Feature-sensitive 3D Shape Matching

Andrey Sharf
Tel Aviv University
asotzio@post.tau.ac.il

Ariel Shamir
Interdisciplinary Center Hertzliya
arik@idc.ac.il

Abstract

Three dimensional shape matching plays an important role in many of today's applications. Nevertheless, shape matching is a difficult problem since there is no unique measure that defines shape similarity and since computing shape distance using various measures is an elaborate task. In this paper we present a new framework for matching shapes, represented by union of spheres hierarchies. Our approach is "feature-sensitive" since it extends the usual geometry-based matching by adding sensitivity to shape topology, shape features (sharp angles, chemical attributes) and their relative positioning on the shape. Our method can be used in conjecture with other geometric matching methods as a pre or post-processing filtering stage, or it can be used as a stand-alone feature-sensitive matching.

1 Introduction

The fundamental objective of shape matching is to measure the similarity between two given shapes using some distance measure. Thus, similarity measures are at the core of every shape matching algorithm. Recent work on shape matching describe some effective geometric similarity measures for shape matching. For a survey see [12, 18]. Shape signatures based on geometric measures [8, 9] or topological measures [2, 10, 15] are often used for effective discrimination between shapes. In [3] solid CAD models are matched based on labels and text describing their machining features. In the area of molecular biology, early works perform comparison of structures based on RMS [11] and geometric hashing [7]. In [13, 16] local features such as torsion angles and curvature provide a "symbolic" description of the molecule for similarity computation. Nevertheless, there are cases when more elaborate measures are needed such as the location and relative positioning of various shape features. In molecular biology or computer aided design, shapes often have high genus and contain important features of various types. These can include holes, tunnels, voids, rings and helixes. Their number as well as relative positioning are an important factor in the resemblance of two shapes. Moreover, while geometric constrains are often defined by the user to guide the matching of two shapes, topological and volumetric constraints are more difficult to define manually.

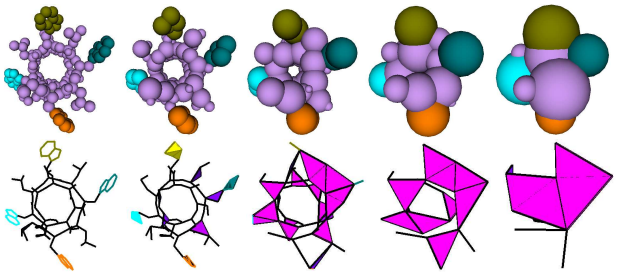
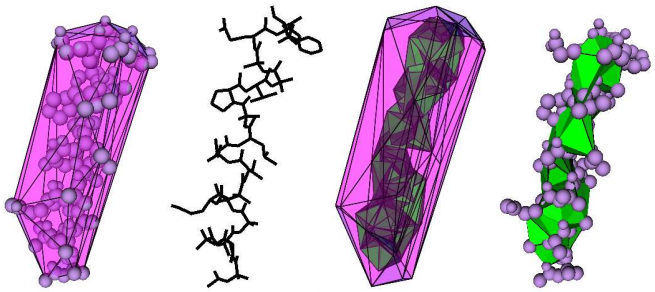


Figure 1. Feature-based hierarchy of molecule (top) and the corresponding zero alpha-shape skeleton (bottom).

In this work we propose a feature-sensitive matching algorithm. Our aim is to define a good matching criteria based on shape features which will combine topology, geometry, feature characteristics and positioning. To establish such capabilities we represent objects using a composition of volumetric elements and their skeleton. We use the union of spheres (UoS) representation due to its simplicity and since its topology is explicitly defined by a simple skeletal structure called *zero alpha-shape* [5]. The most natural application for UoS models is molecular representation composed from union of atoms represented by spheres. Nevertheless, other popular domains such as CAD and model querying benefit from our framework by approximating their volume using UoS representation (e.g. following [17]).

We use a multi-resolution hierarchical representation of the UoS, based on the zero alpha-shape skeleton (for details see [14]). However, in this work we construct the shape's hierarchy in a constrained manner. The construction preserves the shape's features throughout the hierarchy by constrained coarsening of each feature within itself and propagating features information bottom up. The resulting construction is a feature-sensitive segmentation of the shape. Based on this representation, our algorithm creates an "ef-



(a) (b) (c) (d)

Figure 2. Spiral tunnel extraction.

ficient" adaptive cut that serves as a "sufficient" coarse approximation of the shape. This cut adapts its feature approximations to the user demands. Such cuts generate a fast and valid initial alignment, and a coarse match between the shapes. Consequently, matching and alignment are both refined progressively as we advance to finer levels of detail. Finally, the distance between the two shapes is calculated using some metric between the mapped spheres of the two UoS shape representations.

The rest of the paper is composed as follows. Section 2 describes the UoS model and the algorithms used for automatic feature extraction. Section 3 describes the feature-sensitive shape matching algorithm and Section 4 presents our results. We conclude in Section 5.

2 Shape Features Representation

To define a feature-sensitive matching procedure some shape analysis is needed in the preprocessing stage. Shape features can be indicated manually, however, many times this can become a complicated task. In order to automate this analysis and enable feature definition, an elaborate representation of the shape is needed. For this purpose we use the union of spheres (UoS) representation. The UoS complex has been studied extensively in [4, 5]. It has been shown that the topology of the object itself is encapsulated in the topology of its dual skeletal structure called the *zero alpha-shape*. In this section we present some feature extraction procedures defined on the UoS and its skeleton. Since features are extracted directly from the UoS and its dual skeleton, mapping of spheres to features is done simply by adding a pointer from a sphere to its feature.

Topological features: We extract spheres composing topological features such as tunnels and voids. This is done by adapting the algorithm presented by [6]. Following the example in Figure 2, we first subtract the zero alpha-shape complex (b) of the UoS from the regular triangulation convex hull of the spheres (a). The resulting simplicial complex (c) is the complement of the zero alpha-shape to the regular triangulation convex hull. In this complementary complex a tunnel is a complex that shares two or more simplices with the convex hull boundary while a void, has no simplex on the convex hull boundary.

Sharp features: To automatically extract sharp features, we exploit the correspondence between adjacent simplices on the boundary of the zero alpha-shape and their dual spheres. When adjacent spheres have small variation in their radii we find a high correlation between sharp angles on the spheres' surface, and sharp angles on their dual zero alpha-shape simplices. Hence, we define a sharp feature as a sharp angle in the skeleton.

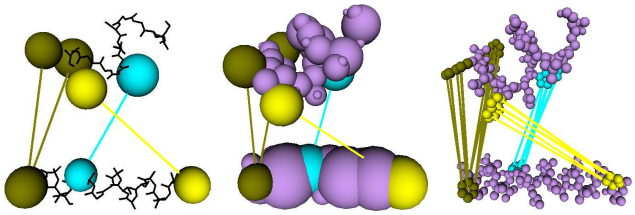
Descriptive high-level features: Features such as alpha helixes in molecules, textures and materials in graphics models, and user specified substructures in CAD, can be specified easily using the UoS and its zero alpha-shape. In our examples we extract carbon rings sub-structures by walking over the zero alpha-shape complex and detecting loops of some length which involve carbon atoms. We extract volumetric information regarding narrow vs. wide regions in the shape by walking over the simplicial complex and examining the neighborhood of each vertex.

3 Feature-sensitive Progressive Match

We first give an overview of our algorithm. In a preprocessing step, we compute for each object its hierarchical representation (Figure 1). The matching starts by creating two feature-sensitive adaptive cuts of each shape (Figure 3 (b)), based on given feature weights. For each cut we create a symbolic approximation which is a composition of the the shape features in the adaptive cut and the non-feature regions (Figure 3 (a)). Next, we perform an exhaustive search to find the best alignment and match between the two symbolic levels (Figure 3 (a)). We refine the match and alignment progressively by descending down the two hierarchies (Figure 3 (c)). We refer the reader to [14] for details regarding the alignment and progressive refinement steps. In the next sections we describe the feature sensitive nature of our matching algorithm.

3.1 Feature-sensitive Multi-Resolution

Our progressive matching algorithm relies on the hierarchical representation of the objects and we follow [14] in the definition of a hierarchical UoS representation. However, using general purpose hierarchy with no feature information may result in inadequate or poor matching. Therefore, instead of global coarsening based on geometry alone, we modify the algorithm to support a local feature-based coarsening scheme. Each feature is coarsened independently by performing coarsening operations only between spheres belonging to the same feature (or to no feature). We use a fea-



(a) (b) (c)

Figure 3. Matching algorithm illustrated on features of 2 molecules (colored): (a) Symbolic level. (b) Adaptive cut. (c) Refinement.

ture separation criteria: when a feature is represented by a small enough set of spheres (most often one sphere) during coarsening, it is defined as *fully separated* from all other shape parts. When a feature is fully separated it then becomes part of the general coarsening frame. In this manner, our clustering scheme maintains the natural segmentation of the shape to features and preserves the inner shape structure of each feature. The resulting hierarchy is feature-sensitive while still preserving the overall shape (see Figure 1).

3.2 Adaptive Cuts and Feature Separation

The first step of our matching algorithm is the creation of an initial adaptive cut of the multi-resolution graph hierarchy forming the *base approximation*. Since this approximation is used as the starting point of the matching algorithm, it has an extensive effect on the final matching result. Therefore, the adaptive cut should provide an "efficient" and "sufficient" approximation of the shape. It should have the following properties: small cardinality (number of spheres), provide a good approximation of the shape and provide appropriate *feature separation*.

In a given matching query the user specifies the measures that define the similarity between objects and their importance in terms of weights. Thus, each feature or type of feature is given a weight which defines its importance in the total distance measure. Since the approximation of shapes should correspond to the distance measure of the matching, we derive the initial adaptive cut based on the features weights. We define the *feature approximation distance* as the distance between the bounding sphere of the feature in the finest level of detail and the bounding sphere of the feature in the current level. In order to reach the suitable feature-sensitive adaptive cut according to the given weights, we refine each feature until its feature approximation distance is below a given threshold (see Figure 3(b)). Denote $s_i^j(k)$ as the i^{th} sphere at level j belonging to feature k, $B()$ as the bounding sphere and $w(k) \leq 1$ as the weight of feature k, then:

$$B(\cup s_i^j(k))/B(\cup s_i^0(k)) \leq 1/w(k)$$

Since the multi-resolution construction is feature-sensitive, usually little refinement is needed to reach the appropriate feature approximation. Hence, the resulting adaptive cut is both "efficient" in the number of spheres and "sufficient" in the approximation ratio of the weighted features.

We achieve further simplification of the shape by converting the adaptive cut into a symbolic representation (Figure 3(a)). We replace all spheres $s_i^j(k)$ representing a feature in the adaptive cut with a representative sphere enclosing the feature $B(\cup s_i^j(k))$. Non feature regions of the shape are added to this symbolic representation at the coarsest resolution. We start the matching algorithm using the symbolic representation and then refine it in a progressive manner.

3.3 Shape Distance Measure

The matching procedure defines a mapping between spheres of the two UoS shape representations. The distance between the two shapes is calculated as the sum of distances between the mapped spheres. The distance $D(s_i, s_j)$ between each pair of spheres (s_i, s_j) is a weighted sum of three factors: the power-distance between the two spheres defined as: $\pi(p_i, q_j) = ||p_i - p_j||^2 - w_i^2 - w_j^2$, the difference in volumes ($|V_i - V_j|$), and the feature distance:

$$D(s_i, s_j) = W_1\pi(p_i, q_j) + W_2|V_i - V_j| + W_3 D_t(s_i, s_j)$$

For feature distance computation, we examine each sphere and the feature it belongs to. If the two spheres do not belong to the same feature, a penalty is added to their distance. Otherwise the difference between the feature sizes (bounding box) is defined as their feature distance. The matching weights (W_1, W_2, W_3), define the total distance between the spheres (s_i, s_j), thus defining the importance of each component for similarity.

4 Experiments and Results

In the first set of experiments we use our matching algorithm as a post-processing filter to molecular search engines such as CATH and SCOP [1]. In Table 1, we retrieved a family of molecules and applied our matching algorithm on the results. The table is symmetric and the similarity results are shown by the colored bars in the lower left part. The green part denotes geometric similarity and the red part denotes features similarity (number of C-rings and positioning). The first, second and last molecules have good geometrical similarity. However, based on features(parentheses denote #rings), the first, third and last are more similar. Hence, even when similar geometry is involved, our feature-sensitive metric differentiates between

	1	2	3	4
1(4)	-0-			
2(3)		-0-		
3(4)			-0-	
4(4)				-0-

Table 1. Molecular family matching.

	1	2	3	4
1(9,2)	-0-			
2(5,0)		-0-		
3(3,1)			-0-	
4(4,1)				-0-

Table 2. Dissimilar molecules matching.

the molecules. Nevertheless, our algorithm separates between general molecules based on specific features and their specific location. In Table 2, all molecules differ both in features (parentheses denote #rings,#spirals) and geometry. The last two molecules are the best match, since they both have one spiral and similar number and location of rings.

In the second set of experiments we used our matching algorithm for 3D shape retrieval. A 3D model along with matching criteria are given as the query, and the best matched models from the data-base are returned. Table 3 summarizes the results of several queries to our data base using different metrics. For each query shape, we perform three queries with different similarity metrics. We use only feature distance in the top row (red), only geometric distance in the middle row (green), and a combination of both at the bottom row. In (a) the main features are topological, hence the top row returns shapes with similar topology (five holes) even if the geometry is different. In (b), the top row detects shapes with similar sharp features (the bridge).

5 Conclusions and Future Work

We have presented a method for matching and calculating the distance between feature-enhanced 3D objects. We use the UoS representation along with its zero alpha-shape skeleton. We tested our algorithm on 3D molecular data as well as on feature-enhanced CAD models, and found it to

Query	Metric	1st result	2nd result	3rd result
(a)				
(b)				

Table 3. Shape matching

be effective in differentiating objects and finding similarity in a feature-sensitive manner.

Possible future directions involve sub-parts matching and articulated object matching where an object has several degrees of freedom defining a valid configuration space.

References

[1] P. D. Bank. http://www.rcsb.org/pdb.

[2] S. Biasotti, S. Marini, M. Mortara, G. Patan, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In *DGCI*, pages 194–203, 2003.

[3] V. Cicirello and W. Regli. Machining feature-based comparisons of mechanical parts. In *SMI*, pages 176–185, 2001.

[4] H. Edelsbrunner. Weighted alpha shapes. Technical Report 1760, University of Illinois at Urbana-Champaign, 1992.

[5] H. Edelsbrunner. The union of balls and its dual shape. In *Proceedings of 9th Annual ACM Symposium on Computational Geometry*, pages 218–231, 1993.

[6] H. Edelsbrunner, M. Facello, and J. Liang. On the definition and the construction of pockets in macromolecules. *Discrete Applied Mathematics*, 88:83–102, 1998.

[7] D. Fischer, R.Nussinov, and H.Wolfson. 3D substructure matching in protein molecules. In *Combinatorial Pattern Matching*, pages 136–150. Springer-Verlag, 1992.

[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. A search engine for 3D models. *ACM Transactions on Graphics (TOG)*, 22:83–105, 2003.

[9] T. Funkhouser, R. Osada, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications ACM SIGGRAPH*, pages 154–166. IEEE, 2001.

[10] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Computer Graphics, Proceedings SIGGRAPH'01*, pages 203–212, 2001.

[11] A. Lesk. A toolkit for computational molecular biology ii.on the optimal superposition of two sets of coordinates, 1986.

[12] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[13] S. Rackovsky and H. Scheraga. Differential geometry and polymer conformations 2.development of a conformational distance function. *Macromolecules*, 13:1440–1453, 1980.

[14] A. Shamir, A. Sharf, and D. C. Or. Enhanced hierarchical shape matching for shape transformation. *Journal of Shape Modeling*, 9:203–222, 2003.

[15] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *SMI*, pages 130–139, 2003.

[16] W. Taylor. Patterns in protein sequence and structure. *Springer Series in Biophysics*, 7, 1992.

[17] G. Treece, R. Prager, and A. Gee. Volume-based three-dimensional metamorphosis using sphere-guided region correspondence. *The Visual Computer*, 17:397–414, 2001.

[18] R. Veltkamp. Shape matching: similarity measures and algorithms. Technical Report UU-CS-2001-03, Utrecht University, the Netherlands, 2001.