Affine Invariant Comparison of Molecular Shapes with Properties

Xiaoyu Zhang
California State University, San Marcos
59 PUBLICATIONS 753 CITATIONS

Chandrajit L. Bajaj
University of Texas at Austin
534 PUBLICATIONS 12,234 CITATIONS

Affine Invariant Comparison of Molecular Shapes with Properties

Xiaoyu Zhang[*]
Department of Computer Science
Cal State University San Marcos

Chandrajit L. Bajaj[†]
Department of Computer Sciences and ICES
University of Texas at Austin

Nathan Baker [‡]
Department of Biochemistry and Molecular Biophysics
Washington University in St. Louis

ABSTRACT

We present an algorithm of matching 3D shapes in an affine-invariant way using multi-resolution dual contour trees enhanced by geometrical, topological, and functional properties. Dual contour trees (DCT) are simplified structures generated from contour trees of volumetric functions. Each node of a DCT corresponds to a set of connected arc segments of the original contour tree, which is a connected sub-volume bounded by contours. DCT nodes are enhanced with geometrical, topological, and functional attributes. Similarities among 3D molecular shapes are computed by matching nodes from multi-resolution DCT's and their attributes. This algorithm shows good performance in clustering proteins into different families compared to standard inner-product methods.

Keywords: multi-resolution, dual contour tree, volumetric function matching

1 INTRODUCTION

The structures of proteins and other large molecules are being determined at dramatically increasing rate through structural genomic and other efforts. Effective comparison and classification of these biomolecules are very important for the understanding of their structural and functional properties. For example, a major resource is the Protein Data Bank (PDB) [8]. There are now more than twenty thousand entries in the PDB and new protein structures are being added daily.

Traditional protein comparison methods are usually based on the similarity of sequences [17] or the three-dimensional structures of biomolecular chains [11]. While such methods have proven to be very useful techniques for comparing protein structures, they lack a description of the chemical heterogeneity that gives (in part) a particular protein structure its unique molecular function.

A closely related problem is shape matching of objects in $\mathbb{R}^3$. A common approach for shape matching is to segment the shape into basic features and match those basic features and their spatial relationships [6, 7, 9, 16, 27]. Another approach is to compute geometrical signatures of a surface and compare them with the signatures of the other surface. Various signatures have been used for shape matching, such as curvature distribution [2, 37], shape distribution [3, 33, 35], and coefficients of functional expansions [1, 26]. However, most shape matching methods are applied only to relatively simple objects.

Because of the high complexity of molecular shapes, as evidenced by the example in Figure 1 (b), their comparsion pose many new challenges. Molecular surfaces are often modeled as contours of some scalar functions f, e.g. electron density field or solvent accessibility in $\mathbb{R}^3$ [10, 31]. We call such a function as a molecular structural function. The electron density is often computed as the sum of the contributions from individual atoms. For simplicity, the contribution of an individual atom is usually approximated by a Gaussian function or spline function [21].

Important properties of biomolecules are often quantitatively calculated as volumetric functions. The two most commonly used functions are the distribution of electron density and electrostatic potential although other properties have also been used [20]. It is shown that the electrostatic properties are conserved for proteins in the same families and superfamilies [29].

In order to avoid modeling the complex structures of molecular surfaces, we choose to match the volumetric function f and enhance the comparison with the properties of biomolecules, e.g. electrostatic potential.

A standard method for comparing volumetric functions is to compute their inner product. Since the functions related to biomolecules are square-integrable, the L^2 norm is commonly used. The inner product of two functions f and g and the length of a function f in the L^2 norm are defined as:

$$\langle f, g \rangle = \int_V f(\mathbf{x}) \cdot g(\mathbf{x}) d\mathbf{x}$$
$$\|f\| = \langle f, f \rangle^{1/2}$$

Based on the definition of norms, similarity metrics are then defined for molecular functions such as electron density and electrostatic potential. The most popular metrics are Hodgkin index [24] $S_H(f, g)$ and Carbo index [12] $S_C(f, g)$:

$$S_H(f,g) = \frac{\langle f, g \rangle}{\|f\|^2 + \|g\|^2}$$
$$S_C(f,g) = \frac{\langle f, g \rangle}{\|f\| \cdot \|g\|}$$

The inner product method usually produces good matching results if molecules are initially aligned appropriately. However, it is not trivial to geometrically align large molecules [11] and the Carbo/Hogkin metrics are not invariant under affine transformations. The alignments of molecules are often done manually, otherwise one has to search the six-dimension rotational and translational space to find the best alignment. Such search in the six-dimension space is expensive and usually does not guarantee to always find the best match.

To avoid the expensive search in the six-dimension space, it is necessary to have new metrics based on affine-invariant signatures of the volumetric functions. One example is the contour spectrum [4], which is a set of numerical metrics for the level sets of the function, such as the area of the surfaces, the volume enclosed by the surfaces, and the integral of the gradient on the surfaces.

[*] xiaoyu@csusm.edu
[†] bajaj@cs.utexas.edu
[‡] baker@biochem.wustl.edu

Although it provides some useful information about the 3D functions, especially for some distinctive level sets, the contour spectrum alone is not sufficient to match and classify volumetric functions.

Another approach is to use some topological structures of the volumetric functions, such as Morse complex [5, 18] and contour tree (CT) [28]. Similar topological structures are defined for surfaces and are applied to shape matching [23], where Multi-resolution Reeb Graphs (MRG) are computed. Both Morse complex and contour tree are related to the critical points of the volumetric function f. The *critical points* in the domain $\mathscr{M}$ of f are the positions where the gradient vanishes, i.e. $\nabla f = 0$. Molecular structural functions usually have a large number of critical points because of their complexity.

While the Morse complex is invariant under the rigid body transformations, it is difficult to construct and has very complicated structures. Although some recent research attempts to make the Morse complex more manageable by contracting critical points and simplifying the Morse complex [15, 19], it is still not clear how to directly match two Morse complexes.

Although contour trees are affine-invariant, they are also very complicated due to the vast number of critical points in the molecular functions, as shown by the example in Figure 1. It is very difficult to directly define a quantitative metric for contour trees and compare them because of the complexity.

In this paper, we construct a simplified structure, called dual contour tree (DCT), from the original contour tree; define new geometrical, topological, and functional attributes for the DCT nodes; and efficiently compare 3D molecular shapes by matching multi-resolution hierarchies of DCT's. The major steps of the algorithm are:

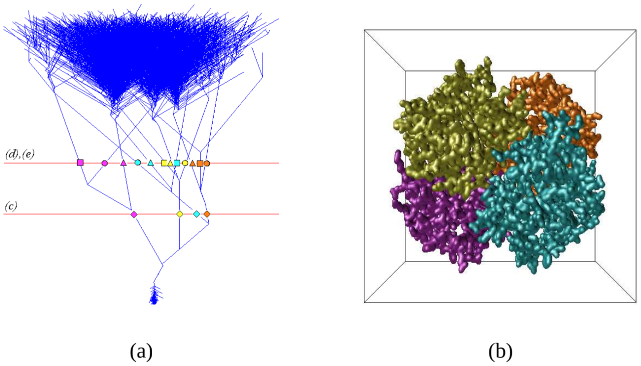
1. Construct the finest-level DCT's from contour trees of molecular structural functions.
2. Compute attributes for the nodes in the DCT's.
3. Build multi-resolution hierarchies of the DCT's to facilitate matching.
4. Match pairs of multi-resolution DCT's and compute the similarity scores.

Section 2 introduces some relevant background on contour trees. Section 3 shows how to build a DCT from a contour tree and how to construct the multi-resolution hierarchy. Next we describe the attributes defined for DCT nodes and how to calculate them. We discuss the algorithm of matching the multi-resolution DCT's and computing the similarity scores in section 4. Section 5 then discusses the implementation and results for comparing some sample proteins from the Protein Data Bank.

2 Background

Consider a continuous scalar field f defined on a domain $\mathscr{M}$, $f : \mathscr{M} \to \mathbf{R}$. In the rest of this paper, we assume $\mathscr{M}$ is decomposed into a simplicial complex and f is approximated by a piece-wise linear function. The function f on $\mathscr{M}$ is then completely determined by the values of the n vertices of the mesh $\mathscr{M}$. For a point inside a simplex, its function value is approximated as a linear interpolation of the values on the vertices. The *functional range* of f is the interval between the minimum and maximum values of the function f: $[f_{min}, f_{max}]$.

For a scalar value $w \in [f_{min}, f_{max}]$, the level set of the field f at w is the subset of points $L(w) \subset \mathscr{M}$, $f(\mathbf{x}) = w \, \forall \, \mathbf{x} \in L(w)$. A level set may have several connected components, called *contours*. While w scans monotonically through the entire range $[f_{min}, f_{max}]$ of f, the evolution of the homology classes of the level sets is well studied in Morse theory [22, 32]. The topology of the level set $L(w)$ changes only at the critical points in $\mathscr{M}$, whose corresponding functional values are called critical values.



(a)

(b)

Figure 1: The contour tree (a) and a segmented level set (b) of the density map of a hemoglobin molecule.

Without loss of generality, the critical points are assumed to be non-degenerate in Morse theory, i.e. only a single critical point at each critical value. This assumption can be enforced by small perturbations of the function values. If the critical points are non-degenerate, the *Hessian* $H(a)$ at a critical point a has non-zero real eigenvalues. The index of the critical point a is the number of negative eigenvalues of $H(a)$. For a function in $\mathbb{R}^3$ there are four types of critical points: index 0 (minima), 1 and 2(saddle points), and index 3 (maxima). A new contour is created when the isovalue w crosses the critical value of a minimum. When w crosses an index 1 saddle, a new connection is formed between two contours. When w crosses an index 2 critical value, a hole in the solid $f(\mathbf{x}) \leq w$ disappears. When w crosses a maximum value, a contour vanishes.

A contour tree captures the topological changes of the level sets for the entire functional range $[f_{min}, f_{max}]$ of f. It was introduced by Kreveld et al. [28] to find the connected components of the level sets and seeds cells for contour generation. Each node of the CT corresponds to a critical point and each arc corresponds to a contour class connecting two critical points. A contour class is a maximal set of continuous contours which have the same topology and do not contain critical points. The CT of the electron density map of the hemoglobin molecule is shown in Figure 1 (a). Each leaf node of the CT represents the creation or deletion of a contour at a local minimum or maximum and each interior node represents the joining and/or splitting of two or more components or topology changes at the saddle points. A cut on the arc $(v_1, v_2) \in \mathscr{T}$ by an isovalue $v_1 \leq w \leq v_2$ corresponds to a connect 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 .

The Betti number β_k $(k = 0, 1, \ldots)$ of a simplicial complex is the rank of its k-dimensional homology group and intuitively measures the number of k-dimensional holes. The Betti numbers provide affine-invariant topological information about the level sets. Contours of the same topology have the same Betti numbers, which change only at critical points. Only the first three Betti numbers $(\beta_0, \beta_1, \beta_2)$ of a surface in $\mathbb{R}^3$ are non-zero, where β_0 corresponds to the number of connected components; β_1 corresponds to the number of independent tunnels; β_2 represents the number of voids enclosed by the surface. For example, a sphere has the Betti numbers $(\beta_0, \beta_1, \beta_2) = (1, 0, 1)$ while a torus has the triple $(\beta_0, \beta_1, \beta_2) = (1, 2, 1)$. The Betti triplets are related to Euler characteristic χ of the surface: $\chi = \beta_0 - \beta_1 + \beta_2$.

Carr et al. [14] 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 [34]. In this paper, we use those algorithms to compute enhanced contour trees and then construct DCT structures to compare molecular functions with properties.

3 Dual Contour Trees

In this section we introduce dual contour trees as a simplified structure constructed from contour trees, which can be used to compare molecular structural functions.

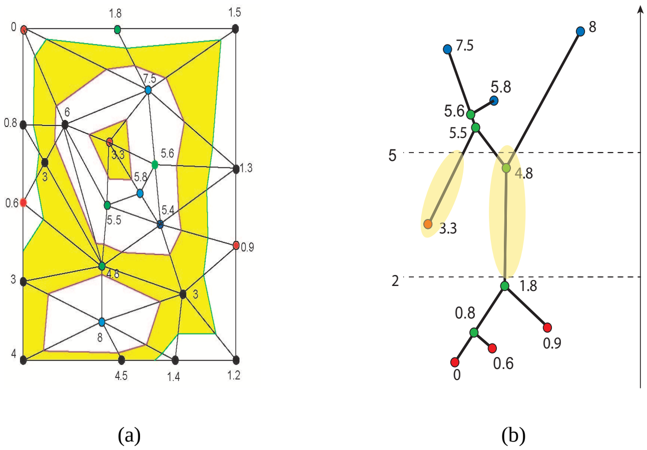


Figure 2: A simple example of 2D mesh and its contour tree. (a) The critical points are marked in different colors: minima in red, saddle points in green and maxima in blue. Two level sets with isovalues of $w_1 = 2$ and $w_2 = 5$ are drawn upon the mesh and the sub-area $V_{2\to5}$ of $2 \le w \le 5$ are shaded in yellow. (b) the two connected components of $V_{2\to5}$ correspond to the arc segments highlighted in the contour tree respectively.

Figure 2 shows a simple example of 2D function f and its contour tree. The contour tree in Figure 2 (b) is cut at two isovalues $w_1 = 2$ and $w_2 = 5$, which correspond to the level sets shown in Figure 2 (a). In this example, the level set $L(w_2)$ at $w_2 = 5$ has three contours (colored magenta) and $L(w_1)$ at $w_1 = 2$ has a single contour (colored green). Those contours bound a subset $V_{w_1 \to w_2}$ of the mesh $\mathscr{M}$, where $\mathbf{x} \in V_{w_1\to w_2}$ iff $w_1 \le f(\mathbf{x}) \le w_2$. $V_{w_1\to w_2}$ has two connected components and is colored yellow in Figure 2 (a). Each connected component of $V_{w_1\to w_2}$ corresponds to a set of connected arc segment cut by w_1 and w_2, as highlighted in Figure 2 (b). Therefore there exists a one-to-one mapping between each set of connected arc segments of a contour tree to a connected subregion (or sub-volume in $\mathbb{R}^3$) in $\mathscr{M}$.

If we cut the functional range $[f_{min}, f_{max}]$ of the field f into N intervals at isovalues $w_1, w_2, ..., w_{N-1}$, then the contour tree of f is partitioned into sets of connected arc segments, each of which is mapped to a connected sub-volume in $\mathscr{M}$. The distribution of those connected sub-volumes contains important information about the original function f. The distribution of sub-volumes is represented by a dual contour tree (DCT), in which every connected sub-volume becomes a node and two nodes are connected by an edge if the corresponding sub-volumes are adjacent (sharing the same contour at their boundaries). Since a node in the DCT corresponds to a set of connect arc segments in the CT, the DCT can be considered as the ”dual” structure of the CT. In the other sense, a DCT can be thought as a ”discrete contour tree” because it is constructed by a set of discrete cutting isovalues $w_1, w_2, ..., w_{N-1}$.

In order to construct the DCT, we first cut the functional range $[f_{min}, f_{max}]$ of the field f into N intervals

$$r_0 = [f_0, f_1), r_1 = [f_1, f_2), \ldots, r_{N-1} = [f_{N-1}, f_N]$$

, where $f_0 = f_{min}$ and $f_N = f_{max}$. Without loss of generality, we can assume isovalues $f_1 < f_2 < \cdots < f_{N-1}$ are not equal to any critical value. Those isovalues cut the CT into N intervals as shown in Figure 3 (a). An arc $e = (v_1, v_2)$ in CT is cut by f_i if $f(v_1) < f_i$ and $f(v_2) > f_i$ for some $0 < i < N$. The arc e may be cut by multiple isovalues and subdivided into several segments.

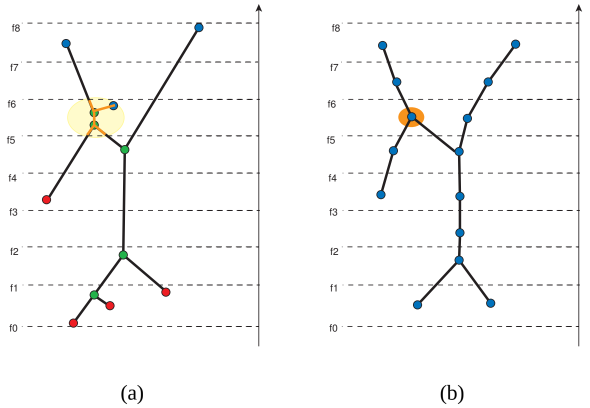


Figure 3: (a) An example of contour tree divided into 8 segments. (b) The dual contour tree constructed from the contour tree in (a).

The DCT also has N intervals and a node d of the DCT at interval i corresponds to a disjoint set of connected arc segments in interval i of the CT. For example, the highlighted arc segments in the interval $[f5, f6]$ of the CT in Figure 3 (a) is mapped to the highlighted node of the DCT in Figure 3 (b). We call the set of connected arc segments for the DCT node d as $Arcs(d)$ and use the union-find data structure to find all disjoint sets $Arcs(d)$ in level i of the CT. The pseudo-code is shown in Figure 4. Because the function f is continuous, a dual node d at interval i can only connect to nodes in intervals $i-1$ and $i+1$. A DCT node d at interval i is connected to a node d' in interval $i+1$ if and only if there exit arc segment $e \in Arcs(d)$ and $e' \in Arcs(d')$ that are adjacent in the CT. The corresponding sub-volume of the DCT node d at level i is bounded by contours of lower and upper isovalues of the interval.

The CT and DCT of the example function in Figure 2 are shown in 3, where the functional range is divided into 8 intervals. The DCT has simpler structure than the CT because small undulations of the function are eliminated in the DCT. On the other hand, the structure of the DCT converges to the CT if we increase the number and decrease the size of cutting intervals.

The DCT structure can be further simplified by restricting it to a smaller functional range $[w_1, w_2] \subset [f_{min}, f_{max}]$. For example we can consider only the portion of the DCT in Figure 3 (b) within the subrange $[f_4, f_6]$. Of course we would cut $[f_4, f_6]$ into more intervals to get more detailed information about the function f in this range. The rationale of using a subrange is that only certain feature regions of the 3D volume are important for comparing molecular shapes, e.g. the regions near the surfaces of biomolecules. Those feature regions often map to a subrange of the structural function f. Examples include the functions for solvent accessibility, electron

```
$\mathscr{D}$ = new DCT($N$) // $N$ is the number of intervals
SubdivideArcs($T$) // $T$ is the contour tree
for $n=0$ to $n=N-1$ {
    $UF$ = new UnionFind
    for each arc $e_i$ in interval $n$ {
        $UF$.newSet($e_i$)
        for each arc $e_j \in r_n$ AND $e_i \cap e_j \neq \emptyset$ AND $j < i$ {
            $n_i$ = $UF$.find($e_i$)
            $n_j$ = $UF$.find($e_j$)
            $UF$.union($n_i$, $n_j$)
        }
    }
    for each disjoint set $S$ in $UF$ {
        $d$ = newNode($S$)
        $\mathscr{D}$.addNode($d$)
        foreach $d'$ in interval $n-1$ and adjacent to $d$ {
            $\mathscr{D}$.addEdge($d$, $d'$)
        }
    }
}
```

Figure 4: The pseudo-code of constructing the DCT.

density, and electrostatic potentials. The extreme values of f usually only happen deep inside the molecules and in very small volumes and have little effect on determining the molecular structures. We can focus on the important feature regions and avoid noises by restricting the functional range and using intervals of adaptive sizes.

The code to construct the DCT for a restricted range $[w_1, w_2]$ is almost identical to that in Figure 4. The portions of the 3D volume with $f(x) \notin [w_1, w_2]$ are excluded from the DCT. We divide the subrange $[w_1, w_2]$ into N intervals, which can be of adaptive sizes, and simply ignore the parts of CT that are beyond $[f_1, f_2]$.

3.1 Multi Resolution DCT

The multi-resolution hierarchy of the DCT is constructed by merging its adjacent functional intervals recursively. Without loss of generality, we assume D is the finest DCT and D has $N = 2^k$ intervals. The overall complexity of the finest DCT and the hierarchy can be controlled by an appropriate number N. The finest DCT D is called D_k and the next coarser DCT D_{k-1} is constructed by merging pairs of adjacent intervals and nodes in the intervals. Intervals $2i$ and $2i+1$ of D_k are merged to form interval i of D_{k-1}. If nodes d_1 and d_2 in intervals $2i$ and $2i+1$ of D_k are connected, d_1 and d_2 are merged into a single node d' in interval i of D_{k-1}. d' is called the parent of d_1 and d_2; d_1 and d_2 are called the children of d'. The attributes of the merged node is computed from those of its children, which is discussed in the next section. Notice more than two nodes in the finer DCT may be merged to be a node in the coarser DCT. More precisely a disjoint set of connected nodes in intervals $2i$ and $2i+1$ of D_k is merged into a node in interval i of D_{k-1}. For example in Figure 5, four nodes in intervals $[f_4, f_5]$ and $[f_5, f_6]$ of the finest DCT are merged into a single node in interval $[f_4, f_6]$ of the next coarser DCT. The union-find data structure can again be used to find disjoint sets of connected nodes in the merged intervals.

The merging process can be recursively applied to the coarser DCT's until there is only one big interval. So we get a multi-resolution hierarchy of $k+1$ dual contour trees $\{D_k, D_{k-1}, \ldots, D_0\}$. Figure 5 shows the hierarchy of the DCT in Figure 3 by merging the initial 8 intervals into 4, 2, and 1 intervals recursively. The complexity of the DCT's at coarser levels is significantly reduced and the hierarchy makes it a lot easier to match nodes in different DCT's. A match algorithm based on the multi-resolution DCT's is discussed in section 4.

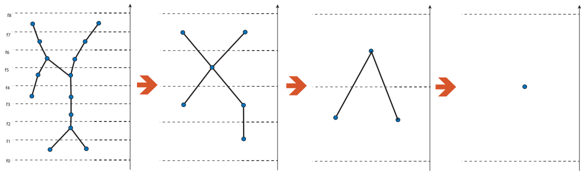


Figure 5: Building the hierarchy of a multi-resolution contour tree.

If a DCT is constructed using a restricted subrange $[w_1, w_2]$, there may be multiple nodes even in the coarsest DCT D_0 because the volume satisfying $f(x) \in [w_1, w_2]$ may have many disconnected components. Numerous nodes in the coarsest DCT D_0 may complicate the matching process. However, most of those nodes are typically very small in size and can be pruned as noises. We prune the nodes in D_0 with volume less than a given threshold and their children in the hierarchy as well. In a test set of more than 200 proteins, only several proteins have more than one node in the coarsest level DCT after conservative pruning.

3.2 Node Attributes

In order to quantitatively measure the similarities of DCT's, we need to define numerical attributes for the DCT nodes. We compute for the DCT nodes geometrical, topological, and functional attributes that are affine invariant. The structural function is used to represent molecular shapes and construct the DCT's. Additional functions are treated as properties defined on the shape and used for computing functional attributes of the DCT nodes. For example, we can use the solvent accessibility as the structural function and electrostatic potential as a property to enhance the comparison of molecular shapes. Similar biomolecules would have similar attributes for their DCT nodes while unrelated ones have very different attributes. This method of combining structural and property information of biomolecules produces very good results in comparing and clustering proteins.

We next describe the attributes for a DCT node m, which corresponds to a connected sub-volume $V_m \subset \mathscr{M}$. We first look at the geometrical and topological attributes that are related to the structure of V_m:

- $V(m)$: The normalized volume of V_m, assuming the sum of all sub-volumes has total volume 1. Matched DCT nodes should have similar volumes.

- $I(m) = \left(I_x, I_y, I_z\right)$: The principal values of the moment of inertia for the sub-volume 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 $\vec{x}_c = \left(x_1^c, x_2^c, x_3^c\right)$ 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 computed as the attribute $I(m) = \left(I_x, I_y, I_z\right)$, where $I_x \leq I_y \leq I_z$. $I(m)$ provides some information about the overall shape of V_m.

- $B(m) = \{B_1(m), B_2(m)\}$: The Betti numbers for the lower and upper boundaries of the sub-volume V_m. $B_1(m)$ and $B_2(m)$ have three non-zero values $(\beta_0, \beta_1, \beta_2)$. The Betti numbers can be computed from an augmented CT [34]. If the lower or upper boundary of V_m consists of multiple contours, its Betti numbers are the sum of those of individual contours.

We then look at functional attributes induced by the property p distributed over the molecular shape, e.g. electrostatic potential. Here we calculate the multipole expansion of the distribution of the property over the sub-volume V_m. The multipole expansion provides numerical information about the property distribution that can enhance the comparison of biomolecules since closely matched proteins should have similar electrostatic potential distribution as well.

- $P(m)$: The integral of the property p over the sub-volume V_m, $P(m) = \int_{V_m} pd^3\vec{x}$ is the first term of the multipole expansion.

- $\vec{D}(m)$: The dipole moment of the property p over the sub-volume 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 principle axis of V_m for computing similarity scores.

- $Q(m) = (Q_1, Q_2, Q_3)$: The quadrupole moment of p over the sub-volume 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 use the principal values of $Q_{ij}(m)$, $Q_1 \le Q_2 \le Q3$, as attributes of a DCT node.

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

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

We only need to compute the attributes for the nodes in the finest DCT because the attributes of a node m in the coarser level of the hierarchy can be easily calculated from those of its children. If the node m has children $m_1, \ldots, m_l$ in the finer level, the volume $V(m)$ is the sum of volumes of its children:

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

If the node m is in interval r_i that is merged from intervals r_{2i} and r_{2i+1} of the finer DCT, the lower (upper) boundary of m is the union of the lower (upper) boundaries of its children in interval r_{2i} (r_{2i+1}). Therefore $B_1(m)$ and $B_2(m)$ can be computed as

$$\begin{aligned} B_1(m) &= \sum_{m_i \in r_{2i}} B_1(m_i) \\ B_2(m) &= \sum_{m_i \in r_{2i+1}} B_2(m_i), \end{aligned}$$

where the sum of B_1 and B_2 is applied to each of the Betti number independently.

As for the functional attributes due to the property function, 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 how to compute the attributes for the nodes in the finest DCT.

While Betti numbers $B(m)$ can be computed from the augmented CT, other attributes like $V(m)$ and $P(m)$ require actually determining the connected sub-volume $V_m \subset \mathscr{M}$ corresponding to the node m. Since the domain $\mathscr{M}$ is assumed to be a simpicial complex, we need to find the set of simplices in $\mathscr{M}$ that interest V_m: $U(m) = \{s \in \mathscr{M} | s \cap V(m) \neq \emptyset\}$, and accumulate the contributions of each simplex $s \in U(m)$ to the attributes of m. The set $U(m)$ can be obtained by a propagation scheme that starts from an initial simplex $s_0 \in U(m)$ because of the connectivity of V_m. The propagation algorithm is as follows:

1. Initialize $U(m) = \emptyset$ and add s_0 into a queue Q.

2. Remove the first element s from Q and add s to $U(m)$.

3. For each neighboring simplex s' of s, add s' to Q if $s' \notin U(m)$, $s' \notin Q$, and $s' \cap V_m \neq \emptyset$.

4. Repeat step 2 and 3 until Q is empty.

Then the volume $V(m)$ of V_m is the sum of contributions from the simplices $s \in U(m)$:

$$V(m) = \sum_{s \in U(m)} V(s \cap V_m),$$

where the volume of $s \cap V(m)$ can be evaluated efficiently as a B-spline function [4]. The center of mass for V_m is

$$\vec{x}^c = \frac{1}{V(m)} \sum_{s \in U(m)} \int_{s \cap V_m} \vec{x} d^3\vec{x},$$

where the integral of the linear function $\vec{x}$ over $s \cap V(m)$ is evaluated in a similar way to the volume. Similar expressions can be derived and evaluated for the other attributes $I(m), P(m), D(m)$, and $Q(m)$.

The remaining question is how to find the starting simplex $s_0 \in U(m)$. It is the same problem of finding the seed to propagate a contour corresponding to a cut on an arc α of the CT [13]. This can be further reduced to find a edge $e \in \mathscr{M}$ that intersects V_m because any simplex containing the edge e can be chosen as the staring simplex s_0. If the arc α has at least one end point v with degree 1 (minimum or maximum) or 2 (regular point) in the CT, we can always search from v and find an edge $e \in \mathscr{M}$ intersecting $V(m)$.

The problem becomes a little more complicated if both end points of the cut arc α are saddle points of degree > 2 in the CT. The arc α can be tagged with a point $p \in \mathscr{M}$ adjacent to one end point v of the arc α in the CT [13]. Then the corresponding edge (v, p) gives the right direction in $\mathscr{M}$ to search the starting simplex. Please refer to [13] for more detail.

4 MATCHING ALGORITHM

4.1 Similarity Metrics

Next we define a similarity metric between two DCT nodes. The similarity between two nodes m and n is defined based on their attribute vectors $\tilde{m}$ and $\tilde{n}$ as the weighted average of the similarities of individual components:

$$\begin{aligned} \langle m,n \rangle &= w_1 \langle V(m), V(n) \rangle + w_2 \langle B(m), B(n) \rangle \\ &+ w_3 \langle I(m), I(n) \rangle + w_4 \langle P(m), P(n) \rangle \\ &+ w_5 \langle D(m), D(n) \rangle + w_6 \langle Q(m), Q(n) \rangle, \end{aligned} \tag{1}$$

where the weights satisfying $0 \le w_i \le 1$ and $\sum w_i = 1$ control the relative importance of different attributes for comparison. The individual terms in the equation above are computed as following:

- $\langle V(m), V(n) \rangle = 1 - \frac{|V(m) - V(n)|}{\max(V(m), V(n))}$: the similarity score of the volumes.

- $\langle B(m), B(n) \rangle = \frac{1}{3} \sum_{i=0}^{2} \frac{\min(\beta_i(m), \beta_i(n))}{\max(\beta_i(m), \beta_i(n))}$: the similarity of the Betti numbers of lower and upper boundaries.

- $\langle I(m), I(n)\rangle = 1 - \frac{\max_{j=1,2,3}(|I_j(m)-I_j(n)|)}{\max(I_1(m),I_1(n))}$: the similarity of the moment of inertia.
- $\langle P(m), P(n)\rangle = 1 - \frac{|P(m)-P(n)|}{\max(|P(m)|,|P(n)|)}$: the similarity of the integrals of properties.
- $\langle D(m), D(n)\rangle = 1 - \frac{|D(m)|-|D(n)|}{\max(|D(m)|,|D(n)|)}$: the similarity of the dipole moments.
- $\langle Q(m), Q(n)\rangle = 1 - \frac{\max_{j=1,2,3}|Q_j(m)-Q_j(n)|}{\max(|Q_1(m)|,|Q_1(n)|)}$: the similarity of the quadruple moments.

The maximum similarity score between two nodes is clearly 1, which is achieved when they have exactly the same attributes.

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

The similarity score may be negative because $\langle P(m),P(n)\rangle$ can be negative if $P(m)$ and $P(n)$ have opposite signs as well as $\langle Q(m),Q(n)\rangle$.

We match two DCT's D and D' by finding and matching node pairs $m_i \in D$ and $n_i \in D'$. The detail of matching algorithm is discussed in the next section. The similarity score between the DCT's D and D' is computed from the scores of matched node pairs:

$$\langle D, D'\rangle = \sum_i \frac{1}{2}(V(m_i)+V(n_i)) \cdot \langle m_i, n_i\rangle\,, \tag{2}$$

where the similarity $\langle m_i, n_i\rangle$ of a node pair is weighted by the average of their normalized volume. So bigger weights are given to larger nodes and the similarity score between D and D' satisfies $\langle D, D'\rangle \le 1$.

For a multi-resolution hierarchy $H = \{D_0, D_1, \ldots, D_k\}$ with $k+1$ levels, its DCT D_i at level i is matched to the DCT D'_i at the same level of another hierarchy H'. The similarity between H and H' is evaluated as the average of the similarity scores of DCT's from level 1 to k:

$$\langle H, H'\rangle = \frac{1}{k}\sum_{i=1}^{k} \langle D_i, D'_i\rangle, \tag{3}$$

where we did not consider the level 0 because D_0 and D'_0 often have only one node and has a similarity score close to 1. The similarity score $\langle H, H'\rangle$ is used to measure the similarity between the molecular shape functions with properties, which clearly satisfies $\langle H, H'\rangle \le 1$.

4.2 Matching Algorithm

The molecular shape functions are compared by matching their multi-resolution hierarchies of DCT's. The matching process is performed from the coarsest to the finest level of the hierarchies, and we assume that the multi-resolution hierarchies H and H' have the same number of levels. A DCT node $m \in H$ and $n \in H'$ must satisfy following restrictions if they form a matched pair (m,n):

- The nodes m and n don't belong to any other pairs.
- m and n must belong to the DCT's at the same level of the hierarchies, i.e. $m \in D_i \subset H$ and $n \in D'_i \subset H'$.
- 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 D_{i-1} and D'_{i-1}. The only exception is level 0, at which 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 $D_i \subset H$ and $D'_i \subset H'$ at level i $(i = 0, \ldots, k)$ are as follows,

1. Add all nodes of the DCT D_i into a priority queue Q, in which the nodes are 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 restrictions 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 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 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 similarity score $\langle H, H'\rangle$ as in equation 3.

4.3 Complexity Analysis

The time complexity of matching and comparing two molecular functions f_1 and f_2 based on multi-resolution DCT's can be divided into three parts.

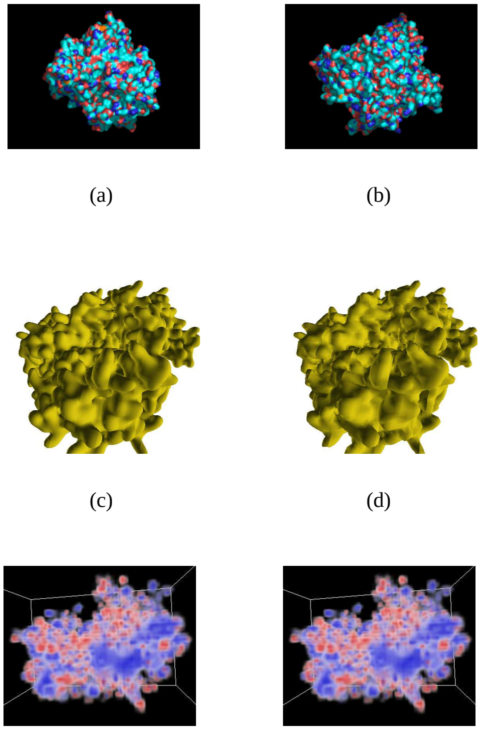
- Time for constructing the enhanced CT: $O(m + n\log n)$ based on the CT algorithms in [14,34], where m is number simplices and n is the number of vertices in $\mathscr{M}$.
- Time for constructing the DCT and its hierarchy: $O((\log n + m)\cdot D)$ in the worst case, where D is the number of nodes in the finest DCT. D is much smaller than n and m and can be controlled by the number of subdivided intervals. Most time of DCT construction is spent on computing node attributes.
- Time for matching the multi-resolution DCT's: $O(D_1 \cdot D_2)$ in the worst case, where D_1 and D_2 are the number of nodes in the DCT's for f_1 and f_2 respectively. This step can be very fast because D_1 and D_2 are much smaller than the original data size.

The first two steps only need to be done once off-line and the DCT's can be stored with molecular functions for future matchings. In comparison, the standard inner-product method requires $O(n^3)$ time for each matching in a naive implementation or $O(n^2\log n)$ time if FFT method is used. The DCT method is clearly a lot faster because the number of nodes D is much smaller than the number of vertices n. Our implementation of the DCT matching algorithm on a PC takes only seconds to match two molecular structural functions with millions of data points.

5 Implementation and Results

The effectiveness of the DCT matching algorithm is based on the assumption that closely related biomolecules have similar features in their structures and properties. The DCT algorithm segments the 3D structural function of biomolecules into smaller feature elements, i.e. the nodes in the DCT's. Those feature elements, enhanced by properties such as electrostatic potentials, are compared to offers insights about relations among proteins. The matching algorithm using multi-resolution DCT is implemented in C++ on a PC based on our earlier implementations of the enhanced CT algorithms.

We first conduct some experiments to test the effectiveness and correctness of the DCT matching algorithm. The DCT matching algorithm has several good features: it is invariant under affine transformations; it does not require same sampling rate for molecular functions; it is resistant to small noises in input data. Three proteins are randomly selected to demonstrate those features. Two different molecular functions are used for each protein. We use electron density as the structural function for those molecules and electrostatic potential as the property. Some visualizations of the molecular functions and properties are shown in Figure 6. Figure 6 (a) and (b) show the level sets of electron density colored by atoms for *Mouse Acetylcholinesterase Complex* (PDB ID: 1MAH), where (b) is generated from (a) by a rotation. Figure 6 (c) and (d) show the level sets of electron density sampled at different rate for *2-Phospho-D-Glycerate Hydrolase* (PDB ID: 1EBG). The sampling rate of (c) is 96 × 96 × 96 and (d) is 64 × 64 × 64. Figure 6 (e) and (f) are volume rendering for the electrostatic potentials of *c-AMP-Dependent Protein Kinase* (PDB ID: 2CPK), where positive values are colored in blue, negative values in red, and values near 0 in white. The data in Figure 6 (f) is generated by adding some small random noise to the original data in Figure 6 (e). All those functions are initially sampled on regular grids, and each cell is then broken into six simplices for CT construction.


(a) (b)

(c) (d)

(e) (f)

Figure 6: Visualizations of molecular functions of three proteins and their variants.

The results of the DCT analysis are summaried in Table 1. Clearly the results of the DCT matching algorithm is invariant under rotational transformations. Due to the added noise, the molecular function in (f) has 72255 critical points, approximately 20 times more than the number of critical points in the original volume (e) 3597. However (e) and (f) have near perfect matching because most critical points induced by noise are suppressed in the DCT nodes, which illustrates the robustness of the DCT method to small perturbations in the functions. The functions in (c) and (d) have very good though not perfect similarity score, which shows the fact that sampling a molecular function at different rates can change the structures and topologies of its level sets.

	(a)	(b)	(c)	(d)	(e)	(f)
(a)	1.00	1.00	0.73	0.72	0.45	0.45
(b)	1.00	1.00	0.73	0.72	0.45	0.45
(c)	0.73	0.73	1.00	0.95	0.33	0.33
(d)	0.72	0.72	0.95	1.00	0.44	0.44
(e)	0.45	0.45	0.33	0.44	1.00	1.00
(f)	0.45	0.45	0.33	0.44	1.00	1.00

Table 1: Matching results of the example volumes shown in figure 6. See text for descriptions of the various datasets.

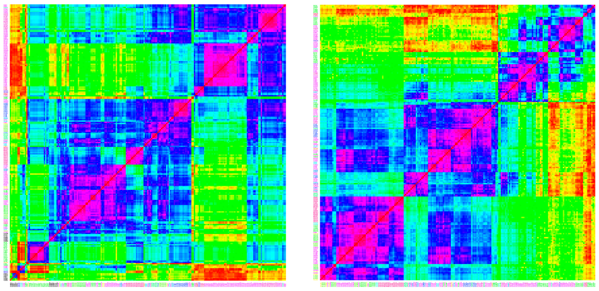

(a) (b)

Figure 7: (a) Clustering result of 242 protein chains by (a) the DCT matching algorithm and (b) the Carbo index method. Axes labeled by chain ID and colored as cholinesterase-like proteins (red), enolase-like proteins (green), ferritin-like proteins (gold), kinases (blue), superoxide dismutases (magenta). Matrix colors range from purple (similarity index 0) to red (similarity index 1).

We also tested our protein matching algorithm on a set of 242 protein chains assembled from several different protein families where extensive, non-redundant structural data was available. The purpose of this analysis was to assess the ability of the DCT method to discriminate between protein families. First, sets of cholinesterase-like proteins and kinases were taken from pre-assembled families in the CE database [36]. Additionally, sets of structures were assembled corresponding to the enolase, ferritin, and superoxide dismutase families studied by Livesay et al in their electrostatic conservation analyses [29]. We use solvent accessibility as the structural function and electrostatic potential as property for each protein in the test set. Multi-resolution DCT's are constructed for each sample and the pair-wise similarity scores of the sample proteins are calculated as described in the proceeding sections. Here we use a restricted functional range $[0.3, 0.7]$ to select sub-volumes near the molecular surfaces. The weights in Equation (1) are estimated using a small set of known proteins. These pairwise scores are then sent to the simple UPGMA clustering algorithm [17] to cluster the protein into similar families. The clustering result is shown in Figure 5 (a) with the protein chains labeled on the axes and color-coded by family. The clustering result is very close to the ones generated by the standard inner-product method using Carbo index, while the actual similarity scores may vary. These initial results are very encouraging as they indicate the feasibility of molecular comparison without structural alignment and thereby enable a new class of comparisons of electrostatic properties between proteins with little sequence similarity. Such comparisons will provide useful information about protein-ligand and protein-protein interactions through the characterization of electrostatic ”motifs”. Electrostatic motifs have already been shown to be useful tools for detecting function, particularly in the identification of DNA binding proteins [25, 30].

6 Conclusions

In this paper we presented an algorithm of matching molecular structures using affine-invariant multi-resolution DCTs for volumetric structural functions with properties. The DCT matching algorithm has the good properties of being affine-invariant and not sensitive to small noises in the functions. Applied to compare and classify a set of protein structures, it demonstrates good performance comparable to the standard inner-product methods. But it is much more efficient because no pair-wise structural alignment is required.

The future directions include using better tree matching algorithms and attributes for the DCT’s. We are looking at more sophisticated geometrical attributes and investigating the possibility of using the DCT’s to align molecular structures.

7 Acknowledgements

The work of XYZ and CB was supported in part by NSF grants ACI-0220037, CCR-9988357, EIA-0325550 and a subcontract from UCSD 1018140 as part of the NSF-NPACI project, Interaction Environments Thrust. NAB would like to acknowledge very useful discussions with J. E. Nielsen, J. A. McCammon, and P. E. Bourne. This work was supported in part by NIH grant GM069702-01, an NPACI grant, and an Alfred P. Sloan Research fellowship to NAB.

References

[1] Arbter, K., Snyder, W. E., Burhardt, H., and 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.

[2] Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K., and Mitchell, J. S. B. An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 3 (1991), 209–216.

[3] Ashbrook, A. P., Thacker, N. A., Rockett, P. I., and 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.

[4] Bajaj, C., Pascucci, V., and Schikore, D. The contour spectrum. In *Proceedings of the 1997 IEEE Visualization Conference*. 1997.

[5] Bajaj, C. L., Pasucci, V., and Schikore, D. Visualization of scalar topology for structural enhancement. In *IEEE Visualization ’98*, D. Ebert, H. Hagen, and H. Rushmeir, Eds. 1998, pp. 51–58.

[6] Basri, R., Costa, L., Geiger, D., and Jacobs, D. Determining the similarity of deformable shapes. *Vision Research 38* (1998), 2365–2385.

[7] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 4 (2002), 509–522.

[8] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic Acids Res 28*, 1 (2000), 235–42. 0305-1048 Journal Article.

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

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

[11] Bourne, P. E., and Weissig, H., Eds. *Structural Bioinformatics*. Methods of Biochemical Analysis. John Wiley and Sons, New Jersey, 2003.

[12] Carbo, R., Leyda, L., and Arnau, M. How similar is a molecule to another? an electron density measure of similarity between two molecular structures. *International Journal of Quantum Chemistry 17*, 6 (1980), 1185–9. Article.

[13] Carr, H., and Snoeyink, J. Path seeds and flexible isosurfaces - using topology for exploratory visualization. In *Joint Eurographics - IEEE TCVG Symposium on Visualizations* (2003).

[14] Carr, H., Snoeyink, J., and Axen, U. Computing contour trees in all dimensions. In *Computational Geometry : Theory and Applications* (2001).

[15] Cazals, F., Chazal, F., and Lewiner, T. Molecular shape analysis based upon the morse-smale complex and the connolly function. In *ACM Symposium on Computational Geometry* (2003).

[16] Dey, T. K., Giesen, J., and Goswami, S. Shape segmentation and matching with flow discretization. In *Workshop on Algorithms and Data Structures* (2003).

[17] Durbin, R., Eddy, S., Krogh, A., and Mitchinson, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, New York, 1998.

[18] Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V. Morse-smale complexes for piecewise linear 3-manifolds. In *ACM Symposium on Computational Geometry* (2003).

[19] Edelsbrunner, H., Letscher, D., and Zomorodian, A. Topological persistence and simplification. In *41st Symposium on Foundations of Computer Science* (Redondo Beach, CA, 2001).

[20] Ferre, F., Ausiello, G., Zanzoni, A., and Helmer-Citterich, M. Surface: a database of protein surface regions for functional annotation. *Nucleic Acids Research 32* (2004), D240–D244.

[21] Grant, J., and Pickup, B. A gaussian description of molecular shape. *Journal of Physical Chemistry 99* (1995), 3503–3510.

[22] Hart, J. C. Morse theory for implicit surface modeling. In *Proc. Visualization and Mathematics ’98* (1998), Springer-Verlag.

[23] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. Topology matching for fully automatic similarity estimation of 3d shapes. In *Siggraph 2001* (Los Angeles, USA, 2001), pp. 203–212.

[24] Hodgkin, E. E., and Richards, W. G. Molecular similarity based on electrostatic potential and electric field. pp. 105–10. International Symposium on Quantum Biology and Quantum Pharmacology Marineland, FL, USA 12-14 March 1987 Int. J. Quantum Chem. Quantum Biol. Symp. (USA).

[25] Jones, S., van Heyningen, P., Berman, H. M., and Thornton, J. M. Protein-dna interactions: A structural analysis. *Journal of Molecular Biology 287*, 5 (1999), 877–96.

[26] Kazhdan, M., Funkhouser, T., and 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.

[27] Körtgen, M., Park, G.-J., Novotni, M., and Klein, R. 3d shape matching with 3d shape contexts. In *The 7th Central European Seminar on Computer Graphics* (April 2003).

[28] Kreveld, M. V., Oostrum, R. V., Bajaj, C., Schikore, D., and 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.

[29] Livesay, D., Jambeck, P., Rojnuckarin, A., and Subramaniam, S. Conservation of electrostatic properties within enzyme families and superfamilies. *Biochemistry 42*, 12 (2003), 3464–3473.

[30] Luscombe, N. M., Austin, S. E., Berman, H. M., and Thorn-

TON, J. M. An overview of the structures of protein-dna complexes. *Genome Biol 1*, 1 (2000), REVIEWS001.

[31] MEZEY, P. *Shape in Chemistry: An Introduction to Molecular Shape and Topology*. John Wiley, New York, August 1993.

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

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

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

[35] RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision* (1998), IEEE Computer Society, p. 59.

[36] SHINDYALOV, I. N., AND BOURNE, P. E. An alternative view of protein fold space. *Proteins 38*, 3 (2000), 247–60.

[37] SONTHI, R., KUNJUR, G., AND 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.