Volume Graphics (2006)
T. Möller, R. Machiraju, T. Ertl, M. Chen (Editors)

Complementary Shape Comparison with Additional Properties

Xiaoyu Zhang†

Department of Computer Science, California State University San Marcos

Abstract

We present an algorithm for comparing 3D shapes by considering their pockets in the complementary space. The pockets of a closed compact surface can be represented by a 3D volumetric function. Multi-resolution dual contour trees are constructed from the pocket functions to efficiently match them in an affine-invariant way. DCTs are simplified data structures computed from contour trees (CT) of 3D functions. The DCTs capture the important features of the volumetric functions and are not sensitive to noises. Each node of a DCT corresponds to an interval volume and is tagged with geometrical, topological, and functional attributes. Similarities among shapes are compared by matching nodes from multi-resolution DCTs and calculating the score based on the attributes of the matched nodes. This method is particularly useful for comparing complicated molecular shapes, for which other properties such as electrostatic potentials can be added as additional attributes to improve performance.

Categories and Subject Descriptors (according to ACM CCS): I.5.3 [Computing Methodologies]: Pattern Recognition˚USimilarity Measures; J.3 [Computer Applications]: Life And Medical Sciences;

Keywords: shape matching, pocket function, dual contour tree, affine-invariant, multi-resolution

1. Introduction

3D shape matching is an important problem in the graphics and other areas. For example, a protein structure may be represented by the shape of its molecular surface. Effective comparison and classification of these highly complicated 3D molecular shapes are very important for the understanding of their structural and functional properties. The structures of proteins and other large molecules are being determined at dramatically increasing rate through structural genomic and other efforts [BWF*00]. Because of the high complexity of molecular shapes, their comparison poses new challenges.

In this paper we introduce a new algorithm for comparing 3D shapes by considering their pockets in the complementary space. Instead of establishing correspondence between the original shapes, which can be very difficult for complex shapes like molecular surfaces, we try to match the surface pockets and compute the similarities. Pockets are the main features on surfaces and also often referred to as "depressions" or "holes". Pockets are of particular importance to proteins and other macromolecules because biochemical reactions often take place in the protected yet accessible regions of pockets. Therefore shape matching using pockets may give clues to understand the protein structures and functions.

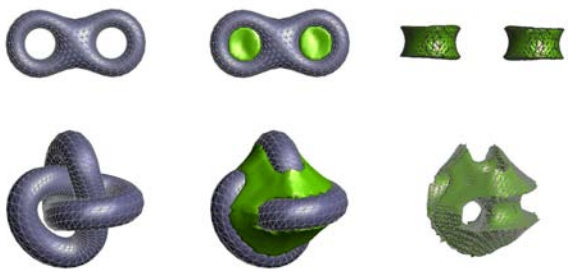


Figure 1: *Example surfaces and their pocket envelopes*

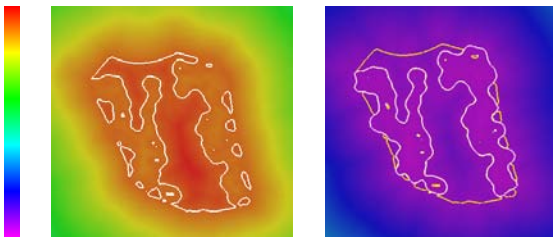
† xiaoyu@csusm.edu

The key idea of our approach is to represent the surface pockets as a volumetric shape function and construct a rotation-invariant data structure called dual contour trees (DCT) to efficiently compare the pocket functions.

We apply a level set marching method to compute the pocket functions [ZB06], while other methods may also be justified. A propagation front marches out from original surface S at a constant speed to a final shell surface T. During the marching the topology of the front would change and the propagation is irreversible. T is chosen to be a front far enough away such that it has the simple topology as a sphere and its topology would not be changed by further propagation. The surface pockets can be extracted through a backward propagation from T towards the original surface S. The backward marching front F is not allowed to penetrate S and stops when it reaches S. Pockets are defined as the regions bounded between final backward marching front F and the original surface S. Rather than representing the pockets as a set of surface envelopes, one can compute a volumetric pocket function. Assume $d_S(\mathbf{x})$ is the signed distance function (SDF) of the original surface S and $d_T(\mathbf{x})$ is that of the shell surface T described above, where $d_S(\mathbf{x})$ is positive if $\mathbf{x}$ is outside S but $d_T(\mathbf{x})$ is positive for $\mathbf{x}$ inside T. Then the pocket function $f_P(\mathbf{x})$ is defined as

$$f_P(\mathbf{x}) = \min(d_S(\mathbf{x}), d_T(\mathbf{x}) - t), \tag{1}$$

where t is the marching distance from S to T. The pocket function $f_P(\mathbf{x}) > 0$ only for points outside S and not reachable by backward propagation from T, i.e. points in pockets. The bounding envelopes of pockets are then computed as the level set $\phi_P(x) = \varepsilon$, where $\varepsilon > 0$ is a small constant for numerical stability. Figure 2 (a) shows a 2D color-mapped pocket function and corresponding pockets and in Figure 2 (b) we superimpose the pockets (yellow curves) onto the original surface (white curves) [ZB06], which come from a slice of the molecular surface of the "Bacteriochlorophyll Containing Protein". The result matches very well with our intuition about pockets and holes.



(a) (b)

Figure 2: *A 2D example of pocket function and extracted pockets: (a) the pocket function is displayed as a color-mapped image; (b) the color-map shows the SDF of the original surface (white curves).*

A standard method for comparing volumetric functions is to compute their inner product. The L^2 norm is commonly used to compute the inner product of two functions f and g. Similarity metrics are then defined based on the definition of norms. The most popular metrics are Hodgkin index $S_H(f,g) = \frac{\langle f,g\rangle}{\|f\|^2+\|g\|^2}$ and Carbo index $S_C(f,g) = \frac{\langle f,g\rangle}{\|f\|\cdot\|g\|}$.

The inner product method is clearly not rotational-invariant. 3D volumetric functions f and g can be aligned relative to each other with six-dimensional freedom of rotations and translations. The inner product $\langle f,g\rangle$ should be computed for the best alignment. However, it is not trivial to geometrically align large molecules [BW03]. The alignments are often done manually, otherwise one has to search the six-dimension rotational and translational space to find the best alignment. Such search is expensive and usually does not guarantee to find the best alignment.

In this paper, we construct an affine-invariant structure, called dual contour tree (DCT), to represent the pocket function. DCTs are simplified data structures computed from contour trees (CT) [KOB*97] of 3D functions. The DCTs capture the important features of the volumetric functions and are not sensitive to noises. Each node of a DCT corresponds to an interval volume and is tagged with geometrical, topological, and functional attributes. one can efficiently compare 3D shapes by matching multi-resolution hierarchies of DCTs. The major steps of the complementary shape comparison algorithm are:

1. Compute a pocket function to represent the main features of 3D shapes [ZB06].
2. Compute a contour tree (CT) for the volumetric pocket function [CSA03].
3. Construct the finest-level dual contour tree (DCT) from the CT in the previous step.
4. Compute the geometrical, topological, and functional attributes for the nodes in the DCT.
5. Build a multi-resolution hierarchy of the attributed dual contour tree (MACT) by merging adjacent functional intervals.
6. Match two MACTs and compute their similarity score.

The remainder of the paper is organized as follows. Section 2 gives an overview of existing shape matching algorithms and introduces some relevant background. Section 3 describes the methods of constructing the DCT, computing its nodal attributes, and building the multi-resolution hierarchy. Section 4 presents the steps to match two MACTs and compute the similarity scores. Section 5 then discusses the implementation and some empirical results of our method.We conclude in section 6.

2. Related Work and Background

Shape matching is traditionally done by finding correspondences between the compared shapes. As discussed earlier, spatial alignment can be very expensive. A common approach is to segment the shapes into basic parts and match

those parts and their spatial relationships [DGG03, BCGJ98, BMP02, BM92, KPNK03]. This scheme may be applied to match complementary space features like pockets and use the spatial relationships of matched pockets to align the original shapes.

Another approach is to compute some fixed length vectors as the *shape descriptors* or *signatures* for 3D shapes and compute the similarity metrics of shapes using the distance between the shape descriptors. Various descriptors have been used for shape matching, such as curvature distributions [ACH*91, SKG97], shape distributions and histograms [OFCD01, ATRB95, KFR04], and coefficients of functional expansions [ASBH90, KFR03] etc. The descriptors can be pre-computed and are almost always rotation-invariant. The shape descriptors have been applied to retrieve similar shapes from a 3D shape database [FKMS05]. As far as we know, no descriptor has been developed for complementary space features like pockets, and the shaper descriptor have not yet been applied to complex 3D shapes like molecular surfaces.

If we represent the shape with a continuous volumetric function f, e.g. the pocket function, defined on a 3D domain $\mathscr{M}$, $f : \mathscr{M} \rightarrow \mathbf{R}$. The original surface S is often a special level set of f. The *functional range* of f is the interval between the minimum and maximum values of f: $[f_{min}, f_{max}]$. Signatures can be computed for volumetric shape functions as well. One example is the contour spectrum [BPS97], which is a set of histograms for the level sets of the function, such as their areas and the volumes enclosed by the level sets. Although the contour spectrum is certainly rotation-invariant, it has limited success in comparing volumetric shape functions.

While a isovalue w scans monotonically through the functional range $[f_{min}, f_{max}]$ of f, the evolution of the homology classes of the level set $L(w)$ is studied in Morse theory [Mil63]. The *critical points* of f are the positions where the gradient vanishes, i.e. $\nabla f = 0$, and they are assumed to be non-degenerate in Morse theory. The topology of $L(w)$ changes only at the critical points of f, and the corresponding functional values are called critical values.

Another approach of comparing the volumetric shape functions is to use some affine-invariant topological structures, such as Morse complexes [EHNP03] and contour trees (CT) [KOB*97]. Similar surface topological structures such as Multi-resolution Reeb Graphs (MRG) are defined and used for shape matching [HSKK01]. Both Morse complexes and contour trees are related to the critical points of the volumetric function f. A complex shape function usually has a large number of critical points, many of which are caused by small noises. Because of the complexity of the shape function, it is difficult to directly compute the Morse complex and even harder to establish correspondences between them.

A contour tree (CT) [KOB*97] captures the topological changes of the level sets for the entire functional range $[f_{min}, f_{max}]$ of f. Each node of the CT corresponds to a critical point and each arc connecting two critical points corresponds to a set of continuous contours of the same topology. A cut on a CT arc (v_1, v_2) at the isovalue w $(f(v_1) \leq w \leq f(v_2))$ corresponds to a connected component (contour) of the level set $L(w)$. So the number of connected components for the level set $L(w)$ is equal to the number of cuts to the CT at the value w .

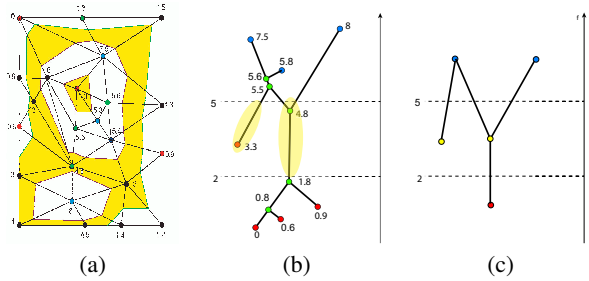
Although the CT is affine-invariant, it is very difficult to determine correspondences between two CTs due to the vast number of nodes corresponding to the critical points of f. An example of CT is shown in Figure 4 (f).

Carr et al. [CSA03] present an efficient two-pass scheme to compute a CT in $O(m+n\log n)$ time, where m is the number of simplices and n is the number of vertices in the mesh $\mathscr{M}$. The CT can be enhanced by tagging arcs with topological information such as the Betti numbers of the corresponding contour classes [PCM02]. In the next section, we describe the algorithm of constructing the DCT and nodal attributes to compare volumetric shape functions , based on enhanced CTs computed with those CT algorithms.

3. DCT Algorithm

In order to compare volumetric shape functions, we introduce dual contour trees (DCTs) as a simplified structure constructed from CTs.

3.1. Dual Contour Tree



(a) (b) (c)

Figure 3: *A simple 2D example function, its contour tree, and dual contour tree. (a) The function values are labeled on vertices and values in a simplex are linear interpolations of vertex values. (b) The CT of the function. Critical points are colored differently: minima in red, saddle points in green, and maxima in blue. Two interval volumes within the functional interval* $[2,5]$ *are highlighted in yellow. (c) The three-range DCT constructed by the cuts at* $w_1 = 2$ *and* $w_2 = 5$.

We illustrate the idea of constructing a DCT using a simple 2D example in Figure 3. Figure3 (a) shows a function f defined on a 2D mesh $\mathscr{M}$ whose vertices are labeled with function values and Figure 3 (b) shows the contour tree of f. The contour tree is cut at two isovalues $w_1 = 2$ and $w_2 = 5$,

and the two cuts correspond to two level sets shown in Figure 3 (a). The level set $L(w_2 = 5)$ has three connected components (colored in magenta) and $L(w_1 = 2)$ has a single contour (colored in green). There are two interval volumes between the cut w_1 and w_2. An *interval volume* is a connected region of the domain $\mathscr{M}$, in which the function value of f lies between two specific isovalues. The interval volumes are highlighted in Figure 3, and clearly they are bounded by contours at isovalues w_1 and w_2. Each interval volume corresponds to a set of connected CT arc segments between the cuts at w_1 and w_2, as highlighted in Figure 3 (b).

Each interval volume becomes a node in the dual contour tree (DCT) and two DCT nodes are connected by an edge if the corresponding interval volumes are adjacent (sharing the same contour at their boundaries). The DCT constructed from the CT and the two cuts in Figure 3 (b) are illustrated in 3 (c). The steps to construct a DCT from a given CT are described below:

1. Divide the functional range $[f_{min}, f_{max}]$ of the function f into N intervals, which cut the CT arcs into segments in N ranges. Without loss of generality, we can assume that CT arcs are not cut at critical values.
2. For each range i $(1 \leq i \leq N)$, we use a Union-Find data structure to assign all cut CT arc segments in the range into disjoined sets. Each set of connected arc segments becomes a DCT node at level i.
3. If a DCT node n at level i and a DCT node m at level $i-1$ contain segments from a shared CT arc, their corresponding interval volumes are adjacent. An edge is insert between n and m. Clearly edges only exist between DCT nodes in adjacent ranges.

The DCT provides a simpler representation of the original function than the CT by eliminating small undulations in a functional range while preserving potentially-significant features like high mounds and deep pockets. The complexity of the DCT can be controlled by selecting the number of ranges N. The structure of the DCT converges to the CT as the number of ranges increases and the size of ranges decreases.

We often want to focus on comparing particular regions of the volumetric shape functions. For instance, we are only interested in the pocket regions of the pocket function $f_P(\mathbf{x})$, i.e. the functional range $f_P(\mathbf{x}) > 0$. This can be easily achieved by restricting the total functional range of the DCT to a subrange $[f_1, f_2] \subset [f_{min}, f_{max}]$. In our algorithm of complementary shape comparison with pocket functions, we choose the DCT functional range as $[\varepsilon, \max(f_P(\mathbf{x}))]$, where $\varepsilon \geq 0$ is a constant. The steps to construct a DCT for a subrange $[f_1, f_2]$ is almost the same. We simply ignore CT arc segments outside the subrange $[f_1, f_2]$. However, notice that the DCT is no longer necessarily a single tree but a graph of multiple trees because the 3D volume satisfying $f(\mathbf{x}) \in [f_1, f_2]$ are not always connected.

3.2. Node Attributes

In order to quantitatively measure the similarities of DCTs, we define attributes based on the volumetric shape function and additional properties, e.g. electrostatic potential of molecules. A DCT node m corresponds to a connected interval volume $V_m \subset \mathscr{M}$. We first look at some geometrical and topological attributes related to the shape of V_m:

- $vol(m)$: The volume of the interval volume V_m. If comparison is desired for shapes of different scales, normalized volume may be used instead, by assuming the total volume of the shape function domain is one. Matched DCT nodes should have similar volumes.
- $area(m)$: The area of the surfaces bounding V_m. It may be also normalized for cross-scale comparisons.
- $I(m) = (I_1, I_2, I_3)$: The principal values of the moment of inertia for V_m. The moment of inertia tensor is defined as

$$I_{ij} = \int_{V_m} (x_i - x_i^c)(x_j - x_j^c) d^3\vec{x} = \int_{V_m} x_i x_j d^3\vec{x} - V(m) x_i^c x_j^c,$$

  where $\vec{x}_c = (x_1^c, x_2^c, x_3^c)$ is the center of mass for V_m. The principal axes of I_{ij} are calculated and the diagonal values of I along the principal axes are recorded as a triplet $I(m) = (I_1, I_2, I_3)$, where $I_1 \geq I_2 \geq I_3$. $I(m)$ provides information about the overall shape of V_m.
- $B(m) = \{B_l(m), B_u(m)\}$: The Betti number attribute representing the topologies of the lower and upper bounding surfaces for V_m. $B_l(m)$ and $B_u(m)$ are triplets containing the three possibly non-zero Betti numbers $(\beta_0, \beta_1, \beta_2)$ for 3D surfaces. If a bounding surface consists of multiple contours, its Betti triplet is the sum of those of individual contours.

Other geometrical and topological attributes may also be added to the DCT node. Actually all the shape descriptors [ACH*91, SKG97, OFCD01, ATRB95, KFR04, ASBH90, KFR03] can be potentially treated as node attributes. Compared to one shape descriptor for a entire shape, the added level of DCT structure of the volumetric shape function would make the comparison more accurate.

Another advantage of our method is to incorporate additional properties into comparison. Particularly for comparing molecular shapes, additional properties such as electrostatic potential, electron density, and solvent accessibility are very important for finding structurally and functionally similar bio-molecules. Not accidentally those properties in pockets are most important because biochemical reactions often take place in the protected yet accessible regions of pockets.

We consider electrostatic potential as an example property, which is another volumetric function defined on the same domain. Here we calculate the multi-pole expansion of the property function distributed over the interval volume V_m. Following descriptors may be added to the DCT node as functional attributes:

- $P(m)$: The integral of the potential p over the interval vol-

ume V_m, $P(m) = \int_{V_m} p d^3\vec{x}$ is the first term of the multipole expansion.

- $\vec{D}(m)$: The dipole moment of the potential p over V_m is a vector $\vec{D}(m) = \int_{V_m} p \cdot (\vec{x} - \vec{x}^c) d\vec{x}$. We use the magnitude of $\vec{D}(m)$ and its angle relative to the main principle axis of V_m as node attributes.
- $Q(m) = (Q_1, Q_2, Q_3)$: The quadropole moment of the potential p over V_m is a tensor defined as

$$\begin{aligned} Q_{ij}(m) &= \int_{V_m} p \cdot (x_i - x_i^c)\left(x_j - x_j^c\right) d^3\vec{x} \\ &= \int_{V_m} p \cdot x_i x_j d^3\vec{x} - x_i^c D_j(m) - x_j^c D_i(m), \end{aligned}$$

where $D_i(m)$ is the ith component of the dipole moment. We again choose the principal values of $Q_{ij}(m)$, $Q_1 \geq Q_2 \geq Q3$, as attributes of a DCT node.

The attributes of the DCT node m can be then summarized into a vector $\tilde{m}$ as following:

$$\tilde{m} = \{vol(m), area(m), I(m), B(m), P(m), D(m), Q(m)\}.$$

3.3. Multi-resolution Hierarchy

In order to facilitate the comparison of attributed DCTs, they can be further organized in a hierarchical multi-resolution form. This Multi-resolution Attributed Contour Tree (MACT) is constructed from a fine DCT by merging its adjacent functional intervals recursively.

Without loss of generality, we assume a finest DCT D has $N = 2^k$ functional intervals. The DCT at the next coarser resolution would have $N/2$ intervals, each of which is merged from two intervals of the finer DCT. A set S of connected DCT nodes in the two combined intervals are merged into a single node n in the coarser DCT. This can be achieved again by using a Union-Find data structure. The node n is called the parent of nodes in the set S, which are the children of n. The merging process can be recursively applied to the coarser DCTs until there is only a single interval spanning the entire functional range under consideration. If we call the finest DCT D_k and the next coarser one D_{k-1} etc, then we get a sequence of increasingly coarser DCTs $(D_k, D_{k-1}, ..., D_0)$. Figure 4 (c) and (d) show the DCTs at two different levels of the hierarchy. The complexity of the DCTs at coarser levels is significantly reduced and the hierarchy makes it much easier to find correspondences between DCT nodes of two shape functions.

As mentioned earlier, a DCT of a restricted subrange $[f_1, f_2]$ may be a graph consisting of multiple trees in the finest level. In the coarsest level DCT D_0, each individual tree is merged in to a single node. Numerous nodes in the coarsest DCT D_0 may complicate the matching process. However, most of those nodes are very small in size and can often be pruned as noise if their volumes are under certain threshold. Pruning a lower resolution DCT node shall remove all its child nodes from finer DCTs as well. In Figure 4 (d) for the pocket function of the unbinding "Staphylococcal Nuclease" (PDB code 1KAA), only four nodes are left in D_0 after pruning, where size threshold is set as 1% of the total pocket volume.

The attributes of a node in the coarser level of the hierarchy is evaluated recursively from those of its children. Most attributes discussed in section 3.2 are additive. For example, if a node m has children $m_1, ..., m_l$ in the finer level DCT, the volume attribute $vol(m)$ is just the sum of volumes of its children:

$$V(m) = \sum_{i=1}^{l} V(m_i).$$

The Betti numbers attributes of the finest DCT can be computed from an augmented CT [PCM02]. Consider a merged DCT node m from its children in two intervals of the finer DCT. The lower bounding surface of m is the union of the lower boundaries of its children in the lower interval. Therefore $B_l(m)$ is the sum of $B_l(m)$'s of those children, and $B_u(m)$ can be computed similarly.

As for the functional attributes from the additional property functions, the values for the node m can be calculated from those of its children as well. For example, $P(m) = \sum_{i=1}^{l} P(m_i)$. Similar but more involved equations exist for $D(m)$ and $Q(m)$. Next we describe the matching algorithm based on the multi-resolution hierarchy of DCTs.

4. Matching Algorithm

4.1. Similarity Metrics

First we look at the similarity metric between two DCT nodes if the correspondence is established. The similarity $\langle m, n \rangle$ between two nodes m and n is defined based on their attribute vectors $\tilde{m}$ and $\tilde{n}$ as a weighted average of the similarity metrics of individual attributes:

$$\langle m, n \rangle = \sum_i w_i \cdot \langle a_i(m) a_i(n) \rangle, \tag{2}$$

where $\langle a_i(m) a_i(n) \rangle)$ is the similarity metric of the ith attribute in the vector $\tilde{m}$ and $\tilde{n}$ and the weights satisfying $0 \leq w_i \leq 1$ and $\sum w_i = 1$ control the relative importance of different attributes in the comparison. The similarity metric of individual attributes is defined as follows such that the exactly same attributes achieve the maximum value = 1.

$$\begin{aligned} \langle vol(m), vol(n) \rangle &= 1 - \frac{|vol(m) - vol(n)|}{\max(vol(m), vol(n))} \\ \langle B(m), B(n) \rangle &= \frac{1}{3} \sum_{i=0}^{2} \frac{\min(\beta_i(m), \beta_i(n))}{\max(\beta_i(m), \beta_i(n))} \\ \langle I(m), I(n) \rangle &= 1 - \frac{\max_{j=1,2,3}(|I_j(m) - I_j(n)|)}{\max(I_1(m), I_1(n))} \end{aligned}$$

$$\langle P(m),P(n)\rangle = 1-\frac{|P(m)-P(n)|}{\max(|P(m)|,|P(n)|)}$$
$$\langle D(m),D(n)\rangle = 1-\frac{|D(m)|-|D(n)|}{\max(|D(m)|,|D(n)|)}$$
$$\langle Q(m),Q(n)\rangle = 1-\frac{\max_{j=1,2,3}\left|Q_j(m)-Q_j(n)\right|}{\max\left(|Q_1(m)|,|Q_1(n)|\right)}$$

The similarity score of functional attributes may be negative, e.g. $\langle P(m),P(n)\rangle < 0$ if $P(m)$ and $P(n)$ have opposite signs and so is $\langle Q(m),Q(n)\rangle$. This feature gives penalty to molecular pockets with similar shape but different functional properties. The maximum similarity score between two nodes is clearly 1, achieved when they have exactly the same attribute vectors.

$$\langle m,n\rangle \le \langle m,m\rangle = \langle n,n\rangle = 1$$

If the correspondences between nodes in two DCTs D and D' have been established, the similarity score between D and D' is computed from the scores of matched node pairs:

$$\langle D,D'\rangle = \frac{\sum_i (vol(m_i)+vol(n_i))\cdot\langle m_i,n_i\rangle}{\sum_i vol(m_i)+vol(n_i)}, \qquad (3)$$

where the similarity score $\langle m_i,n_i\rangle$ of a matched node pair $m_i \in D$ and $n_i \in D'$ is weighted by the average of their normalized volumes. Therefore bigger weights are given to larger interval volumes and the similarity score $\langle D,D'\rangle \le 1$.

For a multi-resolution hierarchy of dual contour trees (MACT) $M = \{D_k, D_{k-1}, \ldots, D_0\}$ with total $k+1$ levels, D_i at level i is matched to the DCT D'_i at the same level of the other MACT M'. The correspondences are found in a order from coarser DCTs to finer DCTs as described in section 4.2. The similarity score $\langle M,M'\rangle$ between MACTs M and M' is evaluated as the average of the scores of DCTs from level 0 to k:

$$\langle M,M'\rangle = \frac{1}{k+1}\sum_{i=0}^{k}\langle D_i,D'_i\rangle. \qquad (4)$$

The similarity score $\langle M,M'\rangle$, clearly satisfying $\langle M,M'\rangle \le 1$, is the measure of the similarity between two volumetric shape functions, particularly the pocket functions in this paper.

4.2. Matching Algorithm

The volumetric shape functions are compared by matching their MACTs. The matching process is performed from the coarsest to the finest level of the hierarchies, where we assume that the MACTs M and M' have the same number of levels. The matching algorithm attempts to find the maximal set of matched MACT node pairs between two MACTs M and M'. The DCT nodes $m \in M$ and $n \in M'$ of a matched pair must satisfy following conditions:

- The nodes m and n do not belong to any other pairs.
- m and n must belong to the DCTs at the same level of the hierarchies, i.e. $m \in D_i \subset H$ and $n \in D'_i \subset H'$, where D_i and D'_i have the same number of functional intervals.
- m and n must belong to the same functional interval of D_i and D'_i.
- The parent $p(m)$ of m and $p(n)$ of n are also a matched pair $(p(m),p(n))$ in the coarser level DCTs. The only exception is D_0, whose nodes have no parents.

We use a greedy algorithm to find the pairs of matched nodes, starting from level 0 of the hierarchies. The steps to match the DCT $D_i \subset M$ and $D'_i \subset M'$ at level i ($i = 0,\ldots,k$) are as follows,

1. Add all nodes of the DCT D_i into a priority queue Q, ranked by their volumes.
2. Remove the node m with the highest priority from Q. Search for the best matching node n from possible candidates in the other DCT D'_i, constrained by the conditions mentioned above. The best match should have the highest score $\langle m,n\rangle$ weighted by their average volumes.
3. If a node n is found, the pair (m,n) is added to the set of matched pairs at resolution level i and n is also removed from future consideration.
4. Repeat step 2 and 3 until the queue Q is empty or no more candidates available in D'_i.
5. Calculate the similarity score $\langle D_i D'_i\rangle$ by using the pairs of matched nodes in level i.
6. Repeat the steps 1 to 5 from level $i = 0$ to k. Calculate the final score $\langle M,M'\rangle$ as the similarity measure between two shapes.

Next we look at the time complexity of the complementary shape comparison algorithm. The time can be divided into that of a off-line process of constructing the pocket function and its multi-resolution DCTs and that of an on-line matching process.

We here focus on the off-line time complexity of constructing the DCT and its hierarchy. In the worst case, the complexity is $O((\log n + m)\cdot D)$, where m is number simplices and n is the number of vertices in the function domain $\mathscr{M}$ and D is the number of nodes in the finest DCT. D can be controlled by the number of functional intervals and is usually much smaller than n and m. In our experiments, most DCT construction time is spent on computing various node attributes of the finest DCT. The DCTs can then be stored with the shape functions for future comparisons. Please refer to cited literatures for the time complexity of pocket function and CT generation.

The time complexity of the on-line matching algorithm is $O(D_1 \cdot D_2)$ in the worst case, where D_1 and D_2 are the number of nodes in the finest DCTs. This step usually is very fast in our experiments and takes only seconds because D_1 and D_2 are much smaller than the original data size. If time is critical in shape comparison, a early termination approach can be adopted to stop the matching process after a few coarser level DCTs are compared and the time limit has been reached.

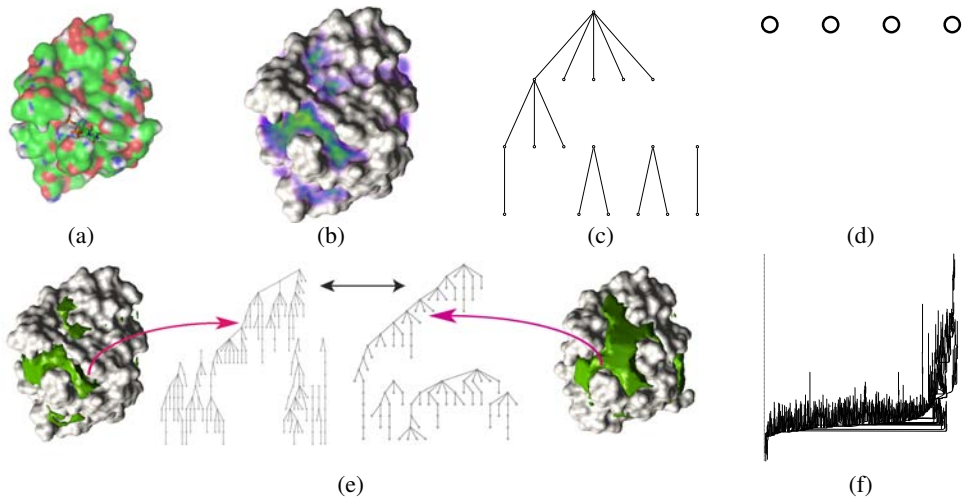


Figure 4: *Shape comparison of the protein "Staphylococcal Nuclease" in the binding (PDB code: 1ATT, (e) left) and unbinding (PDB code: 1KAA, (e) right) state.*

5. Implementation and Results

We implemented the complementary shape matching method in C++. The code is portable across platforms. The effectiveness of the DCT matching algorithm depends on the selection a good shape function. The pocket function in the complementary space appears to have the potential of being a good shape function, especially for molecular shapes.

We first test our implementation on a subset of protein structures downloaded from the Protein Data Bank (PDB). The molecular surfaces of the proteins are computed as a level set of a synthetic electron density scalar function in space R^3 [Bli82]. Figure 4 shows the results of comparing two protein shapes. "Staphylococcal Nuclease" (PDB code 1A2T) is a protein for nucleic acid binding and binds two ligands "S-(Thioethylhydroxy)Cystine"(CME) and "Thymidine-3',5'-Diphosphate"(THP). The protein 1A2T and one of its bound ligand (THP) are drawn in Figure 4 (a) while the other ligand is on the back side. Due to the bound ligands, the shape of "Staphylococcal Nuclease" (1A2T), especially the pocket regions, has changed from that of its unbound sibling (PDB code 1KAA). We compute the pocket function of 1A2T and 1KAA and compare them using the multi-resolution DCT algorithm described in this paper. Figure 4 (b) uses volume rendering to display the pocket function of 1ATT over its molecular surface. The correspondences between the pocket regions of 1ATT and 1KAA are easily established by using the multi-resolution DCT hierarchy, as illustrated in Figure 4 (e). The complementary shape matching makes the evident structural differences between two proteins, which are not obvious by looking at the original shapes directly. The CT of the 1ATT pocket function is shown in Figure 4 (f), which is too complex to be compared to each other. Two lower-resolution DCTs of the protein 1KAA are shown in Figure (c) (four intervals) and (d) (one interval).

We also test robustness of our algorithm by modulating the original shapes with small random noises. Such noises may significantly increase the number of critical points in the pocket functions, sometimes by more than twenty times. However, such noises are effectively removed from DCTs and the modulated shapes are almost perfectly matched to original ones.

6. Conclusions

In this paper we presented a novel algorithm for shape matching by using surface pockets in the complementary space as a volumetric shape function and compute an affine-invariant multi-resolution dual contour tree to compare shape functions with properties. The DCT algorithm segments the 3D shape function into smaller feature elements, i.e. the DCT nodes. Those feature elements combined with geometrical and topological attributes and additional properties such as electrostatic potentials, are shown to be very effective for comparing complex structures like molecular surfaces. It can also be applied to general 3D shapes.

Further improvements of the method may include adding more shape descriptors to the DCT nodes, e.g. the ones described in the cited literatures. Another future work is to build a shape database of pocket functions and corresponding multi-resolution attributed dual contour trees (MACT) for all known protein structures in PDB, in order to facilitate the study of protein structures and functions.

Acknowledgements

The author would like to thank Dr. Chandrajit Bajaj for useful discussions and valuable helps for this work.

References

[ACH*91] ARKIN E. M., CHEW L. P., HUTTENLOCHER D. P., KEDEM K., MITCHELL J. S. B.: An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 3 (1991), 209–216.

[ASBH90] ARBTER K., SNYDER W. E., BURHARDT H., HIRZINGER G.: Application of affine-invariant fourier descriptors to recognition of 3-d objects. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 7 (1990), 640–647.

[ATRB95] ASHBROOK A. P., THACKER N. A., ROCKETT P. I., BROWN C. I.: Robust recognition of scaled shapes using pairwise geometric histograms. In *Proceedings of the 6th British conference on Machine vision (Vol. 2)* (1995), BMVA Press, pp. 503–512.

[BCGJ98] BASRI R., COSTA L., GEIGER D., JACOBS D.: Determining the similarity of deformable shapes. *Vision Research 38* (1998), 2365–2385.

[Bli82] BLINN J. F.: A generalization of algebra surface drawing. *ACM Transactions on Graphics 1*, 3 (1982), 235–256.

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 2 (1992), 239–256.

[BMP02] BELONGIE S., MALIK J., PUZICHA J.: Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4 (2002), 509–522.

[BPS97] BAJAJ C., PASCUCCI V., SCHIKORE D.: The contour spectrum. In *Proceedings of the 1997 IEEE Visualization Conference*. 1997.

[BW03] BOURNE P. E., WEISSIG H. (Eds.): *Structural Bioinformatics*. Methods of Biochemical Analysis. John Wiley and Sons, New Jersey, 2003.

[BWF*00] BERMAN H. M., WESTBROOK J., FENG Z., GILLILAND G., BHAT T. N., WEISSIG H., SHINDYALOV I. N., BOURNE P. E.: The protein data bank. *Nucleic Acids Res 28*, 1 (2000), 235–42. 0305-1048 Journal Article.

[CSA03] CARR H., SNOEYINK J., AXEN U.: Computing Contour Trees in All Dimensions. *Computational Geometry: Theory and Applications 24*, 2 (2003), 75–94.

[DGG03] DEY T. K., GIESEN J., GOSWAMI S.: Shape segmentation and matching with flow discretization. In *Workshop on Algorithms and Data Structures* (2003).

[EHNP03] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Morse-smale complexes for piecewise linear 3-manifolds. In *ACM Symposium on Computational Geometry* (2003).

[FKMS05] FUNKHOUSER T., KAZHDAN M., MIN P., SHILANE P.: Shape-based retrieval and analysis of 3d models. *Commun. ACM 48*, 6 (2005), 58–64.

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T.: Topology matching for fully automatic similarity estimation of 3d shapes. In *Siggraph 2001* (Los Angeles, USA, 2001), pp. 203–212.

[KFR03] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), Eurographics Association, pp. 156–164.

[KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Shape matching and anisotropy. *ACM Trans. Graph. 23*, 3 (2004), 623–629.

[KOB*97] KREVELD M. V., OOSTRUM R. V., BAJAJ C., SCHIKORE D., PASCUCCI V.: Contour trees and small seed set for isosurface traversal. In *Proceedings: Thirteen ACM Symposium on Computational Geometry*. ACM Press, 1997, pp. 212–219.

[KPNK03] KÖRTGEN M., PARK G.-J., NOVOTNI M., KLEIN R.: 3d shape matching with 3d shape contexts. In *The 7th Central European Seminar on Computer Graphics* (April 2003).

[Mil63] MILNOR J.: *Morse Theory*, vol. 51 of *Annals of Mathematics Studies*. Princeton University Press, 1963.

[OFCD01] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Matching 3d models with shape distributions. In *Proceedings of the International Conference on Shape Modeling & Applications* (2001), IEEE Computer Society, p. 154.

[PCM02] PASCUCCI V., COLE-MCLAUGHLIN K.: Efficient computation of the topology of level sets. In *IEEE Visualization 2002* (2002).

[SKG97] SONTHI R., KUNJUR G., GADH R.: Shape feature determination usiang the curvature region representation. In *Proceedings of the fourth ACM symposium on Solid modeling and applications* (1997), ACM Press, pp. 285–296.

[ZB06] ZHANG X., BAJAJ C.: Molecular surface feature analysis and quantitative visualization, 2006.