



Three-dimensional Krawtchouk descriptors for protein local surface shape comparison

Atilla Sit^a, Woong-Hee Shin^b, Daisuke Kihara^{b,c,d,*}

^a Department of Mathematics and Statistics, Eastern Kentucky University, Richmond, KY 40475, USA

^b Department of Biological Sciences, Purdue University, West Lafayette, IN 47907, USA

^c Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

^d Department of Pediatrics, Cincinnati Children's Hospital Medical Center, University of Cincinnati, Cincinnati, OH 45229, USA

ARTICLE INFO

Article history:

Received 27 December 2018

Revised 26 April 2019

Accepted 8 May 2019

Available online 8 May 2019

Keywords:

3D image retrieval

Local image comparison

Region of interest

Discrete orthogonal functions

Krawtchouk polynomials

Weighted Krawtchouk polynomials

3D Krawtchouk moments

Protein surface

Ligand binding site

Pocket comparison

Structure-based function prediction

ABSTRACT

Direct comparison of three-dimensional (3D) objects is computationally expensive due to the need for translation, rotation, and scaling of the objects to evaluate their similarity. In applications of 3D object comparison, often identifying specific local regions of objects is of particular interest. We have recently developed a set of 2D moment invariants based on discrete orthogonal Krawtchouk polynomials for comparison of local image patches. In this work, we extend them to 3D and construct 3D Krawtchouk descriptors (3DKDs) that are invariant under translation, rotation, and scaling. The new descriptors have the ability to extract local features of a 3D surface from any region-of-interest. This property enables comparison of two arbitrary local surface regions from different 3D objects. We present the new formulation of 3DKDs and apply it to the local shape comparison of protein surfaces in order to predict ligand molecules that bind to query proteins.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Moment-based approaches have become very popular in 2D [1,2] and 3D [3,4] image processing due to their compact representation of images. A moment-based approach characterizes a 2D or 3D image by considering its shape as a mathematical function and computes integral of the function multiplied by specific base functions. The approach has been used in many problems including reconstruction, detection, pattern recognition, and compression of images. The theory of moment invariants in 2D has been well established since the foundation of algebraic Hu invariants [5]. Sadjadi and Hall [6] extended the algebraic 2D invariants to 3D and explicitly derived the second order moment invariants, which were later reproduced by Guo [7]. Using a group theoretic approach, Lo and Don [8] constructed twelve complex moment invariants including both second and third order moments. Galvez and Canton [9] defined the 3D moments by evaluating them on the 3D object's

surface and extracting global descriptors from normalized surface shapes. An extension of moment invariants to n -dimension can be found in Mamistvalov's work [10], in which the zeroth and second order moment invariants of n -dimensional regular solids were established. Other examples of 3D moments that are invariant to rotation and blur were provided by Flusser et al. [11].

We have recently developed a set of 2D local moment invariants based on the discrete Krawtchouk polynomials and successfully applied them to the comparison of local image patches [12]. Krawtchouk polynomials were used for the first time in image analysis by Yap et al. [13]. In our previous 2D work [12], while constructing a set of local descriptors that are rotation, position, and size independent, we have also preserved their ability to extract features from any local interest region in an image.

In this paper, we extend our previous 2D work to 3D for local comparison of 3D surface shapes. Our new method is based on 3D Krawtchouk polynomials. 3D Krawtchouk moments were earlier defined and used in content-based search [14] and retrieval of 3D objects [15]. Despite the compact representation and discriminative powers of these moments, the theory of invariants based on 3D Krawtchouk polynomials has not been well studied. Also, the

* Corresponding author.

E-mail addresses: atilla.sit@eku.edu (A. Sit), shin183@purdue.edu (W.-H. Shin), dkihara@purdue.edu (D. Kihara).

very critical local retrieval property of the 3D moments has been noticed in [16], but much of the focus is given to their fast computation.

We propose a new approach on this long-standing issue of local image comparison by constructing 3D Krawtchouk descriptors (3DKDs) for describing local 3D surfaces. The new formulation has many advantages over many similar moment-based approaches, such as TRS invariants [17] and Zernike descriptors [18]: 1) Krawtchouk polynomials are defined on a discrete space, so the moments derived from them do not carry any error due to discretization unlike many other moments related to continuous functions. 2) These polynomials are orthogonal; each moment brings in a new feature of the image, where minimum redundancy is critical in their discriminative performance. Moreover, they are directly defined in the image coordinate space, and hence their orthogonality property is well retained in the computed moments. 3) They are complete with a finite number of functions (equal to the image size) while many other polynomial spaces have infinitely many members. 4) They have the ability to retrieve local image patches by only changing the resolution of reconstruction and using low order moments. 5) The location of the patch can also be controlled by changing three parameters and hence shifting the region-of-interest along each dimension. 6) We also prove that these moments can be transformed into local descriptors, which are invariant under translation, rotation, and scaling. Therefore, using only a small number of invariant descriptors per image will make it possible to develop an efficient method for quick local image retrieval.

Moment-based approaches, particularly Krawtchouk moments, are very useful for representing biological and medical images as they are pixelized or voxelized data. In medical imaging, such as computerized tomography (CT) scan and magnetic resonance imaging (MRI), objects are observed at different viewpoints and local images need to be extracted and examined. In digital pathology, for instance, pathologists are interested in information about specific structures rather than the whole image. Thus, it is necessary to construct moment invariants that do not change by translation, rotation, and scaling and can retrieve local image patches or subimages.

Local shape search methods have many applications also in structural biology, which deals with 3D structures of biomolecules. An important application is the identification of ligand molecules (i.e., small chemical compounds including drug molecules) that bind to local protein surface regions, which is important for predicting biological function of proteins [19,20] and for computational drug design [21,22]. Ligand molecules that bind to a local surface region in a protein can be predicted by finding similar local regions (binding pockets) of known ligand-binding proteins in the protein structure database. In this work, we applied the developed 3DKDs for the protein ligand binding pocket comparison. A ligand binding pocket is represented as a combination of overlapping local surface patches, each of which is characterized by its geometric shape. The shapes of surface patches are compactly represented by 3DKDs. The method is benchmarked on a dataset, which contains a total of 463 proteins that bind to at least one of 11 ligand molecules. Overall, the 3DKD-based method showed better performances than those obtained by the previously developed binding prediction methods: Pocket-Surfer [23] and Patch-Surfer2.0 [20].

This paper is organized as follows. In Section 2, we give a brief background of one-dimensional Krawtchouk polynomials. After introducing the 3D weighted Krawtchouk polynomials and their moments in Section 3, we present the theory and formulation of our new 3D Krawtchouk descriptors in Section 4. In Section 5, we provide a detailed scheme for efficient computation of these descriptors. In Section 6, we show numerical results from local surface recognition performances of 3DKDs using protein structures placed in different orientations. Finally, we discuss the application

of 3DKDs on the comparison of ligand binding pockets on protein surfaces. We finish the paper with a conclusion and summary of this work in Section 7.

2. Krawtchouk polynomials

We start with introducing one-dimensional Krawtchouk polynomials, which can also be found in [13]. A more general and abstract form of these polynomials was provided as Hahn polynomials in [24].

The Krawtchouk polynomials of degree n are defined as

$$K_n(x; p, N) = \sum_{i=0}^n a_{i,n,p,N} x^i = {}_2F_1\left(-n, -x; -N; \frac{1}{p}\right) \quad (1)$$

where $x, n = 0, \dots, N$, $N > 0$, $p \in (0, 1)$ and the function ${}_2F_1$ is the hypergeometric function which is defined as:

$${}_2F_1(a, b; c; z) = \sum_{i=0}^{\infty} \frac{(a)_i (b)_i}{(c)_i} \frac{z^i}{i!}. \quad (2)$$

The symbol $(a)_i$ in (2) is the Pochhammer symbol given by

$$(a)_i = a(a+1)(a+2) \dots (a+i-1) = \frac{\Gamma(a+i)}{\Gamma(a)}. \quad (3)$$

Note that the series in (2) terminates if either a or b is a non-positive integer. Hence, the polynomial coefficients $a_{i,n,p,N}$ in (1) can be obtained by simplifying the summation. It is shown in [13] that the range of Krawtchouk polynomials expands rapidly with the increase of the degree. Besides, these polynomials are not numerically stable for large values of N . Hence, a more stable set of polynomials can be obtained from the classical Krawtchouk polynomials by normalizing with the norm and scaling by the square root of a weight function [13]. The weighted Krawtchouk polynomials is then defined by

$$\tilde{K}_n(x; p, N) = K_n(x; p, N) \sqrt{\frac{w(x; p, N)}{\rho(n; p, N)}}, \quad (4)$$

where

$$w(x; p, N) = \binom{N}{x} p^x (1-p)^{N-x}, \quad (5)$$

$$\rho(n; p, N) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n}. \quad (6)$$

The set of weighted Krawtchouk polynomials

$$\tilde{S} = \{\tilde{K}_n(x; p, N) : n = 0, \dots, N\} \quad (7)$$

becomes a complete orthonormal set of basis functions on the discrete space $\{0, \dots, N\}$ with the orthonormality condition

$$\sum_{x=0}^N \tilde{K}_n(x; p, N) \tilde{K}_{n'}(x; p, N) = \delta_{nn'}. \quad (8)$$

To compute the weighted Krawtchouk polynomials, the three-term recurrence relation given in [13] can be used. Such a recursive computation is shown to be more efficient than computing high-degree polynomials directly using (1) and (4). However, due to error propagation, computing polynomials recursively may still be numerically unstable for large N as noted by Zhang et al. [25]. To achieve numerical stability, we use symmetry and bi-recursive algorithm given in [25].

3. Three-dimensional weighted Krawtchouk moments

In this section, we give a brief formulation of 3D weighted Krawtchouk moments, which are also introduced in [14,15]. Note that the functions \bar{K}_n defined by (4) are orthonormal in the one-dimensional discrete set $\{0, \dots, N\}$, but they can be easily extended to three-dimension as follows:

Let

$$A = \{0, \dots, N\} \times \{0, \dots, M\} \times \{0, \dots, L\} \quad (9)$$

be a discrete field in the 3D space. We define the set of 3D weighted Krawtchouk polynomials on A as

$$\bar{S} = \{\bar{K}_n(x; p_x, N) \cdot \bar{K}_m(y; p_y, M) \cdot \bar{K}_l(z; p_z, L) : n = 0, \dots, N, m = 0, \dots, M, l = 0, \dots, L\}. \quad (10)$$

Note that \bar{S} is orthonormal on A with the orthonormality condition

$$\sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L \bar{K}_n(x; p_x, N) \bar{K}_m(y; p_y, M) \bar{K}_l(z; p_z, L) \cdot \bar{K}_{n'}(x; p_x, N) \bar{K}_{m'}(y; p_y, M) \bar{K}_{l'}(z; p_z, L) = \delta_{nn'} \delta_{mm'} \delta_{ll'}, \quad (11)$$

which follows immediately from the orthonormality of 1D functions given by (8). Let $f(x, y, z)$ be a 3D function defined on the grid A in (9). The 3D weighted Krawtchouk moments of order $n + m + l$ of $f(x, y, z)$ are defined by

$$\bar{Q}_{nml} = \sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L f(x, y, z) \bar{K}_n(x; p_x, N) \bar{K}_m(y; p_y, M) \bar{K}_l(z; p_z, L). \quad (12)$$

Note that by using (11) and solving (12) for $f(x, y, z)$, the 3D function $f(x, y, z)$ can be written in terms of the 3D weighted Krawtchouk polynomials, i.e.,

$$f(x, y, z) = \sum_{n=0}^N \sum_{m=0}^M \sum_{l=0}^L \bar{Q}_{nml} \bar{K}_n(x; p_x, N) \bar{K}_m(y; p_y, M) \bar{K}_l(z; p_z, L). \quad (13)$$

This means that the object $f(x, y, z)$ can be reconstructed perfectly if all the moments \bar{Q}_{nml} are used for $n = 0, \dots, N, m = 0, \dots, M, l = 0, \dots, L$. An approximate reconstruction $\hat{f}(x, y, z)$ of $f(x, y, z)$ can be written as

$$\hat{f}(x, y, z) = \sum_{n=0}^{\hat{N}} \sum_{m=0}^{\hat{M}} \sum_{l=0}^{\hat{L}} \bar{Q}_{nml} \bar{K}_n(x; p_x, N) \bar{K}_m(y; p_y, M) \bar{K}_l(z; p_z, L). \quad (14)$$

where $0 \leq \hat{N} \leq N, 0 \leq \hat{M} \leq M, 0 \leq \hat{L} \leq L$.

Fig. 1 presents some reconstructions of 3D binary images using 3D weighted Krawtchouk polynomials for $\hat{N}, \hat{M}, \hat{L}$ values of 5, 10, 25, and 50, and different (p_x, p_y, p_z) triplets. The 3D polygonal models for the horse and the mug image are downloaded from Princeton Shape Benchmark [26] and voxelized using the algorithm in [27]. The 3D weighted Krawtchouk moments from the original image are first computed using (12), and then these moments are used in (14) for reconstructing the image. The center of a local region corresponding to (p_x, p_y, p_z) is at (x_c, y_c, z_c) , where $x_c = Np_x, y_c = Mp_y$, and $z_c = Lp_z$. These points are (97,126,167) near the horse's mouth and (169,145,100) near the mug's handle. Since $N = M = L = 200$ in this example, the (p_x, p_y, p_z) triplets at these centers will correspond to (0.485,0.630,0.835) and (0.845,0.725,0.500), respectively.

As can be seen from left to right in Fig. 1, the reconstructions start at a local region corresponding to (Np_x, Mp_y, Lp_z) and expand as larger values of \hat{N}, \hat{M} , and \hat{L} are used. Theoretically, using

$\hat{N} = N = 200, \hat{M} = M = 200$, and $\hat{L} = L = 200$, the original image will be fully reconstructed regardless of the choice of (p_x, p_y, p_z) . Using smaller numbers for \hat{N}, \hat{M} , and \hat{L} , the reconstructed surfaces contain only local information, which may actually be more useful for local comparison of 3D images. The parameters p_x, p_y , and p_z play a vital role here to determine the center of local region-of-interest.

In the third and fourth row of Fig. 1, we show the voxelized surface of a protein, nucleosome recognition module of imitation SWI ATPase from fruit fly (*Drosophila melanogaster*). The atomic structure of this protein is downloaded from the Protein Data Bank (PDB) [28] (PDB ID: 1OFC) and then is voxelized using 3D-Surfer [29]. The radius¹ of this protein is about 46.6 Å (78 unit voxels). In the grid shown, 1 unit corresponds to 0.6 Å. For this example, two (p_x, p_y, p_z) triplets are selected: (0.66,0.67,0.82) and (0.21,0.44,0.31) in the third and the fourth row, respectively. Thus, the reconstruction centers will be $(x_c, y_c, z_c) = (132, 134, 164)$ and $(42, 88, 62)$, respectively, both chosen from salient parts on the surface of the protein. Once again, using smaller numbers for \hat{N}, \hat{M} , and \hat{L} , the reconstructed surfaces contain only local information. Local retrieval of structures may reveal important information about the function of a protein, and this may be used to locally compare protein structures in a large database and quickly identify their ligand-binding sites. This may be very useful for identifying biological functions of proteins and further computational drug design for target proteins. In this paper, we will employ p_x, p_y , and p_z parameters for detecting the local region-of-interest by changing them between 0 and 1.

4. 3D Krawtchouk descriptors

In this section, we introduce a new set of invariants, called 3D Krawtchouk descriptors. We show that these invariants are not only rotation, size, and position independent, but also contain discriminative local features from any region-of-interest in a 3D image. Such invariants in 2D have been introduced in our previous work [12]. In this work, we extend them to 3D to locally compare 3D images.

Let $f(x, y, z)$ be a function representing a 3D image defined on an orthogonal grid A given in (9) and define the 3D weight function corresponding to the triplets $\mathbf{p} = (p_x, p_y, p_z)$ and $\mathcal{N} = (N, M, L)$ by

$$W(x, y, z; \mathbf{p}, \mathcal{N}) = w(x; p_x, N) w(y; p_y, M) w(z; p_z, L), \quad (15)$$

where $x = 0, \dots, N, y = 0, \dots, M$, and $z = 0, \dots, L$. Similarly, we can define the 3D norms corresponding to the triplets \mathbf{p} and \mathcal{N} by

$$\Omega(n, m, l; \mathbf{p}, \mathcal{N}) = \rho(n; p_x, N) \rho(m; p_y, M) \rho(l; p_z, L) \quad (16)$$

for $n = 0, \dots, N, m = 0, \dots, M$, and $l = 0, \dots, L$. Using (4), the 3D weighted Krawtchouk moments \bar{Q}_{nml} in (12) become

$$\bar{Q}_{nml} = [\Omega(n, m, l; \mathbf{p}, \mathcal{N})]^{-\frac{1}{2}} \sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L \tilde{f}(x, y, z) \cdot K_n(x; p_x, N) K_m(y; p_y, M) K_l(z; p_z, L), \quad (17)$$

where

$$\tilde{f}(x, y, z) = [W(x, y, z; \mathbf{p}, \mathcal{N})]^{-\frac{1}{2}} f(x, y, z). \quad (18)$$

Now, substituting K_n, K_m , and K_l in (17) by their definitions from (1), reordering summations, and grouping terms, we obtain

¹ The radius is calculated as the maximum of all distances between the center of mass of the protein and each amino acid, i.e., the radius of the smallest sphere that inscribes the protein.

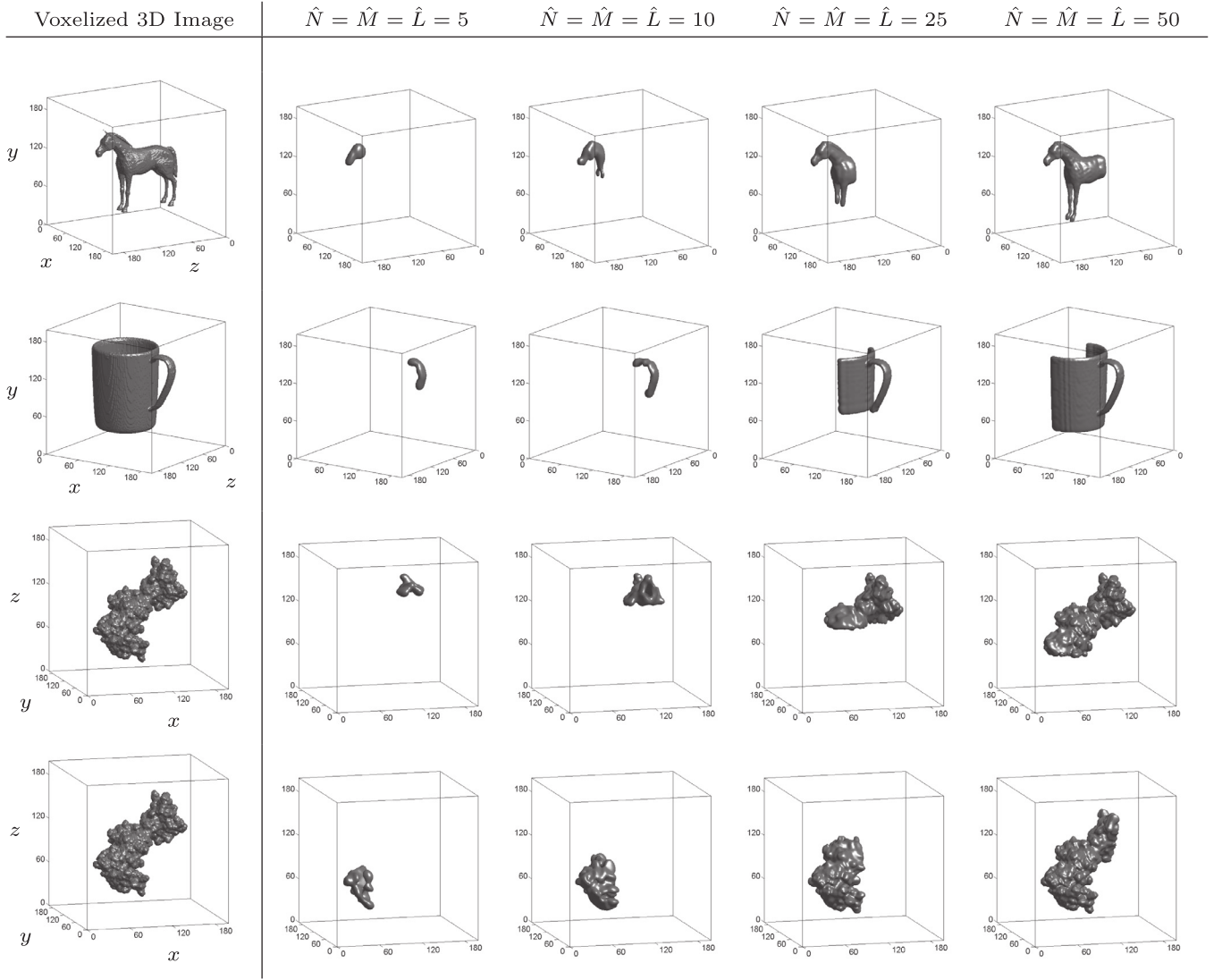


Fig. 1. Examples of 3D binary images and their reconstructions using (14) with $\hat{N} = \hat{M} = \hat{L} = 5, 10, 25$, and 50 and different (p_x, p_y, p_z) triplets. The voxel size for each box is 200^3 . (p_x, p_y, p_z) triplet plays the critical role here in determining the center of local region-of-interest in an image. (p_x, p_y, p_z) was set to $(0.485, 0.630, 0.835)$ and $(0.845, 0.725, 0.500)$ for the horse and the mug image, respectively, to obtain local reconstructions centered at the horse's mouth and the handle of the mug. The manually selected isosurface levels for the images from top to bottom are $0.33, 0.355, 0.195$, and 0.195 , respectively.

$$\bar{Q}_{nml} = [\Omega(n, m, l; \mathbf{p}, \mathcal{N})]^{-\frac{1}{2}} \times \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l a_{i,n,p_x,N} a_{j,m,p_y,M} a_{k,l,p_z,L} \tilde{M}_{ijk}, \quad (19)$$

where

$$\tilde{M}_{ijk} = \sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L \tilde{f}(x, y, z) x^i y^j z^k \quad (20)$$

are the geometric moments of the auxiliary function in (18).

Notice that the geometric moments \tilde{M}_{ijk} and hence the weighted Krawtchouk moments \bar{Q}_{nml} are not invariant under translation, rotation, and scaling. The translation invariant *central moments* of $\tilde{f}(x, y, z)$ can be defined as

$$\tilde{\mu}_{nml} = \sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L \tilde{f}(x, y, z) (x - \tilde{x})^n (y - \tilde{y})^m (z - \tilde{z})^l, \quad (21)$$

where $\tilde{x} = \tilde{M}_{100}/\tilde{M}_{000}$, $\tilde{y} = \tilde{M}_{010}/\tilde{M}_{000}$, and $\tilde{z} = \tilde{M}_{001}/\tilde{M}_{000}$ are the coordinates of the centroid of $\tilde{f}(x, y, z)$.

If $\tilde{\mu}_{nml}$ are the central moments, then we can define geometric moments of $\tilde{f}(x, y, z)$, which are invariant under translation and scaling as follows:

$$\tilde{\eta}_{nml} = \frac{\tilde{\mu}_{nml}}{(\tilde{M}_{000})^{\frac{n+m+l}{3}+1}}. \quad (22)$$

Obtaining rotation invariant geometric moments of $\tilde{f}(x, y, z)$ is, however, not as straightforward. To achieve rotational invariance, we need to find a unique rotation matrix \tilde{R} that would rotate the auxiliary image $\tilde{f}(x, y, z)$ so that its principal axes lie in the x, y, z -directions, respectively. In this work, we did not perform any manual rotation to achieve invariance to rotation; we needed the elements of \tilde{R} to compute the rotation invariants. When the auxiliary image $\tilde{f}(x, y, z)$ is centered at the origin, the principal axes of $\tilde{f}(x, y, z)$ can be defined as the eigenvectors of the inertia matrix

$$\tilde{\mathbf{I}} = \begin{bmatrix} \tilde{I}_{xx} & \tilde{I}_{xy} & \tilde{I}_{xz} \\ \tilde{I}_{yx} & \tilde{I}_{yy} & \tilde{I}_{yz} \\ \tilde{I}_{zx} & \tilde{I}_{zy} & \tilde{I}_{zz} \end{bmatrix}, \quad (23)$$

where

$$\begin{aligned} \tilde{I}_{xx} &= \tilde{\mu}_{020} + \tilde{\mu}_{002}, \quad \tilde{I}_{yy} = \tilde{\mu}_{200} + \tilde{\mu}_{002}, \quad \tilde{I}_{zz} = \tilde{\mu}_{200} + \tilde{\mu}_{020}, \\ \tilde{I}_{xy} &= \tilde{I}_{yx} = -\tilde{\mu}_{110}, \quad \tilde{I}_{xz} = \tilde{I}_{zx} = -\tilde{\mu}_{101}, \quad \tilde{I}_{yz} = \tilde{I}_{zy} = -\tilde{\mu}_{011}. \end{aligned} \quad (24)$$

Here, instead of the inertia matrix $\tilde{\mathbf{I}}$, the covariance matrix can also be used as they have the same set of eigenvectors, perhaps, for a difference in sign. We prefer using the inertia matrix because it is commonly used for 3D rigid bodies and more related to our work. Note that $\tilde{\mathbf{I}}$ is a symmetric matrix with real eigenvalues $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3\}$ and orthogonal eigenvectors $\{\tilde{u}_1, \tilde{u}_2, \tilde{u}_3\}$ such that

$$\tilde{\mathbf{I}}\tilde{u}_i = \tilde{\lambda}_i\tilde{u}_i \quad \text{for } i = 1, 2, 3. \quad (25)$$

The eigenvectors $\{\tilde{u}_1, \tilde{u}_2, \tilde{u}_3\}$ define the columns of the rotation matrix $\tilde{\mathbf{R}}$ that aligns the principal axes with the standard xyz coordinate system. However, for each eigenvalue $\tilde{\lambda}_i$, both \tilde{u}_i and $-\tilde{u}_i$ are eigenvectors, so they define eight different rotation matrices $\{\pm\tilde{u}_1, \pm\tilde{u}_2, \pm\tilde{u}_3\}$ specifying the same principal axes [9].

Although it is possible to reduce this ambiguity to four combinations by only keeping right-handed coordinate systems [9], a heuristic approach is still needed to obtain a unique standard rotation matrix. In our work, we will use the invariants to locally compare 3D image surfaces. For this reason, we first locate points on the surface of the 3D image. Then for each local patch around these points, a direction vector can be specified using the vertex normal at the surface point, pointing away from the surface. Among the eight rotations, the unique rotation matrix $\tilde{\mathbf{R}}$ can be chosen, for example, as the one rotating the vertex normals to the octant in which x , y , and z coordinates are all nonpositive. Once $\tilde{\mathbf{R}}$ is selected, we can define geometric moments of $\tilde{f}(x, y, z)$, which are invariant under rotation, translation, and scaling by

$$\begin{aligned} \tilde{v}_{ijk} &= (\tilde{M}_{000})^{-\frac{i+j+k}{3}-1} \sum_{x=0}^N \sum_{y=0}^M \sum_{z=0}^L \tilde{f}(x, y, z) \\ &\cdot (\tilde{\phi}_1(x, y, z))^i (\tilde{\phi}_2(x, y, z))^j (\tilde{\phi}_3(x, y, z))^k, \end{aligned} \quad (26)$$

where $\tilde{\phi}_1(x, y, z) = \tilde{R}_{11}(x - \tilde{x}) + \tilde{R}_{12}(y - \tilde{y}) + \tilde{R}_{13}(z - \tilde{z})$, $\tilde{\phi}_2(x, y, z) = \tilde{R}_{21}(x - \tilde{x}) + \tilde{R}_{22}(y - \tilde{y}) + \tilde{R}_{23}(z - \tilde{z})$, and $\tilde{\phi}_3(x, y, z) = \tilde{R}_{31}(x - \tilde{x}) + \tilde{R}_{32}(y - \tilde{y}) + \tilde{R}_{33}(z - \tilde{z})$.

Fig. 2 shows 2D views of the square root of the 3D weight function $W(x, y, z; \mathbf{p}, \mathcal{N})$ in (15) for $\mathcal{N} = (200, 200, 200)$ and five different $\mathbf{p} = (p_x, p_y, p_z)$ triplets. To obtain the 2D views, we flatten the function by summing slices along each of the three dimensions. Note that the coverage of each function differs as the parameters $\mathbf{p} = (p_x, p_y, p_z)$ change. The coverage is the largest for $\mathbf{p} = (0.5, 0.5, 0.5)$ and becomes smaller when \mathbf{p} approaches faces of the grid. Different weight functions result in loss of translation and scale invariance. One way to overcome this problem is to determine a unique suitable \mathbf{p} triplet, say $\mathbf{p}^* = (p_x^*, p_y^*, p_z^*)$, and use the corresponding weight $W(x, y, z; \mathbf{p}^*, \mathcal{N})$ for every local region-of-interest by shifting the graph of W to that location, in other words, use the translated weight $W^*(x, y, z; \mathbf{p}^*, \mathcal{N}) = W(x^*, y^*, z^*; \mathbf{p}^*, \mathcal{N})$ with $x^* = x - Np_x^* + Np_x$, $y^* = y - Mp_y^* + Mp_y$, and $z^* = z - Lp_z^* + Lp_z$. Whenever (x^*, y^*, z^*) is situated outside the grid, we set $W^*(x, y, z; \mathbf{p}^*, \mathcal{N}) = 0$. From now on in this paper, we will set $\mathbf{p}^* = (0.5, 0.5, 0.5)$ due to the largest coverage and round shape of the corresponding weight function, which is also critical for rotational invariance. In order to preserve the round shape of the weight function, we will always use a cubical grid, i.e., $N = M = L$ or set $\mathcal{N} = (N, N, N)$. Hence, the center of the grid will be at $C = (N/2, N/2, N/2)$. In order to shift the centroid of the auxiliary image $\tilde{f}(x, y, z)$ to the grid center C , \tilde{v}_{ijk} in (26) is modified to

fied to

$$\begin{aligned} \tilde{\lambda}_{ijk} &= (\tilde{M}_{000})^{-1} \sum_{x=0}^N \sum_{y=0}^N \sum_{z=0}^N \tilde{f}(x, y, z) \left[\tilde{\phi}_1(x, y, z) / (\tilde{M}_{000})^{\frac{1}{3}} + N/2 \right]^i \\ &\cdot \left[\tilde{\phi}_2(x, y, z) / (\tilde{M}_{000})^{\frac{1}{3}} + N/2 \right]^j \left[\tilde{\phi}_3(x, y, z) / (\tilde{M}_{000})^{\frac{1}{3}} + N/2 \right]^k. \end{aligned} \quad (27)$$

Using the binomial expansion and rearranging sums, $\tilde{\lambda}_{ijk}$ can be written as

$$\begin{aligned} \tilde{\lambda}_{ijk} &= \sum_{r=0}^i \sum_{s=0}^j \sum_{t=0}^k \binom{i}{r} \binom{j}{s} \binom{k}{t} (N/2)^{i+j+k-r-s-t} (\tilde{M}_{000})^{-\frac{r+s+t}{3}-1} \\ &\cdot \sum_{x=0}^N \sum_{y=0}^N \sum_{z=0}^N \tilde{f}(x, y, z) (\tilde{\phi}_1(x, y, z))^r (\tilde{\phi}_2(x, y, z))^s (\tilde{\phi}_3(x, y, z))^t. \end{aligned} \quad (28)$$

Thus,

$$\tilde{\lambda}_{ijk} = \sum_{r=0}^i \sum_{s=0}^j \sum_{t=0}^k \binom{i}{r} \binom{j}{s} \binom{k}{t} (N/2)^{i+j+k-r-s-t} \tilde{v}_{rst} \quad (29)$$

is a linear combination of invariants \tilde{v}_{rst} where

$$\begin{aligned} \tilde{v}_{rst} &= (\tilde{M}_{000})^{-\frac{r+s+t}{3}-1} \sum_{x=0}^N \sum_{y=0}^N \sum_{z=0}^N \tilde{f}(x, y, z) \\ &\cdot (\tilde{\phi}_1(x, y, z))^r \cdot (\tilde{\phi}_2(x, y, z))^s \cdot (\tilde{\phi}_3(x, y, z))^t, \end{aligned} \quad (30)$$

for $r = 0, \dots, i$, $s = 0, \dots, j$ and $t = 0, \dots, k$. Therefore, these new geometric moments are rotation, translation, and scale invariant, and yet centered at the point $(N/2, N/2, N/2)$. If we set $p_x = 0.5$, $p_y = 0.5$, and $p_z = 0.5$ in (19) and replace \tilde{M}_{ijk} by their invariant counterparts $\tilde{\lambda}_{ijk}$ from (27), we obtain a new set of moments which are invariant under rotation, translation, and scaling, i.e.,

$$\begin{aligned} \tilde{Q}_{nml} &= [\Omega(n, m, l; \mathbf{p}^*, \mathcal{N})]^{-\frac{1}{2}} \\ &\times \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l a_{i,n,0.5,N} a_{j,m,0.5,N} a_{k,l,0.5,N} \tilde{\lambda}_{ijk}, \end{aligned} \quad (31)$$

where $\mathbf{p}^* = (0.5, 0.5, 0.5)$ and $\mathcal{N} = (N, N, N)$. This new set of moments will be called *3D Krawtchouk descriptors* and referred as 3DKDs in the rest of the paper. Note that 3DKDs still depend on the number N , so it is important to use the same grid size while performing the local comparison of 3D images.

5. Computation of descriptors

The descriptors defined in (31) requires precomputation of $\tilde{\lambda}_{ijk}$ in (29) which is a linear combination of geometric moments \tilde{v}_{rst} given by (30). Notice that computation of \tilde{v}_{rst} requires exponentiation of three 3D functions $\tilde{\phi}_1$, $\tilde{\phi}_2$, and $\tilde{\phi}_3$, and then element-wise multiplication of four 3D functions, and finally summation over three variables. Thus, direct computation of \tilde{v}_{rst} in (30) can be quite time-consuming, especially when the grid size $\mathcal{N} = (N, N, N)$ is large. To save computational time, we will first separate the z variable from the x and y variables by rewriting $(\tilde{\phi}_1(x, y, z))^r$, $(\tilde{\phi}_2(x, y, z))^s$, and $(\tilde{\phi}_3(x, y, z))^t$ using the binomial expansion as follows:

$$(\tilde{\phi}_1(x, y, z))^r = \sum_{\varepsilon_1=0}^r \binom{r}{\varepsilon_1} (\tilde{A}_1(x, y))^{\varepsilon_1} (\tilde{D}_1(z))^{\varepsilon_1}, \quad (32)$$

$$(\tilde{\phi}_2(x, y, z))^s = \sum_{\varepsilon_2=0}^s \binom{s}{\varepsilon_2} (\tilde{A}_2(x, y))^{\varepsilon_2} (\tilde{D}_2(z))^{\varepsilon_2}, \quad (33)$$

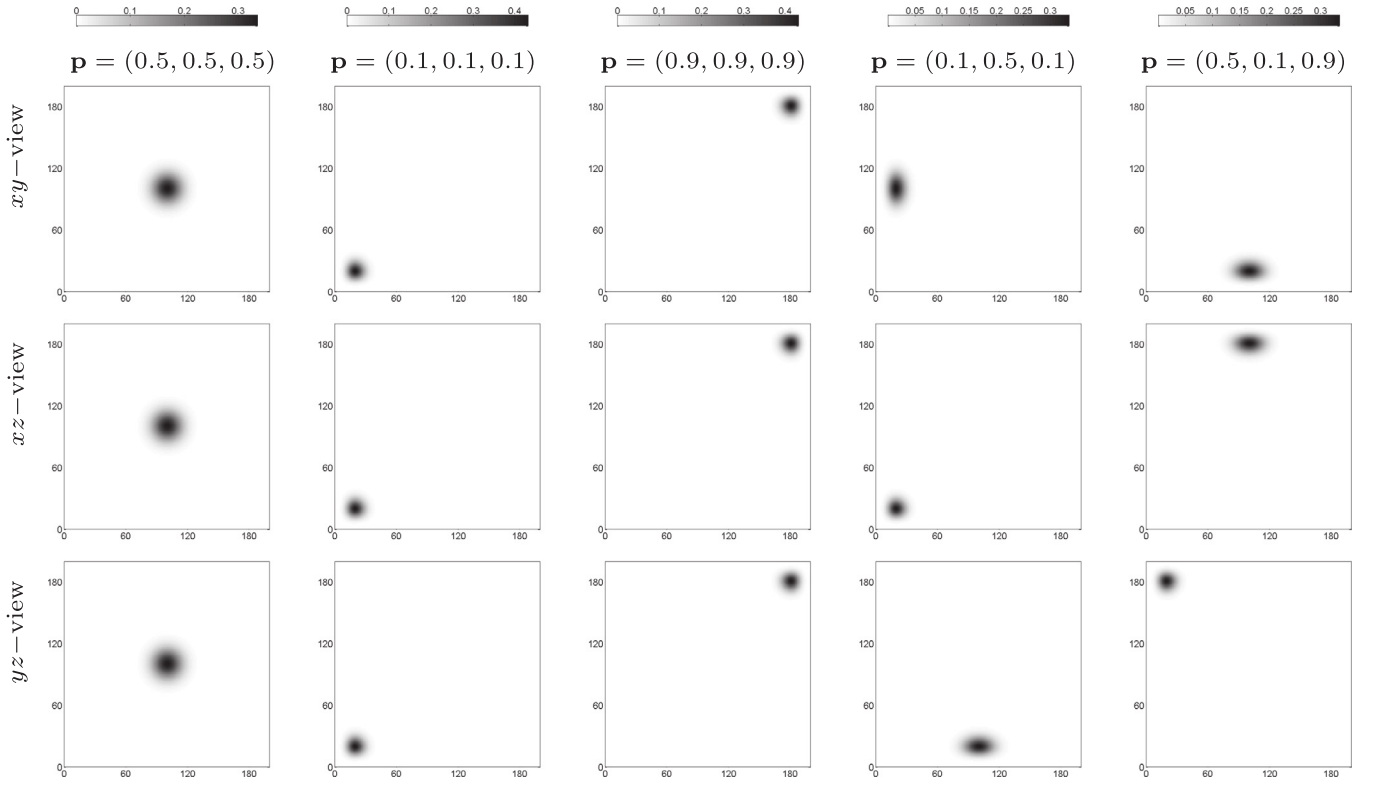


Fig. 2. 2D views of the square root of the weight function in (18), flattened by summing slices along each dimension. The summations are performed along z , y , and x -axis, respectively from top to bottom. Density plots are shown for five different choices of $\mathbf{p} = (p_x, p_y, p_z)$ triplets. $N = (200, 200, 200)$ in all cases. The gray scale colormaps at the top show the intensity of plots in each column.

$$(\tilde{\phi}_3(x, y, z))^t = \sum_{\varepsilon_3=0}^t \binom{t}{\varepsilon_3} (\tilde{A}_3(x, y))^{t-\varepsilon_3} (\tilde{D}_3(z))^{\varepsilon_3}, \quad (34)$$

where $\tilde{A}_\tau(x, y) = \tilde{R}_{\tau 1}(x - \tilde{x}) + \tilde{R}_{\tau 2}(y - \tilde{y})$ and $\tilde{D}_\tau(z) = \tilde{R}_{\tau 3}(z - \tilde{z})$ with $\tau = 1, 2, 3$.

Hence, given a voxelized 3D surface function $f(x, y, z)$, a triplet $\mathbf{p} = (p_x, p_y, p_z)$ corresponding to a point on the surface, and the surface normal at that point, an efficient computation of 3DKDs in (31) can be performed in the following steps.

1. Compute the auxiliary image $\tilde{f}(x, y, z)$ using (18).
2. Compute the function

$$\tilde{T}(x, y; \varepsilon_1, \varepsilon_2, \varepsilon_3) = \sum_{z=0}^N \tilde{f}(x, y, z) (\tilde{D}_1(z))^{\varepsilon_1} (\tilde{D}_2(z))^{\varepsilon_2} (\tilde{D}_3(z))^{\varepsilon_3} \quad (35)$$

for $0 \leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \leq 5$. Note that the z variable is eliminated in this step and the rest of the computations will be carried out in the x and y variables only.

3. Compute

$$\tilde{T}_3(x, y; \varepsilon_1, \varepsilon_2, t) = \sum_{\varepsilon_3=0}^t \binom{t}{\varepsilon_3} (\tilde{A}_3(x, y))^{t-\varepsilon_3} \tilde{T}(x, y; \varepsilon_1, \varepsilon_2, \varepsilon_3) \quad (36)$$

for $0 \leq \varepsilon_1 + \varepsilon_2 + t \leq 5$.

4. Compute

$$\tilde{T}_2(x, y; \varepsilon_1, s, t) = \sum_{\varepsilon_2=0}^s \binom{s}{\varepsilon_2} (\tilde{A}_2(x, y))^{s-\varepsilon_2} \tilde{T}_3(x, y; \varepsilon_1, \varepsilon_2, t) \quad (37)$$

for $0 \leq \varepsilon_1 + s + t \leq 5$.

5. Compute

$$\tilde{T}_1(x, y; r, s, t) = \sum_{\varepsilon_1=0}^r \binom{r}{\varepsilon_1} (\tilde{A}_1(x, y))^{r-\varepsilon_1} \tilde{T}_2(x, y; \varepsilon_1, s, t) \quad (38)$$

for $0 \leq r + s + t \leq 5$.

6. Compute

$$\tilde{v}_{rst} = (\tilde{M}_{000})^{-\frac{r+s+t}{3}-1} \sum_{x=0}^N \sum_{y=0}^N \tilde{T}_1(x, y; r, s, t) \quad (39)$$

for $0 \leq r + s + t \leq 5$.

Finally, we perform another couple of steps to compute 3DKDs of the order up to 5.

7. Compute $\tilde{\lambda}_{ijk}$ in (29) for $0 \leq i + j + k \leq 5$ using \tilde{v}_{rst} from step 6.
8. Compute \tilde{Q}_{nml} in (31) for $0 \leq n + m + l \leq 5$ using $\tilde{\lambda}_{ijk}$ from step 7.

Separating the z variable as described above makes the computations very efficient. The computational performance of the algorithm will be discussed at the end of Section 6.

6. Results and discussion

In this section, we test the local discriminative performance of 3DKDs. We use three feature vectors of descriptors

$$\begin{aligned} K_3 &= \{\tilde{Q}_{nml} : 0 \leq n + m + l \leq 3\} \\ K_4 &= \{\tilde{Q}_{nml} : 0 \leq n + m + l \leq 4\} \\ K_5 &= \{\tilde{Q}_{nml} : 0 \leq n + m + l \leq 5\} \end{aligned} \quad (40)$$

namely, the descriptors of order up to 3, 4, and 5, that are computed using the algorithm summarized in Section 5. The number

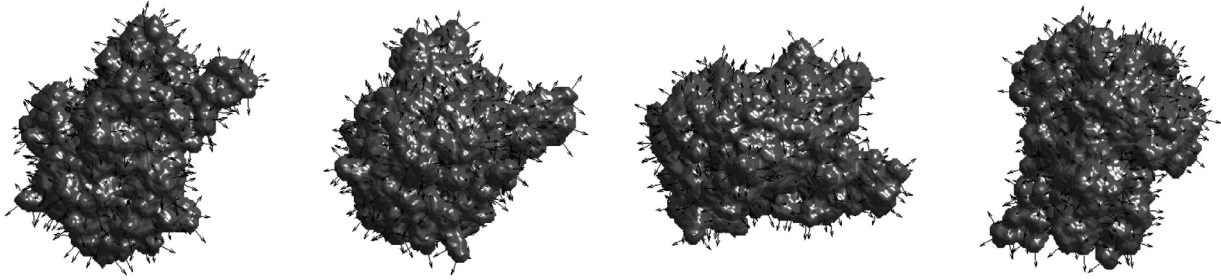


Fig. 3. Query protein surface (1gco.pdb, left) and three target surfaces obtained from the query protein by rotating it using different rotation matrices. The vertex normals on each surface are also demonstrated. Each protein surface is placed in a voxel grid of size 170^3 .

of elements in K_3 , K_4 , and K_5 is 20, 35, and 56, respectively. The seven descriptors (\tilde{Q}_{000} , \tilde{Q}_{100} , \tilde{Q}_{010} , \tilde{Q}_{001} , \tilde{Q}_{011} , \tilde{Q}_{101} , \tilde{Q}_{110}) involved in the normalization process were removed, because they take a constant value irrespective of the 3D patch we are working with. Thus, the actual number of elements in K_3 , K_4 , and K_5 is 13, 28, and 49, respectively.

As the similarity measure, we use the (squared) Euclidean distance between feature vectors of the same size, namely

$$d(v^q, v^t) = \sum_{i=1}^T (v_i^q - v_i^t)^2 \quad (41)$$

where v^q and v^t are the feature vectors for a query and a target object, respectively, to be compared, and T is dimension of the feature vector.

6.1. Local comparison Test I

We first test 3DKDs for comparison of local patches on protein surfaces. We have downloaded the PDB file of a protein (PDB ID: 1GCO) from PDB [28], generated a voxelized surface of the protein using 3D-Surfer [29], and placed it in a voxel grid of size 170^3 . Using the GETPOINTS subroutine of LZerD docking suite [30], we have specified 1608 vertex points on the protein surface as well as the normal vectors at these points pointing outside the surface. We have then reduced the number of points to 500 so that the minimal distance between neighboring points is more than 3 Å, i.e., 5 unit voxels (See Fig. 3, left). Each of these points and the normal vectors were used to represent the center of a patch on the surface. These normal vectors were also used for determining the unique rotation matrix \tilde{R} as described in Section 4. The 3DKDs corresponding to each patch were computed and stored as the query dataset. Then the protein structure and the 500 surface points were rotated so that each patch moves to a different location and orientation (see Fig. 3). We have used three different rotation matrices S_1 , S_2 , and S_3 and obtained three target sets. S_1 rotates the query protein 90° about the x -axis, S_2 rotates the query protein first 90° about the x -axis, then 45° about the y -axis, and then 30° about the z -axis, and S_3 is a randomly generated rotation matrix. Using S_1 , S_2 , and S_3 , the 3DKDs corresponding to each patch in the target sets were computed and stored as Target 1, Target 2, and Target 3, respectively.

Each of the 500 feature vectors in the first set was queried and compared with the 500 feature vectors in a target set. The results were ranked using the Euclidean distance. If the same patch as the query ranks top (Top 1), it was labeled as “correctly classified”. Otherwise, we look at the top five results in the ranking (Top 5), or the top ten results (Top 10). We have then calculated the recognition accuracies as

$$r = \frac{\text{Number of “correctly classified” queries}}{\text{Total number of query inputs}} \quad (42)$$

whose denominator is equal to 500.

We have obtained very high recognition accuracies ranging between 93.6% and 97% for the Top 1 case (see Table 1). The recognition accuracies stay between 97.6% and 99% for the Top 5 case. For the Top 10 case, it reaches up to between 98% and 100%. The results show that the 3DKDs are successful in local patch comparison for this small problem. For the Top 1 case, K_5 performs best with all three targets. K_4 gives the highest recognition accuracies for all targets when our classification criterion is relaxed to Top 5 or Top 10.

6.2. Local comparison Test II

We also test the local performance of 3DKDs on a more difficult problem as follows. The surface grid, vertex points and normal vectors of the query protein, namely 1608 query patches, are generated as before. For the three target sets, the target proteins and their voxelized surface grids were also generated as before. However, the surface points and vertex normals of the target sets were redistributed using GETPOINTS so that we had a new set of points and normal vectors different than the ones obtained before. This occurs due to a randomized subroutine of the program. This time, we have obtained 1557, 1557, and 1546 such points in Target 1, Target 2, and Target 3, respectively. For each point and the corresponding surface patch, the 3DKDs were computed and stored. The number of query patches is then further reduced so that each query patch remains with at least five neighboring target patches when both query and target proteins are considered superimposed. Here, by a neighbor, we mean that the physical distance between centers of the query patch and the target patch is less than 3 Å (5 unit voxels). Hence, the number of query patches to be compared with those in the target sets now depends on each individual target protein, and we obtain 737, 736, and 701 such query patches corresponding to Target 1, Target 2, and Target 3, respectively.

Each of the feature vectors in the query set was compared with those from the corresponding target set. The results are ranked using the Euclidean distance and collected for three target sets. If one of the five neighbors of the query patch ranks within top k , it is labeled as “correctly classified”. We computed recognition accuracies for $k = 1, \dots, 16$ where $k = 16$ approximately corresponds to the top 1-percentile.

The results are shown in Fig. 4. Compared to the results of the Test I (Table 1), the accuracies dropped about 25% points in the results corresponding to $k = 1$ (Top 1 results). When we only look at the top result ($k = 1$) in the rankings, K_5 is the most successful set of descriptors. The performances of K_3 and K_5 are comparable and higher than K_4 for almost all k values in all three cases. In Test II, we have actually tested how each 3DKD vector is tolerant to slight changes in the location of patch centers. From Fig. 4, it is clear that the descriptors in K_4 , in particular, the 4th order descriptors that are not in K_3 , are quite sensitive to such patch center shifts. This deficiency appears to be corrected in K_5 with the addition of 5th order descriptors.

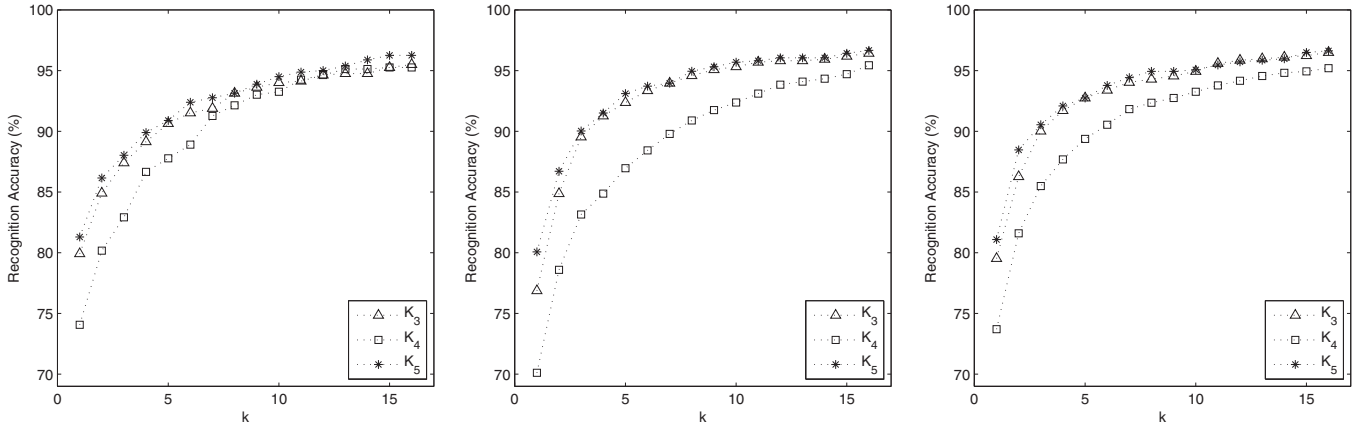


Fig. 4. Test II – Recognition accuracies (%) vs. the number of closest patches in ranking (k) for S_1 , S_2 , and S_3 from left to right, respectively.

Table 1

Test I – Recognition accuracies (%).

3DKD vector	Target 1			Target 2			Target 3		
	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10	Top 1	Top 5	Top 10
K_3	96.8	98.2	98.6	95.4	97.6	98.0	95.8	98.4	98.6
K_4	96.2	99.0	100.0	93.6	98.0	98.8	94.8	98.4	99.0
K_5	97.0	98.0	98.6	95.6	97.6	98.0	96.0	98.2	98.6

6.3. Comparison with other descriptors

Next, we compare the discriminative performances of 3DKDs with 3D Gaussian-Hermite Moments (3DGHMs) [31] and 3D Zernike descriptors (3DZDs) [18] computed for local protein surface patches. For these comparisons, we used the query and Target 1 proteins from Test I, both placed in a voxel grid of size 170^3 . To compute 3DGHMs, we first computed and analyzed the one-dimensional Gaussian-Hermite polynomials $\hat{H}_n(x/\sigma)$ using the recursive algorithm provided in [32]. The Gaussian-Hermite polynomials have a scale parameter σ that controls the size of the reconstructed region in an image. To choose the correct σ value, we compared the graph of the weighted Krawtchouk polynomial of degree 0, namely $\tilde{K}_0(x; p, N-1)$ with $p = .5$ and $N = 170$, and the graph of Gaussian-Hermite polynomial of degree 0, namely $\hat{H}_0(x/\sigma)$, for different σ values. The Gaussian-Hermite polynomial is defined on the interval $[-1, 1]$ sampled by the same number points ($N = 170$). We found that for $\sigma = 0.1083$, the widths of the two curves, i.e., the intervals that are not mapped to zero by both functions, were identical. This way we ensure that 3DKDs and 3DGHMs use the same local information. After choosing the σ value, we multiplied the voxelized surface of the protein by 3D Gaussian as follows:

$$\hat{f}(x, y, z) = e^{-[(x-x_s)^2 + (y-y_s)^2 + (z-z_s)^2]/(2\sigma^2)} f(x, y, z), \quad (43)$$

where (x_s, y_s, z_s) are the coordinates of the surface point mapped to the domain $[-1, 1] \times [-1, 1] \times [-1, 1]$. We then computed the center of mass of $\hat{f}(x, y, z)$ and the unique rotation matrix \hat{R} using the formulas provided in Section 4. By using the matrix \hat{R} , we achieved invariance to rotation by means of principle axes normalization. We finally computed the 3D Gaussian-Hermite moments of the normalized auxiliary surface $\hat{f}(x, y, z)$ using the method outlined in [31].

To compute 3DZDs, we used the same query and Target 1 proteins from Test I. The 3D Zernike descriptors are not able to extract local features from an object directly as 3DKDs do; yet Zernike functions are defined as continuous functions whose domain is the unit ball. For this reason, we first mapped each patch into the unit ball by considering each surface point as the patch center and plac-

ing a sphere of 6 Å radius around each point. 6 Å here corresponds to 10 unit voxels; the size of a typical ligand-binding pocket on a protein surface. The patch cut from the surface is then mapped into the unit ball so that the center of mass of the cropped patch is placed at the coordinate origin. The geometric moments and hence the 3D Zernike moments of the order up to 12 and 15 (3DZD_12 and 3DZD_15, respectively) are computed for each patch using the algorithm provided in [33]. Finally, the rotation invariant 3DZDs are computed from these moments using the formula provided in [18]. The above work is repeated using a larger sphere for each patch (9 Å radius = 15 unit voxels) since there are some well-known ligands binding to proteins by forming larger pockets. In order to make the comparison fairer, we have recomputed 3DKDs (K_5) and 3DGHMs using the same surface function but taking on zero value at voxels that remain outside the spheres used above. By this occlusion, we ensure that 3DKDs and 3DGHMs both use the same local information as 3DZDs do.

The recognition accuracies for 3DKDs (K_5), 3DGHMs, and 3DZDs using different order of descriptors and patch radii are shown in Table 2. The results show that our method was clearly better than 3DZDs and 3DGHMs in local feature extraction, even when 3DZDs are allowed to use more invariants. The Top 1 prediction in our method is 90%, whereas it is only between 7.4%–8.8% in 3DZDs and 13% in 3DGHMs. Increasing the order of invariants to 15 (i.e., the vector size to 72) did not significantly improve the performance of 3DZDs. All methods gave higher recognition accuracies when the patch size was increased from 6 to 9 Å but the performance increase in 3DGHMs is remarkable. The recognition accuracy for 3DGHMs increased from 13% to 91%, which makes 3DGHMs comparable 3DKDs. However, 3DKDs outperformed both 3DZDs and 3DGHMs in all cases shown in Table 2.

6.4. Binding ligand prediction

Finally, we test 3DKDs on binding ligand prediction for proteins, which is one of the important tasks in bioinformatics as it addresses a central question in molecular biology, protein function [19,20], and has real-life application in computational drug

Table 2Comparison of K_5 with 3D Gaussian-Hermite Moments (3DGHMs) and 3D Zernike Descriptors (3DZDs).

Feature vector	Vector size	Patch radius					
		6 Å (10 voxels)			9 Å (15 voxels)		
		Top 1 (%)	Top 5 (%)	Top 10 (%)	Top 1 (%)	Top 5 (%)	Top 10 (%)
3DKD- K_5	49	90.0	97.2	98.0	95.6	97.8	98.6
3DGHM	56	13.0	38.6	54.4	91.0	96.6	97.2
3DZD_12	49	7.4	20.0	29.0	26.0	46.2	58.0
3DZD_15	72	8.8	18.6	28.2	26.4	50.4	61.8

Table 3

The ligand pocket benchmark dataset.

Binding ligand molecule	AMP	ATP	FAD	FMN	FUC	GAL	GLC	HEM	MAN	NAD	PLM
Average size (Å)	6.4	7.6	11.9	7.4	3.8	4.2	3.9	8.7	4.1	10.4	8.7
Number of query pockets (497)	44	44	82	49	7	15	27	146	33	39	11
Number of patches (11039)	619	840	2663	978	49	125	195	4089	248	1095	138
Average number of patches	14.1	19.1	32.5	20.0	7.0	8.3	7.2	28.0	7.5	28.1	12.5

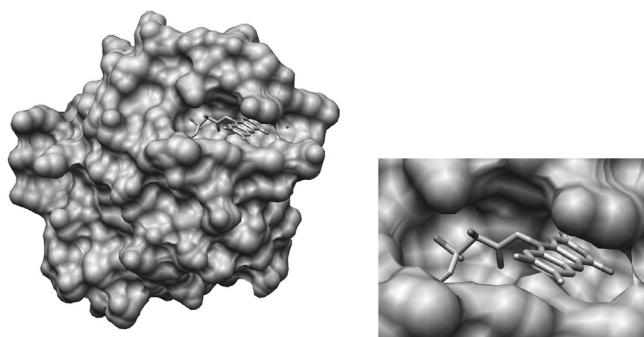


Fig. 5. An example of a ligand binding pocket on a protein surface. Receptor protein, left: FMN-binding domain of human cytochrome P450 reductase, PDB ID: 1B1C. Binding ligand, right: FMN (Flavin mononucleotide). Images were rendered with UCSF Chimera [34].

design [21]. Ligand molecules that bind to a local surface region in a protein can be predicted by finding similar local pockets of known binding ligands in the structure database. An example of a ligand binding pocket is demonstrated in Fig. 5. In order to test 3DKDs on binding ligand prediction, we have constructed a benchmark dataset of 463 proteins already known to bind to 11 different ligands. See Fig. 6 for these ligand structures. For each protein in the dataset, surface vertices and normals were generated using the GETPOINTS subroutine as in the previous tests. In the dataset, we also have the PDB file of each protein, which not only contains the coordinates of all atoms in the protein, but also those that belong to the bound ligand. For each atom in the ligand structure, we have selected the nearest surface vertex on the protein and annotated it with the bound ligand type. The collection of all such points and the 3DKDs of the patches around these points were all stored in a ‘patch database’ of 11,039 patches together with their annotations of binding ligands. Thus, for each query pocket, a database search was performed for each of the patches in the query pocket, and a patch score was assigned to each patch based on the database rankings.

We query total 497 pockets for this task. See Table 3 for the number of query pockets for each ligand type. In Table 3, we have also listed the number of patches that corresponds to a ligand type. The number of patches associated with each pocket depends on the size of the pocket. By the average size of a pocket, we mean the radius of a sphere that encapsulates all surface vertices annotated with that ligand type. As can be seen from Table 3, there is a high correlation between the size of a typical pocket and the aver-

age number of patches included in that pocket. For each patch in the query pocket, we compute a patch score based on the following formula:

$$\text{Patch_score}(p, F, k) = \sum_{i=1}^k (\delta_{l(i),F} \log(n/i)) \cdot \frac{\sum_{i=1}^k \delta_{l(i),F}}{\sum_{i=1}^n \delta_{l(i),F}}, \quad (44)$$

where $l(i)$ denotes the ligand type (e.g. AMP, ATP, etc.) of the i -th closest patch to the query, n is the number of patches in the patch database, and the function $\delta_{l(i),F}$ is equal to 1 if i -th patch is of type F , and 0 otherwise. This formula is used before as a pocket score for binding-ligand prediction in [19]. The first term in (44) is to only involve k closest patches in the patch database to each patch from the query pocket, assigning a higher score to a patch with a higher rank. The second term is to normalize the score by the number of patches of the same type F included in the patch database so that the results are not biased in favor of highly populated ligand types in the database.

For each of the patches that belong to the query pocket and k values from 1 to 300, patch scores are computed and then summed to obtain a unique pocket score for each ligand type F as follows:

$$\text{Pocket_score}(P, F, k) = \sum_{j=1}^{N_p} \text{Patch_score}(p_j, F, k), \quad (45)$$

where p_j , $j = 1, \dots, N_p$ are the patches within the pocket P , and N_p is the number of such patches. Thus, a pocket has a certain number of patches (see the last row of Table 3) and the score for a query pocket P for a ligand type F is computed as the sum of the score of each patch for the ligand F .

For each query pocket, we computed the pocket score for each of the 11 ligand type in the database. The ligand with the highest Pocket_score was predicted to bind to the query pocket. We then compared these 11 scores and looked at the largest one (Top 1) and the largest three (Top 3) to obtain the number of successful predictions (see Table 4). For each ligand type, the number of successful cases was divided by the number of query pockets of that type in the pocket database to obtain prediction accuracies. For each prediction, the results are shown for the k value which maximizes the average prediction accuracy. It turned out that the average prediction accuracy is maximized for small values of k as shown in Table 4.

When we look at the average prediction accuracies, K_5 performs best (with 41.2% correct prediction) among the 3DKDs. For the individual ligand types, K_3 performs best only for FUC and HEM, while K_4 gives the highest accuracies for AMP, ATP, FMN, FUC, GLC,

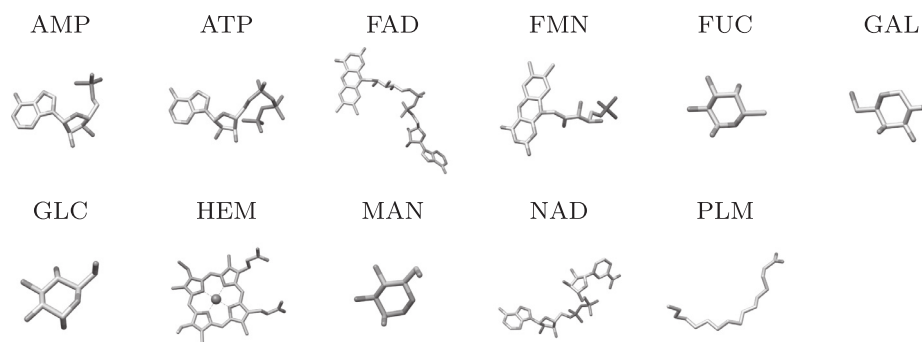


Fig. 6. Eleven ligand structures known to bind to proteins in the benchmark dataset. The ligand structures are shown for adenosine monophosphate (AMP), adenosine triphosphate (ATP), flavin adenine dinucleotide (FAD), flavin mononucleotide (FMN), fucose (FUC), galactose (GAL), glucose (GLC), heme (HEM), mannose (MAN), nicotinamide adenine dinucleotide (NAD), and palmitic acid (PLM). Images were rendered with UCSF Chimera [34].

Table 4

Binding ligand prediction accuracies (%) using 3DKDs and comparison with other methods.

Rank	Descriptor	<i>k</i>	AMP	ATP	FAD	FMN	FUC	GAL	GLC	HEM	MAN	NAD	PLM	Average
Top 1	3DKD- K_3	6	11.4	20.5	48.8	20.4	85.7	20.0	25.9	91.1	45.5	33.3	18.2	38.2
	3DKD- K_4	2	15.9	27.3	32.9	22.4	85.7	26.7	29.6	80.1	48.5	30.8	27.3	38.8
	3DKD- K_5	3	13.6	27.3	48.8	22.4	85.7	26.7	22.2	87.0	48.5	43.6	27.3	41.2
	Pocket-Surfer (3DZD+PS)		0.0	21.4	50.0	0.0	21.4	38.9	0.0	87.5	38.9	60.0	92.3	37.3
	Patch-Surfer2.0 (3DZD)		0.0	0.0	8.5	0.0	0.0	0.0	7.4	98.6	36.4	2.6	0.0	14.0
	Patch-Surfer2.0 (3DZD+GD)		34.1	20.4	78.0	42.8	0.0	6.7	18.5	90.4	54.5	33.3	18.2	36.1
Top 3	Random		8.6	8.6	16.2	9.6	1.3	2.8	5.3	29.2	6.4	7.6	2.0	8.9
	3DKD- K_3	19	34.1	50.0	90.2	38.8	85.7	40.0	51.9	98.6	75.8	76.9	36.4	61.7
	3DKD- K_4	5	47.7	54.5	73.2	40.8	85.7	46.7	55.6	96.6	84.8	64.1	54.5	64.0
	3DKD- K_5	4	45.5	56.8	76.8	49.0	85.7	66.7	51.9	95.9	78.8	71.8	36.4	65.0
	Pocket-Surfer (3DZD+PS)		77.8	100.0	90.0	16.7	85.7	80.6	80.0	100.0	72.2	100.0	92.3	81.4
	Patch-Surfer2.0 (3DZD)		11.4	15.9	92.7	38.8	14.3	6.7	37.0	100.0	63.6	84.6	0.0	42.3
	Patch-Surfer2.0 (3DZD+GD)		61.4	59.1	90.2	79.6	0.0	80.0	88.9	96.6	93.9	94.9	45.4	71.8
	Random		23.8	23.9	41.4	26.3	3.7	8.4	15.2	64.8	18.1	21.5	5.8	23.0

Table 5

Confusion table for K_5 .

	AMP	ATP	FAD	FMN	FUC	GAL	GLC	HEM	MAN	NAD	PLM
AMP	6	7	6	4	0	3	1	9	1	5	2
ATP	5	12	1	0	3	2	4	3	1	7	6
FAD	1	6	40	1	3	1	6	12	2	6	4
FMN	3	1	5	11	3	1	5	9	1	5	5
FUC	0	0	0	0	6	0	1	0	0	0	0
GAL	0	1	1	1	3	4	4	0	0	0	1
GLC	2	0	2	3	5	3	6	1	3	2	0
HEM	2	3	1	1	0	1	1	127	0	0	10
MAN	0	1	1	1	9	2	1	0	16	1	1
NAD	3	5	4	1	1	0	3	3	0	17	2
PLM	0	1	2	0	0	1	1	2	1	0	3

and MAN. Among the 3DKDs, K_5 is the one giving the best average prediction and highest prediction accuracies (including ties) for almost all ligand types except for AMP, GLC, and HEM. In Table 5, we take a closer look at the results for K_5 by forming the confusion matrix with the true positives being along the diagonal. According to Table 5, seven AMP queries are predicted as ATP, and similarly, five ATP queries are predicted as AMP. This may be excused due to similar shapes of AMP and ATP (only differing from each other by two phosphate groups.) Other similar shaped pairs can be observed among sugar molecules FUC, GAL, GLC, and MAN. FUC as a query gives one false positive (GLC), while GAL as a query is confused seven times with other sugar molecules (three times with FUC and four times with GLC). GLC as a query is confused

five times with FUC, three times with GAL, and three times with MAN. Similarly, MAN as a query gives twelve false positives from the sugar group (FUC, nine times; GAL, two times; and GLC once). Thus, the confusion between ligands by 3DKDs is quite reasonable, capturing similarity of binding pockets of similar ligand molecules. In the results for Top 3, K_5 is still the one showing the best average performance among the 3DKDs, and giving the highest average accuracy and comparable results for almost all ligand types.

In Table 4, we also provide a comparison of our approach with a former binding prediction method named Pocket-Surfer [23]. In Pocket-Surfer, 3DZDs are used as shape descriptors of a pocket. In addition to shape information, the pocket size is also utilized in Pocket-Surfer as a classification measure by an optimal weighted

Table 6Number of query pockets and prediction accuracies (%) of holo and apo proteins using 3DKD- K_5 .

Number of Pockets	Form	AMP	ATP	FAD	FMN	FUC	GAL	GLC	HEM	MAN	NAD	Total
	Holo	12	20	2	2	3	8	12	6	7	15	87
	Apo	7	11	2	2	1	6	8	4	4	7	52
Rank	Form	AMP	ATP	FAD	FMN	FUC	GAL	GLC	HEM	MAN	NAD	Average
Top 1	Holo	42.9	18.2	0.0	0.0	0.0	16.7	12.5	0.0	75.0	57.1	22.2
	Apo	50.0	50.0	0.0	0.0	100.0	50.0	33.3	50.0	71.4	86.7	49.1
Top 3	Holo	71.4	81.8	0.0	0.0	0.0	50.0	50.0	25.0	75.0	85.7	43.9
	Apo	58.3	90.0	50.0	0.0	100.0	75.0	83.3	83.3	71.4	86.7	69.8

average of scores from both shape and pocket size. In the current work, we only used shape information from 3DKDs without employing the pocket size in the scoring functions. When Top 1 is the classification criterion, 3DKDs predict better than Pocket-Surfer for seven ligand types and the average prediction. With Top 3 classification, Pocket-Surfer performs better than 3DKDs (K_5) for eight (out of eleven) ligand types. We believe that this is partly due to the fact that 3DKDs employ shape information only, whereas Pocket-Surfer also used pocket size, which is often critical to distinguish ligands of the different sizes. We also show results with random prediction, in which each query pocket is scored based on a randomly shuffled pocket database, averaged over 3000 randomizations. It is clear that 3DKDs outperform the random prediction in all cases.

In addition to Pocket-Surfer, Patch-Surfer2.0 [20] was also benchmarked. Patch-Surfer2.0 is a surface patch-based binding ligand prediction program using 3DZDs. It implements several physicochemical features such as electrostatic potential and hydrophobicity. To make the comparison fairer, we only used shape information by turning off the physicochemical features. In Table 4, 3DKDs were compared with two variants of Patch-Surfer2.0. The first variant used only 3DZDs describing shape information (Patch-Surfer2.0 (3DZD)). The other one also considers the geodesic distance (GD) information between patch centers (Patch-Surfer2.0 (3DZD+GD)), which provides additional information of relative positions of patches in a pocket. When Top 1 accuracy was considered, 3DKDs performed on average better than Patch-Surfer2.0. In Top 3 prediction results, 3DKDs showed a better overall performance than Patch-Surfer2.0 (3DZD). Patch-Surfer2.0 improved its performance when GD was added as a feature, showing higher average accuracy over 3DKDs.

We also tested 3DKDs on apo form of the pockets. An apo form is a ligand-unbound conformation of a pocket and thus it may have a different shape from the corresponding holo form, a ligand-bound form. This test is to mimic a more realistic situation, where a ligand needs to be identified from an apo form of a target pocket. PDB files of apo forms of pockets of the 11 ligands were identified as follows: From PDB files of the 463 holo proteins, we extracted related PDB IDs from the records at REMARK 900 in the PDB files. This resulted in 87 pockets in apo-form. Apo-form structures for PLM were not found in PDB. To define binding sites of the apo structures, they were superimposed to their holo structure partners using TM-align [35]. Coordinates of the binding ligand in the holo structure were copied to the apo-form structure after the superimposition. The benchmark results using 3DKDs on the apo-form pockets using K_5 are given in Table 6.

Overall, prediction accuracies for the apo-form pockets were higher than the holo-form counterparts for both Top 1 and Top 3 prediction results. This is somewhat surprising because, in general, finding ligands for apo-forms is more difficult than holo-forms. But we observed the same trend in our earlier paper on Patch-Surfer2.0 [20]. Some related works that take flexibility of structures into account can be found in [36] using diffusion distances, and in [37] combining local and global shape descriptors.

Table 7Computational times (seconds) for 3DKDs (K_5).

	Laptop	PC
Descriptors	0.5789	0.2733
Pocket Score	0.4156	0.1112
Search	3.5763e-05	1.0907e-05

Table 7 shows the time taken for computing 3DKDs of order up to 5 (K_5) as outlined in steps (1)–(8) in Section 5. The programs were tested on two different platforms: a Windows laptop with i3 CPU of 2.53 GHz and 4 GB memory using MATLAB 2009b 64-bit, version 7.9, and on a Linux workstation with Xeon CPU of 3.60 GHz and 94 GB memory and using MATLAB 2016b 64-bit, version 9.1. On each computer, MATLAB is limited to a single computational thread to perform its jobs. The first row of Table 7 shows the average CPU times spent on computing 3DKDs of order up to 5, where the average is taken among 1608 local protein surface patches from Test I. The computation of 3DKDs can be done in half a second in a laptop, whereas it finishes twice faster on the PC. In the second row of Table 7, we also show the computational time from the binding prediction test, assuming that the 3DKDs of all 11,039 patches in the patch database are precomputed and stored. Given a typical query pocket, the time it takes to compute the pocket scores in (45) for 11 ligand types and $k = 1, 2, \dots, 10$ is about 0.4 s in a laptop, and 3.7 times faster in the second platform.

7. Conclusion

In this paper, we have developed a novel set of local descriptors, three-dimensional Krawtchouk descriptors (3DKDs), for identification and comparison of local regions of 3D voxelized oriented surfaces. Our approach is based on 3D Krawtchouk moments. While obtaining the rotation, size, and position independent invariants, we have preserved the critical ability of Krawtchouk moments to extract local features of a 3D surface from any region-of-interest. The locality property is due to the 3D weight function given in the definition of Krawtchouk polynomials. The weight contains three parameters p_x , p_y , and p_z , shifting the center of the local surface region along the x , y , and z -axes, respectively. We have noticed that, for each triplet (p_x, p_y, p_z) , the coverage of the weight function is different, which prevents Krawtchouk moments from being translation and rotation invariant. To overcome these problems, we have computed the 3D weight function corresponding to (0.5,0.5,0.5) (center of the 3D grid) and used it for other local regions by translating the graph of the weight function as needed. To achieve rotational invariance, we utilized the surface normal at the vertex of the local surface region and computed the eigenvectors of the local inertia matrix. Among the eight possible different orientations, we have chosen the one positioning the vertex normal in a fixed octant in 3D space. We have also provided a detailed scheme for efficient computation of 3D Krawtchouk descriptors.

We have tested the discriminative performances of 3DKDs on three test problems. For each test, we have used K_3 , K_4 , and K_5 ,

namely 3DKDs of the order up to 3, 4, and 5, respectively. In the first test, the results have been comparable, while K_5 has the best recognition accuracies for predicting the top match. The second test demonstrated that K_5 , among the 3DKDs, is the most robust set of descriptors to small changes in patch location. We have also compared 3DKDs with 3DZDs and 3DGHMs. 3DKDs show better recognition accuracies than 3DZDs and 3DGHMs in all cases reported. As the third test, we have employed 3DKDs for prediction of ligand binding sites on protein surfaces. 3DKDs showed better average performance than Pocket-Surfer and Patch-Surfer2.0 when Top 1 prediction was considered. From the results of the second and the third tests, we conclude that 3DKDs are more sensitive than 3DZDs and 3DGHMs to subtle changes in shape. The results on the binding ligand prediction were obtained by only considering geometric shape information of protein surface. Therefore, further improvement is expected by integrating other features, such as the electrostatic potential or other physicochemical properties, to characterize protein surface regions, which is analogous to representing color images rather than black-and-white images of protein surfaces.

Acknowledgments

This work was supported by the National Science Foundation (DMS1614777 and DMS1614661). DK also acknowledges support from National Institutes of Health, (R01GM123055) and the National Science Foundation (CMMI1825941) and the Purdue Institute for Drug Discovery. The authors would like to thank Xiaolei Zhu for the help in voxelizing protein surfaces and Juan Esquivel-Rodriguez for the help in running LZerD software.

References

- [1] J. Žunić, P.L. Rosin, V. Ilić, Disconnectedness: a new moment invariant for multi-component shapes, *Pattern Recognit.* 78 (2018) 91–102.
- [2] R. Benouini, I. Batioua, K. Zenkour, A. Zah, H. El Fadili, H. Qjidaa, Fast and accurate computation of racah moment invariants for image classification, *Pattern Recognit.* 91 (2019) 100–110.
- [3] D.F. Atrévi, D. Vivet, F. Duculty, B. Emile, A very simple framework for 3d human poses estimation using a single 2d image: comparison of geometric moments descriptors, *Pattern Recognit.* 71 (2017) 389–401.
- [4] L. Luciano, A.B. Hamza, Deep learning with geodesic moments for 3d shape classification, *Pattern Recognit. Lett.* 105 (2018) 182–190.
- [5] M.-K. Hu, Visual pattern recognition by moment invariants, *IRE Trans. Inf. Theory* 8 (2) (1962) 179–187.
- [6] F.A. Sadjadi, E.L. Hall, Three-dimensional moment invariants, *IEEE Trans. Pattern Anal. Mach. Intell.* 2 (1980) 127–136.
- [7] X. Guo, Three dimensional moment invariants under rigid transformation, in: *International Conference on Computer Analysis of Images and Patterns*, Springer, 1993, pp. 518–522.
- [8] C.-H. Lo, H.-S. Don, 3-D moment forms: their construction and application to object identification and positioning, *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (10) (1989) 1053–1064.
- [9] J. Galvez, M. Canton, Normalization and shape recognition of three-dimensional objects by 3d moments, *Pattern Recognit.* 26 (5) (1993) 667–681.
- [10] A.G. Mamistvalov, n -Dimensional moment invariants and conceptual mathematical theory of recognition n -dimensional solids, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (8) (1998) 819–831.
- [11] J. Flusser, J. Boldys, B. Zitová, Moment forms invariant to rotation and blur in arbitrary number of dimensions, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (2) (2003) 234–246.
- [12] A. Sit, D. Kihara, Comparison of image patches using local moment invariants, *IEEE Trans. Image Process.* 23 (5) (2014) 2369–2379.
- [13] P.-T. Yap, R. Paramesran, S.-H. Ong, Image analysis by Krawtchouk moments, *IEEE Trans. Image Process.* 12 (11) (2003) 1367–1377.
- [14] A. Mademlis, A. Axenopoulos, P. Daras, D. Tzovaras, M.G. Strintzis, 3D content-based search based on 3D Krawtchouk moments, in: *Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, IEEE, 2006, pp. 743–749.
- [15] P. Xiang, C. Qihua, L. Zhi, Content-based 3D retrieval by Krawtchouk moments, in: *International Conference Image Analysis and Recognition*, Springer, 2006, pp. 217–224.
- [16] A. Mesbah, M. El Mallahi, Z. Lakhili, H. Qjidaa, A. Berrahou, Fast and accurate algorithm for 3D local object reconstruction using Krawtchouk moments, in: *2016 5th International Conference on Multimedia Computing and Systems (ICMCS)*, IEEE, 2016, pp. 1–6.
- [17] J. Flusser, T. Suk, B. Zitová, 2D and 3D image analysis by moments, John Wiley & Sons, 2016.
- [18] M. Novotni, R. Klein, 3D Zernike descriptors for content based shape retrieval, in: *Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, ACM, 2003, pp. 216–225.
- [19] L. Sael, D. Kihara, Detecting local ligand-binding site similarity in nonhomologous proteins by surface patch comparison, *Proteins Struct. Funct. Bioinf.* 80 (4) (2012) 1177–1195.
- [20] X. Zhu, Y. Xiong, D. Kihara, Large-scale binding ligand prediction by improved patch-based method Patch-Surfer2.0, *Bioinformatics* 31 (5) (2015) 707–713.
- [21] M. Rosenberg, A. Goldblum, Computational protein design: a novel path to future protein drugs, *Curr. Pharm. Des.* 12 (31) (2006) 3973–3997.
- [22] W.-H. Shin, C.W. Christoffer, J. Wang, D. Kihara, PL-Patchsurfer2: improved local surface matching-based virtual screening method that is tolerant to target and ligand structure variation, *J. Chem. Inf. Model.* 56 (9) (2016) 1676–1691.
- [23] R. Chikhi, L. Sael, D. Kihara, Real-time ligand binding pocket database search using local surface descriptors, *Proteins Struct. Funct. Bioinf.* 78 (9) (2010) 2007–2028.
- [24] E.G. Karakasi, G.A. Papakostas, D.E. Koulouriotis, V.D. Tourassis, Generalized dual hahn moment invariants, *Pattern Recognit.* 46 (7) (2013) 1998–2014.
- [25] G. Zhang, Z. Luo, B. Fu, B. Li, J. Liao, X. Fan, Z. Xi, A symmetry and bi-recursive algorithm of accurately computing krawtchouk moments, *Pattern Recognit. Lett.* 31 (7) (2010) 548–554.
- [26] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton shape benchmark, in: *Proceedings Shape Modeling Applications*, 2004, IEEE, 2004, pp. 167–178.
- [27] S. Patil, B. Ravi, Voxel-based representation, display and thickness analysis of intricate shapes, in: *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*, IEEE, 2005, pp. 6–pp.
- [28] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, The protein data bank, *Nucleic Acids Res.* 28 (1) (2000) 235–242.
- [29] D. La, J. Esquivel-Rodríguez, V. Venkatraman, B. Li, L. Sael, S. Ueng, S. Ahrendt, D. Kihara, 3D-SURFER: Software for high-throughput protein surface comparison and analysis, *Bioinformatics* 25 (21) (2009) 2843–2844.
- [30] J. Esquivel-Rodríguez, V. Filos-Gonzalez, B. Li, D. Kihara, Pairwise and Multimeric Protein-Protein Docking Using the LZerD Program Suite, in: *Protein Structure Prediction*, Springer, 2014, pp. 209–234.
- [31] B. Yang, T. Suk, M. Dai, J. Flusser, G.A. Papakostas, 2D and 3D image analysis by Gaussian-Hermite moments, *Moments Moment Invariants-TheoryAppl.* 1 (2014) 143–173.
- [32] B. Yang, M. Dai, Image reconstruction from continuous Gaussian-Hermite moments implemented by discrete algorithm, *Pattern Recognit.* 45 (4) (2012) 1602–1616.
- [33] A. Sit, J.C. Mitchell, G.N. Phillips, S.J. Wright, An extension of 3D Zernike moments for shape description and retrieval of maps defined in rectangular solids, *Comput. Math. Biophys.* 1 (2013) 75–89.
- [34] E.F. Pettersen, T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, T.E. Ferrin, UCSF Chimera visualization system for exploratory research and analysis, *J. Comput. Chem.* 25 (13) (2004) 1605–1612.
- [35] Y. Zhang, J. Skolnick, TM-Align: a protein structure alignment algorithm based on the TM-score, *Nucleic Acids Res.* 33 (7) (2005) 2302–2309.
- [36] Y.-S. Liu, Q. Li, G.-Q. Zheng, K. Ramani, W. Benjamin, Using diffusion distances for flexible molecular shape comparison, *BMC Bioinform.* 11 (1) (2010) 480.
- [37] A. Axenopoulos, D. Rafailidis, G. Papadopoulos, E.N. Houstis, P. Daras, Similarity search of flexible 3D molecules combining local and global shape descriptors, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 13 (5) (2016) 954–970.

Atilla Sit is currently an Assistant Professor with the Department of Mathematics and Statistics, Eastern Kentucky University, Richmond, KY, USA. He received the B.S. degree from Middle East Technical University, Ankara, Turkey, the M.S. degree from Bogazici University, Istanbul, Turkey, and the Ph.D. degree from Iowa State University, Ames, IA, USA, in 2001, 2005, and 2010, respectively. His current research interests include image analysis, orthogonal systems, special functions, protein structure determination and classification.

Woong-Hee Shin is currently a Post-Doctoral Researcher of the Kihara Laboratory with the Department of Biological Sciences, Purdue University, West Lafayette, IN, USA. He received the B.S. and Ph.D. degrees from the Seoul National University, South Korea, in 2008 and 2014, respectively. His current research topics are developing a structure-based virtual screening program using molecular surface descriptors and a ligand conformation sampling method using pseudo pockets.

Daisuke Kihara is currently a Professor with the Department of Biological Sciences and the Department of Computer Science, Purdue University, West Lafayette, IN, USA. He received the B.S. degree from the University of Tokyo, Japan, in 1994, and M.S. and Ph.D. degrees from Kyoto University, Japan, in 1996 and 1999, respectively. His research projects include biomolecular shape comparison, computational drug design, protein tertiary structure prediction, protein-protein docking. He is named Showalter University Faculty Scholar from Purdue University in 2013.