algorithms

Article

A Non-Gradient and Non-Iterative Method for Mapping 3D Mesh Objects Based on a Summation of Dependent Random Values

Ihar Volkau [1,*], Sergei Krasovskii [1], Abdul Mujeeb [1] and Helen Balinsky [2,*]

[1] HP–NTU Digital Manufacturing Corporate Lab, Nanyang Technological University, Singapore 637460, Singapore; krasovskii.sergei.gen@gmail.com (S.K.); amujeeb@ntu.edu.sg (A.M.)
[2] Workforce Solutions, HP Inc., Bristol BS1 6NP, UK
[*] Correspondence: volkau.ihar@ntu.edu.sg (I.V.); helen.balinsky@hp.com (H.B.)

Abstract: The manuscript presents a novel non-gradient and non-iterative method for mapping two 3D objects by matching extrema. This innovative approach utilizes the amplification of extrema through the summation of dependent random values, accompanied by a comprehensive explanation of the statistical background. The method further incorporates structural patterns based on spherical harmonic functions to calculate the rotation matrix, enabling the juxtaposition of the objects. Without utilizing gradients and iterations to improve the solution step by step, the proposed method generates a limited number of candidates, and the mapping (if it exists) is necessarily among the candidates. For instance, this method holds potential for object analysis and identification in additive manufacturing for 3D printing and protein matching.

Keywords: 3D object matching; 3D object identification; surface analysis; feature extraction; amplification of extrema

Citation: Volkau, I.; Krasovskii, S.; Mujeeb, A.; Balinsky, H. A Non-Gradient and Non-Iterative Method for Mapping 3D Mesh Objects Based on a Summation of Dependent Random Values. *Algorithms* **2024**, *17*, 248. https://doi.org/10.3390/a17060248

Academic Editors: Alicia Cordero and Juan Ramón Torregrosa Sánchez

Received: 2 May 2024
Revised: 31 May 2024
Accepted: 3 June 2024
Published: 6 June 2024

1. Introduction

Modern methods of modeling, digitization, and visualization of 3D objects brought explosive growth to the number of models accessible in different databases and on the Internet. This increase has allowed the development of new methods, fast progress in 3D object classification and identification, and new approaches in 3D object search and retrieval systems. The quick, accurate, and efficient identification of 3D objects is a practical problem in robotics, self-driving cars, medical image analysis, computer vision, 3D printing, protein matching, etc.

Two standard methods are usually used to represent and visualize 3D shapes: either a 3D model is represented as a surface (for example, as polygonal/triangular meshes) or a voxel representation. Each approach has its advantages and drawbacks.

In reviews [1,2], a plethora of methods for 3D search for both surface and voxel representations are considered. The approaches are assessed with respect to several content-based 3D shape retrieval criteria, such as properties of dissimilarity measures, discrimination abilities, ability to perform partial matching, robustness and efficiency, shape representation requirements, and the necessity of pose normalization.

This work is directed towards solving the optimization problem of identifying two 3D objects, namely, deciding if they are the same or distinctive. In case objects O_1 and O_2 are identified as the same, it is necessary to find a rigid transformation to map them together. The problem of object identification is different from the problem of classification when it is necessary to find if the objects have the same properties intrinsic to the same class.

The rigid transformation should minimize some distance metrics between the transformed original object O_1 and the sample O_2. If this distance is less than some predefined threshold, consider the objects similar and indistinct.

Let us assume that a 3D object is defined by its triangular mesh. Different ways can be found to map objects to each other rigidly. One state-of-the-art method is the ICP (iterative closest point for point cloud registration) algorithm [3,4]. It is based on iterative gradient calculation and improvement of the solution.

In [5], the problem of shape matching is reduced to comparing probability distributions, and a method for calculating shape signatures for arbitrary (possibly degenerate) 3D polygonal models is proposed. The key idea is to represent the signature of an object as a distribution of some geometric properties (features) of the object (for example, the distance between two random points on the surface). This signature should allow distinguishing between classes of objects (for example, cars and airplanes) in a medium-sized database. Of course, the method based on such an idea can be used as a preliminary classifier, but it does not solve the problem of identifying objects. In [6], the mapping transformation is calculated by maximizing the overlap of objects' features.

In [7], query methods were developed that are simple enough for novice users, and relatively reliable matching algorithms were used to work with arbitrary polygonal models based on 3D sketches, 2D sketches, 3D models, and text keywords.

The neural network approach is usually used to solve the problem of object classification and check if the object under analysis owns the common properties of the objects of some class. Ref. [8] attempts to improve both volumetric convolutional neural networks (CNNs) based on volumetric representations of the surface and multi-view CNNs, which are based on representations with multiple representations. The authors introduce two different network architectures of volumetric CNNs and investigate multi-view CNNs with filtering having multiple resolutions in 3D. In [9], a patch convolutional neural network (PCNN) is proposed to search for 3D models based on representations.

Another direction to solve the mapping problem by applying neural networks is deep learning-based point cloud registration. A review of the recent progress and overview of methods can be found in [10].

2. Relevant Methods: State of the Art

Earlier works on shape representation analysis for object identification can be divided into two broad categories. The first category uses methods such as PCA to align the model in a canonical coordinate system. Then, it defines the representation of the shape by relating it to this orientation. Such methods include, among others, moments for object surfaces [11] and Extended Gaussian Images [12]. Kazhdan [13] (Appendix B) mentions that PCA-based methods are unstable due to the multiplicity of eigenvalues and sensitivity to outliers. The second category of methods defines representation invariants with respect to rotation and includes methods such as Shape Histograms [14] and Shape Distributions [5].

In this work, we adhere to ideas based on decomposing some function given on a sphere into the sum of spherical harmonics. The works of [7,13,15,16] gave the impetus for developing such an approach to the problems of classifying 3D objects and retrieving their information from databases.

In [17,18], tools for searching for 3D objects were presented. The model is represented as a polygonal grid and serves as a query, and similar objects are extracted from a collection of 3D objects. The first stage of the algorithm is normalization (estimation of the pose) when the models are transformed into a canonical coordinate system. Then, feature vectors are extracted and compared with vectors obtained from normalized models in the search space. Using metrics in the feature vector space, the nearest neighbors are calculated and ranked. The objects extracted in this way are displayed for inspection, selection, and processing. Feature vectors are based on rays cast from the object's center of mass. The distance from the center of mass to the surface in the ray's direction sets the function value on the sphere. To evaluate the pose, a modified Karhunen–Loeve transformation is introduced, taking into account not only vertices or polygonal centroids from 3D models but also all points in polygons of objects.

In [19], the 3D model's symmetry descriptors are presented as a set of spherical functions that describe the measure of the model's rotational and reflective symmetry with respect to each axis passing through the center of mass.

Papadakis et al. [20] present a methodology for reconstructing a 3D shape based on spherical harmonics, which ensures invariance to scaling and rotation. Rotation normalization is performed using continuous principal component analysis and its application to the model surface normals. The 3D model is represented as a set of spherical functions. This representation comprises the intersections of the surface with rays originating from the origin as well as points in the direction of each ray that are closer to the origin than the farthest intersection point.

Ref. [21] presents a 3D shape descriptor that uses a set of panoramic views of a 3D object. The views describe the position and orientation of the object's surface in 3D space. A panoramic view of the 3D object is obtained by projecting it onto the surface of the cylinder parallel to one of its three main coordinate axes and is centered on the object's centroid. The object is projected onto three perpendicular cylinders aligned axes to capture the global shape of the object. The corresponding 2D discrete Fourier transformation and a 2D discrete wavelet transformation are found for each projection.

Ref. [22] presents the decomposition of spherical harmonics for spherical functions to represent 3D triangulated star-shaped objects. After splitting the surface into star-shaped sections, this result can be extended to any triangular object. The evaluation of the spherical harmonic coefficients is performed by edge integration using the Monte Carlo method, which distinguishes this work from voxel methods.

In [23], after translating and scaling the 3D model, the concentric spheres are used to analyze the object, and a new set of functionals is defined and applied for each sphere. This results in a vector of feature descriptors. This vector is invariant to rotation and, hence, suitable for matching 3D models. Weight coefficients are assigned to each descriptor to improve the mapping.

In [24], descriptors of 3D shapes are generated using functions on a sphere. A set of vector descriptors invariant to rotation is extracted using spherical harmonics. First, the models' size is normalized, and then a sample of data is taken from the surface of the 3D model and converted into point cloud data. Next, spherical harmonics are applied to this point cloud to obtain invariant rotation descriptor vectors. The method proved resistant to noise on the model's surface and allowed us to compare similarities of 3D models.

The essential part of identifying the similarity of two objects is to find a rotation that best combines two sets of vectors (or two clouds of points). Algebraic approaches to solving the problem of optimal rotation were proposed in [25–28]. These results are exact but require knowing the correspondence among the points in these two sets.

In [29], the spherical harmonics of a pair of images are related to each other by a shift theorem, which uses an irreducible representation of the rotation group. Using this theorem, Euler angles are extracted. Regarding the required number of spherical coefficients, it is assumed that they are global image encoders and that rotations can be estimated using a simple reference table of combinations of coefficients and angles.

Esteves et al. [30] solve the equivariance problem of 3D rotation using convolutional neural networks. The authors model 3D data with spherical functions and propose a spherical convolutional network that implements point convolutions on a sphere. The paper demonstrates that networks with low bandwidth without increasing the dataset size can demonstrate performance comparable to state-of-the-art networks.

In [31], the fundamental problem of finding the shape and registering a point cloud for the point clouds with zero point-to-point correspondences is considered. Using the trained optimization process, the authors propose a method to represent 3D point clouds by spherical Gaussians and a rotation-invariant convolution.

In [32], spherical harmonic functions are directly used to simulate irregular 3D particles. Discrete surface points of irregularly shaped 3D particles are represented by spherical harmonic functions with a limited number of harmonic coefficients to restore the particle's morphology. These spherical harmonic functions are used to detect overlap between particles and calculate the interparticle contact forces, moments, and particle motions in various engineering and industrial processes.

The analysis of spherical harmonics was initially used to solve problems in geophysics, potential theory, and mathematical physics [33]. Later, it was found to apply to the classification and identification of 3D objects.

An increasing number of 3D models of buildings are hosted on web platforms for model sharing. Therefore, based on data reuse, in [34,35], an approach to coding and searching 3D models of buildings using point clouds obtained by airborne light detection and ranging (LiDAR) systems is proposed. To encode LiDAR point clouds with sparse, noisy, and incomplete sampling, the authors introduce an encoding scheme based on a set of low-frequency spherical harmonic basis functions. Descriptors are extracted from spatial histograms and used to search for 3D models.

The state-of-the-art approach to solving the optimization problem of mapping objects to each other is the ICP algorithm. The solution is found by step-by-step optimizing the loss function defined at the sets of points of both objects. Based on gradient search, the ICP iteratively finds the best correspondence among the subset of points of the objects. The latest modification of the ICP is Go-ICP [36], where Local ICP is based on a branch-and-bound (BnB) scheme.

In [37,38], global optimization methods based on implicit enumeration were mentioned. As per their taxonomy, these methods belong to the space-covering group, which aims at implicitly exploring the whole feasible region. Such approaches could be iterative and, at the same time, non-gradient methods. These methods started with the simplex method [39] and the algorithms in [40,41] and do not require calculating the gradient; they are still iterative methods. There are several modifications of the Powell method, such as COBYLA [42], which employs linear polynomial approximations to the loss and constraint functions by interpolation at the vertices of simplices.

Another one-pass approach to calculating the mapping of two 3D objects is based on training a deep learning network. The trained network calculates the mapping in a non-gradient and non-iterative way. Such approaches could be found in [43–47], where the neural network is initially trained to extract local spatial features from each of the 3D objects under comparison. Then, the mapping is calculated based on these features.

If the number of possible extrema at the object's surface is relatively small, it is not prohibitively expensive to find the matching of the extrema for both objects and check how good these mappings of the objects are. As the possible number of extrema combinations is assumed not high, as an option, the mapping can be found even by exhaustive search without the calculation of gradients in a non-iterative way.

This is the idea of the proposed method for solving the identification problem and finding the mapping of the objects. Let us use a spherical transformation to map the 3D object in the feature space (encoding), and for partial reconstruction (decoding), let us define a 3D shortest distance function. The values of this function are treated as random values. The summation of random values gives us several extrema (see Sections 3.9 and 3.10). Calculating the mappings for the corresponding extrema for two 3D objects under comparison allows us to find the required transformation or tells us that the objects are different.

In the "Results and Discussion" section, a comparison of the proposed method with a deep learning approach (using [43] as an example) is given.

3. Method

Let us assume that there are two 3D objects: O_1 and O_2. Each of these objects is represented by its triangular mesh. It is necessary to define whether these two objects are identical (up to some calculation error). If they are similar (this analysis considers the surface and the internal architecture), find the transformation matrix (rotation and translation) to map them.

Let us reiterate that the problem to be solved is not to identify if the class of the objects is the same (for example, for two gears with different numbers of teeth) but to check whether either two objects could be transformed to each other by a rigid transformation.

Initially, let us describe an overview of an identification algorithm, and every step will be explained in detail in the following sections.

It is necessary to compare two 3D objects, O_1 and O_2, for identity. Each object's model mesh is watertight and has consistent winding and outward-facing normal vectors (hereafter, let us define this set of features as "correct geometry"). A corresponding 3D shortest distance function (SDF) is calculated for each object. The SDF value at the 3D point is the shortest distance from this point to the object's surface. Spherical harmonic coefficients represent the calculated SDF function. Using expansions of degree L, we partially reconstruct the SDF functions related to objects O_1 and O_2.

First, the energies of the corresponding degrees of spherical harmonic coefficients are compared to analyze the objects' similarity. Then, extrema amplification is utilized, and next, an attempt to align the most distinguishable extrema of both partial reconstructions is conducted. If the objects match with the acceptable quality, the matrix of the object mapping is calculated.

Preprocessing. For object O_1, let us denote its surface as S. The surface (the object's mesh) is assumed to have the correct geometry.

P1. Shift the origin of the system of coordinates to the centroid (the center of mass of the object's surface).

P2. Scale O_1 so that none of its vertices are outside the sphere of radius sixteen, and at least one vertex is on the sphere surface. Let the coefficient of scaling be f_{scale}.

P3. Place n concentric spheres (shells) centered at the origin with radii $R_1, \ldots, R_n$ (we use $n = 9$ and $i = 1, 3, \ldots, 17$). These spheres cut O_1.

P4. On the i-th sphere, $i = 1, \ldots, n$, a continuous function $f^{(i)}(\theta, \varphi)$ is defined, and θ, φ are spherical coordinates. SDF $f^{(i)}(\theta, \varphi)$ is the shortest distance from the point (θ, φ) to the surface S (see the Section 3.1 and [48]).

P5. At each sphere, a grid of points $(\theta_j, \varphi_j), j = 1, \ldots, k$, is defined. For this grid, we can use, for example, a t-design [49], Fliege–Maier [50], or an icosphere [51] set of points (see Section 3.2). These points are employed for low-error integration of spherical functions.

P6. At each sphere, using $f^{(i)}(\theta_j, \varphi_j)$, let us calculate the spherical harmonic coefficients (or Fourier–Legendre coefficients) $f_{lm}^{(i)}, l = 0, 1, 2, \ldots, -l \le m \le l$ (see the definition (3) below). Let us calculate spherical coefficients up to degree $l \le 10$.

P7. At each i-th sphere, let us partially reconstruct the function $\tilde{f}^{(i)}(\theta_j, \varphi_j)$ using these coefficients $f_{lm}^{(i)}$ for $l = 1, \ldots, 10$ degrees and the icosphere grid (θ_j, φ_j) (see the Section 3.6).

$$\tilde{f}^{(i)}(\theta_j, \varphi_j) = \sum_{l=0}^{10} \sum_{m=-l}^{l} f_{lm}^{(i)} Y_{lm}(\theta_j, \varphi_j). \tag{1}$$

P8. Next, on the unit sphere, let us add all the functions $\tilde{f}^{(i)}(\theta_j, \varphi_j)$ over the entire set of concentric spheres as follows:

$$\tilde{f}(\theta_j, \varphi_j) = \sum_{i=1}^{n} \tilde{f}^{(i)}(\theta_j, \varphi_j) \tag{2}$$

and define a surface $\left(\theta_j, \varphi_j, \tilde{f}(\theta_j, \varphi_j)\right)$. The surface points are connected the same way as the corresponding points of the icosphere.

P9. Find the extrema points and saddle points of the surface $\tilde{f}$ of the previous item.

P10. Build the convex hull of the points found in the previous item.

P11. Order these points by not increasing their radius vector lengths. Denote this sequence as F.

We repeat the same preprocessing for object O_2 and denote the corresponding calculated function with replacement f to g, obtaining, for example, $\widetilde{g}(\theta_j, \varphi_j)$, etc.

Steps of the algorithm. Let us find a transformation matrix that maps O_1 to O_2, if it exists.

A1. Let us compare the scaling coefficient f_{scale} and g_{scale}. If the objects are not equal (up to some tolerance), they are different. If we want to check the similarity of two objects while ignoring the scale, go to the next step.

A2. For each sphere $i = 1, \ldots, n$, calculate and compare energy for each degree l for both objects. If

$$\left| \sum_{m=-l}^{l} \left| f_{lm}^{(i)} \right|^2 - \sum_{m=-l}^{l} \left| g_{lm}^{(i)} \right|^2 \right| > \text{Tolerance}$$

for at least one degree l, then the objects are different; otherwise, go to the next step.

A3. Let us choose vector f_1. This is the longest vector from sequence F that has not yet been used by the algorithm. If there is no such vector, go to step A8.

A4. Look for subsequent vectors f_2 from F, which forms an angle of at least 10 degrees with f_1. For each such vector, pair f_1 and f_2 and go to the next step; otherwise, go to step A3.

A5. Find the furthest point f_3 of the convex hull from the plane α, which contains the origin and vectors f_1 and f_2. Find, if any, the corresponding points, g_1, g_2, and g_3, in sequence G. These points in G should have the same length of vectors and angles among vectors (up to some tolerance) as the vectors in F. The scalar triple product of (f_1, f_2, f_3) and (g_1, g_2, g_3) should have the same sign.
If there are no such vectors, go to step A3 and look for the next vector f_1.

A6. Calculate the rotation matrix using the Kabsch algorithm [25,26] to map the triplet of non-coplanar vectors (f_1, f_2, f_3) to (g_1, g_2, g_3). If this (or similar, up to some tolerance) rotation matrix was calculated and analyzed before, go to step A3, and look for the next vector f_1.

A7. Using the rotation matrix found, rotate O_2 to $\overline{O}_2$, and calculate the vector of spherical harmonic coefficients $\overline{g}_{lm}^{(i)}$ for $\overline{O}_2$. In the ideal case, objects O_1 and $\overline{O}_2$ should coincide. For each sphere i, we calculate the cosine similarity $\text{CS}^{(i)}$ (i.e., find the angle between two vectors), $\overline{g}_{lm}^{(i)}$, and $f_{lm}^{(i)}$. To evaluate the quality of the mapping, we introduce two metrics (see Section 3.3) as follows:

$$M_1 = \min_{i=1,\ldots,n} \text{CS}^{(i)}, \qquad M_2 = \sum_{i=1}^{n} \left(1 - \text{CS}^{(i)} \right).$$

The rotation matrix found is considered an acceptable solution if M_1 is more than some tolerance (we set it as $\cos(10^0) = 0.9848$, with the perfect match being 1.0). This means that the cosine similarity of each sphere is greater than this value.
If there are several acceptable solutions, the best of them is defined as the one having the minimal M_2 (the minimal sum of deviations from the perfect match of coefficients for all the spheres). If M_2 is small (we chose $M_2 < 0.02$), we assume that the best acceptable solution has been found; stop the search. If step A7 was executed more than 30 times, go to step A8; otherwise, go to step A3.

A8. Suppose none of the acceptable solutions has been found. In that case, the objects are different; otherwise, the objects are considered the same, and the best rotation matrix found at step A7 is the solution. Return the best acceptable solution found and stop the algorithm.

Thirty repetitions are, in some sense, overkill. The rationale for choosing this number and its sufficiency will be discussed in Sections 3.8–3.10.

Let us describe the steps of the algorithm in detail.

3.1. Spherical Harmonics

Spherical harmonics are the basis functions for irreducible representations of a group of rotations SO(3) in $\mathbb{R}^3$. These orthonormal functions form a basis of the Hilbert space of square-integrable function $L^2_{\mathbb{R}}(S^2)$.

Let us denote S^2 a unit sphere. Any square-integrable function $f : S^2 \to \mathbb{R}$ can be represented as a sum [33] as follows:

$$f(\theta,\varphi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} f_{lm} Y_{lm}(\theta,\varphi)$$

where spherical harmonics coefficients (or Fourier–Legendre's coefficients) are

$$f_{lm} = \int_0^{2\pi} \left(\int_0^{\pi} \sin\theta \, f(\theta,\varphi) Y_{lm}(\theta,\varphi) \, d\theta \right) d\varphi \tag{3}$$

and spherical harmonics Y_{lm} are defined by

$$Y_{lm}(\theta,\varphi) = \begin{cases} (-1)^m \sqrt{2} \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^{|m|}(\cos\theta) \ \sin(|m|\varphi), & m < 0, \\ \sqrt{\frac{2l+1}{4\pi}} P_l^m(\cos\theta), & m = 0, \\ (-1)^m \sqrt{2} \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m(\cos\theta) \ \cos(m\varphi), & m > 0. \end{cases}$$

Here, P_l^m are associated Legendre's polynomials as follows:

$$P_l^m(t) = (-1)^m \left(1 - t^2\right)^{\frac{m}{2}} \frac{d^m}{dt^m} P_l(t), \quad l \geq 0, \ \ |m| \leq l,$$

and $P_l(t)$ are Legendre's polynomials, which can be represented by the Rodrigues differential form as follows:

$$P_l(t) = \frac{1}{2^l l!} \frac{d^l}{dt^l} \left(1 - t^2\right)^l.$$

Under rotation operation, a spherical harmonic of degree l and order m transforms into a linear combination of spherical harmonics of the same degree. For each degree l, the basic functions Y_{lm}, $-l \leq m \leq l$ form a set of irreducible representations of the group SO(3) of dimensions $(2l+1)$. This means that any rotation R will be a self-homeomorphism in this subspace.

Let us denote the result of a rotation R around the center of the sphere (around the origin of the system of coordinates) applied to sphere S^2 as S^2_R and the function $f = f(\theta,\varphi)$ calculated on a rotated sphere S^2_R as $g : S^2_R \to \mathbb{R}$. The rotation of sphere S^2_R corresponds to the rotation R of the object.

First, let us translate object O_i the way that the centroid of O_i is in the center of the coordinates. For now, we assume that objects O_1 and O_2 are the same, and O_2 is the rotated version of O_1. We would like to find the matrix of rotation R. Next, we calculate f_{lm} and g_{lm}, the coefficients of spherical harmonics for degrees l up to some $L_{\max}$ for objects O_1 and O_2. The useful property of energies for any degree L is that the energy is not changed by rotation. As basic functions Y_{lm} for each l form a subspace of dimension $(2l+1)$, the total power of function f in the subspace corresponding to the degree l does not depend on the rotation R.

$$\sum_{m=-l}^{l} |f_{lm}|^2 = \sum_{m=-l}^{l} |g_{lm}|^2$$

This directly follows from Parseval's theorem [33] and the properties of the group $SO(3)$. The coefficients for every degree l for the rotated object could be found using, e.g., the Wigner D-matrix [52].

Functions f and g could be expanded on a unit sphere S^2 as

$$f(\theta,\varphi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} f_{lm}Y_{lm}(\theta,\varphi), \quad g(\theta,\varphi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} g_{lm}Y_{lm}(\theta,\varphi) \tag{4}$$

where

$$\begin{aligned} f_{lm} &= \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \, \sin\theta \, g(\theta,\varphi) Y_{lm}(\theta,\varphi), \\ g_{lm} &= \int_0^{2\pi} d\varphi \int_0^{\pi} d\theta \, \sin\theta \, g(\theta,\varphi) Y_{lm}(\theta,\varphi). \end{aligned}$$

If we only consider a specific degree l in Formula (4), we will obtain a partial reconstruction of the functions f and g for this degree. This reconstruction will use $2l+1$ coefficients g_{lm} and f_{lm}, $m = -l, \ldots, l$ for the whole expansion.

Expansions f and g are given by arrays of their spherical harmonic coefficients, $\{f_{lm}\}$ and $\{g_{lm}\}$. Let us denote the partial expansion of the degree L of the function f as f_L and g_L for g as follows:

$$f_L(\theta,\varphi) = \sum_{l=0}^{L}\sum_{m=-l}^{l} f_{lm}Y_{lm}(\theta,\varphi), \qquad g_L(\theta,\varphi) = \sum_{l=0}^{L}\sum_{m=-l}^{l} g_{lm}Y_{lm}(\theta,\varphi). \tag{5}$$

Function f_L is defined by coefficients $\{f_{lm}\}$ with $l \le L$. The norm of deviation of functions f_L and g_L could be defined as a cosine similarity of arrays $\{f_{lm}\}$ and $\{g_{lm}\}$ for $l \le L$.

As we postulated, function g is a result of rotation R of function f, and the results of the partial reconstruction will have the same relationship—they are the rotated versions of each other. For every degree l (as well as for any combinations of such degrees), the partial reconstructions of f and g in $\mathbb{R}^3$ could be mapped by the same rotations R.

3.2. Numerical Calculation of Spherical Coefficients

Let us introduce several different grids for numerical calculations for the method proposed.

The numerical integration (step P5) of the square-integrable function defined on the two-dimensional sphere is performed using quadrature formulae. This function is evaluated at some nodes in the sphere. For example, this set of points could be Fliege–Maier's set of points on the sphere for low-error integration of spherical functions [50]. The mentioned grid can be used for integration by directly summating the function evaluated at grid points and weighted with the respective weights. Another option could be to use t-designs [49], which constitute uniform arrangements on the sphere for which spherical polynomials up to degree t can be integrated precisely by summating their values at the points defined by the t-design. One more way could be to use a grid defined by the icosphere [51].

The Gauss' quadrature formulae with the mentioned designs provide fast and accurate calculations of spherical harmonic coefficients. In [50], it was shown that for pre-computed nodes and weights, the integration error drops to 10^{-12} for many functions.

Another approach could be performing the spherical harmonic transform (SHT) in the least-squares sense. Using values $f\left(\theta_j,\varphi_j\right)$ (see Formula (4)) at nodes $\left(\theta_j,\varphi_j\right)$ on the grid, the spherical harmonic coefficients f_{lm} could be computed using the Moore–Penrose pseudo-inverse of a matrix [53]. The following least-squares problem is solved with respect to variables f_{lm} as follows:

$$f(\theta_j,\varphi_j) = \sum_{l=0}^{L_{\max}}\sum_{m=-l}^{l} f_{lm}Y_{lm}(\theta_j,\varphi_j), \qquad j = 0, \ldots, p.$$

In step P7, it is necessary to reconstruct a function at the sphere by its spherical harmonic coefficients. This reconstruction requires calculating the function at the preferably uniform grid. We used a tessellation [51] built from an icosahedron. To obtain a tessellation, the following steps are performed: (i) divide each triangular face of the icosahedron into four smaller triangles by connecting the midpoints of the three sides; (ii) normalize the new triangle vertices on the unit sphere; and (iii) repeat these steps until the desired solution is reached.

After applying five such iterations to the original icosahedron, we obtained an icosphere containing 10,242 vertices. Each vertex of this triangular tessellation has six adjacent triangles, except for the original twelve icosahedral vertices with five adjacent triangles.

3.3. Comparison of Two Objects: Optimization Function

We have two 3D objects, O_1 and O_2, to be compared for identity. Let us mathematically define the meaning of the word "identity" used in this work.

We assume that a 3D object O has the surface S with a correct geometry. The object is scaled the way that the centroid of its surface is at the origin of the system of coordinates, and the object itself fits into a sphere of radius 16 and touches this sphere. The number 16 was arbitrarily chosen as a parameter of the method.

Let us place n concentric spheres (shells) centered at this origin. We use a set of nine spheres with radii of 1, 3, 5, ..., 17. On each i-th sphere, $i = 1, \dots, n$, and a continuous function $f^{(i)}(\theta, \varphi)$ is defined. This function is called the shortest distance function (SDF), and its value is the shortest distance from the point (θ, φ) of this i-th sphere to the surface S. This distance could have a positive or negative sign. The distance is considered positive if the point (θ, φ), for which the distance to the surface S is measured, is outside this surface, and negative otherwise. The SDF is a continuous function on S^2.

Let us assume that we found the function $f = \{f^{(i)}\}$, where $f^{(i)} = \{f^{(i)}_{lm}\}$ corresponds to object O_1 spherical coefficients of i-th sphere, and the function $g = \{g^{(i)}\}$, $g^{(i)} = \left\{g^{(i)}_{lm}\right\}$ corresponds to O_2.

Let us denote

$$M_0^{(i)}(l) = \left| \sum_{m=-l}^{l} |f^{(i)}_{lm}|^2 - \sum_{m=-l}^{l} |g^{(i)}_{lm}|^2 \right|.$$

Due to the Parseval equation [33], if the objects are the rotated version of each other, the absolute value of the difference of energies for expansions f and g for degree l should be equal to zero as follows:

$$M_0^{(i)}(l) = 0.$$

Due to inevitable errors of numerical calculations, we assume that for the rotated versions of objects, the difference should be less than or equal to some threshold T_{energy}

$$M_0^{(i)}(l) \le T_{\text{energy}} \tag{6}$$

for all the degrees l used for partial reconstruction. In the calculations, the difference $M_0^{(i)}(l)$ is considered negligible if it is either below 5% of the first object's energy or below 0.01 (the approach is the same as the Python function *isclose* with relative and absolute tolerances). Condition (6) is used to check the energy similarity of both objects in step A2.

It is also possible to use a more general form as follows:

$$M_1 = \min_{i=1,\dots,n} \text{CS}^{(i)}, \tag{7}$$

where

$$\text{CS}^{(i)} = \text{cosine\_distance}\left(\left\{f^{(i)}\right\}, \left\{g^{(i)}\right\}\right)$$

As a condition of similarity, we request that M_1 should be less than the cosine of 10 degrees. This condition is necessary but not sufficient, as shown in [13] (Figure 5 with an airplane).

One more additional loss function used is

$$M_2 = \textstyle\sum_{i=1}^{n} \left(1 - \text{CS}^{(i)}\right), \tag{8}$$

This is the sum of overall deviations of the spherical coefficients at each sphere. We consider the smaller M_2 the better the solution.

The loss functions, M_1 and M_2, are used in step A7.

If we assume that function g is the result of the application of function f to sphere S^2_R (sphere S^2 rotated by transformation R), then $g = fR$. We would like to find the transformation R to map object O_1 (corresponding to function f) to O_2 (corresponding to function g). This transformation R can be defined as

$$R = \underset{R}{\operatorname{argmin}} \|g - fR\|$$

In case the best R found gives the cosine similarity measure M_1 above some predefined threshold T_{CS}, we consider objects O_1 and O_2 as nonidentical. We chose T_{CS} corresponding to 10 degrees.

One more measure of mapping accuracy could be the Root-Mean-Square Deviation between two reconstructions (5).

3.4. *Behind the Comparison: Usage of Spherical Harmonics as Structural Patterns*

Let us denote the partial reconstruction of function g for degree l as g_l. The same denotation will be used for f.

Let us start with a toy example and assume that at degree l for function f has only one coefficient with order m, say, $l = 3$ and $m = 2$; see Figure 1.

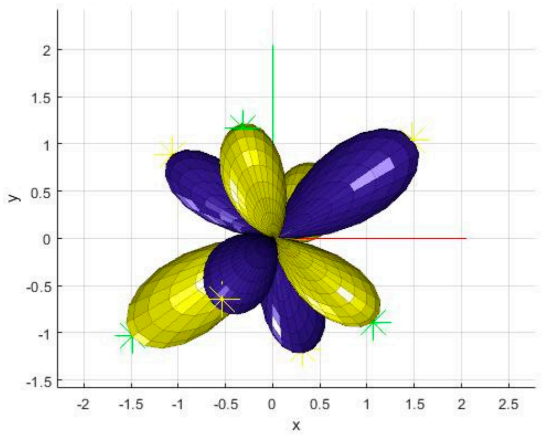

Figure 1. Spherical harmonic of order (3,2) with the extrema marked. The yellow areas correspond to the negative values of the function, and the blue ones correspond to the positive ones.

The partial reconstruction f_l for degree l will be the spherical harmonic Y_{lm} multiplied by the corresponding coefficient f_{lm}. Let us assume that this coefficient is equal to 1. Let us take a look at the visualization of a spherical harmonic Y_{lm}. We can see that the number of extrema points on the harmonic's surface is not large. The reconstruction g_l will look geometrically (spatially) the same; hence, there is the same number of maxima for the original and rotated spherical harmonic. If the number of maxima is relatively small, we can find the mapping of the maxima of f_l to the maxima of g_l using an exhaustive search. We can preliminarily order these maxima by their values (length of vectors) and angles between different surface extrema points to speed up the process. This ordering would allow us to find the correct mapping faster with fewer combinations to be checked.

If, in the toy example, the harmonic under consideration has a continuous rotation symmetry, like the spherical harmonic in order (3,0) (Figure 2, see a shape in the center of the last row), then the extrema and the saddle points of the harmonic surface could be sampled at the grid with the chosen grid step. The middle part (the "equatorial" line) will obtain limited points. Matching of respective extrema points at both objects could still be performed using an exhaustive search. The accuracy of matching depends on the size of the grid step.

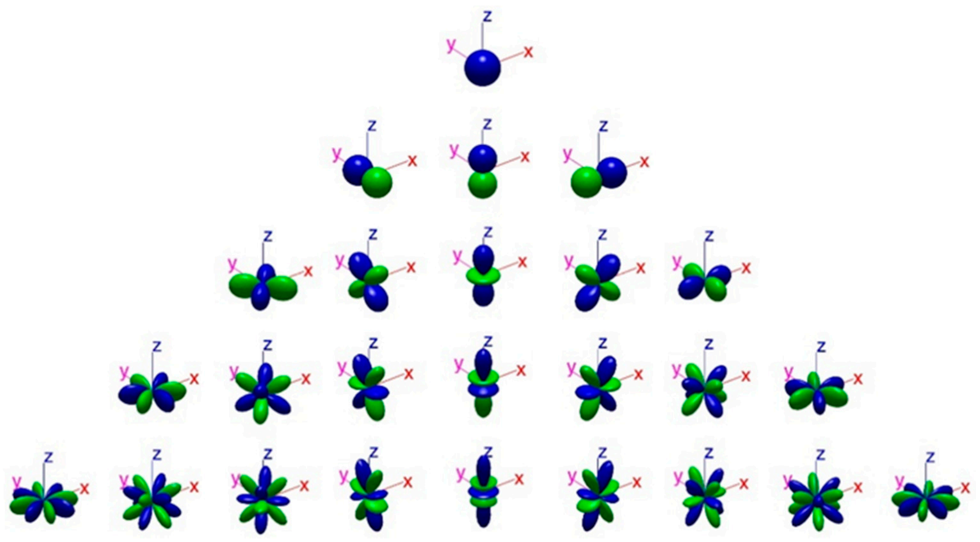

Figure 2. Spherical harmonics (degrees from 0 to 4). The green areas correspond to the negative values of the harmonics, and the blue ones correspond to the positive ones.

Suppose several non-zero coefficients exist in the expansion for some degree l (Formula (4)). In that case, the number of surface extrema for each harmonic is limited. For the partial reconstruction (4), the number of extrema in the sum of weighted spherical harmonics participating will also be bounded. The same consideration is valid for the sum of all the degrees participating in the partial reconstruction (5).

We will show that under some assumptions, the sum of all the partial reconstructions at all the shells mapped to the unit sphere will have the expected number of top-value extrema to be relatively small (on average, below thirty). We define top-value extrema as those that are among the largest by absolute value. A more formal definition will be given in Section 3.8.

3.5. Non-Gradient Methods

The non-gradient method could be applied if either the direction of function descent is calculated without taking a gradient, like in [41] the simplex or Powell methods [40,42], or if the number of points to be checked for being the extremum is relatively small and we can examine them one by one. We will follow the second approach for the problem of mapping two 3D objects.

If we look at spherical harmonics inside correspondent degrees (see Figure 2), we can notice that when the band has a low number, the spherical harmonics have a relatively limited number of local extrema (the local peaks on the 3D surface of the harmonic). When we find a weighted sum of these harmonics, the number of local extrema at the resulting surface will also be limited and relatively small.

Considering that function f is a weighted sum of spherical harmonics, and the spherical harmonics of different degrees are independent, we can say that function f itself is the sum of partial reconstructions corresponding to every degree. These partial reconstructions for degree l for 3D objects O_1 and O_2 will also be rotated at the same angle as the original objects, O_1 and O_2. This observation is accurate for any specific degree or combination of degrees. Also, it is accurate for whatever square-integrable function on the surface of a sphere is analyzed. This means that exactly the same angle of rotation will be calculated (up to the error of calculation) for every partial reconstruction of degree l on every concentric sphere.

Having a moderate number of local extrema at the partial reconstruction for level l for O_1 and O_2 and considering that all the rotations are around the center of coordinates, the angle of rotation to rotationally match the peaks of reconstructed surfaces to each other could be performed, for example, by an exhaustive search of all possible configurations in a low-dimensional space. This calculation provides a rotation matrix to match O_1 and O_2. Instead of the exhaustive search (due to a limited number of top-valued extrema generated), the solution could be found by analyses of only several possible combinations. The measure of the accuracy of mapping could be, in addition to M_1 (Formula (7)) and M_2 (Formula (8)), the Root-Mean-Square Deviation between two reconstructions.

3.6. Efficient Calculation of the Shortest Distance Function (SDF)

The SDF function calculation we used is based on library [48], where instead of the SDF, which computes the distance to all the faces of the surface, this value is approximated by the distance to the faces adjacent to the nearest neighbors of the grid point $\left(\theta_j, \varphi_j, R_i\right)$ on the i-th sphere with radius R_i.

3.7. Building the Surface at Icosphere Nodes

As mentioned in Section 3.3, we cut the object by concentric spheres. Let us consider an SDF function $f^{(i)}(\theta, \varphi)$—the shortest distance from the point (θ, φ) of the i-th sphere to the surface S. Function $f^{(i)}(\theta, \varphi)$ is continuous by parameters θ and φ, as well as by the radius of the sphere and the vector $\left(\theta, \varphi, f^{(i)}(\theta, \varphi)\right)$, which is defined in the spherical coordinate system. If the value of $f^{(i)}(\theta, \varphi) < 0$, the corresponding vector of length $\left|f^{(i)}(\theta, \varphi)\right|$ is directed opposite vector $(\theta, \varphi, 1)$.

At each sphere, a grid of points, $\left(\theta_j, \varphi_j\right)$, $j = 1, \ldots, k$, is defined. We use an icosphere [51] set of points out of 10,242 vertices after five subdivisions. This icosphere has a dense and rather uniformly distributed set of vertices. Using values $f^{(i)}\left(\theta_j, \varphi_j\right)$, we compute (3) the spherical harmonic coefficients $f_{lm}^{(i)}$; then, we partially reconstruct function $\tilde{f}^{(i)}\left(\theta_j, \varphi_j\right)$ on the unit sphere at the icosphere grid $\left(\theta_j, \varphi_j\right)$ for degrees $l = 0, \ldots, 10$ using (1).

As was mentioned in steps P8–P11, on the unit sphere, we add functions $\tilde{f}^{(i)}\left(\theta_j, \varphi_j\right)$ over the entire set of concentric spheres, see Formula (2), and connect the points $\left(\theta_j, \varphi_j, \tilde{f}\left(\theta_j, \varphi_j\right)\right)$ to make a surface C with the adjacency matrix for the icosphere. We find this surface's extrema and saddle points and build the convex hull using the points found. The number of vertices in the convex hull built on these points is usually much smaller than the number of points in the original surface C. These vertices are ordered in sequence F by the length of the radius vector. We obtain a similar sequence G for function g.

Let us form the sequence $\tilde{F}$ of lengths of vectors corresponding to the vertices of the convex hull F. The global maxima of $\tilde{F}$ will have the same values as the global maxima for $\tilde{f}\left(\theta_j, \varphi_j\right)$ and will be reached at the same icosphere grid nodes. The values corresponding to the global maxima will be concentrated in the right tail of the distribution S_n. The number of extrema on a convex hull cannot be more than the number of extrema for the original surface S_n on which it is built.

If objects O_1 and O_2 are similar, then to match them, we can only use the top extrema from F and G and later check the quality of the mapping.

The process described here will be referred to as extrema amplification hereafter.

3.8. Values of Function as Dependent Random Variables

Let us find a set of values of the function $\tilde{f}^{(i)}\left(\theta_j, \varphi_j\right)$ (1) at the icosphere grid $\left(\theta_j, \varphi_j\right)$, and consider these values as a random variable X_i distributed on the segment $\left[f_{\min}^{(i)}, f_{\max}^{(i)}\right]$

from global minima to global maxima. The variable X_i at each sphere $i = 1, \ldots, n$ has finite mathematical expectations and variances as follows:

$$E[X_i] = \mu_i, \quad D[X_i] = E\left[(X_i - \mu_i)^2\right] = \sigma_i^2.$$

Let us discuss the rationale for the summation of functions $\tilde{f}^{(i)}(\theta_j, \varphi_j)$ in Formula (2) and how it ensures that the number of variants to be checked to find the solution is below two on average. In the proposed method, to obtain either a satisfactory result for object rotation (if it exists) or a negative answer otherwise (objects are not identified as similar), it is sufficient to apply the matching algorithm no more than 30 times. This number is confirmed experimentally using the algorithm.

To explain how the idea appeared and what is the logic behind it, we will first carry out the justification under the unrealistic assumption that the random variables X_i corresponding to the distributions of the values of the distance functions $\tilde{f}^{(i)}(\theta_j, \varphi_j)$ on each sphere are independent. Then, the possibility of using the results found for the classes of dependent random variables will be discussed.

We know nothing about the 3D object and its SDF, as no assumptions are made about the object's shape and topology. So, first, let us consider a toy example, then how the addition of constraints affects the result, and finally, how the experimental result is compared with the expected theoretical result.

3.9. Method 1: Central Limit Theorem

Consider the sum of random variables $S_n \equiv \sum_{i=1}^{n} X_i$. Then, in particular

$$E[S_n] = m_n = \sum_{i=1}^{n} \mu_i, \quad D[S_n] = \sigma^2 = \sigma^2(n) \equiv \sum_{i=1}^{n} \sigma_i^2. \tag{9}$$

One of the classical formulations of the central limit theorem is contained, for example, in [54] (p. 383, Theorem 27.2).

Lindenberg's Central Limit Theorem. *Suppose that sequenceX_i, $i = 1, \ldots,$ n is independent and satisfies (9). If Lindenberg's condition*

$$\lim_{n\to\infty} \sum\nolimits_{i=1}^{n} \frac{1}{\sigma^2} \int_{|X_i - \mu_i| \geq \varepsilon\sigma} (X_i - \mu_i)^2 \, dP = 0, \tag{10}$$

holds for all positiveε, then $\frac{S_n - m_n}{\sigma} \to N_{0,1}$ *at* $n \to \infty$.

Here
$$\frac{S_n - m_n}{\sigma} = \frac{(X_1 - \mu_1) + \ldots + (X_n - \mu_n)}{\sqrt{\sum_{i=1}^{n} \sigma_i^2}},$$

and $N_{0,1} \equiv N(0;1)$ denotes the standard Gaussian random variable, i.e., having zero mathematical expectation and variance equal to one. The density of the standard normal distribution is
$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

This theorem follows from the more general Lyapunov theorem [54] (p. 385, Theorem 27.3). While the classical central limit theorem requires finite moments of order 2, Lyapunov's central limit theorem requires finite moments of order $2 + \delta$ for some $\delta > 0$.

Lyapunov's Central Limit Theorem. *Suppose that sequence X_i, $i = 1, \ldots, n$ is independent and satisfies (9). If Lyapunov' condition*

$$\lim_{n\to\infty} \frac{1}{\sigma^{2+\delta}} \sum\nolimits_{i=1}^{n} E\left[|X_i - \mu_i|^{2+\delta}\right] = 0, \tag{11}$$

holds for some $\delta > 0$, then $(S_n - m_n)/\sigma \to N_{0,1}$ at $n \to \infty$.

As shown in [54] (p. 385), the Lindenberg condition (10) follows from the Lyapunov condition (11), but not vice versa.

Note that fulfilling the conditions of the Lindenberg and Lyapunov theorems is only sufficient to satisfy the requirement of independence and not the identical distribution of random variables.

A special case of Lyapunov's theorem is the Bates theorems [55] on the continuous probability distribution of the mean of independent uniformly distributed random variables on the unit interval and the Irwin–Hall theorems [56,57] on the continuous probability distribution for the sum of independent and identically distributed random variables. These theorems also tend to have a normal distribution.

Applying Lyapunov's theorem to the sum $S_n \equiv \sum_{i=1}^{n} X_i$, we find that S_n will approach the normal distribution. The summation is performed per each element of X_i, and S_n will also have 10,242 elements. Due to the trend to the normal distribution, the maximum values at the right tail of the distribution (as well as the minimum values in the left tail) will have a relatively low probability. For example, if the extremum is greater than 3σ, where σ is the variance of this normal distribution, then the probability of encountering these values will be 0.27%. In other words, if we have a large sample from the population of all sums S_n, then the number of extremes in the sample above 3σ will be approximately 0.27%. There are 10,242 points in our set. This means that 99.73% of values will be expected to be less than 3σ in both directions, and approximately no more than 30 values are greater than 3σ. The matching algorithm will use these 30 longest vectors (and the corresponding pyramids—triplets of non-coplanar vectors in steps A5 and A6).

Note that the longest vectors among the extrema can be located not only on the right tail of the S_n distribution but also on the left tail since the terms $f^{(i)}(\theta, \varphi)$ can also take negative values and the local minima of the sum $f(\theta, \varphi)$ can turn out to be negative in value but rather large by the absolute value and, therefore, enter the constructed convex hull based on the known functional property

$$\max f = -\min(-f).$$

In the case of a continuous function on a sphere, the extrema of function f built for object O_1 will correspond to the extrema of function g built for object O_2 as rotation of O_1. The first vector from the list of the extrema of f must have the same length as the first vector from g. However, when the function is sampled on a grid, the extrema of continuous functions are not necessarily located precisely at the grid nodes. Therefore, all grid calculations are performed with a certain error, and extrema for f and g may differ from each other due to the choice of sampling nodes. This is the reason behind checking several vectors' triplets for the solution and choosing the one with the best correspondence.

If a grid on the sphere surfaces is not dense (for example, containing only 100 nodes), the possible error of calculation of the extrema is significant. In such a case, even an exhaustive search on all possible triplets of vectors does not ensure obtaining the correct answer with an acceptable error. For this reason, we use a relatively dense grid.

We realize that in the discussion above, our assumption of the independence of X_i is hardly realistic, but at this stage of the presentation, it is only performed to explain the basis of actions, such as why the idea to sum partial reconstruction functions on all the shells has appeared and where the limited number of steps to finalize the analysis could potentially come from.

The mentioned results about the trend to normal distribution will not only be valid under the assumption of independence of the random variables X_i (X_i corresponds to the spherical functions f_i), as in Lyapunov's theorem formulation. In recent decades, Lyapunov's theorem has also been proven for some classes of dependent random variables, so the trend to normal distribution will also be held for more general cases. Examples of such classes might be the following:

1. Martingale differences [58–60];
2. m-dependent and α-mixing sequences, such as [54], pp. 387–388, and [61], pp. 773–774, [62–64];
3. Mixing conditions [65–69].

The common feature for all these results is the following. The random variables "far away" from each other are almost independent in a certain sense [70]. This happens for the parts of the object under the assumption that these parts weakly depend on each other. There are also classical results by Bruns, Markoff, S. Bernstein, P. Lévy, and Loéve (see [71–74] (chapter VIII)) for dependent random variables. The formulated conditions of these theorems are either very strict or include conditional distributions, which makes their application difficult.

These results are also aligned with the statistical extreme value theory (EVT) [75] (chapter 3) [76]. Both the central limit theorem and the theory of extreme values allow us to estimate the probability of a rare event. This event can be the result of the accumulation of many events or be only one event exceeding some critical threshold [76]. Unfortunately, for these theories, the essential requirement is the identical and independent distribution (i.i.d.) of random variables.

The SDF functions corresponding to the spheres cutting the surface are the distance functions. The SDF is continuous by polar angles and the radius. This means that the neighboring spheres at the points with the same angular coordinates cannot differ by an arbitrarily large value. Therefore, the respective values of SDF functions on spheres are dependent.

In [77], the case of extreme value theory for dependent random variables was considered. It was shown that if the variables are not identically distributed or are dependent, the result is similar to the case of independent identically distributed random variables. However, as usual, the result relies on very restrictive conditions. Checking the feasibility of these conditions is also a non-obvious and sometimes non-formalized process.

3.10. Method 2: Multiplication Rule of Probability and Chebyshev's Inequality

Let us assume that we have a random value distributed at the segment $[a,\ b]$ and also a random sample of this distribution. Let us call the significant extrema those that correspond to the leftmost (rightmost) bin in the histogram, and the length of each bin is ½ * 0.27% of the length of the segment $[a,\ b]$. This arrangement arises from the following considerations. For the normal distribution, the part of the distribution outside the 3σ interval should contain approximately 0.27% of the whole distribution. If we take the uniform distribution at segment $[a,\ b]$, these last bins are expected to receive the same 0.27% of the overall samples. Later, to map the 3D objects, we will only consider the extrema from these last bins.

Let us explore another way to show that the number of significant extrema of summation of two spherical functions will not increase compared to the number of significant extrema of each function. This consideration is based on the probability multiplication theorem and does not even require a Gaussian assumption.

Since the same number of nodes (namely, 10,242) is used on each of n spheres, we divide all intervals, $\left[f_{\min}^{(i)}, f_{\max}^{(i)}\right]$ and $i = 1, \ldots, n$, by the same constant number N_{bin} of bins. Let $P(X_i)$ be the probability of the random variable X_i to be a significant maximum, i.e., to get into the rightmost bin $\left[f_{\max}^{(i)} - \left(f_{\max}^{(i)} - f_{\min}^{(i)}\right)/N_{\text{bin}}, f_{\max}^{(i)}\right]$ or to the leftmost one. Let us reiterate that the length of both bins is 0.27% of the length of segment $\left[f_{\min}^{(i)}, f_{\max}^{(i)}\right]$.

According to the multiplication rule of probability, we have $P(X_1X_2) = P(X_1)P_{X_1}(X_2)$, where $P(X_1X_2)$ is the probability of occurrence of both events, i.e., the probability that the sum $(f_1 + f_2)$ is the significant maximum, and it includes the significant maxima for f_1 and f_2. $P_{X_1}(X_2)$ is the conditional probability of X_1 occurring given that X_2 occurs, i.e., the probability of a point on the sphere to be a significant maximum of f_1, when f_2 is a also significant maximum. In the case of independent random variables X_1 and X_2, the conditional probability is $P_{X_1}(X_2) = P(X_2)$. As we see, in any case, it turns out to be $P(X_1X_2) \leq P(X_1)$ since $P_{X_1}(X_2) \leq 1$. On the other hand, $P(X_1X_2) = P(X_2)P_{X_2}(X_1)$ and $P(X_1X_2) \leq P(X_2)$, thus

$$P(X_1X_2) \leq \min\{P(X_1), P(X_2)\}.$$

By induction, a similar conclusion can be drawn for n random variables as follows:

$$P(X_1X_2\cdots X_n) \leq \min\{P(X_1), \ldots, P(X_n)\}. \tag{12}$$

Due to the multiplication rule of probabilities

$$P(X_1X_2\cdots X_n) = P(X_1)P_{X_1}(X_2)P_{X_1X_2}(X_3)\cdots P_{X_1\cdots X_{n-1}}(X_n).$$

Formula (12) only shows a non-increase of the number of significant extrema in the sum and does not provide a quantitative assessment.

For significant minima, the same considerations are valid.

Chebyshev's inequality allows us to quantify the probabilities $P(X_1)$, …, $P(X_n)$ as being significant extrema in (12).

Chebyshev's Inequality. *If a random variable X has expectation $E[X]$ and variance $D[X]$, then the inequality holds for any $\varepsilon > 0$*

$$P(|X - E[X]| > \varepsilon) \leq \frac{D[X]}{\varepsilon^2}.$$

If we consider the sum S_n from Method 1 as a random variable X (without any assumptions about the dependence or independence of its terms), then, we estimate the probability that the deviation of this quantity S_n from its expectation $E[S_n]$ will be more than three standard deviations in absolute value.

We have $D[S_n] = \sigma^2$ and $\varepsilon = 3\sigma$; then, according to Chebyshev's inequality

$$P(|S_n - E[S_n]| > 3\sigma) \leq \frac{D[X]}{\varepsilon^2} \leq \frac{\sigma^2}{(3\sigma)^2} = \frac{1}{9}.$$

This estimate is not really helpful in practice as it corresponds to the worst-case scenario.

From the two approaches described above, we see that the weaker the restrictions imposed on the class of random variables, the rougher (less accurate) the theoretical estimates are.

For this reason, we decided to analyze the actual data statistics and compare how close they are to the theoretical results (see Section 4).

4. Results and Discussion

We use a dataset out of 1309 STL files collected from ShapeNet (The Princeton Shape Benchmark (PSB), Version 1) [78,79], Princeton ModelNet 40-Class Subset [80], Purdue Engineering Shape Benchmark (ESB) [81], and internet websites. Every model used has a correct geometry.

The extrema amplification process described in Section 3.10 (Method 2) generates the most prominent extrema. The amplification includes summation, which is performed per node for all the shells, looking for the extrema and saddle points, and building the convex hull using those points. We expect that the sum assumes a certain amount of independence of the multiple components (as discussed in Section 3.9 (Method 1)).

We ran the method described on the database of 3D objects for 1309 objects. It was found that in 94% of cases, the first attempt (performed by steps A5–A7 of the algorithm) to align the objects using the amplified extrema was immediately successful. Five attempts were enough to achieve 99% success. The maximum number of attempts was set to 30, and it allowed us to find the solution in all the cases.

Let us find the rationale for this small number of attempts to find the solution.

For each object of the dataset mentioned, we calculate the values

$$f_{\text{spread}} = \sum_{i=1}^{n} \left| \tilde{f}{}^{(i)}_{\max} - \tilde{f}{}^{(i)}_{\min} \right|, \qquad \tilde{f}_{\text{spread}} = \left| \tilde{f}_{\max} - \tilde{f}_{\min} \right|$$

The first one is the sum of the spreads of functions $\tilde{f}{}^{(i)}$ defined by (1), and the second one is the spread of the final sum $\tilde{f}$; see Formula (2).

Let us consider the ratio $\tilde{f}_{\text{spread}}/f_{\text{spread}}$ for the objects of the dataset. It turned out that the expectation and variance of this ratio are 0.90 and 0.08, respectively. It can be interpreted that in 90 percent of cases, there is at least one extremum on each sphere with the same angular coordinates.

The total number of extrema points in the last bins near $\tilde{f}_{\max}$ and $\tilde{f}_{\min}$ appeared following two trends. If the object does not have symmetrical (repeatable) parts, the number of points in the last bins is less than 30 on average. It corresponds to approximately 0.27% of all the points (recall that we use an icosphere with 10,242 vertices).

The second trend manifests itself when the number of points in the last bins is relatively high, and this number can come to above two hundred. This situation appears when some parts of the object are similar or the object has some symmetry (for example, rotational symmetry). These symmetrical/similar parts may contribute a large number of extrema points of the convex hull to the last bins. This trend appears when all right-part probabilities are high in inequality (12). If there is some rotational self-similarity of the object, this number of extrema could be rather large, but all these extrema will define the same transformation of the 3D objects.

From this reasoning, in both cases, we have to try matching only a limited number of the extrema to check the object mapping. The results of running the method on a dataset demonstrate that a few attempts to match the objects (steps A5–A7) are enough to conclude whether the object is the same or different.

Due to grid sampling of the square-integrable functions f and g from (4), the matching could have some errors, and the best matching using metrics M_1 and M_2 (step A7) will define the best acceptable solution (if found).

We propose an algorithm to identify similar objects. It is a non-iterative and non-gradient method of optimization. The current state-of-the-art mapping algorithms are the ICP and Go-ICP.

For complex shapes with many local extrema and tens or hundreds of thousands of vertices, the ICP might be much more time consuming (say, 3 min vs. 30 s of our algorithm) and sometimes stuck in the local quasi-optimal solution (yellow and blue shapes do not match each other's after rotation found by the ICP, Figure 3).

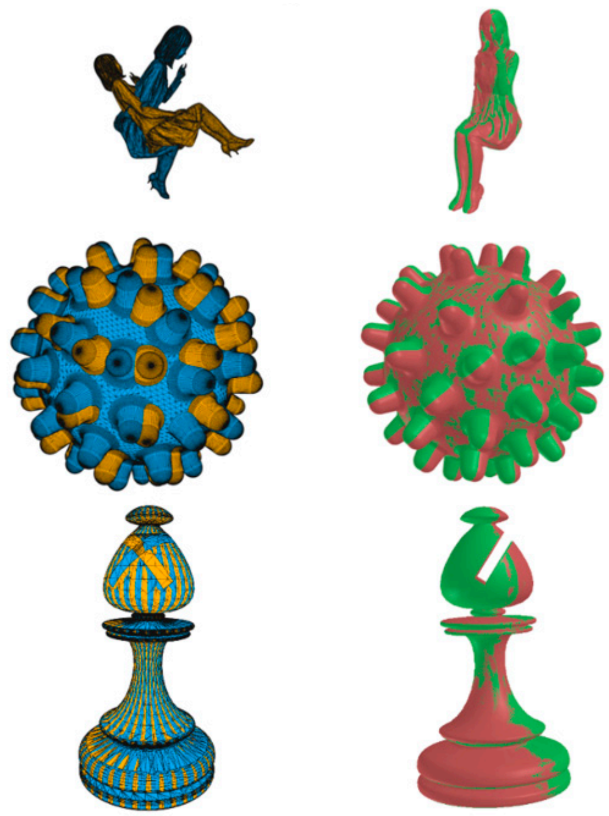

Figure 3. ICP vs. our algorithm. Left side: The results of mapping rotated blue and original gold objects by the ICP. Right side: Mapping of rotated green and original red objects using our algorithm.

Two algorithms (the ICP and ours) use different approaches to search for optimal mapping. Our approach is based on structural correspondences of spherical harmonics. It will not catch and, hence, not align the elements that are described by spherical harmonics of high degrees (greater than $L_{\max}$). The idea to stop at $L_{\max}$ is based on the following rationale. We work with a grid, and the errors in calculating coefficients for higher degrees are affected by the errors made for coefficients of lower degrees.

The accuracy of our algorithm and the ICP is analyzed in the following way.

Initially, we mapped both objects into the sphere of radius 16. The center of the sphere is the centroid of the object's surface, and the farthest from the centroid point of the object mesh lies on the sphere's surface.

When we look for the mapping of two objects as input data for the ICP algorithm, we use the point cloud, which contains all the mesh vertices. The number of vertices in the point cloud may reach millions. The mapping error is the average of all the distances from the mesh vertices of the first object to the surface of the second object after mapping. Suppose the ICP is converged to the global extremum. In that case, the average mapping error is usually small, around 10^{-9} and less, and running time, on average, is around 0.18 s (we used Open3D implementation of the ICP). If the ICP converged to the incorrect optimum, the error could be arbitrarily huge, and the time to complete the run could reach up to 30 min for huge STL files (Figure 3, top and middle row examples).

For our algorithm implementation, the computation time varies from 8 s for relatively not very complex meshes (<50,000 vertices) to below a minute for a huge mesh with more than 1M vertices. This time includes reading the STL file of the model, which could reach a hundred megabytes and even gigabytes in size.

To compare the ICP and our method, we chose the Princeton dataset [78] and a threshold of 0.14 for the maximum mapping error ME (Table 1). This threshold is an average value for ME for the dataset [78] when using our approach. It divides the dataset approximately by half. For 53% of the objects in the dataset, the ICP has the maximum ME below this threshold, and other cases demonstrate relatively high errors. Suppose we do not consider these two datasets separately. In that case, the error received in the second case can make the average error of the ICP method extremely high despite great accuracy for the other half of the data samples.

Table 1. ICP vs. our algorithm. The threshold for maximum ME is 0.14. It divides the dataset [78] into two parts. For ME, the mean value (avg), standard deviation (std), and maximum (max) are calculated for all the considered datasets. Also, the average and the maximum time to process 3D objects by the algorithm (in seconds) are provided. The ICP statistics are given for the whole dataset and parts of the dataset with maximum mapping error max(ME) $\leq$ 0.14 and max(ME) > 0.14. The part of the considered dataset is provided in the column named "% of data".

Description	**Mapping Error**			**Time to Run**		**% Of Data**
	Avg	**Std**	**Max**	**Avg**	**Max**	
Our algorithm	0.14	0.07	2.98	8	51.7	100
ICP (whole dataset)	1.30	0.82	16.52	0.18	5.47	100
ICP for max(ME) $\leq$ 0.14	6×10^{-4}	4.6×10^{-3}	0.13	0.18	5.47	53
ICP for max(ME) > 0.14	2.73	2.06	16.52	0.18	2.79	47

Table 1 shows that our algorithm gave an average error (average distance to the surface) below 0.2 with a standard deviation of this value of less than 0.1. The maximum point-wise deviation for the whole dataset used was below 3. We can see that if we compare our approach with the ICP for the entire dataset, our method looks more accurate. Unfortunately, this commonly accepted type of analysis does not consider that the ICP has two clusters with different behaviors, as Table 1 shows.

Methods with the same setting should be considered to compare the accuracy and efficiency of different approaches correctly. That is why we checked the validity of our algorithm by comparing it with the ICP (which performs a mapping of point clouds and not subsampled data, as neural networks do; see Section 4.1). The ICP appeared to be the best method for comparison, but unfortunately, the settings of both methods are not perfectly the same. The ICP does the mapping of the point clouds to find the best mapping. Our method task is either to find a rigid transformation of two objects or to reject the identity of these objects. Let us see how different the solutions are received. Assume we have two objects, as shown in Figure 4a,b. The ICP method provides the mapping shown in Figure 4c. It maps the animal bodies and ignores the failure to map the absent fangs of cat (b) to cat (a). Our algorithm finds that both objects are not identical and stops processing.

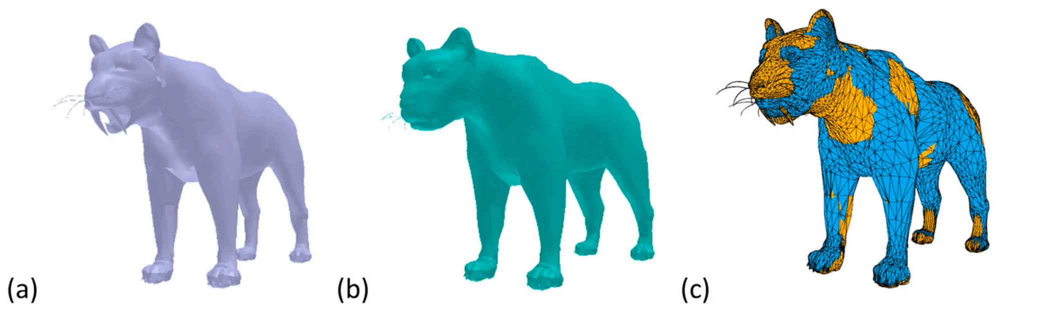

Figure 4. (**a**) A model of a feline with fangs, (**b**) a feline without fangs, (**c**) the ICP mapping of both models.

4.1. Deep Learning Approach vs. Statistical Analytical Approach

The review of the current deep learning approaches shows they have definite constraints. As an example of a deep learning approach, let us consider the ICCV 2019 paper by Wang and Solomon [43]. It describes a non-iterative and non-gradient method based on applying deep learning. The deep learning model should be trained on a dataset of 3D objects, and later, the transformation to map the objects will be produced.

In the mentioned manuscript, the following are pointed out:

- Current deep learning networks can only handle object-level point clouds (each usually has around 500 to 5000 points); this is a common limitation of recent point cloud learning methods.
- The deep learning model can be improved by iteration or "polishing" via the classical ICP (iterative closest point algorithm).
- The deep learning model is trained on some input data. The application of the method to unseen categories of objects shows a sharp drop in accuracy.
- The extraction of the rigid motion is performed by solving 3×3 SVD eigenproblems. This solution might be unstable due to the multiplicity of eigenvalues and sensitivity to outliers, as was shown by Kazhdan [13] (Appendix B).

Usually, the proposed method is compared with the state-of-the-art ICP method and its modification, the Go-ICP method [36]. The ICP method may converge to the global or local extremum. The convergence to the local extremum could show a huge discrepancy with the ground truth solution. Calculation of the average accuracy of converged and non-converged (local extremum convergence) cases could show the inferiority of the ICP, even if it has higher or comparable accuracy in an overwhelming majority of all the cases in comparison with the deep learning method. We think the accuracy should be calculated separately for the category of globally converged data, with the percentage of locally converged cases in the dataset mentioned.

We would like to mention the following:

- Our algorithm does not depend on the end-to-end trained model, and we propose a solution based on a statistical analytical approach.
- It does not matter to our algorithm which class of objects it uses; its accuracy does not depend on whether it has seen this class before.
- Our algorithm was able to process 3D models with tens of millions of vertices and find global extrema, even when the ICP cannot, and the memory requirements for a deep learning approach are excessive.
- It considers the internal architecture of the object, not only the tiny amount of surface points.
- Our algorithm does not use a voxel representation of the object; this representation may result in the removal of the fine details of the object and would make the surface rough. The proposed approach to finding extrema statistically by summating random values is not among classical techniques or modification of the known approaches; to the best of our knowledge, it has not been used before by anyone.

Summarizing the abovementioned, we can conclude the following.

The data preparation for neural network processing requires either voxelization of the object inside the box with a limited number of overall voxels or choosing a limited subset of points to represent the object (usually around 1 K points only). We work directly with the mesh of thousands, and even millions of vertices, and subsampling of this mesh would affect the accuracy of mapping objects with fine details and complex internal structures. Additionally, the comparison result would significantly depend on the choice of these 1 K points. That is why our approach cannot be directly compared to the neural network approaches.

4.2. Cases When Our Approach Does Not Work

It would be naïve to assume that this method would work in all the cases. We found that when the object's surface has many superficial details, which mainly look like a texture, our method may generate the transformation corresponding to substandard results; see Figure 5, top and bottom rows. The issue, likely, is in a large number of components at the surface of the object. The number of degrees $L_{\max} = 10$ used for reconstruction is probably not enough to catch the correct alignment of a large number of similar components. In support of this claim, we can refer to Figure 5, a bottom left cell with a stress ball, where the alignment is acceptable, likely due to the not-so-big number of repeated components.

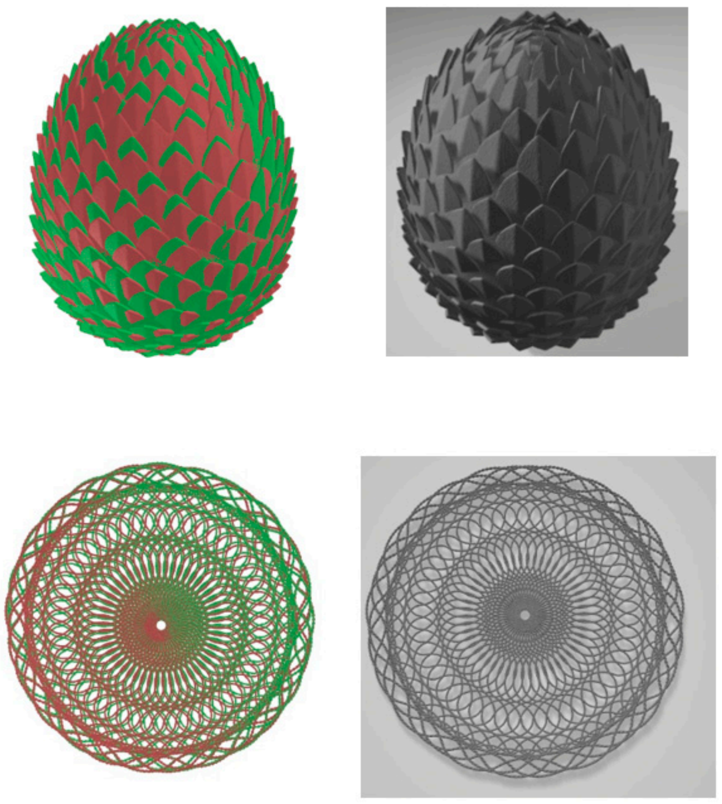

Figure 5. Examples when the mapping calculated by the method proposed is substandard. In each row, the (**right**) figure is the target, and the (**left**) is the result of the mapping of rotated green and original red objects. (**top**) Dragon egg—the surface components (spirally arranged scales) are not aligned. (**bottom**) Tablecloth—the holes of the objects are rotationally misaligned.

4.3. Advantages of Our Method

In our work, we followed up and further developed the ideas described in [7,13,15,16]. In these works, the voxelization of the surface is considered, and the approach is based on the expansion of the indicator function defined on each concentric sphere crossing the surface into a sum of spherical harmonics. The value of the indicator function on the corresponding sphere is 1 if there is an intersection with the surface and 0 otherwise. For the partial reconstruction of discontinuous function by its spherical coefficients, in addition to computational errors associated with surface voxelization and sampling the function on a grid, the Gibbs effect takes place in [82] (chapter 4). This effect manifested near the discontinuity point of a function [83] as a jump in the sum of its Fourier series, which exceeds the jump of discontinuity of the function itself by 17.9%. In contrast, in our work, we consider the triangular surface of the object, not voxels, and only continuous functions $f^{(i)}(\theta,\varphi)$ defined on the concentric spheres are used. Their sum $f(\theta,\varphi) = \sum_{i=1}^{n} f^{(i)}(\theta,\varphi)$ will also be a continuous function as the sum of a finite number of continuous functions.

Modifications made in our approach allow us to improve the accuracy of the identification. The summation of partial reconstructions at the grid nodes over the concentric spheres allows us to obtain the final grid. After summation over all spheres, the extrema on each sphere become even more prominent among the other sum values on the grid. We find the extrema and saddle points on this grid and build a convex hull on the found points. When we look for the mapping to match both 3D objects, we try to map the essential extrema of the respective objects. The amplification of the extrema causes a reduction in the number of mappings to check, so the final result is obtained in a limited (fixed) number of steps without calculation of the gradient.

5. Conclusions

We propose a non-gradient and non-iterative method to map two 3D objects. The statistical analytical method is based on amplifying extrema by summating dependent random values. As a result, a limited number of extrema matchings is obtained. This allows us to identify whether objects are different or the same and to find the transformation matrix in the last case just by checking a very limited number of candidates to be the solution. The method works for objects having millions of vertices and does not require subsampling to the range of several thousand vertices or voxelization as deep learning methods require. It works for all types of objects without the necessity to train the algorithm for some specific class. We hope that the method presented could be extended to other optimization problems.

Author Contributions: Conceptualization, I.V. and H.B.; investigation, methodology, writing original draft, review & editing, I.V., S.K., A.M. and H.B.; project administration, I.V. and H.B.; data curation, formal analysis, software, validation, I.V., S.K. and A.M.; supervision, funding acquisition, H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported under the RIE2020 Industry Alignment Fund—Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as by cash and in-kind contributions from industry partner HP Inc. through the HP-NTU Digital Manufacturing Corporate Lab.

Data Availability Statement: The original data presented in the study are openly available in [34,42,43,78–81].

Conflicts of Interest: The authors declare that this study received funding from HP Inc. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication. In addition, Helen Balinsky is employed by HP Inc. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as potential conflicts of interest.

References

1. Bustos, B.; Keim, D.; Saupe, D.; Schreck, T.; Vranic, D. An experimental comparison of feature-based 3D retrieval methods. In Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004, Thessaloniki, Greece, 9 September 2004; pp. 215–222. [CrossRef]
2. Tangelder, J.W.; Veltkamp, R.C. A survey of content based 3D shape retrieval methods. *Multimed. Tools Appl.* **2004**, *9*, 441–471. [CrossRef]
3. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
4. Zhang, J.; Yao, Y.; Deng, B. Fast and robust iterative closest point. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3450–3466. [CrossRef] [PubMed]
5. Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D. Matching 3D models with shape distributions. In Proceedings of the Proceedings International Conference on Shape Modeling and Applications, Genova, Italy, 7–11 May 2001; pp. 54–166. [CrossRef]
6. Zhang, F.; Zhang, L.; He, T.; Sun, Y.; Zhao, S.; Zhang, Y.; Zhao, X.; Zhao, W. An overlap estimation guided feature metric approach for real point cloud registration. *Comput. Graph.* **2024**, *119*, 103883. [CrossRef]
7. Funkhouser, T.A.; Min, P.; Kazhdan, M.M.; Chen, J.; Halderman, J.A.; Dobkin, D.P.; Jacobs, D.P. A search engine for 3D models. *ACM Trans. Graph.* **2003**, *22*, 83–105. [CrossRef]
8. Qi, C.R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and multi-view CNNs for object classification on 3D data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5648–5656. [CrossRef]
9. Gao, Z.; Shao, Y.; Guan, W.; Liu, M.; Cheng, Z.; Chen, S. A novel patch convolutional neural network for view-based 3D model retrieval. In Proceedings of the 29th ACM International Conference on Multimedia, Need York, NY, USA, 20 October 2021; pp. 2699–2707.
10. Cheng, X.; Liu, X.; Li, J.; Zhou, W. Deep learning-based point cloud registration: A comprehensive investigation. *Int. J. Remote Sens.* **2024**, *45*, 3412–3442. [CrossRef]
11. Elad, M.; Tal, A.; Ar, S. Content based retrieval of VRML objects: An iterative and interactive approach. In *EG Multimedia*; Springer: Vienna, Austria, 2001; pp. 97–108. [CrossRef]

12. Horn, B.K.P. Extended Gaussian images. *Proc. IEEE* **1984**, *72*, 1671–1686. [CrossRef]
13. Kazhdan, M.M.; Funkhouser, T.A.; Rusinkiewicz, S.M. Rotation invariant spherical harmonic representation of 3D shape descriptors. *Eurographics Symp. Geom. Process.* **2003**, *6*, 156–165. [CrossRef]
14. Ankerst, M.; Kastenmüller, G.; Kriegel, H.-P.; Seidl, T. 3D shape histograms for similarity search and classification in spatial databases. In *Advances in Spatial Databases, SSD 1999*; Güting, R.H., Papadias, D., Lochovsky, F., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1651, pp. 207–226.
15. Kazhdan, M.M.; Funkhouser, T.A. Harmonic 3D shape matching, International Conference on Computer Graphics and Interactive Techniques. In Proceedings of the SIGGRAPH 2002, San Antonio, TX, USA, 21–26 July 2002; p. 191. [CrossRef]
16. Kazhdan, M.; Chazelle, B.; Dobkin, D.; Finkelstein, A.; Funkhouser, T. A reflective symmetry descriptor. In *Computer Vision—ECCV 2002*; Lecture Notes in Computer Science; Heyden, A., Sparr, G., Nielsen, M., Johansen, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2351. [CrossRef]
17. Vranic, D.V.; Saupe, D.; Richter, J. Tools for 3D-object retrieval: Karhunen–Loeve transform and spherical harmonics. In Proceedings of the 2001 IEEE Fourth Workshop on Multimedia Signal Processing (Cat. No.01TH8564), Cannes, France, 3–5 October 2001; pp. 293–298. [CrossRef]
18. Saupe, D.; Vranic, D.V. 3D Model retrieval with spherical harmonics and moments. In Proceedings of the Pattern Recognition, 23rd DAGM-Symposium, Munich, Germany, 12–14 September 2001; pp. 392–397. [CrossRef]
19. Kazhdan, M.M.; Funkhouser, T.A.; Rusinkiewicz, S.M. Symmetry descriptors and 3D shape matching. In Proceedings of the Eurographics Symposium on Geometry Processing, Berlin, Germany, 15–17 July 2004. [CrossRef]
20. Papadakis, P.; Pratikakis, I.; Perantonis, S.J.; Theoharis, T. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognit.* **2007**, *40*, 2437–2452. [CrossRef]
21. Papadakis, P.; Pratikakis, I.; Theoharis, T.; Perantonis, S.J. PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. *Int. J. Comput. Vis.* **2010**, *89*, 177–192. [CrossRef]
22. Mousa, M.-H.; Chaine, R.; Akkouche, S.; Galin, E. Efficient spherical harmonics representation of 3D objects. In Proceedings of the 15th Pacific Conference on Computer Graphics and Applications (PG'07), Maui, HI, USA, 29 October–2 November 2007; pp. 248–255. [CrossRef]
23. Zarpalas, D.; Daras, P.; Axenopoulos, A.; Tzovaras, D.; Strintzis, M.G. 3D Model search and retrieval using the spherical trace transform. *EURASIP J. Adv. Signal Process* **2007**, *2006*, 023912. [CrossRef]
24. Ziyang, C. Retrieval of 3D Models Based on Spherical Harmonics. In Proceedings of the 2010 International Conference on Electrical and Control Engineering, Wuhan, China, 25–27 June 2010; pp. 2991–2994. [CrossRef]
25. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A* **1976**, *32*, 922–923. [CrossRef]
26. Kabsch, W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. Sect. A* **1978**, *34*, 827–828. [CrossRef]
27. Horn, B.K.; Hilden, H.M.; Negahdaripour, S. Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A* **1988**, *5*, 1127–1135. [CrossRef]
28. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [CrossRef]
29. Makadia, A.; Daniilidis, K. Direct 3D-rotation estimation from spherical images via a generalized shift theorem. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003. [CrossRef]
30. Esteves, C.; Allen-Blanchette, C.; Makadia, A.; Daniilidis, K. Learning SO_3 equivariant representations with spherical CNNs. *Intern. J. Comput. Vision.* **2017**, *128*, 588–600. [CrossRef]
31. Salihu, D.; Steinbach, E.G. SGPCR: Spherical Gaussian point cloud representation and its application to object registration and retrieval. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 572–581. [CrossRef]
32. Wang, X.; Yin, Z.; Xiong, H.; Su, D.; Feng, Y. A spherical-harmonic-based approach to discrete element modeling of 3D irregular particles. *Intern. J. Numer. Methods Eng.* **2021**, *122*, 5626–5655. [CrossRef]
33. Hobson, E.W. *The Theory of Spherical and Ellipsoidal Harmonics*; Chelsea: New York, NY, USA, 1965.
34. Chen, Y.-C.; Lin, C.-H.; Hsu, P.-C.; Chen, C.-H. Point cloud encoding for 3D building model retrieval. *IEEE Trans. Multimed.* **2014**, *16*, 337–345. [CrossRef]
35. Chen, Y.-C.; Lin, C.-H. Image-based airborne LiDAR point cloud encoding for 3D building model retrieval. *ISPRS—Inter. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *41*, 1237–1242. [CrossRef]
36. Yang, J.; Li, H.; Campbell, D.; Jia, Y. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2241–2254. [CrossRef]
37. Locatelli, M.; Schoen, F. *Global Optimization: Theory, Algorithms, and Applications*; Saunders: Philadelphia, PA, USA, 2013. [CrossRef]
38. Locatelli, M.; Schoen, F. (Global) Optimization: Historical notes and recent developments. *EURO J. Comput. Optim.* **2021**, *9*, 100012. [CrossRef]
39. Spendley, W.; Hext, G.R.; Himsworth, F.R. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics* **1962**, *4*, 441–461. [CrossRef]

40. Powell, M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **1964**, *7*, 155–162. [CrossRef]
41. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [CrossRef]
42. Powell, M.J.D. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*; Gomez, S., Hennart, J.-P., Eds.; Kluwer Academic: Dordrecht, The Netherlands, 1994; pp. 51–67.
43. Wang, Y.; Solomon, J. Deep Closest Point: Learning Representations for Point Cloud Registration. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3522–3531. [CrossRef]
44. Wu, Y.; Liu, J.; Yuan, Y.; Hu, X.; Fan, X.; Tu, K.; Gong, M.; Miao, Q.; Ma, W. Correspondence-Free Point Cloud Registration Via Feature Interaction and Dual Branch. *IEEE Comput. Intell. Mag.* **2023**, *18*, 66–79. [CrossRef]
45. Wu, Y.; Hu, X.; Zhang, Y.; Gong, M.; Ma, W.; Miao, Q. SACF-Net: Skip-attention based correspondence filtering network for point cloud registration. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 3585–3595. [CrossRef]
46. Wu, Y.; Liu, J.; Gong, M.; Liu, Z.; Miao, Q.; Ma, W. MPCT: Multiscale Point Cloud Transformer with a Residual Network. *IEEE Trans. Multimed.* **2024**, *26*, 3505–3516. [CrossRef]
47. Huang, X.; Qu, W.; Zuo, Y.; Fang, Y.; Zhao, X. IMFNet: Interpretable multimodal fusion for point cloud registration. *IEEE Robot. Autom. Lett.* **2022**, *7*, 12323–12330. [CrossRef]
48. Pysdf 0.1.9. SDF: Convert Triangle Mesh to Continuous Signed Distance Function + Some Other Mesh Utilities. Available online: https://pypi.org/project/pysdf/ (accessed on 9 October 2023).
49. Hardin, R.H.; Sloane, N.J.A. McLaren's improved snub cube and other new spherical designs in three dimensions. *Discret. Comput.* **1996**, *15*, 429–441. [CrossRef]
50. Fliege, J.; Maier, U. *A Two-Stage Approach for Computing Cubature Formulae for the Sphere*; Universitat Dortmund: Dortmund, Germany, 1996.
51. Sun, C.; Sherrah, J. 3D symmetry detection using the extended Gaussian image. *Pac. Asian Manag. Inst.* **1997**, *19*, 164–168. [CrossRef]
52. Wigner, E.P. *Group Theory and Its Application to the Quantum Mechanics of Atomic Spectra*; Academic Press: New York, NY, USA, 1959.
53. Penrose, R. A generalized inverse for matrices. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [CrossRef]
54. Billingsley, P. *Probability and Measure. Wiley Series in Probability and Statistics*, 3rd ed.; Wiley: New York, NY, USA, 2012; ISBN 0-471-00710-2/978-1-118-12237-2.
55. Bates, G.E. Joint distributions of time intervals for the occurrence of successive accidents in a generalized Polya urn scheme. *Ann. Math. Stat.* **1955**, *26*, 705–720. [CrossRef]
56. Irwin, J.O. On the frequency distribution of the means of samples from a population having any law of frequency with finite moments, with special reference to Pearson's type II. *Biometrika* **1927**, *19*, 225–239. [CrossRef]
57. Hall, P. The distribution of means for samples of size N drawn from a population in which the variate takes values between 0 and 1, all such values being equally probable. *Biometrika* **1927**, *19*, 240–245. [CrossRef]
58. Billingsley, P. The Lindeberg–Levy theorem for matringales. *Proc. Am. Math. Soc.* **1961**, *12*, 788–792. [CrossRef]
59. Ibragimov, I.A. A central limit theorem for a class of dependent random variables. *Theor. Probab. Appl.* **1963**, *8*, 83–89. [CrossRef]
60. Rosén, B. On the central limit theorem for sums of dependent random variables. *Z. Für Wahrscheinlichkeitstheorie Und Verwandte Geb.* **1967**, *7*, 48–82. [CrossRef]
61. Hoeffding, W.; Robbins, H. The central limit theorem for dependent random variables. In *The Collected Works of Wassily Hoeffding*; Fisher, N.I., Sen, P.K., Eds.; Springer Series in Statistics; Springer: New York, NY, USA, 1994. [CrossRef]
62. Berk, K.N. A central limit theorem for m-dependent random variables with unbounded m. *Ann. Probab.* **1973**, *1*, 352–354. [CrossRef]
63. Diananda, P.H. The central limit theorem for m-dependent variables. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 92–95. [CrossRef]
64. Shang, Y. A central limit theorem for randomly indexed m-dependent random variables. *Filomat* **2012**, *26*, 713–717. [CrossRef]
65. Rosenblatt, M. A central limit theorem and a strong mixing condition. *Proc. Natl. Acad. Sci. USA* **1956**, *42*, 43–47. [CrossRef]
66. Bradley, R.C. Basic properties of strong mixing conditions: A survey and some open questions. *Probab. Surv.* **2005**, *2*, 107–144. Available online: http://eudml.org/doc/223765 (accessed on 9 October 2023). [CrossRef]
67. Kaminski, M. Central limit theorem for certain classes of dependent random variables. *Theory Prob. Its Appl.* **2007**, *51*, 335–342. [CrossRef]
68. Balan, R.; Zamfirescu, I.-M. Strong approximation for mixing sequences with infinite variance. *Electron. Commun. Probab.* **2006**, *11*, 11–23. [CrossRef]
69. Ould-Saïd, E.; Tatachak, A. Strong consistency rate for the kernel mode estimator under strong mixing hypothesis and left truncation. *Commun. Stat.* **2009**, *38*, 1154–1169. [CrossRef]
70. Berkes, I.; Philipp, W. Limit theorems for mixing sequences without rate assumptions. *Ann. Probab.* **1998**, *26*, 805–831. [CrossRef]
71. Bernshtein, S.N. New applications of almost independent quantities. *Izv. Akad. Nauk SSSR Ser. Mat.* **1940**, *4*, 137–150.
72. Bernshtein, S.N. Sums of dependent variables, having mutually almost zero regression. *Dokl. AN SSSR* **1941**, *32*, 303–307.
73. Lévy, P. *Théorie De l'Addition Des Variables Aléatoires*; Gauthier-Villars: Paris, France, 1937.

74. Loéve, M. *Probability Theory*, 3rd ed.; D. Van Nostrand Company, Inc.: Princeton, NJ, USA, 1963.
75. Jacob, M.; Neves, C.; Greetham, D.V. *Forecasting and Assessing Risk of Individual Electricity Peaks, Mathematics of Planet Earth*; Springer: Cham, Switzerland, 2020; pp. 39–60. [CrossRef]
76. Gumbel, E. *Statistics of Extremes*; Columbia University Press: New York, NY, USA, 1958. [CrossRef]
77. Zhang, R.-R.; Zhukovskii, M.E.; Isaev, M.; Rodionov, I.V. Extreme value theory for triangular arrays of dependent random variables. *Russ. Math. Surv.* **2020**, *75*, 968–970. [CrossRef]
78. The Princeton Shape Benchmark. Available online: https://shape.cs.princeton.edu/benchmark/benchmark.pdf (accessed on 15 May 2023).
79. Shilane, P.; Min, P.; Kazhdan, M.M.; Funkhouser, T.A. The Princeton shape benchmark. In Proceedings of the Proceedings Shape Modeling Applications 2004, Genova, Italy, 7–9 June 2004; Giannini, F., Pasko, A., Eds.; IEEE: Toulouse, France, 2004; pp. 167–178. [CrossRef]
80. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1912–1920. [CrossRef]
81. Jayanti, S.; Kalyanaraman, Y.; Iyer, N.; Ramani, K. Developing an engineering shape benchmark for CAD models. *Comput. Aided Design.* **2006**, *38*, 939–953. [CrossRef]
82. Vretblad, A. Fourier Transforms. In *Fourier Analysis and Its Applications*; Axler, S., Gehring, F.W., Ribet, K.A., Eds.; Graduate Texts in Mathematics; Springer: New York, NY, USA, 2003; Volume 223, Chapter 7; pp. 165–195. [CrossRef]
83. Mathews, J.; Walker, R.L.; Davidon, W.C. Mathematical methods of physics. *Am. J. Phys.* **1965**, *33*, 246. [CrossRef]