotes the dimension of the output vector. N assumes 6 for rotation and 2 for translation.

Layer Type	Output Shape	Parameters
Input	$M \times 3$	0
Conv 1D $1 \times 3/64 + \text{ReLU}$	$M \times 64$	256
Conv 1D $1 \times 64/64 + \text{ReLU}$	$M \times 64$	4160
Conv 1D $1 \times 64/64 + \text{ReLU}$	$M \times 64$	4160
Conv 1D $1 \times 64/128 + \text{ReLU}$	$M \times 128$	8320
Conv 1D $1 \times 128/1024 + \text{ReLU}$	$M \times 1024$	132,096
Global Max Pooling 1D	M	0
Dense 512	512	$(M + 1) \times 512$
Batch Normalization	512	2048
ReLU	512	0
Dense 256	256	131,328
Batch Normalization	256	1024
ReLU	256	0
Dropout	256	0
Dense N	N	$257 \times N$
Linear	N	0

4.3. Training Procedure

Deep learning models are characterized by their need for substantial amounts of training data. Nevertheless, the accessibility of fully restored archaeological vessel 3D models is limited due to three factors: the incompleteness of archaeological records, the high costs associated with cataloging, and the time-consuming nature of the process.

An alternative to coping with such data scarcity is to use restored models, such as the ones presented in Figure 3, to produce synthetic training data. Our approach relies on a virtual vessel shattering procedure (see Section 5.1), which can be instantiated using different parameters to deliver an arbitrary amount of synthetic sherds for a particular vessel. Then, point clouds sampled from the sherds created with the virtual shattering algorithm are processed as described in Sections 4.1.1 and 4.1.2 to control the degrees of freedom. Afterward, using the normalized and canonical point clouds as input, the respective target Euclidean transformation is computed using the algorithm described in Section 4.1.3.

As mentioned in Section 4, the PotNet Backbone is specialized into two models, one for rotation and another for translation. Each model is trained independently through an optimization process that minimizes a particular loss function. Therefore, parameter vectors \mathbf{y} gather the rotation components \mathbf{y}_R and translation moments \mathbf{y}_T . As usual, losses relate the target (\mathbf{y}) and predicted ($\hat{\mathbf{y}}$) vectors. Accordingly, the respective loss functions

$\mathcal{L}_R = \text{L2}(\mathbf{y}_R, \hat{\mathbf{y}}_R)$ and $\mathcal{L}_T = \text{L2}(\mathbf{y}_T, \hat{\mathbf{y}}_T)$ are used for each specialized network training. The L2 norm between the target and predicted vectors is shown in Equation (3).

$$\text{L2}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{(\hat{\mathbf{y}} - \mathbf{y})^T (\hat{\mathbf{y}} - \mathbf{y})} \quad (3)$$

5. Experimental Setup

Concerning the experimental procedure, this section provides details about method implementation and dataset description.

5.1. Virtual Shattering Procedure

To address the well-documented high data volume demanded for training deep learning models, we devised a way to automatically produce synthetic sherds. In that regard, we employed the Cell Fracture tool, a native feature of Blender software conceived to break up digital object models into an arbitrary number of sherds. The tool is based on a 3D Voronoi Diagram implementation designed for polyhedron fragmentation [35,36]. Cell Fracture implements a non-deterministic approach, the outcome of which is influenced by some parameters, including the maximum number of fragments provided and a seed for noise generation ranging from 0 to 1. For a regular object like a cube, 0 means more regular shapes, and 1 means more natural shapes for the breaking results. For uneven objects, the Voronoi point distribution is always random. Thus, the results have uneven and natural shapes. So, even when fixing the noise parameter value, the point distribution in the breakage process is still random. In this work, the noise parameter value was fixed at 0.5. The maximum number of sherds in each break was set up to be the upper bound of the total number of real sherds for each vessel.

With the Cell Fracture tool, the 3D model of the restored LV vessel (refer to Figure 3a) was broken up 2000 times, with 60 being the maximum number of sherds (synthetic sherds) in each run. Out of those, the product (synthetic sherds) of 1800 runs constitutes the training set. The synthetic sherds created in the other 200 runs constitute the test set.

The 3D model of the restored MV and SV vessels (refer to Figure 3b,c) were broken up 3300 times each, with a maximum of 30 sherds in each run. The product of 3000 runs composes the training set, while the other 300 runs compose the test set. Performing more breaking runs for these two vessels, when compared to the LV, looks to standardize the number of sherds for these 3 vessels since, for MV and SV, fewer sherds are produced in each break.

5.2. Synthetic Sherd Datasets

Synthetic sherds are normalized and then placed in the canonical position; afterward, the respective target transformation matrices are calculated. In sequence, the Poisson-disk sampling filter [37] of the Meshlab software was applied to every virtual sherd (in canonical position) in order to standardize the number of points in their point clouds. The resulting point clouds were then represented by a 1024×3 matrix.

Part of the synthetic sherds were discarded, and only those sherds with respective canonical rectangular cuboids exceeding 10 cm^3 were selected to compose the datasets. Table 2 shows the number of synthetic sherds used for training and testing the PotNet instances.

Table 2. Quantities of synthetic sherds in the training and testing sets.

Vessel	Train	Test	Total
LV	78,892	8695	87,587
MV	76,664	7816	84,480
SV	77,377	7694	85,071

We recall that 57, 20, and 21 real sherds were selected from the physical shattering of the real LV, MV, and SV vessels, respectively. Those sherds were also used to evaluate the PotNet instances, as reported in Section 6.

5.3. Training

A rotation-translation PotNet set was trained for each vessel independently, resulting in three different instances of the rotation and translation networks. The training procedures employed the same hyperparameter values, as follows. The batch size was 128. The Adam optimizer [38,39] was used with standard values (i.e., the exponential decay rate for the 1st and 2nd moments were 0.9 and 0.999, respectively). The learning rate was fixed at 0.001. Networks were trained until the maximum number of epochs were reached: a total of 1500 for the rotation network and 1000 for the translation network.

5.4. Computational Environment and Requirements

Experiments were performed on a 12th generation Intel Core i9-12900F hexacore using a 5.1 GHz processor computer with 128 GB RAM, running the Ubuntu 22.04.3 LTS operating system, kernel Linux 6.2.0-37-generic, Python 3.10.12, and Tensorflow 2.10. This machine is equipped with a GPU NVIDIA GeForce RTX 4090 with 24 GB VRAM. The VRAM peak requirements were 11.92 MB during inference and 2.4 GB for training. The total inference time was 0.58 s per sherd.

6. Results and Discussion

This section presents and analyzes the outcomes for both the synthetic and real sherd test sets. This analysis concerns prediction quality while bringing about the transformation matrices, \mathbf{T} , that take the sherds canonical point clouds to the normalized space.

After prediction, every point \mathbf{x}_i in the canonical cloud is multiplied by the corresponding \mathbf{T} matrix, resulting in $\hat{\mathbf{x}}'_i$, a predicted point that should fit \mathbf{x}'_i , its homologous aspect in the ground truth normalized cloud. The predicted sherd point positions were compared to the respective reference normalized positions in order to compute the relative errors. Since the predicted and normalized point clouds of a sherd have exactly the same points, the exact error between those two clouds can be computed. The errors are calculated considering each point and each coordinate axis. From the differences of each point position, we computed the Root Mean Square Error (RMSE), given by Equation (4).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{x}}'_i - \mathbf{x}'_i\|_2^2} \quad (4)$$

We also computed the RMSE relative to each coordinate axis, (x, y, z) , considering all points in the test dataset. Referring to Equation (4), instead of considering vector distances, we simply computed the differences of each particular coordinate. In addition, we grouped errors relative to axes x , y , and z to present the standard deviation values (STD).

6.1. Results for the Synthetic Test Bset

The number of sherds in the synthetic test set for each vessel is given in Table 2, while Table 3 presents the values of the RMSE, RMSE (x, y, z) , and STD obtained for the whole synthetic test set. On average, the positioning error between the predicted and normalized clouds of the synthetic sherds is less than 1.9 cm for all vessels (column “RMSE”), which we consider small in relation to the size of the vessels. The standard deviation of the errors for all vessels is not more than 1 mm, showing that the errors in the predictions are quite stable.

Regarding translation, the “RMSE (x, y, z)” column shows that the axis with greater variation is the vertical one (y), which represents the height of the vessel; this coordinate is the harder one to predict. The table shows consistent results for this observation, with the error related to the y axis being the larger one in all cases.

Table 3. Evaluation metrics for the whole synthetic test set of LV, MV, and SV (values in meters).

	RMSE	RMSE (x, y, z)	STD
LV	0.025	(0.01, 0.021, 0.009)	0.001
MV	0.017	(0.008, 0.014, 0.007)	0.0004
SV	0.013	(0.008, 0.009, 0.006)	0.0002

6.2. Results for the Real-World Test Set

The real test set constitutes 57, 20, and 21 real sherds selected from the physical shattering of the real LV, MV, and SV vessels, respectively. Those sherds were 3D scanned, decimated by a factor of 3000, and remeshed (a tessellation algorithm was used to connect the new vertices, producing a new mesh).

In these experiments, with the rotation and translation networks trained on the synthetic sherd datasets, we aimed to validate the proposed method using real objects, showing that it can be employed in real-world problems. At this point, it is important to observe that the real sherds were not used for training the deep networks.

After decimation, each sherd was sampled to 1024 points using the Poisson-disk sampling Meshlab filter, resulting in a denser point cloud representation. Each point cloud was placed in its canonical position and inputted to the trained rotation and translation models. It is worth mentioning that, for the real test set, since each real sherd position comes from its natural position in the reassembled vessels (see Figure 2), obtaining their ground truth normalized positions is trivial by solely eliminating the φ rotation angle (see Figure 5 and Equation (1)).

For visualization purposes, regarding the test set, it is sufficient to multiply the sherd’s point cloud in the predicted position in normalized space by the inverse of the R_φ matrix, meaning rotating the cloud around the y axis counterclockwise, bringing the sherd from the canonical space to the vessel’s space. This will cause the sherd to return to its true position within the vessel’s coordinate system, ensuring that the rotation and translation predictions are maintained. The results are depicted in Figures 7–9.

Despite variations among individual sherds, overall, the qualitative results seen in Figures 7–9 are close to the expected results, showing that the rotation and translation models closely approximate the expected outcomes of a pottery vessel reconstruction process from its sherds.

Table 4 shows the error metrics obtained for the real sherds of LV, MV, and SV vessels, which are consistent with the results obtained for the synthetic test set (shown in Table 3). The standard deviation of the errors is less than 1 mm for all vessels, showing, again, that the prediction errors are stable.

Table 4. Evaluation metrics for the whole real test set of LV, MV, and SV (values in meters).

	RMSE	RMSE (x, y, z)	STD
LV	0.034	(0.009, 0.026, 0.017)	0.0006
MV	0.019	(0.007, 0.01, 0.012)	0.0002
SV	0.023	(0.01, 0.012, 0.013)	0.0003

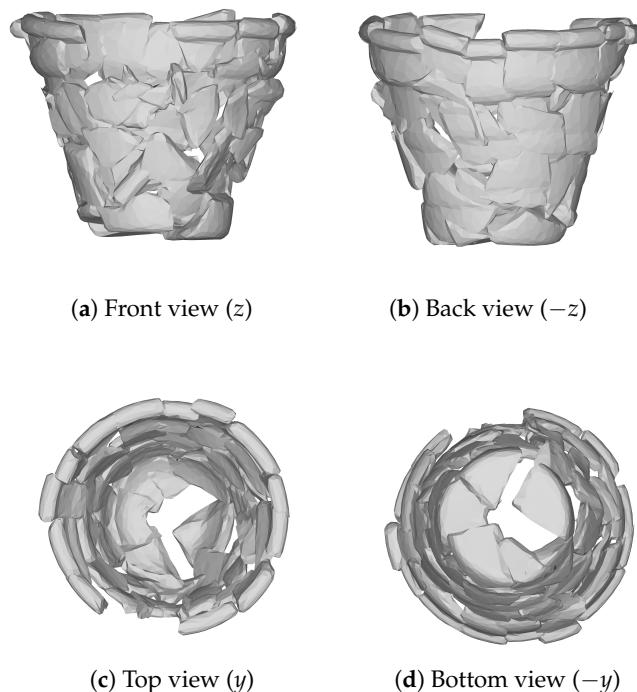


Figure 7. Four different views of the sherds from the LV real-world test set placed at the predicted positions and rotated counterclockwise by the inverse of the R_φ matrix.

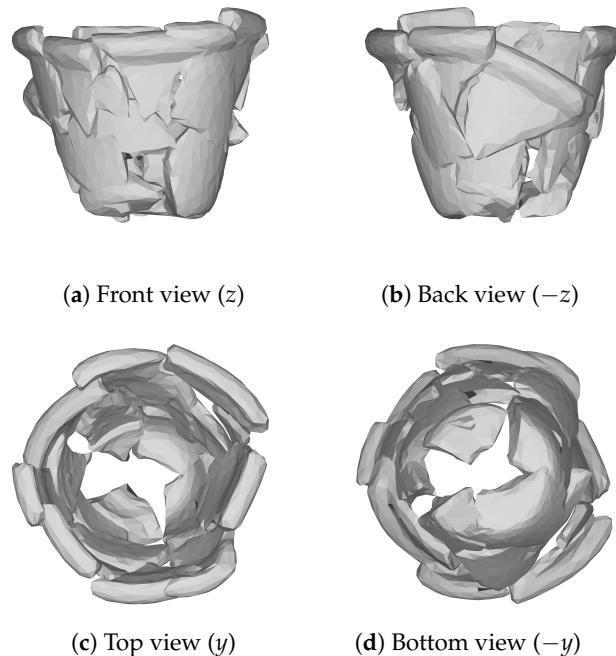


Figure 8. Four different views of the sherds from the MV real-world test set placed at the predicted positions and rotated counterclockwise by the inverse of the R_φ matrix.

We believe that the slightly poorer results for the real sherds are due to the more irregular surfaces of the real-world sherds, as compared to the computer-generated ones, which have smoother, regular surfaces; this results in a more complex distribution of vertices on the real sherds, making network inference more challenging. The small decline in performance observed for real-world fragments may be a positive indication of the generality of the proposal, even if subjected to a training process based on synthetic data of limited quality. This result reinforces its transferability for practical contexts.

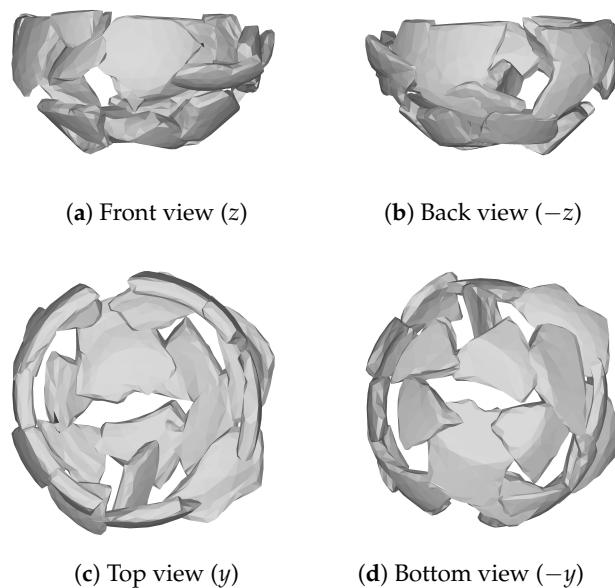


Figure 9. Four different views of the sherds from the SV real-world test set placed at the predicted positions and rotated counterclockwise by the inverse of the R_φ matrix.

Table 5 shows the best, average, and worst predictions for particular real sherds regarding different vessels. The sherds were selected based on their RMSE values. The best cases correspond to the lowest RMSE value, the average cases correspond to the median, and the worst cases correspond to the highest RMSE value.

The sherds indicated in Table 5 are shown in Figures 10–12 (for LV), Figures 13–15 (for MV) and Figures 16–18 (for SV). The sherds in the reference positions are shown in red; the sherds in the predicted positions in normalized space are shown in green.

Table 5. Evaluation metrics for the best, average, and worst cases from the real-world test sets of LV, MV, and SV (values in meters).

	Sherd	RMSE	RMSE (x, y, z)	STD
Best cases	LV29	0.023	(0.013, 0.014, 0.013)	0.0002
	MV11	0.017	(0.012, 0.011, 0.005)	0.0001
	SV14	0.019	(0.013, 0.012, 0.006)	0.0002
Average cases	LV23	0.034	(0.024, 0.021, 0.013)	0.0005
	MV19	0.032	(0.015, 0.025, 0.012)	0.0005
	SV21	0.027	(0.015, 0.02, 0.009)	0.0004
Worst cases	LV21	0.038	(0.026, 0.025, 0.012)	0.0008
	MV17	0.038	(0.028, 0.023, 0.011)	0.0008
	SV11	0.033	(0.023, 0.022, 0.01)	0.0006

The prediction for sherd SV14 (Figure 16), the best case among the SV real sherds, achieved an RMSE of 0.019 m, corresponding to a distance of 1.9 cm between sherd SV14 in the expected and predicted positions. The worst case of MV, MV17, as shown in Figure 15, achieved an RMSE of 0.038 m, corresponding to a distance of 3.8 cm between the expected and predicted positions.

The qualitative results obtained for SV are visually inferior to those obtained for MV, as sherd SV21 (Figure 17) and sherd SV11 (Figure 18) were placed upside down compared to their expected positions. In terms of RMSE, the values of 0.015 (SV21) and 0.021 (SV11) are not very different from those obtained for MV because, being an average, they are

balanced by the results of the most central part of the sherd. The worst case of SV (SV11) had an error of only 0.025 (2.5 cm).

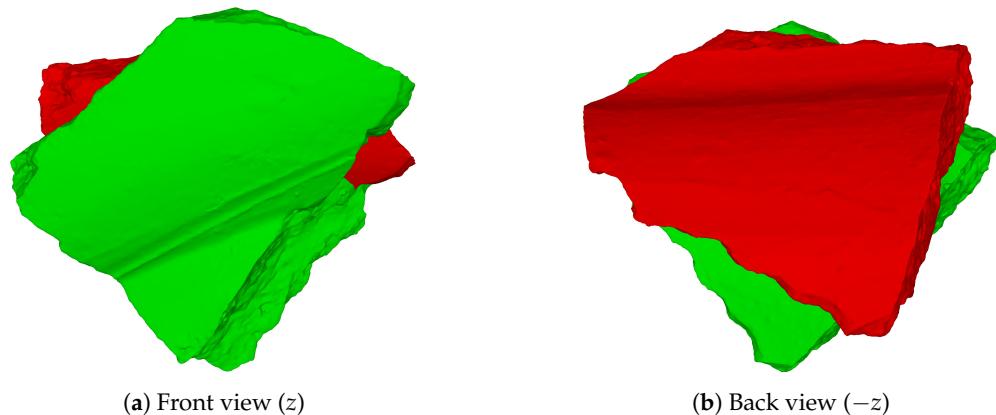


Figure 10. Sherd LV29, the best case for the LV vessel (RMSE = 0.023 m), in the expected position (red) and predicted position (green).

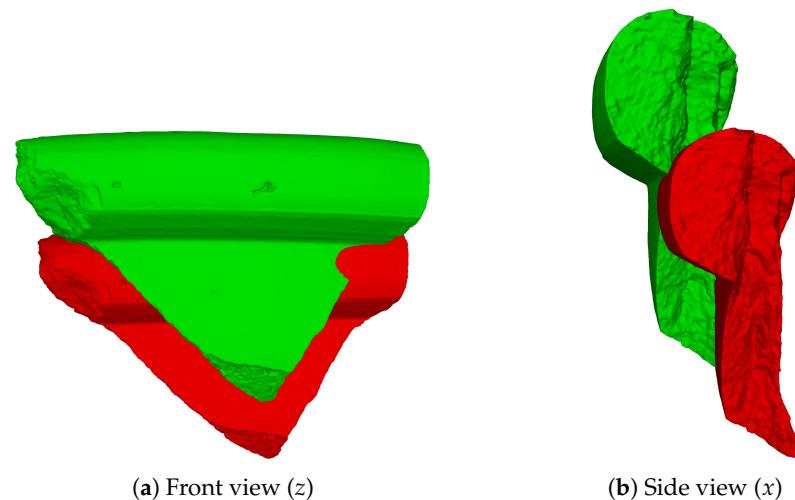


Figure 11. Sherd LV23, the average case for the LV vessel (RMSE = 0.034 m), in the expected position (red) and predicted position (green).

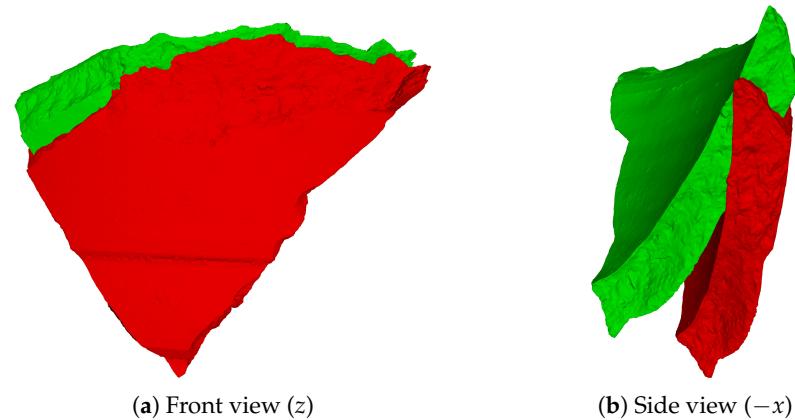


Figure 12. Sherd LV21, the worst case of the LV vessel (RMSE = 0.038 m), in the expected position (red) and predicted position (green).

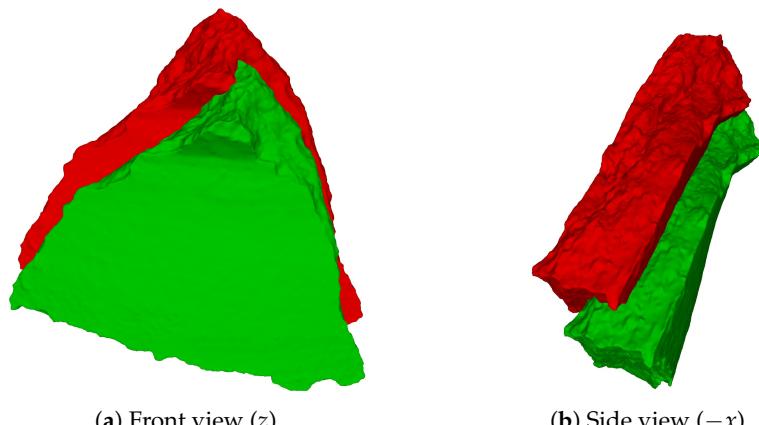


Figure 13. Sherd MV11, the best case for the MV vessel (RMSE = 0.017 m), in the expected position (red) and predicted position (green).

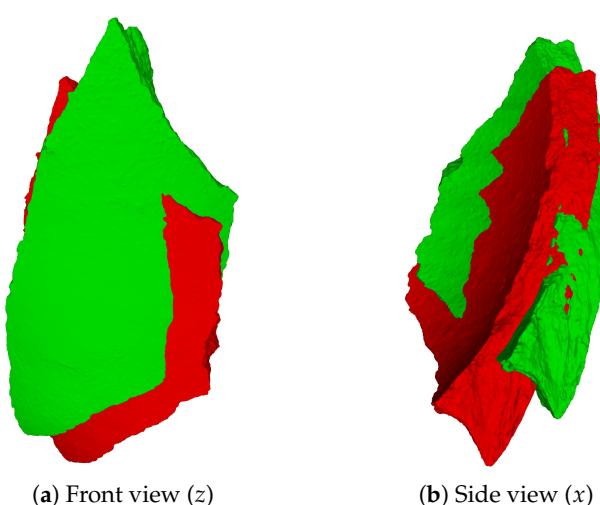


Figure 14. Sherd MV19, the average case for the MV vessel (RMSE = 0.032 m), in the expected position (red) and predicted position (green).

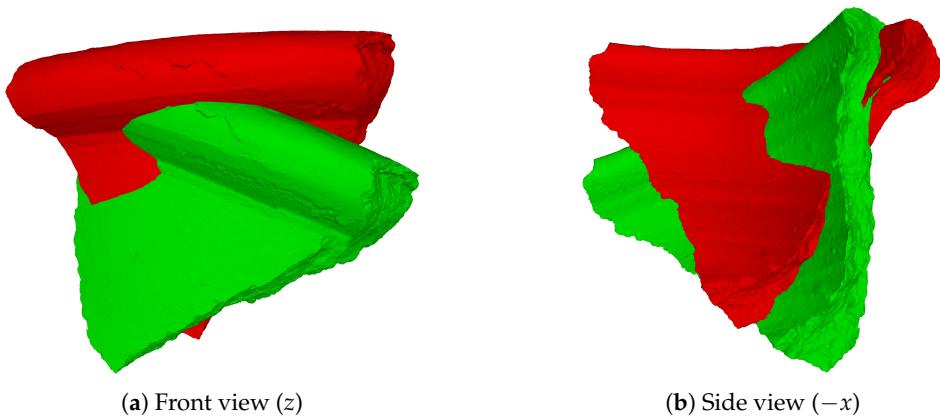


Figure 15. Sherd MV17, the worst case of the MV vessel (RMSE = 0.038 m), in the expected position (red) and predicted position (green).

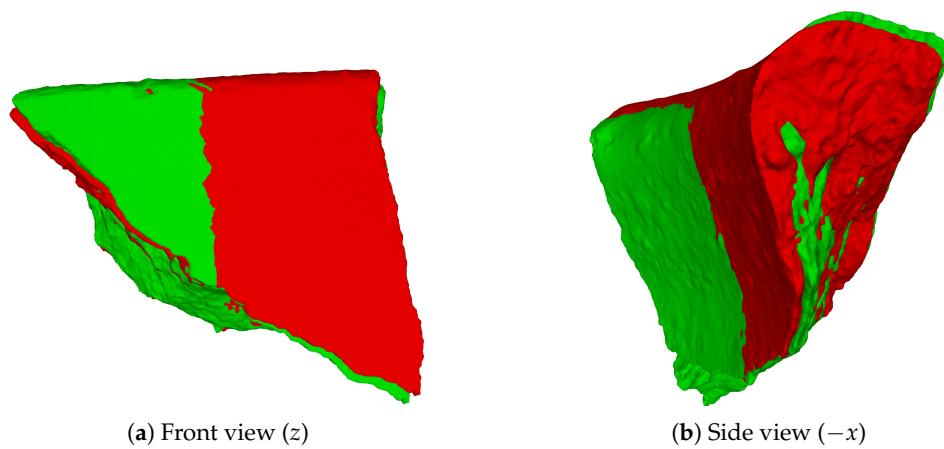


Figure 16. Sherd SV14, the best case for the SV vessel (RMSE = 0.019 m), in the expected position (red) and predicted position (green).

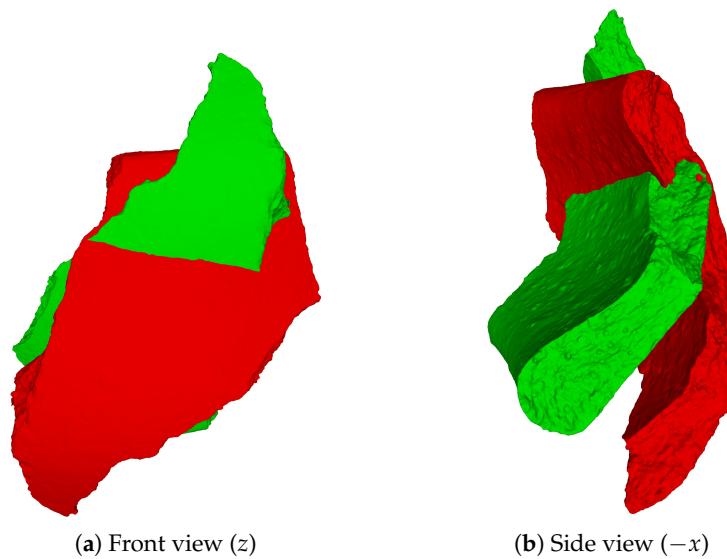


Figure 17. Sherd SV21, the average case for the SV vessel (RMSE = 0.027 m), in the expected position (red) and predicted position (green).

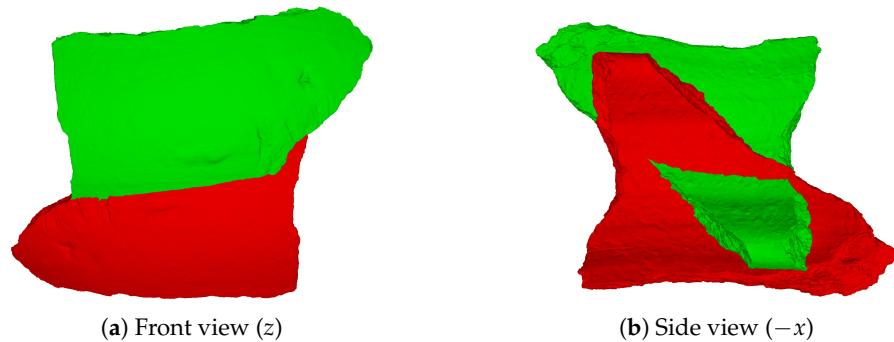


Figure 18. Sherd SV11, the worst case of the SV vessel (RMSE = 0.033 m), in the expected position (red) and predicted position (green).

7. Conclusions

In this work, we propose and evaluate a deep learning-based approach to pottery reconstruction. The proposed method works by inferring the geometric transformation that moves a 3D model of a sherd in a standard, canonical position to its original position relative to the respective vessel's coordinate system.

The method was evaluated using three distinct real vessels. The vessels were first physically broken into real sherds. The sherds from each vessel were digitally scanned, and a 3D point cloud from each sherd was produced. The vessels were then virtually reassembled into 3D models, which were later digitally broken apart a number of times to produce virtual sherds. The virtual sherds were then used to train convolutional neural networks with the proposed PotNet architecture. Two PotNet models were trained for each vessel type: one responsible for predicting translation moments and the other for predicting the rotation parameters of the geometric transformation associated with a single sherd.

In the experiments, the point clouds of the real sherds were subjected to the networks, which predicted the transformation that would move them to their expected positions. The results are considered to be very satisfactory, as the average errors for all vessels were in the centimeter range. The results are also very stable, showing standard deviations in the millimeter range. The small decline in performance observed for real-world sherds should be considered a positive indication of the generality of the proposal, even if subjected to a training process based on synthetic data of limited quality. This result makes us optimistic about the system's transferability to practical contexts. The qualitative results, obtained by visually comparing the sherds at their real and predicted positions, are also satisfactory, as the automatic outcomes are similar to their expected appearances.

One important limitation of this approach that remains to be tackled in the future is that the current setup has not been designed to deal with multiple vessel types simultaneously. We plan to address this point in the future since we know that this limitation is an obstacle to the use of this approach in the field. Therefore, as pointed out in the introduction, in its current version, the manual procedure of separating the sherds by vessel and providing an approximate 3D model of the vessel remains among the best practices that allow this approach to be beneficial. In the continuation of this work, we also plan to investigate enhancements to the proposed PotNet architecture in search of higher positioning accuracies. We also plan to extend the method by proposing a more general architecture, which not only predicts a sherd's position but also delivers a probability of the sherd belonging to a particular vessel shape. Finally, we plan to exploit texture and decorative patterns in order to automatically stitch the sherds of the same vessel together.

Author Contributions: The authors have collaboratively contributed to the paper. In particular, authors agree on the following contributions: conceptualization, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; methodology, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; software, M.F.C.P.; validation, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; formal analysis, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; investigation, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; resources, G.L.A.M. and G.A.O.P.d.C.; data curation, M.F.C.P.; writing—original draft preparation, M.F.C.P.; writing—review and editing, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; visualization, M.F.C.P.; supervision, G.L.A.M. and G.A.O.P.d.C.; project administration, M.F.C.P., G.L.A.M. and G.A.O.P.d.C.; funding acquisition, G.L.A.M. and G.A.O.P.d.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), (Process 88887.849129/2023-00), and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) under call 13/2023, (Process SEI-260003/005981/2024), Basic General Purpose Research Aid.

Data Availability Statement: The code with the network models, including the training and testing scripts, is available at <https://github.com/matfcp/PotNet> (accessed on 7 August 2024).

Acknowledgments: We would like to express our gratitude to Neodent and the Straumann Group for providing the Virtuo Vivo™ intraoral scanner, which enabled the development of this work. We would also like to thank Plinio Cesar dos Santos Silva for facilitating our collaboration with Neodent, as well as Marcelino Jose dos Anjos, Davi Ferreira de Oliveira, and Olga Maria Oliveira de Araújo

for their assistance in assessing the quality of the scanner. The authors also take this opportunity to express their gratitude to FAPERJ for funding this research and to CAPES for supporting this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DGCNN	Dynamic Graph Convolutional Neural Network
ICP	Iterative Closest Point
LV	Large Vessel
MV	Medium Vessel
POCO	Point Cloud-Based Object Classification and Orientation
RMSE	Root Mean Square Error
STD	Standard deviation
SV	Small Vessel
SVD	Singular Value Decomposition

Note

- 1 Here, we use the right-handed coordinate system as our reference coordinate system with the y axis pointing up, coinciding with the rotation axis.

References

- 1 Lee, S.; Yang, Y. Progressive Deep Learning Framework for Recognizing 3D Orientations and Object Class Based on Point Cloud Representation. *Sensors* **2021**, *21*, 6108. [CrossRef] [PubMed]
- 2 Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
- 3 Di Angelo, L.; Di Stefano, P.; Pane, C. An automatic method for pottery fragments analysis. *Measurement* **2018**, *128*, 138–148. [CrossRef]
- 4 Rasheed, N.A.; Nordin, M.J. Classification and reconstruction algorithms for the archaeological fragments. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 883–894. [CrossRef]
- 5 Alagrami, A.; Palmieri, L.; Aslan, S.; Pelillo, M.; Vascon, S. Reassembling Broken Objects Using Breaking Curves. In *Pattern Recognition*; Antonacopoulos, A., Chaudhuri, S., Chellappa, R., Liu, C.L., Bhattacharya, S., Pal, U., Eds.; Springer: Cham, Switzerland, 2025; pp. 197–208.
- 6 Eslami, D.; Di Angelo, L.; Di Stefano, P.; Pane, C. Review of computer-based methods for archaeological ceramic sherds reconstruction. *Virtual Archaeol. Rev.* **2020**, *11*, 34–49. [CrossRef]
- 7 Di Angelo, L.; Di Stefano, P.; Guardiani, E. A review of computer-based methods for classification and reconstruction of 3D high-density scanned archaeological pottery. *J. Cult. Herit.* **2022**, *56*, 10–24. [CrossRef]
- 8 Papaioannou, G.; Karabassi, E.A.; Theoharis, T. Automatic Reconstruction of Archaeological Finds—A Graphics Approach. In Proceedings of the 4th International Conference on Computer Graphics and Artificial Intelligence, Limoges, France, 3–4 March 2000; pp. 117–125.
- 9 Cooper, D.B.; Willis, A.; Andrews, S.; Baker, J.; Cao, Y.; Han, D.; Kang, K.; Kong, W.; Leymarie, F.F.; Orriols, X.; et al. Assembling virtual pots from 3D measurements of their fragments. In Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage, Glyfada, Greece, 28–30 November 2001; pp. 241–254.
- 10 Andrews, S.; Laidlaw, D.H. Toward a framework for assembling broken pottery vessels. In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI/IAAI), Edmonton, AB, Canada, 28 July–1 August 2002; pp. 945–946.
- 11 Kampel, M.; Sablatnig, R. 3D puzzling of archeological fragments. In Proceedings of the 9th Computer Vision Winter Workshop, Piran, Slowenien, 4–6 February 2004; Volume 2, pp. 31–40
- 12 Marie, I.; Qasrawi, H. Virtual assembly of pottery fragments using moiré surface profile measurements. *J. Archaeol. Sci.* **2005**, *32*, 1527–1533. [CrossRef]
- 13 Huang, Q.X.; Flöry, S.; Gelfand, N.; Hofer, M.; Pottmann, H. Reassembling fractured objects by geometric matching. In *ACM SIGGRAPH 2006 Papers*; Association for Computing Machinery: Boston, MA, USA, 2006; pp. 569–578.

14. Palmas, G.; Pietroni, N.; Cignoni, P.; Scopigno, R. A computer-assisted constraint-based system for assembling fragmented objects. In Proceedings of the 2013 Digital Heritage International Congress (DigitalHeritage), Marseille, France, 28 October–1 November 2013; Volume 1, pp. 529–536.
15. Zheng, S.; Huang, R.; Li, J.; Wang, Z. Reassembling 3D thin fragments of unknown geometry in cultural heritage. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *2*, 393–399. [[CrossRef](#)]
16. Stamatopoulos, M.I.; Anagnostopoulos, C.N. 3D digital reassembling of archaeological ceramic pottery fragments based on their thickness profile. *arXiv* **2016**, arXiv:1601.05824.
17. Cohen, F.; Zhang, Z.; Liu, Z. Mending broken vessels a fusion between color markings and anchor points on surface breaks. *Multimed. Tools Appl.* **2016**, *75*, 3709–3732. [[CrossRef](#)]
18. Sakpere, W. 3D Reconstruction of Archaeological Pottery from Its Point Cloud. In *Pattern Recognition and Image Analysis, Proceedings of the 9th Iberian Conference, IbPRIA 2019, Madrid, Spain, 1–4 July 2019; Proceedings, Part I 9*; Springer: Cham, Switzerland, 2019; pp. 125–136.
19. Eslami, D.; Di Angelo, L.; Di Stefano, P.; Guardiani, E. A semi-automatic reconstruction of archaeological pottery fragments from 2D images using wavelet transformation. *Heritage* **2021**, *4*, 76–90. [[CrossRef](#)]
20. Kashihara, K. An Intelligent Computer Assistance System for Artifact Restoration Based on Genetic Algorithms with Plane Image Features. *Int. J. Comput. Intell. Appl.* **2017**, *16*, 1750021:1–1750021:15. [[CrossRef](#)]
21. Anichini, F.; Dershowitz, N.; Dubbini, N.; Gattiglia, G.; Itkin, B.; Wolf, L. The automatic recognition of ceramics from only one photo: The ArchAIDE app. *J. Archaeol. Sci. Rep.* **2021**, *36*, 102788. [[CrossRef](#)]
22. Kim, K.; Hong, J.; Rhee, S.H.; Woo, S.S. Reconstructing the Past: Applying Deep Learning to Reconstruct Pottery from Thousands Shards. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the Applied Data Science and Demo Track: European Conference, ECML PKDD 2020, Ghent, Belgium, 14–18 September 2020; Proceedings, Part V*; Springer: Cham, Switzerland, 2021; pp. 36–51.
23. Kaya, G.; Bilmenoglu, C. Accuracy of 14 intraoral scanners for the All-on-4 treatment concept: A comparative in vitro study. *J. Adv. Prosthodont.* **2022**, *14*, 388–398. [[CrossRef](#)] [[PubMed](#)]
24. Straumann Group. Straumann Virtuo Vivo™. Available online: <https://www.straumann.com/clearcorrect/br/pt/discover/virtuo-vivo.html> (accessed on 11 March 2025).
25. Blender Home Page. Available online: <https://www.blender.org/community> (accessed on 11 March 2025).
26. Garland, M.; Heckbert, P.S. Surface simplification using quadric error metrics. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97), Los Angeles, CA, USA, 3–8 August 1997; pp. 209–216.
27. Muntoni, A.; Cignoni, P. PyMeshLab. Available online: <https://pymeshlab.readthedocs.io/en/latest> (accessed on 11 March 2025).
28. Zhou, Y.; Barnes, C.; Lu, J.; Yang, J.; Li, H. On the continuity of rotation representations in neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5745–5753.
29. Van Loan, C.F. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.* **1976**, *13*, 76–83. [[CrossRef](#)]
30. O'Rourke, J. Finding minimal enclosing boxes. *Int. J. Comput. Inf. Sci.* **1985**, *14*, 183–199. [[CrossRef](#)]
31. Jylänki, J. An Exact Algorithm for Finding Minimum Oriented Bounding Boxes. 2015. Available online: http://clb.confined.space/minobb/minobb_jylanki_2015_06_01.pdf (accessed on 11 March 2025).
32. Dawson-Haggerty, M. Trimesh. Available online: <https://trimesh.org> (accessed on 11 March 2025).
33. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *PAMI-9*, 698–700. [[CrossRef](#)] [[PubMed](#)]
34. Malischewski, S.; Schumann, H.; Hoffmann, D. Kabsch Algorithm. Available online: <https://biomolecularstructures.readthedocs.io/en/latest/kabsch> (accessed on 25 July 2022).
35. Ronnegren, J. Real Time Mesh Fracturing Using 2D Voronoi Diagrams. 2020. Available online: <http://www.diva-portal.org/smash/record.jsf?pid=diva2:1452512> (accessed on 25 July 2022).
36. Mark, d.B.; Otfried, C.; Marc, v.K.; Mark, O. *Computational Geometry: Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2008.
37. Corsini, M.; Cignoni, P.; Scopigno, R. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 914–924. [[CrossRef](#)] [[PubMed](#)]
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
39. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.