

Accurate and Efficient Similarity Search on 3D Objects Using Point Sampling, Redundancy, and Proportionality

Johannes Aßfalg^{1,2}, Hans-Peter Kriegel¹, Peer Kröger¹, Marco Pötke²

¹ Institute for Computer Science, University of Munich, Germany

² sd&m AG, Munich, Germany

{assfalg,kriegel,kroegerp}@dbs.ifi.lmu.de, marco.poetke@sdm.de

Abstract. With fast evolving resources for 3D objects such as the Protein Data Bank (PDB) or the World Wide Web, new techniques, so-called *similarity models* to efficiently and effectively search for these 3D objects become indispensable. Invariances w.r.t. specific geometric transformations such as scaling, translation, and rotation are important features of similarity models. In this paper, we focus on rotation invariance. We first propose a new method of representing objects more accurately in the context of rotation invariance than the well-known voxelization technique. In addition, we extend existing feature-based similarity models by proposing a new spherical partitioning of the data objects based on proportionality and redundancy¹, and generalizing an existing method for feature extraction. A broad experimental evaluation compares our method with existing methods in terms of accuracy and efficiency. In particular, we experimentally confirm that our point sampling method is better suited to represent 3D objects in the context of rotation invariance than voxelized representations. In addition, we empirically show that our new similarity model significantly outperforms competitive rotation invariant models in terms of accuracy as well as efficiency.

1 Introduction

During the last years, more and more 3D models became available, e.g. through the fast-growing protein database PDB [1] or through the World Wide Web [2]. This trend will probably continue and thus, new techniques are required to efficiently and effectively search within such 3D databases.

This paper covers feature-based methods to describe 3D objects, so called *shape descriptors*. Shape descriptors extract numerical features of a 3D object so that the object is mapped to a metric space called *feature space*. The similarity of two spatial objects is then measured by the proximity of their feature vectors.

In this paper, we are particularly interested in shape descriptors that are invariant with respect to rotation, translation, and scaling.

¹ patent pending

We will base our work on an existing shape descriptor (the *volume model*[3]) by applying a combination of new techniques. The original descriptor is already invariant w.r.t. the mentioned transformations because it uses a spherical partitioning of data objects after moving the balance point of an object to the origin of the coordinate system. For a given object, the model extracts the volume of the object from each partition as feature.

The retrieval quality of this very intuitive but simple model can be significantly enhanced by our newly proposed techniques. The first improvement we propose in this paper is a new representation of 3D objects. Instead of the commonly used voxel representation, we use uniformly distributed surface points.

Our second enhancement is a new way of decomposing an object into several partitions. Instead of constructing equidistant shells like in [3], our method creates partitions that are dynamically adapted to the specific shape of an object. In addition, we propose a new technique for the assignment of parts of an object to different partitions. Instead of assigning each part to exactly one partition, our new *redundant assignment* method associates parts of an object to several overlapping partitions.

A third enhancement we are proposing is to use an advanced feature extraction method instead of the rotation invariant but rather simple feature extraction based on the volume of the object in a partition. In particular, we show how the *eigenvalue model* which is presented in [4] for voxelized data working with cubic partitionings, can be generalized for arbitrarily shaped partitionings. We further adopt this generalization to our new object partitioning method.

We empirically show how these newly introduced techniques can be combined, to significantly improve quality of similarity search on 3D models.

The rest of the paper is organized as follows. We review related work on rotation invariant shape descriptors in Section 2. We then discuss the limitations of voxel representation in the context of rotation invariance and present our solution to this problem based on point sampling in Section 3. Section 4 presents our new object partitioning methods and Section 5 describes the generalization of the eigenvalue model. Section 6 summarizes our new method to compute rotation invariant shape descriptors. The experimental evaluation of our methods is presented in Section 7. Section 8 provides conclusions.

2 Related Work

Various methods for the characterization of 3D shapes have been reported. In this section we focus on techniques that result in a rotation invariant representation of an object without requiring a normalization step.

In [5] Osada et al. presented a technique called “D2” to describe 3D shapes. At first the distances of pairs of randomly selected points on the surface of an object are measured. These distances are used to create a histogram that is finally used as a feature vector. This method is intended to distinguish large differences in geometric shape. It is not able to discriminate between objects

that differ only in small details. In addition, a high number of pairwise distances has to be calculated to obtain a robust description of the object.

The idea of the D2 descriptor was refined in [6]. The authors not only calculated the distance between two randomly distributed surface points, but also classified it according to the position of the two points. The line between two points either can lie completely inside the object, completely outside of the object, or inside and outside of the object. This separation yields three histograms instead of one. The presented results show that very similar shape distributions are still derived for dissimilar parts. The complexity of the method is the same as above while the time that has to be spent for each pair of points is even longer since the connecting line has to be classified.

In [7], Kazhdan et al. presented a rotation invariant shape descriptor by applying the spherical Fourier analysis to a number of spherical functions defined on a voxel grid. The concentric spheres constructed around the center of the voxel grid are used to define corresponding spherical functions.

In [8] the authors describe a shape descriptor based on the work in [7]. A spherical function is defined by measuring the distance between the surface of an object and its balance point. This function is afterwards analyzed with the spherical Fourier transform and the Fourier coefficients are used to characterize the object.

As the computation of the Fourier coefficients by means of the SFT algorithm requires a lot of function values we decided to use the method in [7] for our experimental comparisons because it is easier to determine whether or not a voxel is filled than to intersect a large number of rays with a triangle mesh.

Hilaga et al. in [9] introduced a technique to characterize 3D shapes based on so-called Reeb graphs. The better these graphs match, the higher is the similarity between the corresponding objects. However, this method compares the topology of objects rather than their geometry.

An intuitive and rotation invariant technique to describe the 3D shapes of proteins was presented by Ankerst et al. in [3]. Since our method uses a similar technique for object partitioning, we will present details on this method in Section 4. The method also will be included in our experimental comparisons.

3 Representation of 3D Objects

In this section, we discuss how 3D objects can be represented in order to efficiently derive spatial features. Often, 3D objects are given by triangle meshes, i.e. sets of connected triangles. The algorithm in [10] for example calculates a triangulated surface of a protein. Although well suited for the graphical display, this representation is still too complex for the efficient computation of feature vectors. In the following, we discuss voxelization as a method to represent 3D objects and its problems in terms of rotation invariance. Thereafter, we propose a new representation method called *point sampling* that overcomes the shortcomings of voxelization regarding rotation.

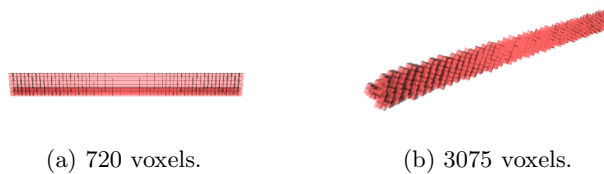


Fig. 1. Rotation variant voxel representation.

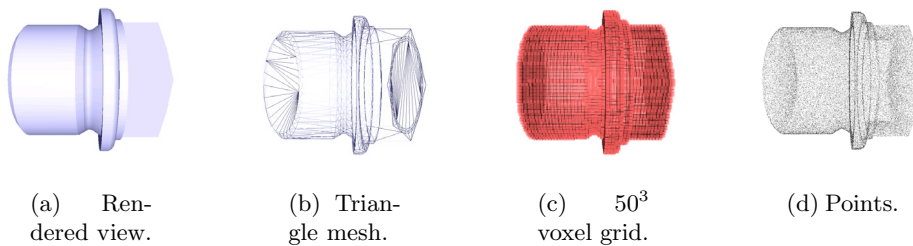


Fig. 2. Different representations of a 3D object.

3.1 Voxelization

A well-known method to represent 3D data for extracting spatial features is voxelization. For voxelization, an object is usually placed into a standardized cube. This cube is partitioned into n^3 small cubes (so-called voxels) using an equidistant grid. A voxel is considered as filled if it intersects a triangle of the triangle mesh. The algorithm proposed by Kaufman in [11] yields a conservative approximation of the object surface by means of voxels. A further advantage of this representation is the possibility to model filled objects by adding interior voxels.

However, this representation is not invariant with respect to rotations, i.e. the number of voxels changes when the object is rotated. This effect is illustrated in Figure 1. As we will demonstrate experimentally, this rotation variant representation of objects reduces the quality of shape descriptors working with voxel input.

3.2 Point Sampling

In this section we introduce another approach to represent a given mesh of triangles. Let $\mathcal{M} = \{T_1, \dots, T_m\}$ be the set of triangles the mesh consists of. For $1 \leq i \leq m$ let A_i denote the area of the triangle T_i and let $P_i, Q_i, R_i \in \mathbb{R}^3$ be the three vertices of T_i . Then in a first step the total area $A = \sum_{i=1}^m A_i$ of all triangles is calculated and the desired number $n \in \mathbb{N}$ of points to be sampled is chosen. Afterwards, the following steps are repeated n times:

Selection of a Triangle. With probability $p = \frac{A_i}{A}$ triangle T_i is selected. This selection is implemented by a preprocessing step assigning to each triangle the sum of the areas of all its predecessors plus its own area, i.e. $A_{sum_i} = \sum_{j=1}^i A_j$. Afterwards, a random number $r \in \mathbb{R}, 0 \leq r \leq A$ is created and the triangle T_j is selected so that $A_{sum_j} \geq r$ and $\nexists i : 1 \leq i < j : A_{sum_i} \geq r$. **Sampling of a Random Point.** According to [12] a random point is created inside triangle T_j .

In contrast to the voxelization method this method is not able to model filled volumes. Nevertheless, our experiments show an improved quality when using shape descriptors working on models that have not been normalized before the shape descriptor is applied.

Figure 2 shows different representations of a 3D object.

4 Partitioning Data Objects

The general idea of our rotation invariant shape descriptors is to partition an object into shells similar to the method proposed in [3] and then extracting spatial features from each partition. We show how this method can be adopted and enhanced in terms of accuracy. Let us note that the following concepts do not rely on a certain kind of object representation, i.e. objects represented as voxels or using sampled points. In the following, we assume that the data objects are represented as points, either center points of voxels or sampled points.

4.1 Construction of the Shells

The shell model to partition the data was originally proposed in [3]. After the computation of the balance point M (cf. Section 5) of a given object O and the radius of the bounding sphere around O the space enclosed by the bounding sphere is divided up into $k \in \mathbb{N}$ shells.

Equidistant Shell Construction. The original approach in [3] constructs k shells whose distance to each other is the same for each pair of adjacent shells. So we will refer to this method as the *equidistant* construction method. Although the innermost shell actually is a sphere, we nonetheless will denote it as a shell. Let $r_{BS} \in \mathbb{R}$ be the radius of the bounding sphere. Then the thickness δ of each shell, i.e. the radius of the inner sphere, can be computed by $\delta = r_{BS}/k$.

For each $i \in \{1, \dots, k\}$ the shell S_i is characterized by its inner radius r_{i-1} and its outer radius r_i where $r_i = \delta \cdot i$.

This method is illustrated in Figure 3. An object is inscribed into its minimal bounding sphere and the bounding sphere is afterwards partitioned into $k = 3$ equidistant shells. This method corresponds to the calculation of so-called *shape histograms* [3].

Proportional Shell Construction.¹ The shell construction method of [3] described above keeps the distance between adjacent shells constant. However, this method does not partition the bounding sphere into areas of equal complexity and thus of equal interest. Shells of very low complexity (i.e. with few

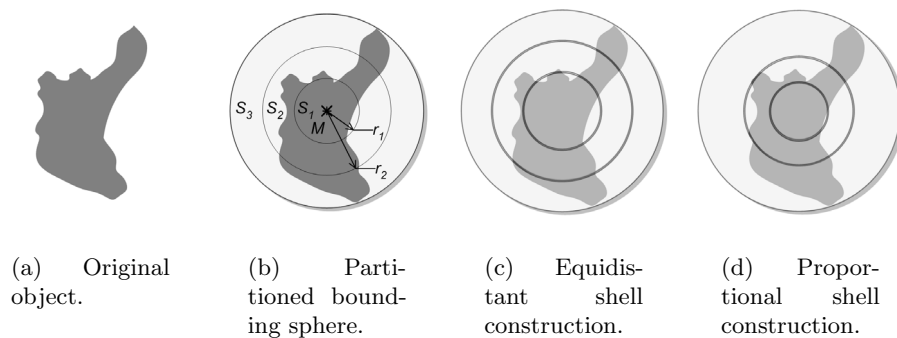


Fig. 3. Shell construction.

points) are most likely of very low interest. The granularity of the partitioning is too fine to extract meaningful spatial features in that case. In the worst case, the equidistant construction method may result in empty partitions that are of no interest. On the other hand, other shells contain many more points. In this case, the granularity of the partitioning may be too coarse to extract meaningful spatial features. The solution we are proposing in the following is to apply a proportional shell construction such that each partitioning has the same level of complexity and interest.

Let $O = \{p_1, \dots, p_n\}$ be an object represented by a set of points (either representing voxels or sampled points as discussed above). Let M be the balance point of O , let r_{BS} be the radius of the bounding sphere around O , and let $k \in \mathbb{N}$ be the desired number of shells to be constructed. Then the number $a \in \mathbb{N}$ of points to be placed into each shell is given by $a = \lfloor n/k \rfloor$.

Now the points of O are sorted in ascending order with respect to the Euclidean distance d_{eucl} of its elements to M such that O can be rewritten as $O = \{p_{s(1)}, \dots, p_{s(n)}\}$, where $d_{eucl}(M, p_{s(1)}) \leq d_{eucl}(M, p_{s(2)}) \leq \dots \leq d_{eucl}(M, p_{s(n)})$.

Let $k \neq 1$. Then the shell S_i , $i \in \{1, \dots, k-1\}$, is characterized by its inner radius r_{i-1} and its outer radius r_i where $r_i = d_{eucl}(M, p_{s(i \cdot a)})$.

The outer radius r_k of the outmost shell S_k is set to r_{BS} . In case of objects represented using point sampling, this radius is not necessarily equal to $d_{eucl}(M, p_{s(n)})$. This is due to the random process with which the sampled points of O have been created. The inner radius r_{k-1} of S_k is calculated as described above.

The radii of the shells are important for our method of feature extraction from the partitions which will be presented in Section 5.

If $n \neq k \cdot a$, the remaining elements of O ($p_{s(k \cdot a + 1)}, \dots, p_{s(n)}$) are assigned to the outmost shell.

Obviously, the proportional shell construction method partitions the bounding sphere into areas of equal complexity and thus of equal interest. The granularity of the partitioning adopts to the shape of the data objects. The method

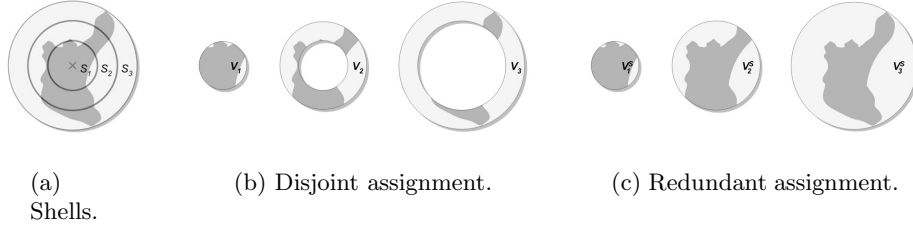


Fig. 4. Disjoint vs. redundant assignment. Cross section of the bounding sphere.

is illustrated in Figure 3(d). While the equidistant shell construction method corresponds to the calculation of shape histograms, the proportional method generates *shape quantiles*.

4.2 Assignment to Shells

After the space enclosed by the bounding sphere has been partitioned, i.e. after the shells have been constructed in an equidistant or proportional way, points of $O = \{v_1, \dots, v_n\}$ (either representing voxels or sampled points) have to be assigned to these partitions.

Let M be the balance point of O and let the bounding sphere be segmented into shells S_1, \dots, S_k , $k \in \mathbb{N}$. Each shell S_i is characterized by its inner radius r_{i-1} and its outer radius r_i , where $r_0 := 0$.

Disjoint Assignment. In [3] a disjoint assignment is proposed. For each shell S_i , a set V_i is created. An element v_j of O is assigned to V_i if and only if $r_{i-1} < d_{eucl}(M, v_j) \leq r_i$.

As every element of O is assigned to exactly one shell, this assignment results in a disjoint decomposition of O into V_1, \dots, V_k . Each set corresponds to the space enclosed by a certain shell. This method is illustrated in Figure 4(b).

Redundant Assignment.¹ The disjoint assignment proposed in [3] has a severe limitation. If two (or more) shells (and their associated elements) are rotated against each other, the resulting feature vector will be exactly the same as the one derived from the unrotated elements. Although the so changed set of points may represent a totally different object, this object will be regarded as very similar to the one described by the unrotated points.

To overcome this weakness we introduce a new method for the assignment of elements to spherical partitions. This method can be applied to both, equidistant and proportional shells.

A set V_i^S is created for each shell S_i . An element v_j of O is assigned to V_i^S if and only if $d_{eucl}(M, v_j) \leq r_i$.

To illustrate the method, imagine $x \in O$, $r_{i-1} < d_{eucl}(M, x) \leq r_i$, i.e. x lies inside the shell S_i . The disjoint assignment method would assign x to only one specific set V_i , whereas the redundant assignment method assigns x to the sets V_i^S, \dots, V_k^S .

It is obvious that rotating a shell against other shells now results in a different feature vector. If for example shell S_i is rotated, the sets V_i^S, \dots, V_k^S will yield different features.

The set V_i^S corresponds to the space enclosed by a sphere with radius r_i that is centered at M . V_k^S represents the space enclosed by the bounding sphere and thus $V_k^S = O$. In Figure 4(c), an example of the partitions resulting from the redundant assignment method is depicted.

4.3 Problems with Thin Shells

No matter how the elements are assigned to the spherical partitions, a problem can occur when the underlying shells are constructed proportionally. If all or almost all points representing an object have the same distance to their balance point, the constructed shells will be very thin.

Let object O be a perfectly triangulated globe, i.e. the triangle mesh of O consists of infinitesimally small triangles and thus the points that are sampled on the surface of O all have the same Euclidean distance to their balance point M , the center of the globe. Sorting the points according to their distance to M will then result in a list whose ordering is in the best case random and in the worst case determined by the creation time of a certain point. The elements of a certain partition will therefore not reflect the globular shape of O . It is more likely that a partition only describes a small (non-globular) area of O .

In reality, due to a triangulation process that is far from being perfect, not all sampled points representing O will have the same Euclidean distance to M . Nonetheless the constructed shells will be very thin and the distribution of points in the partitions will most probably not reflect the globular distribution of the complete set of sampled points. The points assigned to a certain shell will more likely reflect geometric deviations from the ideal surface of the globe due to a non-perfect triangulation of O .

An example of this problem is illustrated in Figure 5(a). The points are marked with four different colors. The colors correspond to the disjoint assignment of the sampled points to different shells. The shape of the object is globular and thus the mentioned problem becomes visible. Instead of being equally distributed, points of the same color are grouped together in certain areas.

A possible solution to this problem is to shuffle the points. Therefore, we introduce a value we call the *blur-distance* β . Let $O = (p_{s(1)}, \dots, p_{s(n)})$ be represented by a list of sampled points. Let M be the balance point of O and let $d_{eucl}(M, p_{s(1)}) \leq \dots \leq d_{eucl}(M, p_{s(n)})$, i.e. the points are sorted in ascending order with respect to the Euclidean distance to M . Thus the blur-distance β is defined by: $\beta = c \cdot d_{eucl}(M, p_{s(n)})$, $c \in [0, 1]$.

The blur-distance is specified as a portion of the maximum distance to M over all elements of O . Our experiments suggest that 0.01 is a good choice for c , so we will set $c = 0.01$ throughout the rest of the paper. The blur-distance specifies regions adjacent to borders between shells. The elements of these regions are afterwards shuffled with regard to their position in the sorted list O . The exact procedure is as follows:

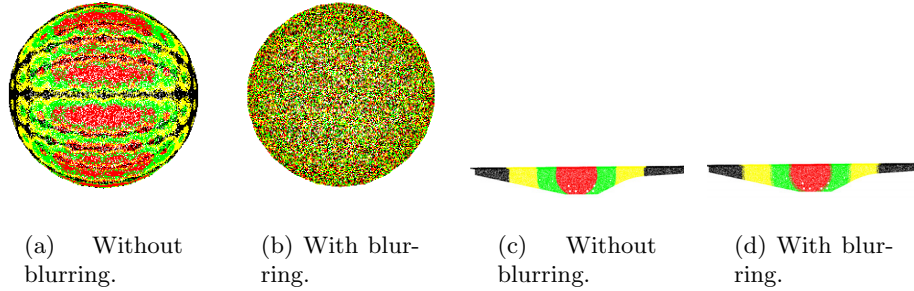


Fig. 5. Impact of blurring on (a),(b) thin shells and (c),(d) thick shells.

1. Let $k \in \mathbb{N}$ be the number of shells to be constructed.
2. The number a of elements to be assigned to each shell is calculated. Then the set $B = \{s(a), s(2a), \dots, s((k-1)a)\}$ contains the indices of the outmost elements (with respect to M) of the shells (except for the outmost shell). These elements mark the border between two adjacent shells.
3. Now we successively use each index $b \in B$ as a starting point in the list O . We proceed downwards in the list until we reach for the first time an index l such that $d_{eucl}(p_b, p_{s(l)}) > \beta$ or until we reach the innermost element of O . In this case, we set $l = 1$.
4. Afterwards, for each $b \in B$ we proceed upwards in the list O until we for the first time reach an index u , such that $d_{eucl}(p_b, p_{s(u)}) > \beta$ or until we reach the outermost element of O . In this case we set $u = n$.
5. As a result, a pair of indices (l, u) was determined for each element of B . Each pair specifies a portion of the list O that now is shuffled (the elements from $p_{s(l+1)}$ to $p_{s(u-1)}$). If two or more of these regions overlap, they are all shuffled at once such that an element of one part of O can be swapped with an element of another part.

This shuffling ensures that the distribution of points of a very thin shell describes all parts of an object that lie in adjacent thin shells. In Figure 5(b) this effect is demonstrated with the globular object mentioned above. Now all shells reflect the spherical shape of the original object. In Figure 5(c) we depicted an object partitioned into thick shells. The clearly defined borders become only slightly blurred when applying the method described above (cf. Figure 5(d)).

5 Extracting Spatial Features

Having partitioned the objects into shells, we have to extract spatial features to build a feature vector. In [3], the volume, i.e. the number of representing points, of the object in a given partition is extracted as feature. The resulting feature

vector consists of one volume measurement in each of the k shells. We refer to this method of extracting spatial features as the *volume model*.

In [4] the eigenvalue model has been introduced which clearly outperforms the volume model in terms of effectiveness. Thus, we argue that it is more accurate to use this eigenvalue model to extract spatial features. The eigenvalue model was originally introduced in [4] but was only applied to a cubic partitioning of a voxelized object. The method is based on PCA, the principal component analysis [13], to analyze a given set of data and extract shape descriptors from a given partitioning. In the following, we describe the basic idea of the eigenvalue model, and generalize a procedure for obtaining feature vectors from the data of a given partition. The sets to be analyzed are the sets of 3D points that lie inside a certain partition.

The eigenvalue model. The basic idea of the eigenvalue model is to obtain the eigenvalues of a set of data within a cubic partitioning as shape descriptors [4]. Originally, it was applied to voxelized data, however, it is quite simple to apply it on objects being represented by sampled points.

Let $V = \{v_1, \dots, v_n\}$ be a set of 3D points, i.e. $v_i \in \mathbb{R}^3, 1 \leq i \leq n$, where $v_i = (v_{i1}, v_{i2}, v_{i3})^T$. If the object is voxelized, the v_i represent the center of voxel v_i , whereas if the object is represented using point sampling, v_i is simply one of the sampled points.

Each $v_i \in V$ is translated such that M afterwards coincides with the origin and afterwards the covariance matrix \mathbf{C} for V is computed as follows:

$$\mathbf{C} = \frac{1}{|V| - 1} \sum_{j=1}^n (\mathbf{v}_j - M) \cdot (\mathbf{v}_j - M)^T, \text{ where } M = \frac{1}{n} \sum_{j=1}^n \mathbf{v}_j \text{ is the balance point.}$$

The covariance matrix can be decomposed as $\mathbf{C} = \mathbf{V}\mathbf{E}\mathbf{V}^T$, where \mathbf{V} is an orthonormal matrix containing the eigenvectors of \mathbf{C} and \mathbf{E} is a diagonal matrix containing the eigenvalues of \mathbf{C} . The eigenvectors are called *principal axes* of V . They describe the three orthogonal axes where the scattering of the elements is greatest. The eigenvalues describe the variance along the three principal axes and thus can be used to characterize the shape of the elements of V . As stated above, in [4], the authors apply this idea to voxelized data. The data objects are partitioned into axis-parallel units. From each unit, the eigenvalues of the voxels (represented as vectors of their center point) are obtained as shape descriptors. Let us note, that this method is not rotation invariant due to the cubic partitioning of the data. The resulting feature vector consists of $3k$ values since for each partition, 3 features (eigenvalues) are extracted.

Generalization and Adoption to Spherical Partitionings. The eigenvalue model as proposed in [4] is applicable on cubic partitionings only and thus is not rotation invariant. In the following, we present a way to normalize the calculated eigenvalues such that eigenvalues of different partitions or even different similarity models can be compared to each other or can be combined with each other. In particular, we show how this model can be applied to spherical partitions like the shells constructed in Section 4. The idea is to express each eigenvalue as a portion of the maximum possible eigenvalue.

Lemma 1. *Let V be a 3D space and let d_{max} be the maximum Euclidean distance two points in V can be apart from each other. The maximum variance Var_{max} that can occur in V is given by $Var_{max} = \frac{1}{2}d_{max}^2$.*

Proof. The highest variance in a certain direction is given by two points that have the highest possible distance from each other and that lie on a line indicating the specific direction. Let therefore $T = \{(a_1, a_2, a_3), (b_1, b_2, b_3)\} =: \{a, b\}$ be a set with two points in a 3D space V and let $d_{eucl}(a, b)$ be the largest Euclidean distance two elements in V can be apart from each other.

The mean value M of T is equal to $(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2}, \frac{a_3+b_3}{2})$. The variance of T is then given by:

$$\begin{aligned} Var(T) &= \frac{\sum_{t \in T} (d_{eucl}(t, M))^2}{|T| - 1} \\ &= \frac{(a_1 - \frac{a_1+b_1}{2})^2 + (a_2 - \frac{a_2+b_2}{2})^2 + (a_3 - \frac{a_3+b_3}{2})^2}{1} \\ &\quad + \frac{(b_1 - \frac{a_1+b_1}{2})^2 + (b_2 - \frac{a_2+b_2}{2})^2 + (b_3 - \frac{a_3+b_3}{2})^2}{1} \\ &= \frac{1}{2}((a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2) = \frac{1}{2} (d_{eucl}(a, b))^2 \end{aligned}$$

Lemma 1 states, that in order to normalize the calculated eigenvalues of V , it is sufficient to determine the largest possible distance d_{max} that can occur in V . Using d_{max} , the maximum possible variance, i.e. the maximum possible eigenvalue, can be computed as indicated in Lemma 1. For example, the maximum distance d_{max} can be the diameter of a shell or the diagonal of a cube. An eigenvalue λ is normalized by the mapping $\lambda \mapsto \lambda/Var_{max}$.

In the following, if we speak of the eigenvalue model, we mean the generalized version of the model, i.e. we will assume that the normalized eigenvalues are obtained as shape descriptors.

The eigenvalue model can now be applied to spherical partitionings such as the shells constructed in Section 4. In both cases of a disjoint or a redundant assignment of points to the shells, the maximum Euclidean distance d_{max} of two elements within a shell S_i or S_i^S , respectively, can be determined by $2r_i$. The maximum variance, i.e. the maximum eigenvalue for scaling, can then be computed according to Lemma 1.

6 Computing Invariant Shape Descriptors

Now we can summarize our proposed methods of computing shape descriptors and discuss whether they are invariant w.r.t. several transformations, including translation, rotation, reflection, and scaling.

The general technique of how invariance with respect to translation, rotation, reflection, and scaling is achieved is the same for all descriptors. Let O be the object whose feature vector is to be calculated. The following steps depend on the type of representation.

Voxel Representation: Let $O = \{v_1, \dots, v_n\}$, $v_i \in \mathbb{R}^3$ be represented by a set of voxels (cf. Section 3.1). Then the balance point $M = \frac{1}{n} \sum_{i=1}^n v_i$ of O is computed. Afterwards, O is moved so that M coincides with the origin. Finally, the minimum bounding sphere centered at M is constructed. The radius r_{BS} of the bounding sphere is determined by $r_{BS} = \max_{v_i \in O} \{d_{eucl}(M, v_i)\}$.

Sampled Points Representation: Let $O = \{p_1, \dots, p_n\}$, $v_i \in \mathbb{R}^3$ be represented by a set of sampled points. In principle the method for the calculation of the center and the radius of the bounding sphere described above can be applied to any set of 3D coordinates. But with regard to the random process, the sampled point set was created with (cf. Section 3.2), we decided to calculate these values with the help of the original triangle mesh. The triangle vertices used to calculate the balance point are weighted by the area of the corresponding triangle. This procedure was proposed in [14]. To determine the radius of the bounding sphere we again use the vertices of the triangles rather than the set of interpolated points. This is due to a problem that may occur when using the sampled points. Consider a very small but long, needle-shaped triangle with one vertex being the most remote point with respect to the balance point. Due to its tininess only one point might be sampled into the triangle. Thus the radius of the bounding sphere may vary significantly depending on the position of this point. Using the vertices of the triangles will result in a radius that is invariant with respect to the randomly generated points.

After these steps, invariance with respect to *translation* is achieved as the balance point of O has been moved to the origin.

Our method partitions the space enclosed by the bounding sphere of a given object into several shells centered at the balance point of the object. In Section 4, we presented two methods to construct these shells, in particular equidistant partitioning as proposed in [3] and proportional partitioning. In addition, we presented two methods to assign the points representing a given object to the shells. The first method uses a disjoint assignment of the points to shells (cf. [3]) whereas the second method uses a redundant assignment.

Afterwards, we apply the eigenvalue model [4] to the elements (voxels or sampled points) of each shell. We presented a generalization of this model in Section 5. As the variance of the elements inside a certain shell does not change even if the object is rotated the resulting eigenvalues are invariant with respect to *rotation*. At the same time, invariance with respect to *reflection* is achieved. The amount of scattering of the elements of a certain shell remains unchanged, if the object is reflected. Finally, the eigenvalues derived for the single partitions are scaled (cf. Section 5) and thus, invariance with respect to *scaling* is achieved.

7 Experiments

In this section, we present our experimental evaluation of the proposed methods. In particular, we will first evaluate the usability of the sampled points representation in comparison to a voxelized representation in the context of rotation invariance. Next, we confirm the superiority of our adoption of the eigenvalue

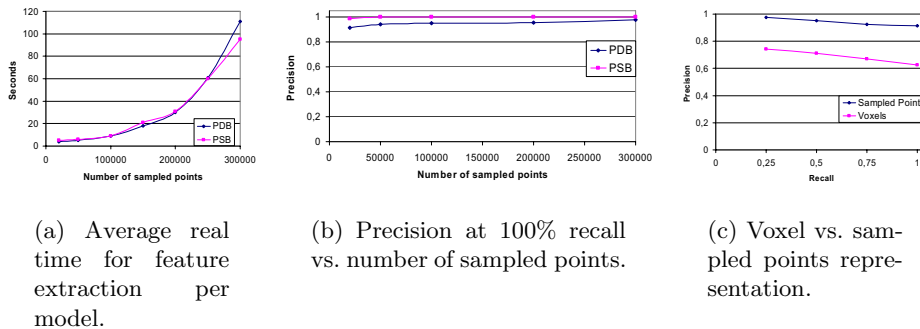


Fig. 6. Evaluation of sample points vs. voxel representation.

model over the volume model. Last but not least, we evaluate the performance of the different shell construction techniques and point assignment strategies described in Section 4 and compare the best shape descriptor with existing work on rotation invariant similarity models.

7.1 Data Sets

For our experiments we used two different real-world data sets that are described in the following.

Princeton Shape Benchmark. The models of this set originate from the Princeton Shape Benchmark Set [2] consisting of 1814 models collected from the World Wide Web. Along with the models, a hierarchical classification is provided that can be used to evaluate the quality of different shape descriptors with precision/recall plots. We decided to only regard the leaves of the classification system. Thus, the set is partitioned into 161 disjoint classes. We will refer to this set of objects as the “PSB set”.

Proteins. A huge amount of 3D protein structure data is available at the online repository of the Protein Data Bank (PDB) [1]. We used the MSMS program [10] to calculate the solvent excluded surface [15] of each molecule. Afterwards the surface was triangulated and so all experiments could be performed on the same type of data. For the classification of the proteins we used the FSSP (Families of Structurally Similar Proteins) classification [16], a well-known classification system for proteins. To create classes of approximately the same size without selecting too many proteins for the test set, we scanned the PDB for classes whose size ranged from 50 to 100 members. This resulted in a set of 3279 proteins in 47 classes which we will refer to as the “PDB set”.

We rotated all models in both sets around a random axis by a randomly determined angle in a preprocessing step to ensure a non-canonical orientation. All experiments were run on an Intel Pentium 4 processor featuring 2.53 GHz and 1 GB RAM.

7.2 Sampled Points vs. Voxel Representation

Our first experiments evaluated the representation of 3D objects. In Section 3 we discussed the limitation of voxelization in the context of rotation invariance and proposed the technique of point sampling to overcome this limitation. To evaluate the usability of voxelization and point sampling for rotation invariance, we randomly selected 100 models from the PDB set and duplicated each selected model four times. In addition, we rotated each copy randomly. Finally the resulting 500 models together with the rest of the PDB set were mapped to a feature space using the eigenvalue model on 8 disjoint and equidistant shells. We applied the same preprocessing to the PSB set.

First we analyzed the influence of the number of sampled points on retrieval quality and computational cost. The task was for each of the 500 preprocessed models to retrieve the 4 corresponding (arbitrarily rotated) models. The experiment was performed for different numbers of sampled points. We measured the average time needed for the feature extraction for one model. The results shown in Figure 6(a) and Figure 6(b) suggest a sensible trade-off between retrieval quality and computational cost is a number of 50,000 sampled points.

We then compared the accuracy of both the sampled points representation and the voxel representation, using precision/recall plots. To confirm our assumption that the representation by voxels is inferior to the sampled points representation we repeated the above described experiments for the 500 duplicated models in both data sets. The models were mapped to a feature space using 20,000 sampled points, and for the 4 possible recall values the average precision was determined. Then the objects of both data sets were represented by a 60^3 voxel grid. The resulting precision/recall plots (cf. Figure 6(c)) confirm our presumption. Although we used a relatively low number of sampled points and a relatively high voxel resolution, the sampled points representation generated significantly higher precision values.

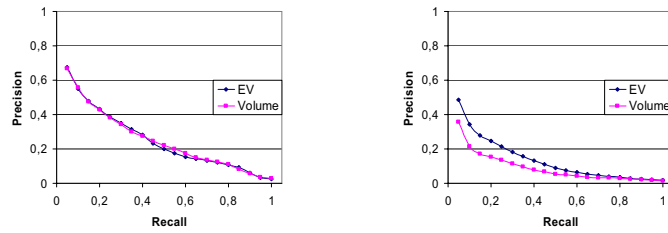
Thus, throughout the rest of our experiments we used the sampled points representation for the 3D objects. Due to our experiments presented above, we set the number of sampled points to 50,000.

7.3 Volume Model vs. Eigenvalue Model

The following experiments evaluate the applicability of our generalization of the eigenvalue model (cf. Section 5). Therefore, we compared the volume model as proposed in [3] with our generalized eigenvalue model. Both models are based on an equidistant shell construction with disjoint point assignment. In case of the eigenvalue model, we used 10 shells whereas in case of the volume model we used the parameter setting that performed best according to [3], i.e. 120 shells.

Figure 7 illustrates the result of the comparison. The precision/recall plot of the volume model is labelled with “Volume” and the results for the eigenvalue model are marked with “EV”.

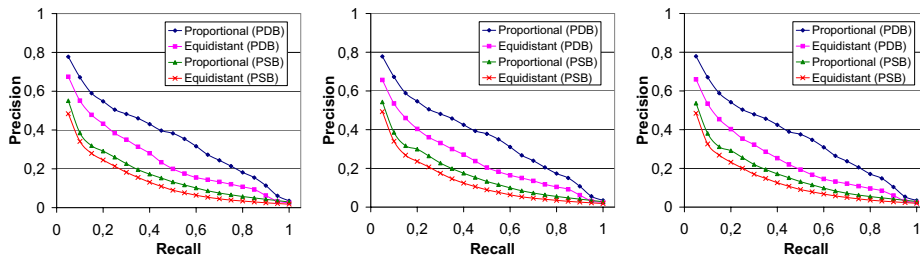
While the two models yield almost the same results on the PDB set (cf. Figure 7(a)), the eigenvalue model yields significantly better results on the PSB



(a) PDB Set.

(b) PSB Set.

Fig. 7. Volume model vs. eigenvalue model.



(a) 8 shells.

(b) 10 shells.

(c) 12 shells.

Fig. 8. Proportional vs. equidistant shell construction.

set (cf. Figure 7(b)). These results underline that the eigenvalue model is more accurate than the volume model. Let us note that in case of the volume model we need 12 times more partitions than using the eigenvalue model resulting in significantly higher waste of resources for the volume model.

Thus, we use the eigenvalue model throughout the rest of the experiments as the method for extracting spatial features.

7.4 Shell Construction and Assignment to the Shells

In this section, we evaluated the different methods of shell construction and shell assignment presented in Section 4.

First we analyzed the impact of the different methods for the shell construction. Figure 8 shows that for both data sets and for different numbers of constructed shells the proportional shell construction significantly outperforms the equidistant method. Let us note that the presented results were achieved with a disjoint assignment step. The same effect occurs when assigning the sampled points redundantly (results are not shown here due to space limitations).

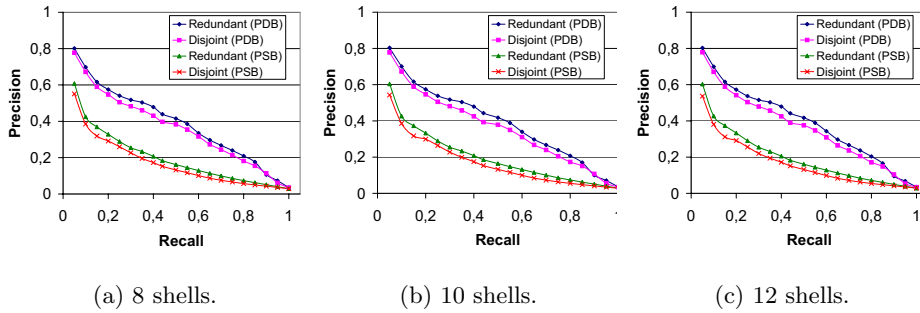


Fig. 9. Redundant vs. disjoint assignment.

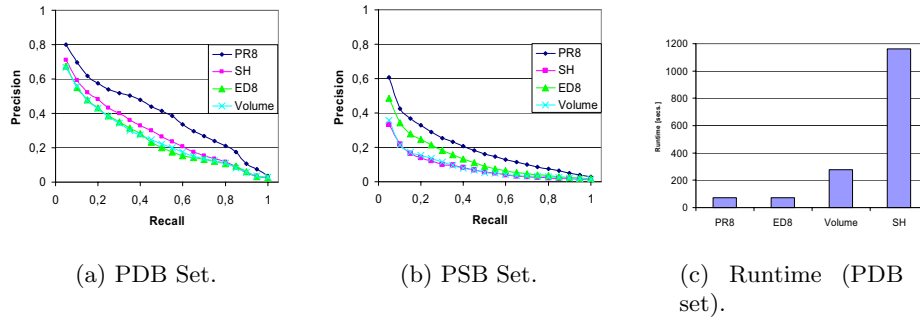


Fig. 10. Results for different shape descriptors.

A further significant improvement in the quality of 3D model retrieval can be observed when using the redundant assignment to the shells. This is shown for different numbers of proportional shells and for both data sets in Figure 9. We empirically verified that this is also true for all considered numbers of equidistant shells (results are not shown here due to space limitations).

7.5 Comparison with Existing Approaches

Last but not least, we compared the best of our proposed shape descriptors with existing rotation invariant methods. In particular, we compared the accuracy of proportional and redundant shells and equidistant and disjoint shells (both combined with the eigenvalue model) with the work presented in [7] and the method proposed in [3]. The latter model corresponds to 120 equidistant and disjoint shells combined with the volume model.

In Figure 10, we show the resulting precision/recall plots for the competing shape descriptors. The plot of the method of [3] is marked with “Volume”, the result of our generalized eigenvalue model applied to 8 equidistant and disjoint

shells is labelled with “ED8”, and the results of our generalized eigenvalue model applied to our 8 proportional and redundant shells are labelled with “PR8”. We furthermore implemented the shape descriptor based on spherical harmonics (marked with “SH” in Figure 10) as described in [7]. The dimensionality of “Volume” is 120, the dimensionality of “ED8” and “PR8” is $3 \cdot 8 = 24$, and the dimensionality of “SH” is 512. As can be seen from Figure 10, the combination of the newly introduced proportional shell construction and the newly introduced redundant assignment in combination with the generalized eigenvalue model and the point sampling representation leads to a significantly higher retrieval quality.

We also determined the efficiency of the competitive methods by measuring the time that was necessary to create a precision/recall plot for the PDB data set. We started the measurement after all features were loaded into main memory. As we employed no index structures, the measured time depends only on the dimensionality of the feature space.

The results depicted in Figure 10(c) show that the presented combination of proportional shell construction and redundant assignment with the generalized eigenvalue model does not outperform the other methods only in terms of accuracy, but also in terms of efficiency. The reason for this performance is the comparatively low dimensionality of the resulting feature space. This is also an advantage if a spatial index structure is applied for further speeding-up similarity queries since the performance of spatial index structures usually deteriorates with increasing data dimensionality.

In summary, we can observe that our newly proposed method of proportional shell construction and redundant shell assignment in combination with our generalized eigenvalue model and our new object representation based on point sampling significantly outperforms existing rotation invariant approaches in terms of accuracy and efficiency.

8 Conclusions

In this paper, we presented different similarity models featuring many important invariances with a focus on rotation invariance. In particular, we discussed the limitation of voxel representations (a well-known standard technique to represent 3D objects) in the context of rotation invariance and proposed a solution to this limitation based on point sampling. Furthermore, we introduced a new similarity model that is based on two key ingredients: (1) a new partitioning of the data objects that extends spherical partitionings using the ideas of redundancy and proportionality; (2) a generalization of the existing eigenvalue model and an adoption of this generalization to the newly introduced spherical partitionings.

Our broad experimental evaluation shows that the sampled point representation is better suited in the context of rotation invariance than the well-known voxel representation. In addition, we showed that our new similarity model using proportional shells with redundant point assignment as object partitioning method and the adoption of the eigenvalue model as feature extraction tech-

nique clearly outperforms existing rotation invariant models in terms of accuracy as well as efficiency.

References

1. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., Bourne, P.: The Protein Data Bank. *Nucleic Acids Research* **28** (2000) 235–242
2. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton Shape Benchmark. In: *Shape Modeling International*, Genova, Italy. (2004)
3. Ankerst, M., Kastenmüller, G., Kriegel, H.P., Seidl, T.: 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. In: *Proc. 6th International Symposium on Spatial Databases*. (1999) 207–226
4. Kriegel, H.P., Kröger, P., Mashael, Z., Pfeifle, M., Pötke, M., Seidl, T.: Effective Similarity Search on Voxalized CAD Objects. In: *Proc. 8th DASFAA*. (2003) 27–36
5. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3D Models with Shape Distributions. In: *International Conference on Shape Modeling and Applications*. (2001) 154–166
6. Ip, C., Lapadat, D., Sieger, L., Regli, W.: Using Shape Distributions to Compare Solid Models. In: *7th ACM/SIGGRAPH Symposium on Solid Modeling and Applications*. (2002) 273–280
7. Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In: *Symposium on Geometry Processing*. (2003) 167–175
8. Vranić, D.V.: An Improvement of Rotation Invariant 3D-Shape Descriptor Based on Functions on Concentric Spheres. In: *Proc. IEEE International Conference on Image Processing*. (2003) 757–760
9. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L.: Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. In: *Proc. ACM SIGGRAPH 2001*. (2001) 203–212
10. Sanner, M., Olson, A., Spehner, J.C.: Fast and Robust Computation of Molecular Surfaces. In: *Proc. 11th ACM Symposium on Computational Geometry*. (1995) C6–C7
11. Kaufmann, A.: An Algorithm for 3D Scan-Conversion of Polygons. In: *Proc. Eurographics*. (1987) 197–208
12. Glassner, A., ed. In: *Graphics Gems*. Academic Press (1990) 24–28
13. Jolliffe, I.: *Principal Component Analysis*. Springer (1986)
14. Vranić, D.V., Saupe, D.: 3D Model Retrieval. In: *Proc. SCCG2000*, Budmerice, Slovakia. (2000) 89–93
15. Sanner, M.: *Modeling and Applications of Molecular Surfaces*. PhD thesis, University of Haute-Alsace, France (1992)
16. Holm, L., Sander, C.: Mapping the Protein Universe. *Science* **273** (1996) 595–602