Computer Vision and Image Understanding 114 (2010) 865–886

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Subspace methods for retrieval of general 3D models

Helin Dutağacı [a,*], Bülent Sankur [a], Yücel Yemez [b]

[a] Department of Electrical-Electronics Engineering, Boğaziçi University, Bebek, 34342 Istanbul, Turkey
[b] Department of Computer Engineering, Koç University, Rumelifeneri, 34450 Istanbul, Turkey

ARTICLE INFO

Article history:
Received 24 November 2008
Accepted 9 May 2010
Available online 15 May 2010

Keywords:
3D shape retrieval
3D shape matching
Principal component analysis
Independent component analysis
Nonnegative matrix factorization
Distance transform

ABSTRACT

In statistical shape analysis, subspace methods such as PCA, ICA and NMF are commonplace, whereas they have not been adequately investigated for indexing and retrieval of generic 3D models. The main roadblock to the wider employment of these methods seems to be their sensitivity to alignment, itself an ambiguous task in the absence of common natural landmarks. We present a retrieval scheme based comparatively on three subspaces, PCA, ICA and NMF, extracted from the volumetric representations of 3D models. We find that the most propitious 3D distance transform leading to discriminative subspace features is the inverse distance transform. We mitigate the ambiguity of pose normalization with continuous PCA coupled with the use of all feasible axis labeling and reflections. The performance of the subspace-based retrieval methods on Princeton Shape Benchmark is on a par with the state-of-the-art methods.

1. Introduction

Rapidly growing databases of 3D models in many domains give rise to a need for efficient indexing tools for various recognition, classification or retrieval tasks. Manual annotation of objects with keywords being inadequate and impractical, automatic indexing and retrieval of 3D objects is the obvious alternative.

In this work, we explore subspace approaches for the shape-based classification and retrieval of complete 3D object models. This approach is based on the conjecture that 3D shapes have a sparse representation in some appropriate subspace, and hence can be represented with many fewer coefficients as compared to their voxel representation. The subspace can be model-driven, thus spanned by a fixed set of basis functions, and among which the most energetic terms are selected. Discrete Fourier transform, discrete cosine transform, discrete wavelet transform are well-known examples in this category. The subspace can be data-driven where the basis functions are generated from the training data itself. In this work, we demonstrate the advantages of data-driven methods, which capture the statistical characteristics of the 3D objects better for the retrieval task, on several well-known subspace techniques, namely principal component analysis (PCA), independent component analysis (ICA) and nonnegative matrix factorization (NMF).

Subspace techniques have commonly been used for characterization of specific categories of shapes, such as 3D faces, body shapes and anatomical structures [1–6]. However, to the best of our knowledge they have not been extensively studied specifically for indexing and retrieval of general 3D models. Subspace methods are powerful in capturing the essence of the shape space and successful in retrieval provided the query and target models are well represented in the training set. They greatly reduce the dimensionality of the models supplying compact representations, which enable fast searching. The feature extraction procedure is also time efficient since it only involves multiplication with a matrix. However, the success of subspace-based retrieval methods, when applied to general 3D models, depends critically upon the quality of the object pose normalization or registration process due to the lack of common natural landmarks (see Fig. 1).

The primary contribution of this paper is a general subspace-based framework for indexing general 3D objects (see Fig. 2). We propose and explore various alternatives for each component of this framework, namely for data representation, object alignment, choice of the subspace and shape matching. In particular, we use continuous PCA (CPCA) [7] for the alignment problem in conjunction with the distance transform of the voxelized models, which provides smoothness and hence an inherent robustness against minor misalignments (Section 3.3). We present mean shape-based and class-based correction schemes to resolve pose ambiguities resulting from CPCA (Section 5.4). For shape matching, we propose a computationally efficient version of the Munkres algorithm, which we refer to as pose assignment strategy, in order to compute the distance between two models by taking into account all possible mirror reflections and axis relabelings (Section 6). As a result, the PCA, ICA and NMF subspaces, when tailored to the needs of a

* Corresponding author. Fax: +90 212 2872465.
E-mail addresses: helindutagaci@gmail.com (H. Dutağacı), bulent.sankur@boun.edu.tr (B. Sankur), yyemez@ku.edu.tr (Y. Yemez).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.05.001

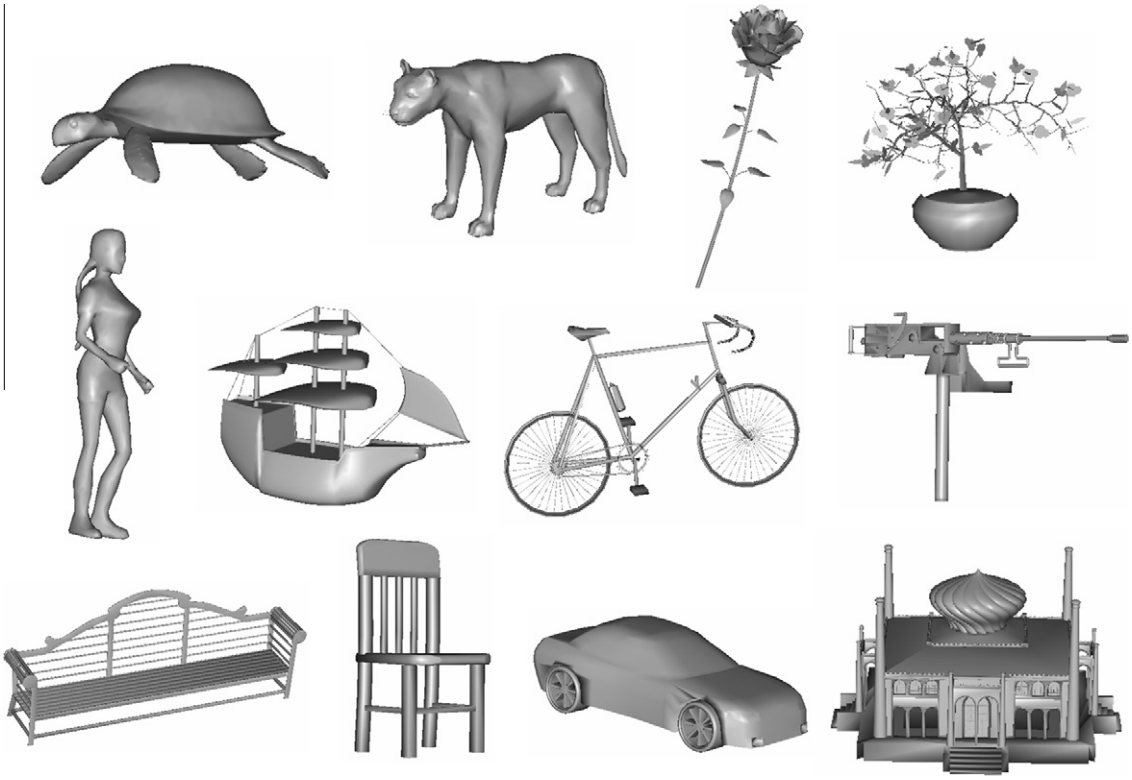


Fig. 1. Samples of general 3D models.

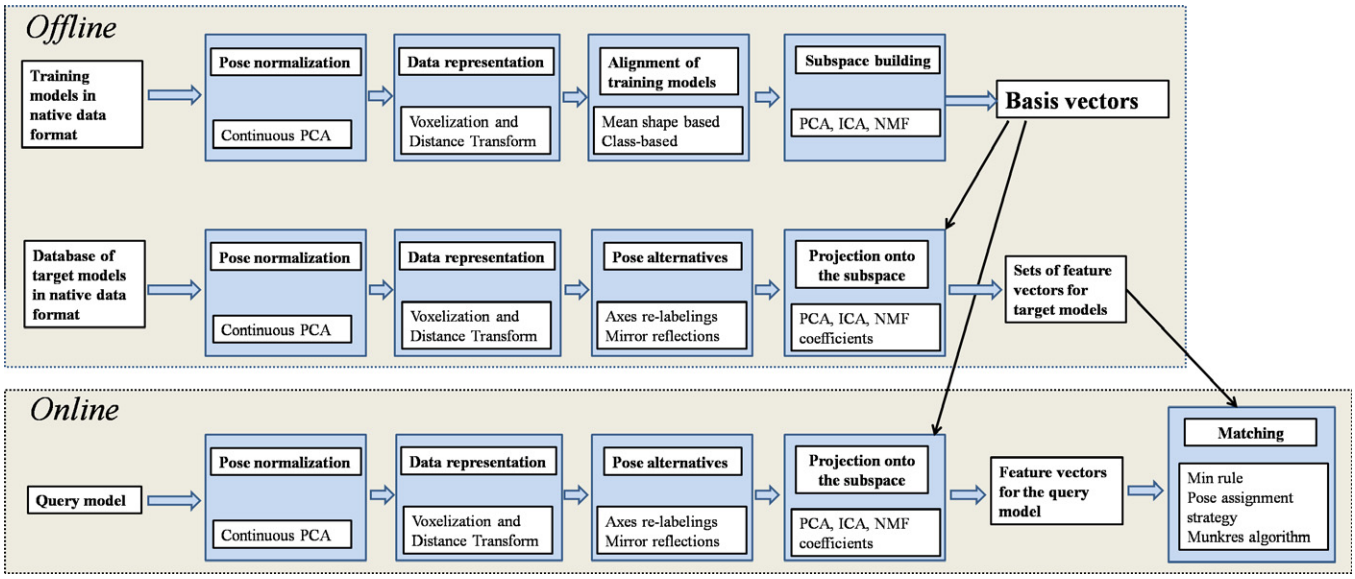


Fig. 2. General framework of subspace-based indexing scheme.

retrieval problem and applied to voxelized 3D shapes (Section 5), provide state-of-the-art performance.The retrieval performance of the proposed framework is demonstrated on Princeton Shape Benchmark (PSB) database. The subspace-based methods, when combined with other state-of-the-art descriptors in the literature, achieve the highest retrieval performance reported so far on PSB test set.

The paper is organized as follows: In Section 2, we overview the related work on shape-based retrieval of 3D models. In Section 3, we give details of the voxelization process and of the distance transform. Direct voxel comparison in Section 4 aims to serve as a baseline method. In Section 5, we describe the subspace techniques and the alignment of the training models for subspace building. In Section 6, we present the matching process between a query model and target models in terms of their pose alternatives. We provide experimental results in Section 7 and finally conclude in Section 8.

2. Related work

The large amount of research carried on the retrieval of general 3D models from large databases within the last 10 years is thor-

oughly categorized and reviewed in a number of survey papers [8–10], and PhD theses [11–14]. In this section, we limit ourselves to the discussion of approaches that have inspired or are directly related to our work.

Converting the surface representation of an object, mostly from a mesh representation into a voxel grid has been suggested by many authors [11,15–23]. Funkhouser et al. suggested the use of a binary voxel grid, where the voxels that intersect the object surface are assigned the value 1 [18]. Vranic has argued that a binary function will result in a loss of important surface information and proposed to use a real-valued function, where each voxel is attributed to a value proportional to the area of the surface patch confined in it [11]. Novotni and Klein [21] suggested to voxelize the surface using radial linear, binary and Gaussian kernels; however they obtained the best results with the binary kernel. Kazhdan et al. proposed the exponentially decaying Euclidean distance transform [24]. One important advantage of the distance transform vis-à-vis plain surface voxels is that it can alleviate the effects of small pose variations on shape matching. Furthermore the functional form and parameters of the distance transform can be optimized to minimize the impact of pose variations in subspaces.

One source of controversy in the 3D model retrieval community concerns the pose invariance problem. Some authors advocate the development of pose-invariant descriptors [18,21,24,25] while others rely on preprocessing for pose normalization and then extract nonpose-invariant features from the normalized representations [11,26–29]. Our methods are in the second category, that is, our subspace-based features are dependent on pose, and we rely on pose correction prior to voxelization. Pose normalization techniques can be listed as PCA, weighted PCA [26,30], continuous PCA [11,7] and PCA on the normals of the model (NPCA) [29]. All these techniques aim to transform objects into a canonical coordinate frame. Among these, CPCA is the most robust method, since it incorporates the whole object surface to the pose normalization via integration over triangles.

The descriptors that are closely related to our subspace-based approach are the representations of the 3D models in some transform domain. In general, transform-based methods assume the following signal model:

$$x = \phi b + N, \tag{1}$$

where x is the data representing the geometry of a model, ϕ is the set of basis vectors onto which x will be projected; b is the coefficient vector and finally N is the observation noise. The aim is to describe the shape in a compact form that preferably possesses an inherent multiresolution nature. One common way to achieve this goal is to use spherical harmonics [11,18,7,31,32,21,33]. Another way, as proposed by Vranic [11], is to characterize the 3D voxel grid by using 3D-DFT. Novotni and Klein have used Zernike functions, which are basically spherical harmonics modulated by appropriate radial functions [21]. Ricard et al. have introduced 3D Angular Radial Transform, which is defined as a product of radial and angular basis functions [19,34].

In any such transform-based representation, the discriminating shape information is subsumed in the coefficients b while ϕ is fixed. Most of these transform domains are constructed in terms of complex exponentials and sinusoids of varying frequency. One of the drawbacks of using harmonics as basis functions is that it is difficult to obtain a compact representation of a 3D shape with high frequency content. If the surface is composed of a series of jagged or highly curved concave and convex parts, as in articulated objects, many coefficients are required to describe the model. In particular, an additional drawback of the spherical harmonics descriptor is the necessity to describe the geometry of the object in terms of functions on a sphere. However, most of the 3D models cannot be mapped onto a single sphere without loss of information.

On the other hand, subspace techniques such as PCA, ICA and NMF are data-driven approaches and solve jointly for the subspace basis vectors and their projection coefficients. So far they have been used only to model objects of the same genre. For example, there is a vast literature of subspace analysis of anatomical structures in the domain of biomedical imaging [1–3]. These techniques usually concentrate on a single structure, such as the corpus callosum, and model the small, but medically significant variations. Likewise, in biometric systems that identify a person from her 3D face geometry, subspace methods are powerful tools for modeling the interpersonal variations [4–6]. There is also research for modeling shape variations of 3D human body via PCA [35] and human torso via PCA and ICA [36].

3. Voxel representation

3.1. Pose normalization

In order to normalize the pose of triangular mesh models before voxelization, we use the continuous principal component analysis (CPCA) technique developed by Vranic et al. [11,7]. The aim of this procedure is to transform the mesh model into a canonical coordinate frame. The model is first translated such that its center of gravity coincides with the origin. Scale invariance is achieved by setting the area-weighted radial distance from the origin to unity. Then, the covariance matrix of the x, y and z coordinates on the object surface is estimated via a continuous integration over all the triangles. The eigenvectors of the covariance matrix are considered as the principal axes, and the object is rotated such that the canonical coordinates coincide with the eigenvectors. The eigenvectors of the covariance matrix are sorted in decreasing order of their eigenvalues, and they are made to correspond to the x, y and z axes of the canonical frame, respectively. This procedure assigns the orientation of the largest spread of the surface points with the x-axis, the next largest spread with y axis and so on. After the order of the axes is determined, the second order moments of the model are used for selecting the positive direction of the axes [11]. A serious drawback of PCA-based normalization is its potential risk to put objects of the same class "out of phase" due to inconsistent axes labelings and reflections [27], as we will address later in Section 5.4 in more detail.

3.2. Binary function in 3D space

The voxelization converts the mesh information into a discrete function regularly sampled in 3D coordinates. The mesh model is placed in a Cartesian grid at some resolution and voxels are assigned the value 1 if the surface passes through it, and 0 otherwise. The voxelization operation involves setting of two important parameters: The first one is the size of the rectangular prism, a 3D windowing function, in which the object will be sampled. The second one is the sampling density, the number of voxel units, along each direction.

Using the bounding box of the object itself as the windowing function, will make the representation sensitive to outliers. Since the number of voxels must be the same for all the objects, fitting an object into its bounding box will scale it with respect to its extremities. Instead, we scale objects such that their area-weighted mean distance (AWMD) from the center of gravity to the surface is set to unity. Then we put the object in a fixed size box and discard all object parts that fall outside the box, as in Fig. 3.

We use a cube as the box; hence we take identical dimensional factors along x, y and z directions. We define the size of the box as half the length of one of its edges. There are obvious trade-offs in the choice of the box size vis-à-vis the normalized scale. The choice

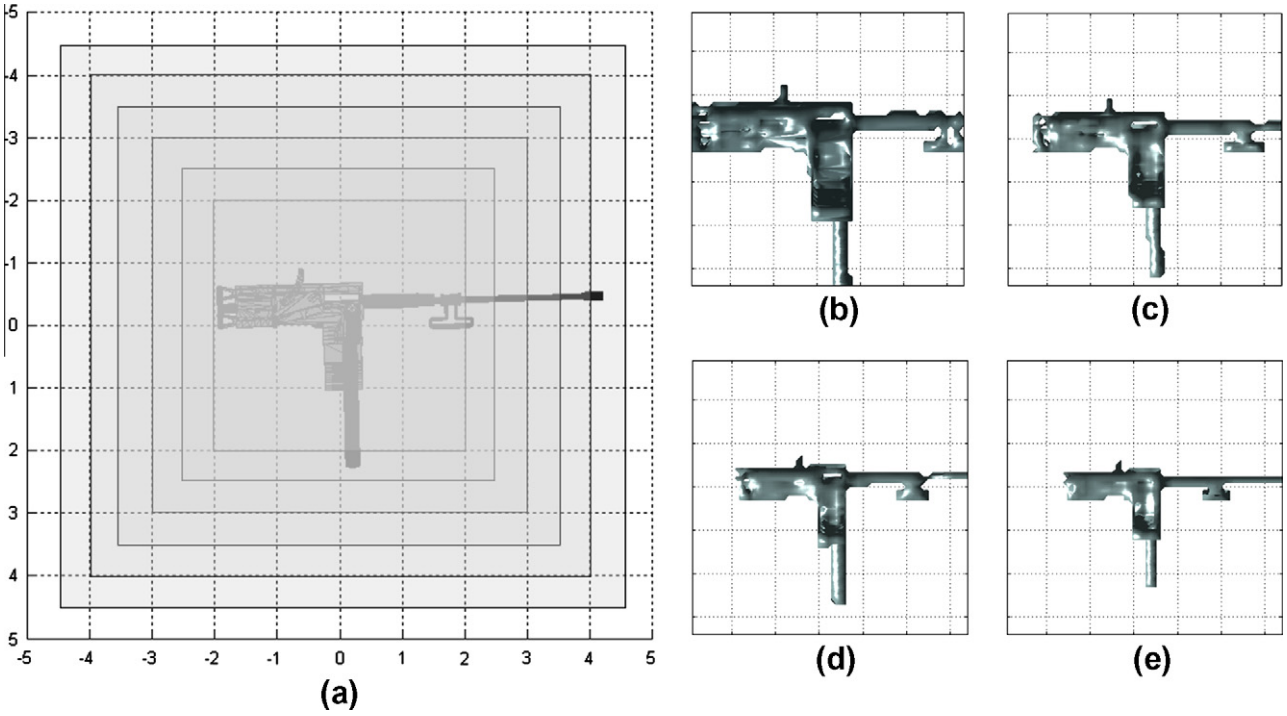


Fig. 3. Selection of the box size for voxelization of the mesh model in (a). The resulting voxel representations are shown on the right, with box sizes of 2 (b), 2.5 (c), 3 (d) and 3.5 (e).

of a large box means larger voxels and a coarser representation; on the other hand, small boxes may crop some important model parts. In extremum, the cube size can be adjusted to encompass all extremities of all the objects in a database. Then we guarantee to have all objects remain within the box, while sacrificing resolution, and for most objects in the database leaving many parts of the box volume empty.

In this work instead, we search for optimum box size that includes a proportion of objects within the box favorable to good classification. We select the box size as some factor of AWMD. Since AWMD is already set to unity for all the models during scale normalization, the size of the box is equal to the factor we choose (Fig. 3). We have used two approaches to set the box size. The first approach examines the histograms of the extremities along x, y and z axes of the pose normalized objects in the database and selects the box size such that the majority of the objects will remain entirely in the box. The second approach calculates the cropped proportion over all the objects in the database in terms of surface area for varying box sizes and then chooses the minimum size that keeps the lost surface proportion below a threshold. We have applied both procedures to the training set of Princeton Shape Benchmark and have chosen the latter method since it is more robust to outliers. Details are given in Section 7.1.

We rasterize scale-normalized objects into grids of RxRxR voxels. The number of voxels determines the level of detail that will be preserved in the voxel representation. The sampling density is a compromise between maintaining class-specific details and glossing over small within-class variations, which are considered as "noise". Too small an R obviously results in a rough voxelization; on the other hand, too large R values, while attaining fine voxelization, may unnecessarily bring forth disparities due to slight pose normalization errors. This issue of mismatching of two similar high resolution models is discussed in [11]. Fig. 4 shows voxelized representations of five models with various selections of R. For these specific examples, representations at resolutions of $R = 32$ or 64 seem to be sufficient to at least visually identify object classes. In our work, we conducted experiments with $R = 32$.

3.3. Functions of the distance transform

One consequence of binary representation is that it is not sufficiently robust against pose perturbations. The distance transform has many advantages over the binary function. First, the representation is smoothed and high-frequency artifacts due to the blocky structure of the binary voxels are suppressed. Thus contradictory indications by the nearby binary voxels of two objects, an artifact of binary voxelization, will be avoided. Second, each voxel in the cube will contribute to the distance computation between two objects.

The distance transform, also known as the distance field, is a function which maps each point in the space to the distance between that point and the nearest non-zero point in the original binary function. We can define the 3D distance transform, $DT_f(p)$ at point $p = (x,y,z)$ of the binary function $f(p)$ as

$$DT_f(p) = \min_{\{\hat{p}, f(\hat{p})=1\}} d(p,\hat{p}).$$

For distance measure $d(p,\hat{p})$ we use the Euclidean distance. The distance transform is zero at the surface of the object and increases monotonically as we move further from the surface. The values can become quite large at the borders of the box. Thus points farthest from object surface will have higher impact on the shape comparison, which is counterintuitive. We therefore prefer to use a function of the distance transform that takes its largest value on the surface of the object and decreases smoothly as one moves away from the surface. We have experimented with the distance transform itself and the following functions of it:

(i) The inverse of the distance transform (IDT):

$$IDT_f(p) = \frac{1}{1 + DT_f(p)}.$$

(ii) The Gaussian of the distance transform (GDT):

$$GDT_f(p) = \exp\left\{-(DT_f(p)/\sigma)^2\right\},$$

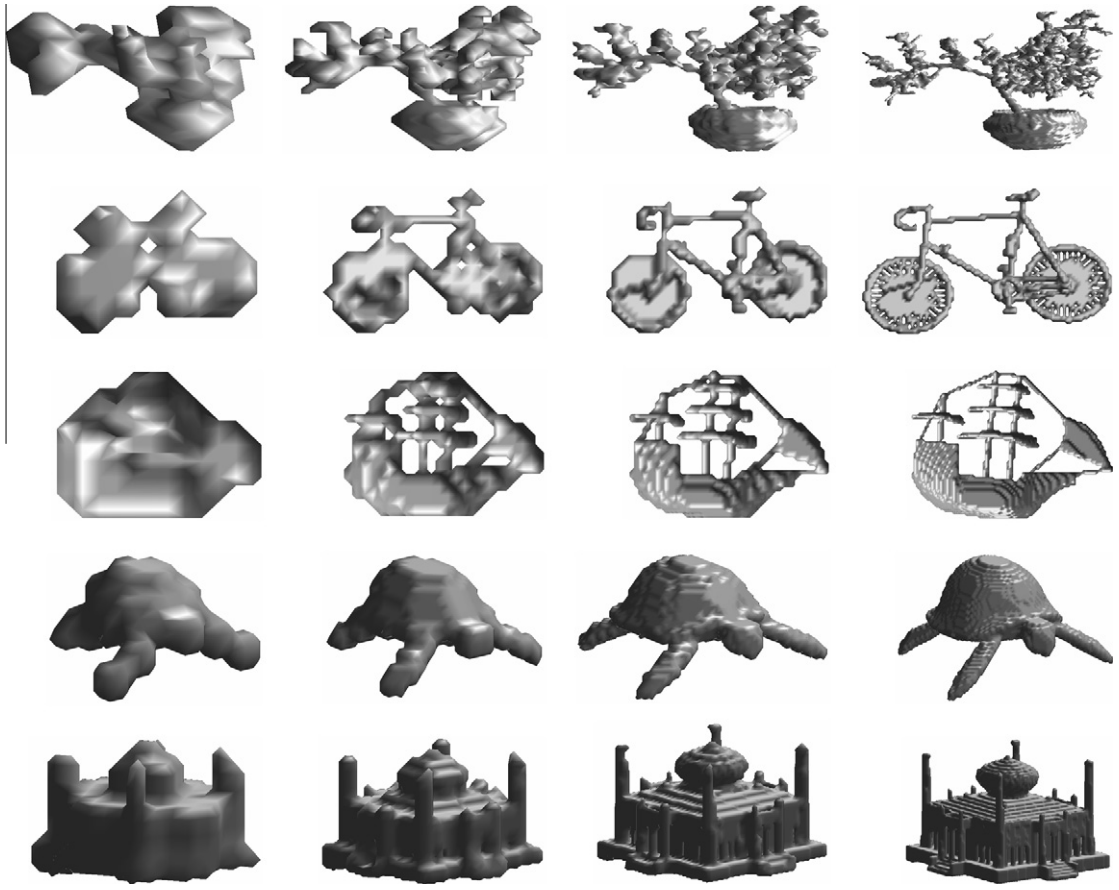


Fig. 4. Models voxelized at resolutions $R = 16$, 32, 64 and 128 from left to right.

where the parameter σ determines the width of the Gaussian profile.

(iii) A piecewise linear function of the distance transform (LDT):

$$LDT_f(p) = \begin{cases} 1 - \frac{DT_f(p)}{k} & \text{if } DT_f(p) \leqslant k; \\ 0 & \text{otherwise.} \end{cases}$$

where the parameter k determines the width of the triangular profile of the linear function.

The experiments that we have conducted with various radii of the Gaussian function show that the inverse of the distance transform gives significantly better results regardless of the resolution of voxel representation. Fig. 5 shows the profiles of the functions with various width parameters, σ and k. The Gaussian and linear profiles are similar in their appearance, and in fact they yield similar retrieval performances (Section 7.2). If their support is small, they decay rapidly toward zero. For larger widths, the GDT varies slowly in the neighborhood of the surface, which in turn causes blurring of the object surfaces (Fig. 6h). The profile of the IDT is significantly different from the others. First, it decays rapidly in the beginning, thus the blurring effect is mitigated; furthermore voxels on the surface gain much more importance. Second, IDT has a larger effective support than GDT and LDT. Therefore the distance information is propagated further away from the object surface, but with attenuated weights as compared voxels proximal to the surface.

Fig. 6 shows the voxel representation of a chair and the slices from various possible 3D functions. The slice from the binary representation carries very little information about the general shape of the model. The same observation is also valid for GDT and LDT functions with small support. On the other hand, IDT seems to provide a good compromise between fast decay rate and large support.

4. Direct voxel comparisons

Direct comparison of objects provides a base-line to measure gains enabled by the feature extraction schemes. The representation modalities can be voxel-wise differences of volumetric models or pixel differences of depth buffers, etc. This gain is expressed in terms of increased discrimination power and decreased search effort. All feature extraction or selection methods focus on class-specific shape characteristics and attenuate irrelevant variations and details. Subspace projection as a feature extraction method provides a controlled way of filtering details nonpertinent to classification. In order to measure the performance gain, if any, of the subspace algorithms, we resort to baseline retrieval performance obtained directly via raw data without any feature extraction attempt.

For the $R \times R \times R$ voxel array representation, we convert this 3D array to a 1D vector, x, using lexicographical ordering with indexing m. The distance of a query model x_i to a target model x_j in the database is the sum of pairwise absolute differences of the attributes of the voxels. We select the L_1 distance first due to its computational simplicity and second due to its appropriateness for high dimensional data comparison [37]:

$$d(x_i, x_j) = \sum_m |x_i(m) - x_j(m)|.$$

The direct comparison of raw data serves first as a baseline system. Second, it is instrumental in tuning parameters such as the box size, sampling resolution, the type of distance transform

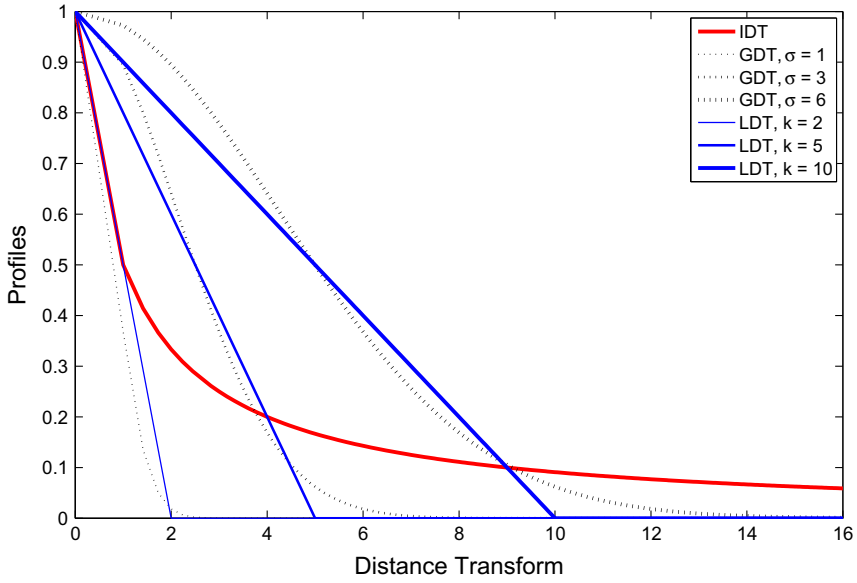


Fig. 5. Profiles of the distance transform functions for various parameter settings.

function as well as the aperture of the GDT or LDT functions. These optimized parameter settings are then used by all the subspace transform methods. Thirdly, direct comparison method guides us to form a well aligned training set, where coherent axis labels and reflections are selected within classes. This procedure is described in Section 5.4.

Calculation of $d(x_i,x_j)$ for every pair of the query and target object becomes very time consuming with increasing number of database objects and for large R. It would be inefficient to use the direct comparison method in an online application such as web-based 3D model retrieval. In general, it is desired to have as compact and informative descriptors as possible, without any significant performance loss. In Section 5 we investigate subspace methods for compacting features.

5. Subspace methods

Given the observation matrix X, subspace methods find a set of vectors that describe the significant statistical variations among the observations. The significant part of an observation x is expressed as the linear combination of basis vectors ϕ:

$$x \approx \phi b,$$

where $b = (\phi^H\phi)^{-1}\phi^H x$. These methods have the additional advantage of greatly reducing the dimensionality of the observations.

Let us assume that we have a set of N training models that are represented as a voxel grid of size $R \times R \times R$. Let x_i be a column vector of length $M = R^3$ formed by some lexicographic ordering of the voxel values of the ith model. The data matrix is then formed as $X = [x_1\ x_2 \cdots x_N]$ and is of size $M \times N$.

5.1. Principal component analysis (PCA)

PCA is an analysis technique that is based on decorrelation of the data using second order statistics. The eigenvectors of the $M \times M$ sample correlation matrix, $C = X X^T$ gives the principal directions of variations. Let $\{u_1,u_2,\ldots,u_K\}$ be the first K eigenvectors of C with corresponding eigenvalues $\{\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_K\}$. These first K leading eigenvectors identify the directions in shape space along which data have the most energy. The amount of information maintained depends on K and the spread of eigenvalues. The projection of an input vector x onto the PCA subspace is given by $a = U^T x$, where U represents the $M \times K$ projection matrix formed as $[u_1\ u_2 \cdots u_K]$.

Notice that we use the correlation matrix instead of the covariance matrix, i.e., non-centered data for PCA analysis contrary to the common practice. Non-centered PCA is much more suitable for data that exhibit high heterogeneity among axes [38]. Each voxel in the 3D grid corresponds to an axis of the vectors of size $M = R^3$. Some shape classes have negligible activity on some subset of voxels, i.e., along the corresponding axes since IDT is close to zero for most of the voxels which are not close to the surface. That is why we consider the data as having high heterogeneity among axes. Choosing non-centered PCA is also validated by our experiments conducted with centered and non-centered data.

Figs. 7 and 8 give visualizations of eigenvectors, or eigenshapes of the training set of the Princeton Shape Benchmark. The first mode of variation is related to the notion of a "mean shape" (see Fig. 7), hence its first coefficient determines the extent the "mean shape" is contained in a given shape. Lopsided models, such as elongated and thin shapes that have voxels concentrated in particular regions, are not well represented by the mean shape, hence the first mode plays a minor role for them. The first mode becomes a thin ellipsoid along the direction of concentration and the shape is generated by the addition of higher modes. In contrast, for compact and fat objects the first mode consists of a larger ellipsoid, and some of the other modes are subtractive, that is, they carve out the inner parts. Notice that the role of the first mode becomes prominent because we are using the shape correlation matrix and the mean shape is not subtracted from models. Large deviations from the grand-mean of the shapes support our choice for the non-centered PCA.

Fig. 8 shows the next largest four modes of variations. The reconstructed modes seen in Fig. 8 are obtained by fixing the first mode $\lambda_1 u_1$ at μ and adding to it higher order shape modes with some gain factor c:$c\lambda_i u_i + \mu$. Notice that the gain can be positive or negative, summing or subtracting volumes from the mean shape. To visualize the reconstructed shapes with mode manipulation we show its slices in three orthogonal directions and also as one isosurface. Note that since we work with IDT data, the resulting basis shapes (eigen-volumes) are real-valued functions in 3D. First, we observe that the basis shapes are nearly symmetric around some of their axes, and this is simply due to the fact that

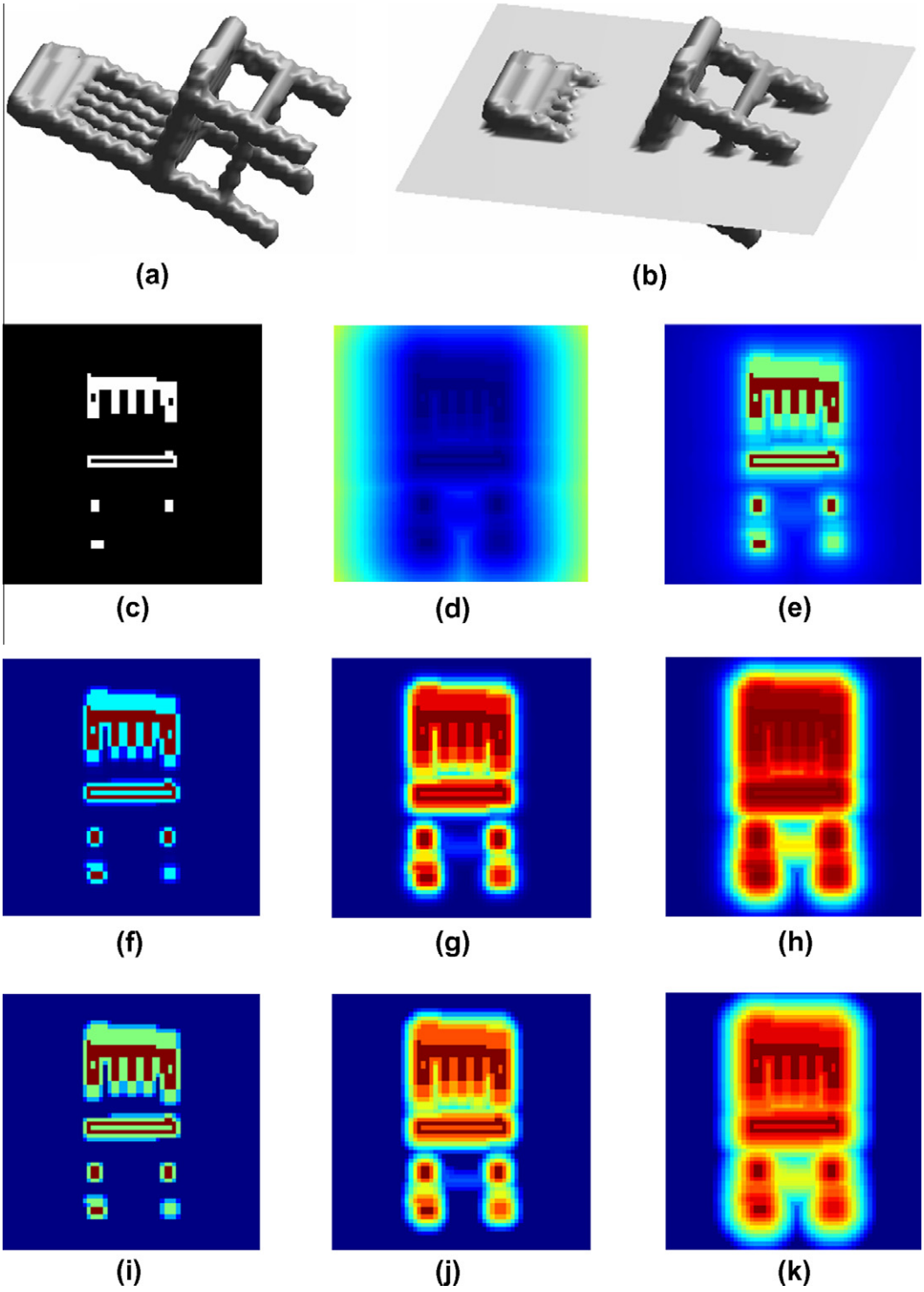


Fig. 6. Slices of the chair model (a) extracted as in (b). Slices from binary function (c), distance transform (d), inverse distance transform (e), Gaussian of the distance transform with $\sigma = 1$ (f), $\sigma = 2$ (g), $\sigma = 6$ (h), piecewise linear function of the distance transform with $k = 2$ (i), $k = 3$ (j), and $k = 10$ (k).

most of the models in the PSB are symmetric. Our second observation is that larger variations give rise to topological changes. Changing the weight of an eigenvector causes gross holes and disjoint parts to appear and disappear. It is interesting to note that the second eigenvector controls the elongation of a model (Fig. 8a). In fact, for negative coefficients the model becomes thinner and more elongated, and for positive values the model becomes more spherical with a hole inside. The generation of a hole, and not of a solid ellipsoid, is due to the fact that we have surfaces instead of filled volumes in the training set. Higher positive gains of the second mode causes splitting of the ellipsoid into two parts. As the coefficient of the third mode sways in negative and positive directions, the shape splits into two elongated parts and parts start to merge, respectively (Fig. 8b). Further positive gain of the third mode results in a torus. The fourth and fifth modes have similar kind of topological effects (Fig. 8c–d). The higher modes contribute finer shape variations, though they are less discernible on the topology and on the global shape.

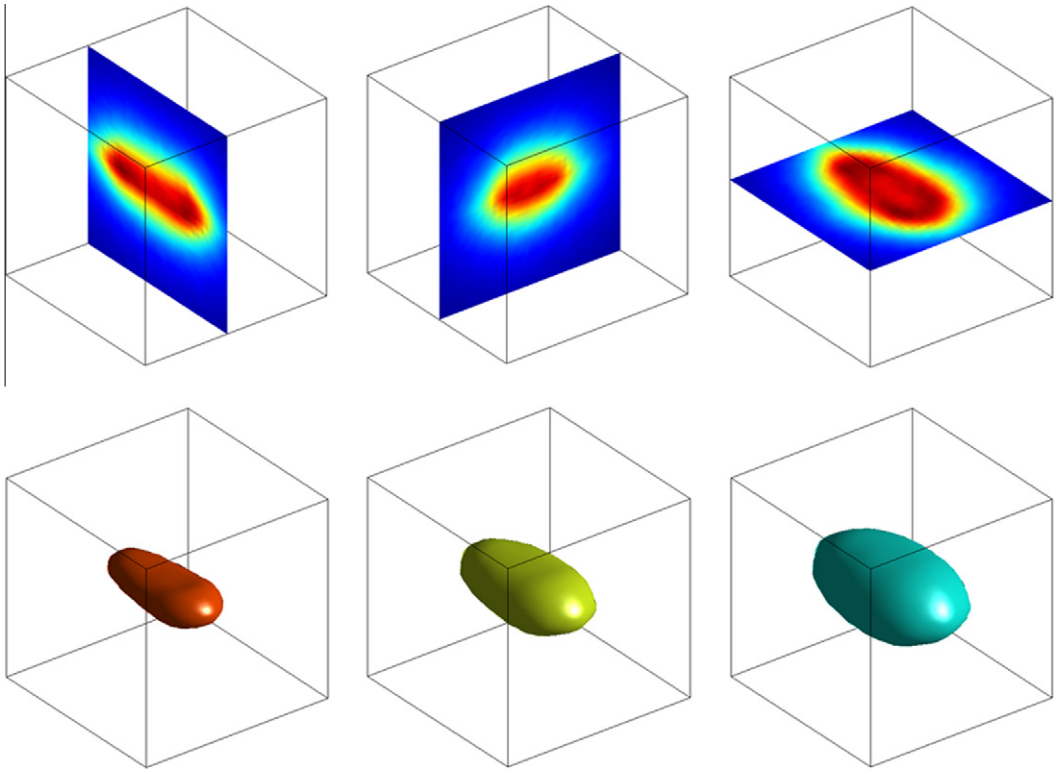


Fig. 7. Visualization of the first eigenvector. First row shows slices from *x*, *y* and *z* axes, from left to right. Second row shows isosurfaces at different levels.

5.2. Independent component analysis (ICA)

ICA is a generalization of PCA in that it removes dependencies of higher order statistics from the data. ICA assumes that the observed signals $\{x_1,x_2,\ldots,x_N\}$ result from linear mixing of K source signals $\{s_1,s_2,\ldots,s_K\}$. The signal model is then $X = AS$ where A is the matrix of mixing coefficients and S contains the sources in its rows. Both the source signals and the mixing coefficients are unknown and need to be estimated. The sources are recovered by a linear transformation, where W is the separating or de-mixing matrix. In this paper, we use ICA architecture II and we estimate W using FastICA algorithm. In ICA2, the mixture coefficients are assumed to be independent whereas the basis functions are not statistically independent. The details can be found in [39]. We estimate the mixing coefficients for the query and target models and use those coefficients as shape descriptors.

Prior to the estimation of the de-mixing matrix, W, it is conventional to reduce the dimensionality of the data matrix via PCA. We select the PCA subspace dimension K in a goal-oriented manner to achieve the best retrieval performance over the training set.

Fig. 9 gives visualizations of 10 of the ICA components obtained from the PSB training set. We see that the basis volumes, as illustrated via their horizontal slices and isosurfaces are totally different from the PCA versions. The ICA2 components, or basis volumes, resemble the models that are present in the training set; whereas PCA components represent very general topological or geometric variations. The PCA coefficients engage several coefficients in order to give a clue about the class while in the ICA representation a few pronounced coefficients can strongly indicate the model class.

5.3. Nonnegative matrix factorization (NMF)

Given a nonnegative data matrix, X, of size $M \times N$, we can factorize it into two nonnegative matrices V and H, such that $X \approx VH$, with sizes $M \times K$ and $K \times N$, respectively. Note that the IDT representation of the shapes guarantees nonnegativeness of X. V contains the basis vectors spanning the subspace and H is constituted of combination coefficients. We use the update rules described by Lee and Seung [40] to estimate the nonnegative $v_{m,k}$ and $h_{k,n}$ factors. The objective function is taken as $\|X - VH\|^2$, where $\|\cdot\|$is the Frobenius norm and the factor matrices are constrained to have nonnegative elements [40].

The NMF representation is more physical and sparse. This is because both PCA and ICA allow basis vectors and their coefficients to assume positive or negative values, and the subspace reconstruction may involve redundant bases that cancel out each other, that is, irrelevant additions and subtractions can occur. This may introduce unphysical artifacts of negative mass in 3D. Since only positive bases and coefficients are involved in NMF, that is, subtractions are not allowed in linear combinations, NMF leads to basis signals that are locally physical and sparse, and it provides a parts-based representation [40]. Visualization of sample NMF components in Fig. 10 verifies this argument. We get sparse basis volumes representing different parts of the models in the training set.

5.4. Axis relabeling and reflection

The most problematic issue with the CPCA normalization is the ambiguity of axis ordering and reflections. We conjecture that most of the misalignment errors are due to inconsistent within-class axis orderings and orientations resulting from normalization procedure. We will demonstrate this fact in Section 7.5 by showing the non-negligible gains in the retrieval performance when more coherent within-class orientations are available. We resolve the axis ordering and reflection ambiguities by generating the set of all 48 possible reflections and orientations of the objects. Notice that the three coordinate axes can be labeled in $3! = 6$ possible ways and, for any given labeling, there are $2^3 = 8$ possible polarity assignments, which result in $6 \times 8 = 48$ possible configurations.

Fortunately, these 48 poses are generated very rapidly by applying array transpositions to the voxel-based representation of 3D

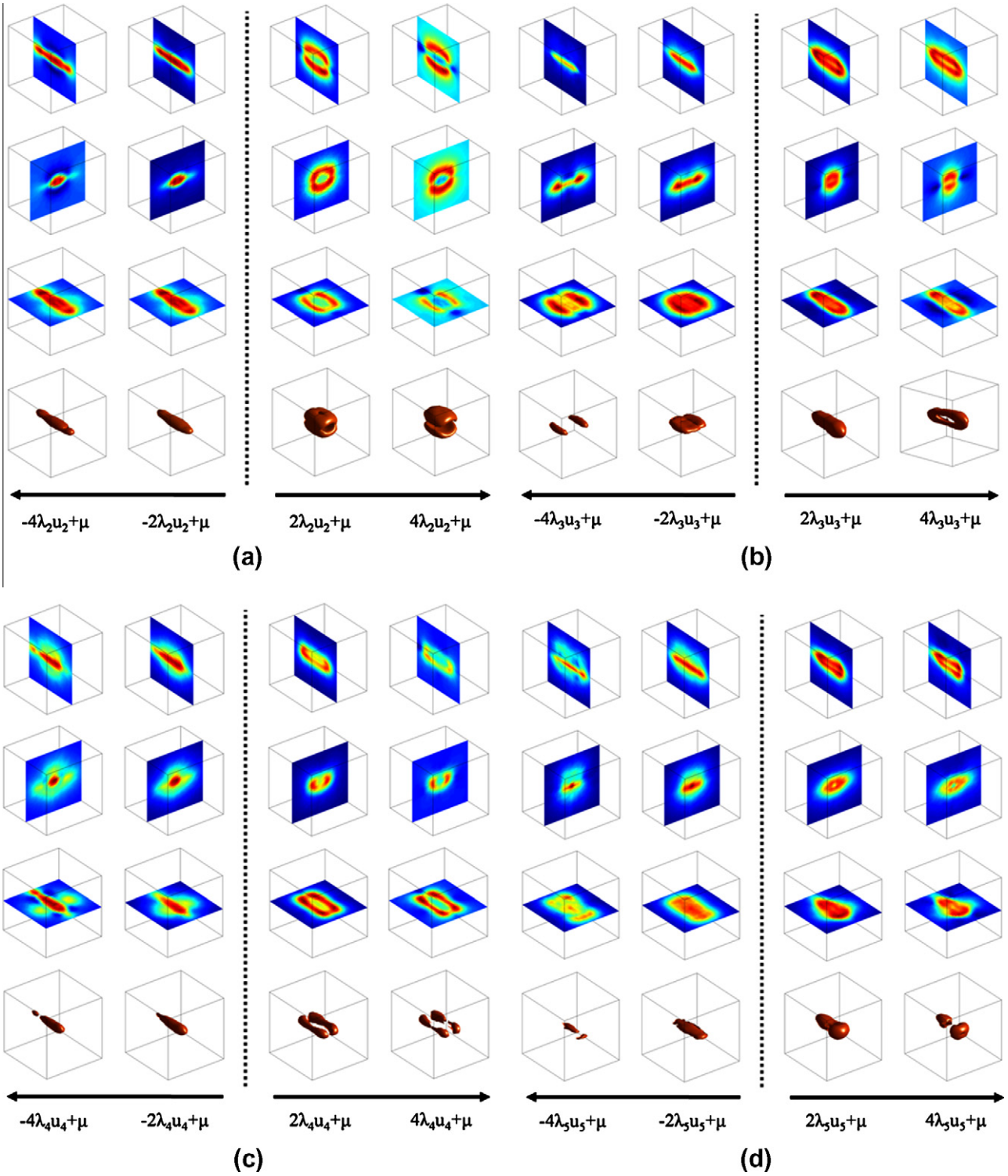


Fig. 8. Second (a), third (b), fourth (c), and fifth (d) modes of variation in PCA. First three rows show slices from *x*, *y* and *z* axes, respectively. Fourth row shows isosurfaces all at the same level.

models. We simply permute the coordinates of the 3D array to alter axes relabeling and flip the array along the three coordinates to obtain reflected representations of the voxel-based models. We will refer collectively to these pose varieties of the voxel-array as 48-Axes Relabeled and Reflected (48-ARR) versions of the model. For the *i*th model in the database, the *r*th ARR version will be denoted as x_i^r, with $r = 1,2,\ldots,48$.

While constructing the data matrix at the training stage, we correct the inappropriate axes ordering and orientations by applying one of the following corrective schemes: We find the most appropriate axis labeling and reflection by: (1) Mean shape based ARR selection, and (2) Class-based ARR selection.

5.4.1. Mean shape based ARR selection

This procedure assumes that the training set is not annotated with class information. In this case, we calculate the mean shape m, by averaging the training samples $\{x_1, x_2, \ldots, x_N\}$. Using direct voxel comparison method, we find the best among the 48-ARR versions

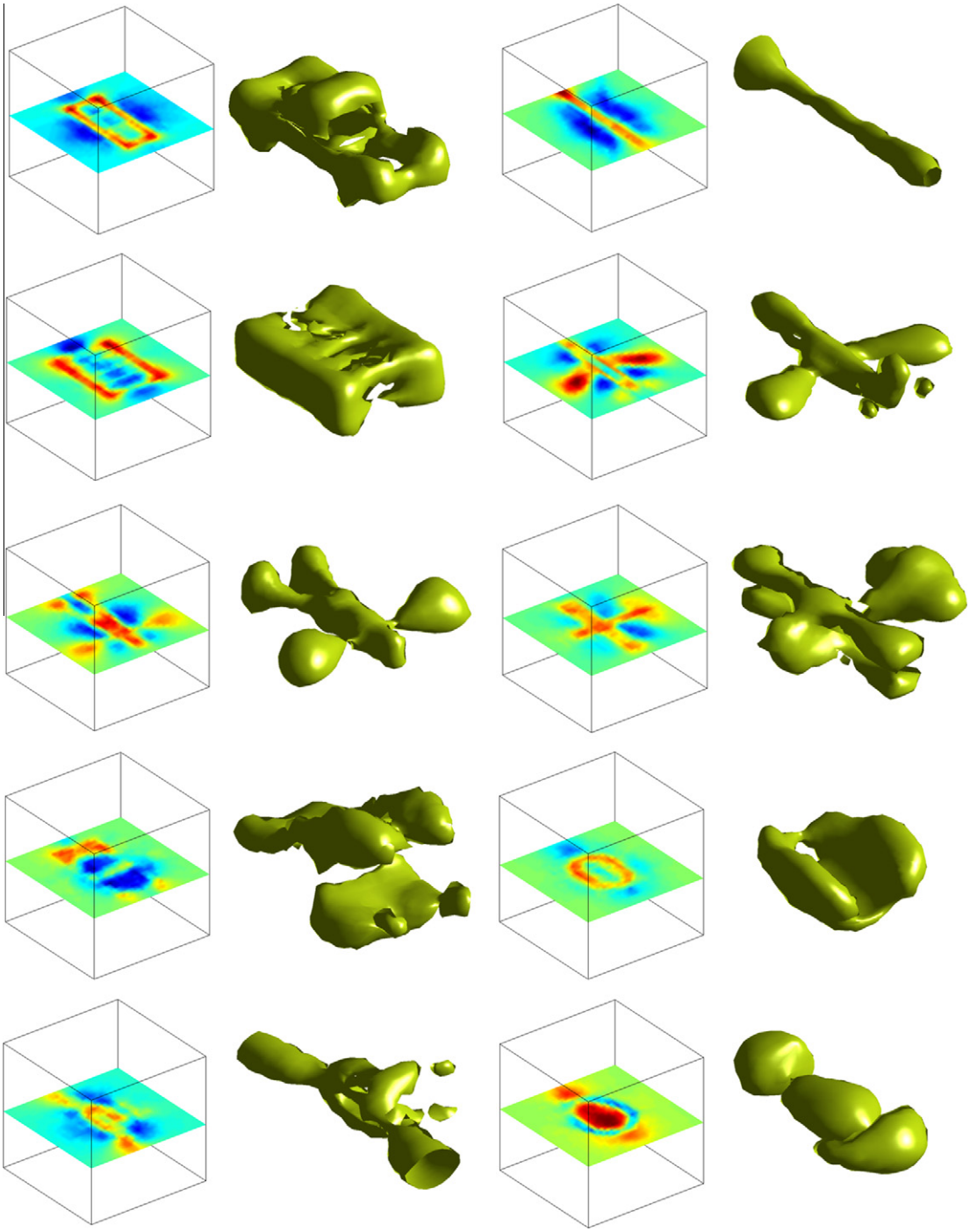


Fig. 9. Visualization of sample ICA2 basis vectors.

of each model from the training set as $r_i = \arg\min_{r=1,2,\ldots,48} |x_i^r - m|$. Then we recalculate the mean and repeat the procedure iteratively until the mean shape is not altered anymore.

5.4.2. Class-based ARR selection

In this procedure, we assume that the training set is equipped with the class information of the models. For each class C, we select an arbitrary member $\bar{x}_C$ as the representative of the class. Then we find the best 48-ARR version of the remaining members of the class via direct comparison of voxels: $r_i = \arg\min_{r=1,2,\ldots,48} |x_i^r - \bar{x}_C|$ for $x_i \in C$. The assumption of class informed databases is not unrealistic, because there are supervised 3D shape applications, such as in face recognition or detection of pathologies of organs where class information is annotated. However, in general purpose retrieval problems, one cannot always expect to find annotated large databases.

6. Matching of models

After the subspace is trained and the bases are formed, the target and query models are projected on the subspace, and these projections are used as the shape descriptor. We apply CPCA to the query and target models, voxelize them and define the IDT functions in the 3D space. For each model we also get the 48-ARR versions and project each one onto the subspace. We have a set of feature vectors, $F_i = \{f_i^1, f_i^2, \ldots, f_i^{48}\}$ for the ith model defined as:

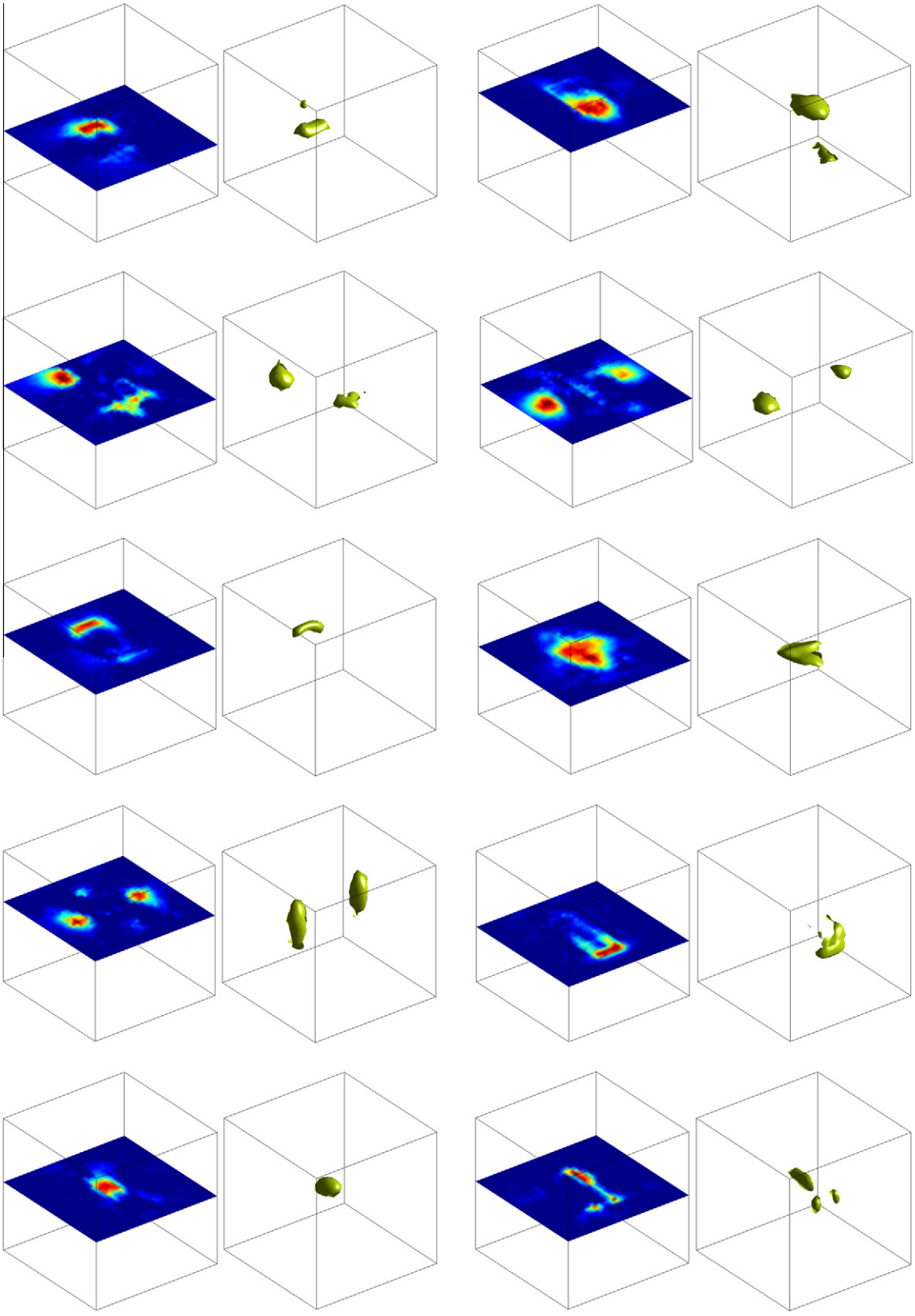


Fig. 10. Visualization of sample NMF basis vectors.

$$f_i^r = P_V x_i^r \quad r = 1, 2, \ldots, 48,$$

where P_V is the projection operator for the subspace spanned by V. In order to assess the dissimilarity between two models i and j, we construct the distance matrix, D of the 48×48 pairings from the two sets, F_i and F_j, such that

$$D_{rq}(F_i, F_j) = d_c\left(f_i^r, f_j^q\right)$$

where, we use cosine distance to compare pairs of feature vectors:

$$d_c\left(f_i^r, f_j^q\right) = \frac{f_i^r f_j^q}{\left|f_i^r\right|\left|f_j^q\right|}.$$

We have also conducted experiments with L_1 and L_2 distances to compare pairs of feature vectors. However, cosine distance yields the best performance since it normalizes the norms of the feature vectors to unity.

After constructing the distance matrix among the feature vectors corresponding to 48-ARR versions of the two models, we either select the minimum of the matrix as the distance between the two models (the MIN rule) or use the following fast variant of the Munkres algorithm, which we call as the pose assignment strategy. We define a cost function of the one-to-one assignment of each 48-ARR version of a model to a 48-ARR version of another model. We initialize the cost to zero. We select the minimum element of the matrix and add its value to the cost function. We set all elements in the row and column of the minimum element to infinity, and search for the next minimum of the distance matrix. We repeat the procedure until all 48-ARR versions of the two models are assigned to each other in a one-to-one manner. The final cost is the accumulated distances between the pairings of the two models. The pose assignment strategy is more robust and it improves the performance significantly as opposed to the MIN rule. In MIN rule we consider only one pair of pose matches, while in the pose assignment strategy we use all the distances between matched pose pairs. The pseudo-code for the pose assignment strategy is as follows:

Step 1: Initialize $COST = 0$;
Step 2: $(\hat{r}, \hat{q}) = \arg\min_{r,q} D_{rq}$
$COST \leftarrow COST + \min_{r,q} D_{rq}$
$D_{\hat{r}q} \leftarrow \infty \quad q = 1, 2, \ldots, 48$
$D_{r\hat{q}} \leftarrow \infty \quad r = 1, 2, \ldots, 48$
Step 3: Stop if all poses (r, q) are assigned to each other (or all $D_{rq} = \infty$). Otherwise go to Step 2.

7. Experimental results

We have conducted our experiments on the database of Princeton Shape Benchmark [41]. The database consists of a training set with 907 models in 90 classes and a test set with 907 models in 92 classes. The training and test sets are disjoint in the sense that they do not have common models. Although most shape classes are common to both, each set includes classes not present in the other.

We use precision–recall curves, discounted cumulative gain (DCG), nearest neighbor (NN), first tier (FT) and second tier (ST) as measures of retrieval performance. A detailed description of these performance measures can be found in [41]. Let C be the class of a query model and $|C|$be the number of models of class C in the target database. Let K be the number of retrieved models and, K_C be the number of models that belong to class C among the K retrieved models.

Recall: Given K and a query model, recall is the ratio of K_C to $|C|$, i.e., the proportion of the correctly retrieved models to all the correct models in the database. Ideally, we expect the recall to increase as $1/|C|, 2/|C|, 3/|C|, \ldots, |C|/|C|$while K increases as $1, 2, 3, \ldots, |C|$, since we would like all the K retrieved models to belong to the class of the query object.

Precision: Given K and a query model, precision is the ratio of K_C to K, i.e., the proportion of the correctly retrieved models to the K retrieved models. Ideally, we expect the precision to be always 1 until K reaches $|C|$; since we would like all the K retrieved models to be the correct objects, i.e., to be from the correct class. The ideal precision–recall curve is then a horizontal line at precision equal to 1 (or 100%).

First tier (FT): First tier is equal to the recall at $K = |C|$. This value of K is the smallest number of the retrieved models that could possibly include all the correct models in the database.

Second tier (ST): Second tier is equal to the recall at $K = 2|C|$.

Nearest neighbor (NN): Nearest neighbor is the rank-1 classification accuracy. This number is either 1 or 0, depending on the class of the first retrieved model; i.e. the closest database model to the query.

Discounted cumulative gain (DCG): Discounted cumulative gain is an evaluation tool where correctly retrieved models are weighted more when they appear earlier in the order of retrieval list. This measure takes into account the fact that a user will be less interested in the correctly retrieved models toward the end of the list. To calculate discounted cumulative gain, we obtain a list, G_k of the retrieved models, where G_k is 1 if the kth model belongs to C and, 0 otherwise. Then the DCG at k is given by

$$\mathrm{DCG}_k = \begin{cases} G_k, & \text{for } k = 1; \\ \mathrm{DCG}_{k-1} + \frac{G_k}{\log_2 k} & \text{for } k = 2, 3, \ldots, k_{\max}. \end{cases}$$

The overall DCG is calculated as

$$\mathrm{DCG} = \frac{\mathrm{DCG}_{k_{\max}}}{1 + \sum_{k=1}^{|C|} \log_2 k}.$$

We note that the performance of a retrieval system is usually expressed in terms of one or more of the measures defined above averaged over all the queries in a test database.

7.1. Selection of the box size

Prior to voxelization of the models in a database (e.g., PSB), we should set the size of the box in which the models will be rastered. We select the box size by inspecting the statistics of the pose-normalized triangular mesh models in the PSB training set. As explained in Section 3.2, we examine the extremities along x, y, and z directions and the surface areas remaining outside the box.

Fig. 11 shows the histograms of the extremities along positive and negative x, y, and z directions in the PSB training set. The extremities are larger in the x direction since PCA pose normalization assigns the orientation with the highest energy, hence dispersion, to the x-axis of the object. The "max" figure in each graph denotes the maximum extremity encountered among the models in the PSB training set. Inspecting the histograms for y and z, we can safely set the box size to 2.5. However, extremities along the x direction go well beyond 2.5 for many models. Table 1 gives the percentage of objects that will be cropped with respect to the choice of the box size. When we select 2.0 for box size, more than half of the objects will be cropped. When the box size is set to 2.5, nearly 25% of the objects will not fit in the box.

However the percentage of cropped objects does not reflect the situation properly since an object is considered to be cropped even if one voxel is excluded by the bounding box. A more faithful assessment of the size of the bounding box would be given by the ratio, a_i, of the cropped surface area of the ith object to that object's total area as a function of box size. Table 1 gives the statistics of the ratio a_i with respect to the box size over the PSB training set. We can observe that less than 1% of the surface of an object will be cropped on average if we select a box size of 2.5. When the average is taken over only the cropped models, the ratio of outside surface area per object is only 2.15%, and the median is half that amount. Based on these observations, we have decided to fix the box size at 2.5 in all the experiments.

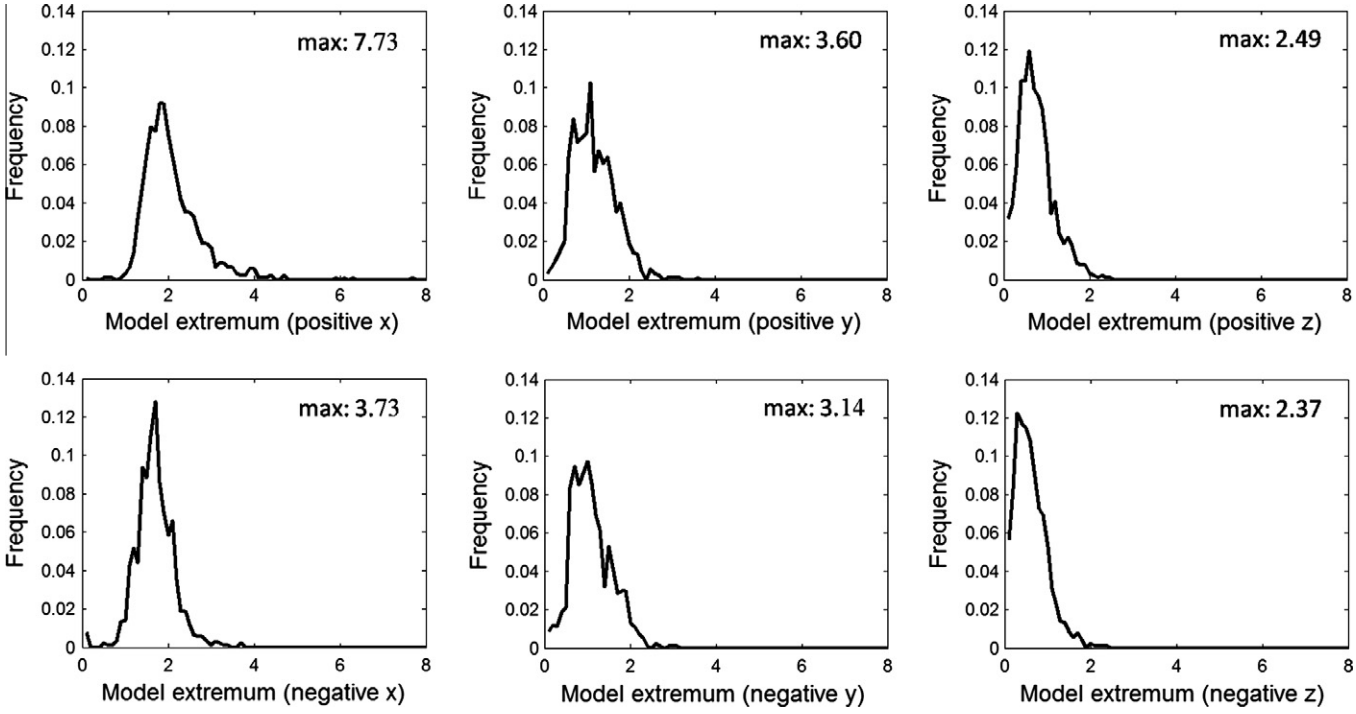


Fig. 11. Histograms of model extrema along positive and negative x, y, and z directions.

Table 1
Statistics of cropped models and cropped surface area percentage (a_i) with respect to box size (PSB training set).

Box size	# Cropped models	% Cropped models	Max(a_i)	Mean(a_i)	Mean(a_i), over cropped models	Median(a_i), over cropped models
2.0	474	52.26	16.62	2.34	4.47	3.82
2.5	211	23.26	8.97	0.50	2.15	1.14
3.0	80	8.82	7.29	0.15	1.69	1.04
3.5	31	3.41	6.44	0.05	1.57	0.88
4.0	13	1.43	4.95	0.02	1.45	0.72
4.5	6	0.66	3.80	0.01	1.92	1.90
5.0	4	0.44	2.63	0.01	1.91	2.22

7.2. Comparison of 3D distance functions

In this section, we compare the 3D distance functions defined in Sections 3.2 and 3.3 with respect to their retrieval performances on the PSB set. The aim of these experiments is to determine the optimal one among these.

In this set of experiments, we use directly the 3D distance fields generated by various distance functions without the use of any subspace technique. The experiments are conducted on the PSB training set using direct comparisons method described in Section 4. Furthermore instead of calculating 48×48 comparison scores between model pairs; we have used only one pose for each model in the matching process. The pose is either the one given by the CPCA or is determined using class-based ARR selection (CbARR). Table 2 gives the NN and DCG values for three resolutions of voxelization; i.e. for $R = 16$, $R = 32$, and $R = 64$. The values under the column entitled as NoARR correspond to the cases without any pose optimization, hence with the use of the pose obtained by CPCA. The values under the CbARR refer to the cases with class-based ARR selection described in 5.4. We omitted the results for a pose selection using mean shape-based ARR selection (MbARR) here, since it results in similar performance ordering of distance transform functions.

Table 2
Performance of 3D functions for various resolutions on the training set of Princeton Shape Benchmark. Direct comparison method is used.

	$R = 16$		$R = 32$		$R = 64$	
	NoARR	CbARR	NoARR	CbARR	NoARR	CbARR
NN						
Binary	47.0	56.4	49.8	58.1	33.8	38.6
DT	48.1	60.4	51.8	63.9	55.1	67.8
IDT	**51.0**	**62.3**	**58.0**	**69.1**	**58.3**	**71.1**
GDT ($\sigma = 1$)	49.2	60.2	53.7	63.6	48.3	67.6
GDT ($\sigma = 3$)	49.4	61.6	56.0	66.2	55.5	67.7
GDT ($\sigma = 6$)	47.9	60.4	54.7	64.1	56.7	67.6
GDT ($\sigma = 10$)	47.3	58.7	52.6	64.9	57.2	67.7
LDT ($k = 2$)	49.6	59.8	54.7	65.9	50.5	67.9
LDT ($k = 5$)	49.7	62.2	56.4	65.5	55.6	67.9
LDT ($k = 10$)	48.1	60.1	54.6	64.7	57.3	67.9
DCG						
Binary	50.5	57.7	50.2	56.9	40.3	44.2
DT	52.0	61.6	54.8	64.5	55.5	65.6
IDT	**53.4**	**62.3**	**56.6**	**66.2**	**57.0**	**66.8**
GDT ($\sigma = 1$)	52.0	60.1	53.5	61.7	48.9	64.6
GDT ($\sigma = 3$)	52.5	61.5	55.3	64.3	55.1	64.4
GDT ($\sigma = 6$)	51.5	60.9	55.2	64.1	55.8	64.4
GDT ($\sigma = 10$)	51.1	60.4	54.7	64.4	55.9	64.4
LDT ($k = 2$)	52.3	60.5	54.0	62.6	50.6	64.7
LDT ($k = 5$)	52.6	61.8	55.3	64.4	55.0	64.7
LDT ($k = 10$)	52.0	61.3	55.4	64.2	55.8	64.7

The most significant result of the experiments is that IDT performs much better than all the other functions at all the three resolutions. The binary function performs poorly as expected. Its performance even deteriorates for increasing voxel resolution, since the finer resolutions result in greater binary mismatch among similar models. Since the shapes of GDT and LDT profiles are similar, their performances for corresponding apertures are close to each other. At resolution 64, small apertures yield poor performance when we do not use class-based ARR selection. Another observation is that we get a much bigger increase in NN as compared to DCG, when we increase the resolution. This is because

fine resolutions favor target models that are very similar to the query model. However for target objects in the same class that do not have good matching details with the query model, an increase in resolution does not necessarily raise their ranks. With IDT, we gain 2 points for NN and 0.6 points for DCG when the voxel resolution goes from 32 to 64. These observations have lead us to adopt $R = 32$ as a compromise resolution in all following experiments.

7.3. Performance analysis of subspace methods

7.3.1. Training phase

In order to select the dimensionality of the PCA, ICA and NMF subspaces, we perform experiments on the training set of Princeton Shape Benchmark. We either leave the training set without any pose correction (NoARR) or apply mean shape-based (MbARR) or class-based ARR selection (CbARR), the latter two with the goal of selecting the best representative of 48-ARR versions of each model in the training set. Once the subspaces and their basis vectors are obtained, we extract the feature vectors corresponding to the 48-ARR versions of each model in the training set. We apply the MIN rule (Section 6) to match the sets of feature vectors of the query and target models for the NoARR and MbARR cases. However, for the CbARR case, we directly use the best representative of 48-ARR versions of each model and do not perform 48×48 comparisons between query and target models.

Fig. 12a–c shows DCG versus subspace dimension curves obtained with PCA, ICA and NMF, respectively. For all subspaces, class-based ARR selection boosts the retrieval performance; since we greatly reduce the 90° pose ambiguities within classes. The performance of PCA remains insensitive to increasing dimension, since higher order PCA coefficients have lower impact on the similarity of the models. With ICA, the performance is quite sensitive to the dimensionality. We have a peak performance at dimension 40, regardless of the alignment scheme of the training models (Fig. 12b). NMF-based retrieval scheme yields stable results with increasing dimension as compared to ICA, although the DCG values fluctuate a little due to random initialization of NMF basis vectors. With these observations, we set the dimension of PCA-based subspace to 100 for the PSB test set experiments. For ICA, we set the dimension either to 40, following the peak of DCG with the training set or to 100 in order to have the same dimension with PCA. Likewise, for the NMF-based experiments, we report the results of dimensions 70 and 100.

7.3.2. Performance on PSB test set

Regardless of which ARR selection scheme we have used in the training phase, we do not use any class information in the experiments conducted over the test set. In Table 3 we give the retrieval

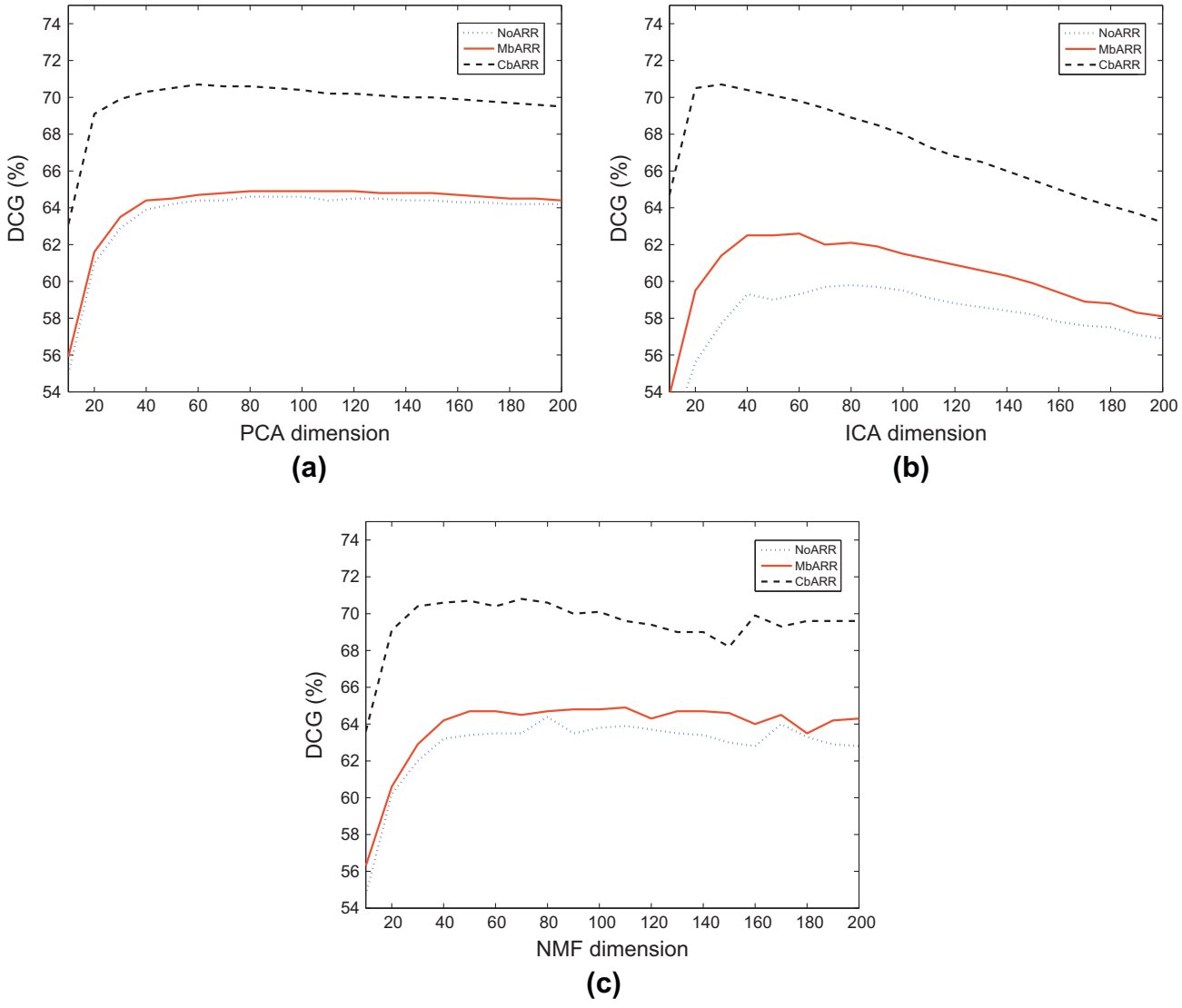


Fig. 12. DCG versus subspace dimension for PCA (a), ICA (b) and NMF (c). Experiments are conducted on the PSB training set.

Table 3
Retrieval performances on the PSB test set. The pose correction is only performed on the PSB training set during the subspace building phase. MIN rule is used to match query and target models.

Subspace	Dimension	Pose correction	NN	FT	ST	DCG
PCA	100	MbARR	61.2	58.0	57.1	61.6
		CbARR	61.7	58.4	56.8	61.5
ICA	40	MbARR	57.8	56.2	53.6	59.8
		CbARR	58.4	55.2	52.9	59.2
	100	MbARR	62.1	59.2	56.8	61.2
		CbARR	62.6	59.8	57.0	61.4
NMF	70	MbARR	61.0	58.1	55.7	61.1
		CbARR	60.3	58.9	56.3	60.7
	100	MbARR	62.0	60.0	56.4	61.5
		CbARR	61.7	59.1	56.9	61.0

Table 4
Retrieval performances on the PSB test set. The pose correction is only performed on the PSB training set during the subspace building phase. Pose assignment strategy is used to match query and target models.

Subspace	Dimension	Pose correction	NN	FT	ST	DCG
PCA	100	MbARR	63.2	37.1	48.1	63.4
		CbARR	63.5	37.0	48.2	63.4
ICA	40	MbARR	66.2	38.4	51.2	65.0
		CbARR	66.4	38.5	50.7	64.8
	100	MbARR	66.5	39.5	51.4	65.5
		CbARR	66.5	39.4	51.5	65.6
NMF	70	MbARR	66.3	38.6	50.3	64.9
		CbARR	66.9	38.5	50.4	64.7
	100	MbARR	66.8	39.0	50.7	65.0
		CbARR	66.9	38.7	50.0	65.0

performances of the three subspaces obtained on PSB test set, with the MIN rule, whereas the results in Table 4 are obtained using pose assignment strategy (Section 6). Clearly, pose assignment strategy provides a significant performance gain.

Table 5
Retrieval performances on the PSB test set when the database models are pre-filtered.

Filtering threshold	Average # of full comparisons per query	Subspace	NN	FT	ST	DCG
No filtering	906	PCA	63.5	37.0	48.2	63.4
		ICA	66.5	39.4	51.5	65.6
		NMF	66.9	38.7	50.0	65.0
0.4	550	PCA	63.7	36.6	47.3	62.9
		ICA	66.4	38.9	50.2	64.9
		NMF	66.9	38.2	49.0	64.3
0.5	665	PCA	63.6	36.8	47.8	63.2
		ICA	66.7	39.3	50.9	65.2
		NMF	66.9	38.4	49.6	64.6
0.6	738	PCA	63.6	37.0	48.0	63.3
		ICA	66.6	39.3	51.2	65.4
		NMF	66.9	38.5	49.9	64.8

Fig. 13 gives the precision–recall curves for the three subspace methods. The curves correspond to the case where we use the pose assignment strategy. When we compare the three subspaces, we can observe that the performance of PCA-based scheme is lower than the ICA and NMF-based schemes. ICA and NMF subspaces give comparable results, although ICA performs slightly better. We can see that the class-based ARR selection of the training set brings almost no gain to the performance on the test set. Another disparity between the training and test cases is about the dimension. Although the ICA performance drops beyond dimension 40 in the retrieval experiments on the training set (Fig. 12b), when we switch to the test set, we have performance gains with higher ICA subspace dimensions. Some classes in the PSB test set are not present in the training set, therefore fine tuning of the parameters for the training set does not necessarily reflect on the test set. However, an inspection of Table 4 reveals that we do not have dramatic dependency on the selection of the pose correction scheme

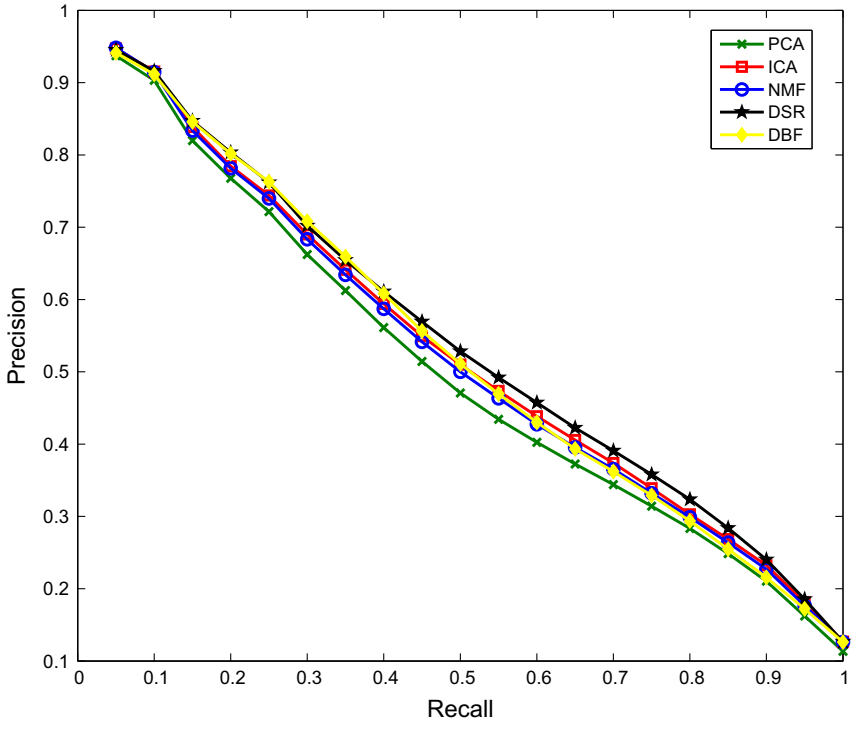


Fig. 13. Precision–recall curves on the PSB test set. Mean shape-based pose correction is applied to the training set. Pose assignment strategy is used to match query and target models.

or the dimension. Therefore we do not need to have a labeled training set to incorporate class-based alignment while constructing the subspace models.

7.4. Filtering the search space

To reduce the computational cost resulting from the calculation of 48×48 comparison scores between model pairs, we propose a filtering method for database models prior to the full comparison with the query model in order to quickly eliminate the irrelevant ones and focus our attention on relevant candidates. The subspace methods studied in this paper provide global shape descriptors in the sense that the entire model is projected onto the basis shapes. Thus, a big difference in the gross shape between two models will result in a high dissimilarity score in the subspace domain. One distinguishing characteristic of the gross shape is the elongation, and we measure it using the ratios of the eigenvalues obtained from the CPCA normalization (Section 3.1).

Here, we use the filtering scheme described in our recent work [42]. Let $\{\alpha_1, \alpha_2, \alpha_3\}$ be the eigenvalues of the covariance matrix of the x, y and z coordinates on the mesh model. These eigenvalues give a measure of the extent of the object in the principal directions. We define two ratios of these eigenvalues as follows: $e_1 = \alpha_2/\alpha_1$ and $e_2 = \alpha_3/\alpha_2$. These ratios can be used as rough

Fig. 14. Class-based ARR selection for bench seat and rectangular table classes. The top red figures are the reference models. Pink figures at the left are the outputs of CPCA-based normalization. Cyan figures under the reference models are the best choice out of the 48-ARR representations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Retrieval performances on the PSB test set assuming that correct axis labeling and reflection of each model are known.

Subspace	Dimension	NN	FT	ST	DCG
PCA	100	70.0	43.0	53.8	68.3
ICA	40	69.6	43.7	55.2	68.6
	100	72.4	43.4	53.9	68.9
NMF	70	70.3	43.7	54.9	68.9
	100	71.4	43.8	55.1	69.2

Table 7
Comparison of subspace methods with the state-of-the-art 3D shape descriptors on PSB test set.

Descriptor	NN	FT	ST	DCG
CRSP	67.9	40.5	52.8	66.8
DSR	66.5	40.3	51.2	66.5
DBF	68.6	39.3	50.0	65.9
ICA	66.5	39.5	51.4	65.5
NMF	66.8	39.0	50.7	65.0
LFD	65.7	38.0	48.7	64.3
PCA	63.2	37.1	48.1	63.4

descriptors of the global geometry of the object. When a query model is introduced to the system, we first calculate the distance between the eigenvalue ratios of the query model and each database model as $\sqrt{(e_1^{query} - e_1^{database})^2 + (e_2^{query} - e_2^{database})^2}$. The models, for which this distance is higher than a filtering threshold t_f, are assumed to be completely different from the query, and hence we do not make further comparison in the subspace domain. Their distance is set to infinity and these database models are cast at the end of the rank list.

Table 5 gives the retrieval results of the subspace method, with no filtering applied and when filtered with different threshold values. CbARR pose correction is used and the subspace dimension is set to 100 for all methods. For the three given threshold values, the retrieval performance remains stable and does not drop significantly while, with a threshold of 0.4 for instance, more than one third of the database models are filtered out on average prior to the full comparison with query. Hence, this simple filtering scheme reduces the retrieval cost by about 40% without a significant loss in the performance.

7.5. The importance of the correct pose

To prove the importance of the within-class coherence of axes labeling and reflection, we carry out a hypothetical experiment where we assume that the best 48-ARR pose of each model is known, and report the corresponding performance results on the PSB test set. We perform a class-based pose correction on the models of the test set using direct comparisons method, then obtain the 48-ARR versions of the IDT representation of a model, and select the one that results in the least L_1 error with the class representative (Fig. 14). Table 6 gives the performance of various descriptors with this ideal case. We can observe the boost in the performance when we compare the results with those in Table 4, and this comparison shows that coherent axes labeling is crucial when PCA normalization is applied to the models. With the pose assignment strategy we try to achieve the ideal results presented in Table 6.

7.6. Comparison with state-of-the-art

In order to demonstrate the potential of subspace techniques for 3D model retrieval, we compare our results to the state-of-the-art methods (Table 7). We select the four top performing methods that were evaluated in [27,28], namely, concrete radialized spherical projection (CRSP) [29], DSR descriptor [11], density based framework (DBF) [27,28], and light field descriptor (LFD)

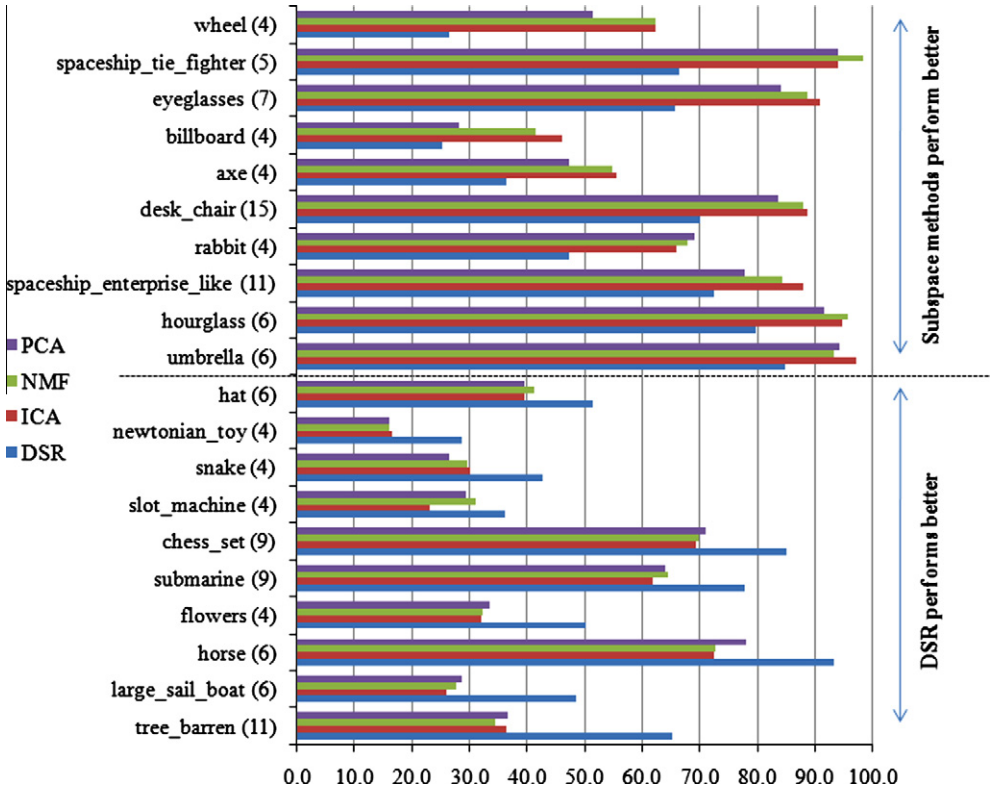


Fig. 15. Class-based DCG values for DSR method vis-á-vis the subspace-based methods.

[43]. The CRSP scheme decomposes the models into a set of spherical functions, which are then encoded using spherical harmonics. The DSR descriptor is a hybrid descriptor that combines depth buffer and silhouette-based descriptors. The DBF characterizes models using multivariate probability density functions of local surface features. The LFD is a collection of views of an object from uniformly sampled points on a sphere. With the exception of LFD, the three methods use CPCA for pose normalization. In CRSP, the normalization is even enhanced with another PCA normalization that is based on surface normals. None of the four descriptors employ learning schemes based on any class information of training or target models. Similarly we abstain in this comparison from any class information and use unsupervised ARR selection based on the mean shape of the training database (Section 5.4). In addition to Table 7, we provide precision–recall curves of the DSR and DBF methods in comparison with our subspace methods (Fig. 13).

7.6.1. Class-based observations

To acquire more insight into the performance values provided in Table 7, one needs to observe how well a specific method performs in response to certain types of objects. To this effect, in Fig. 15 we consider the class-based averages of the DCG values obtained by the DSR method and our subspace-based methods. We calculated the DCG differential between the ICA-based method and DSR method and sorted the classes according to this differential. We chose ICA instead of NMF or PCA, since ICA performs similar to NMF and better than PCA. The top 10 object classes in the figure are those on which ICA-based method performed better than the DSR method whereas the bottom 10 classes are those on which DSR method performed better. Fig. 16 is obtained similarly where the comparisons are between the subspace-based methods and the DBF method. Note that in Figs. 15 and 16; we give the number of models belonging to that class attached to the name of each class.

Based on Figs. 15 and 16, and the retrieved models in response to some query examples (Figs. 18–21) we can make the following comments:

- We observe that ICA and NMF-based methods yield similar DCG values to each other. PCA is usually trailing behind.
- For some classes, all methods, from subspace methods to DSR and DBF, result in poor DCG values; that is lower than 50%. Examples are "newtonian_toy", "snake", "slot_machine", and "flowers" (see the thumbnails in Fig. 17). These are difficult cases mainly because there are very few objects to be retrieved at the top of the list.
- The "newtonian_toy" objects show high intra-class shape variation. They are "semantically" similar in the sense that they both have four balls hanging from a stand. The "snake" class on the other hand shows a different type of shape variety; namely articulation based. None of the methods listed in Table 7 are designed to deal with large articulations within classes. The "flower" class possesses detailed shape varieties due to its branches.
- The problem with the "slot_machine" is rather different: Its shape lacks class-specific details so it is matched to similar looking rectangular shapes. In Fig. 18, in response to a query model "slot_machine" (with green background), we display the most similar three models retrieved by our ICA-based method, DSR and DBF methods.
- In Fig. 15, we see that the highest DCG difference between the subspace methods and the DSR method belongs to the class "tree_barren". In Fig. 19, in response to a query model "tree_barren", we display the most similar seven models retrieved by the ICA-based method and the DSR method. The

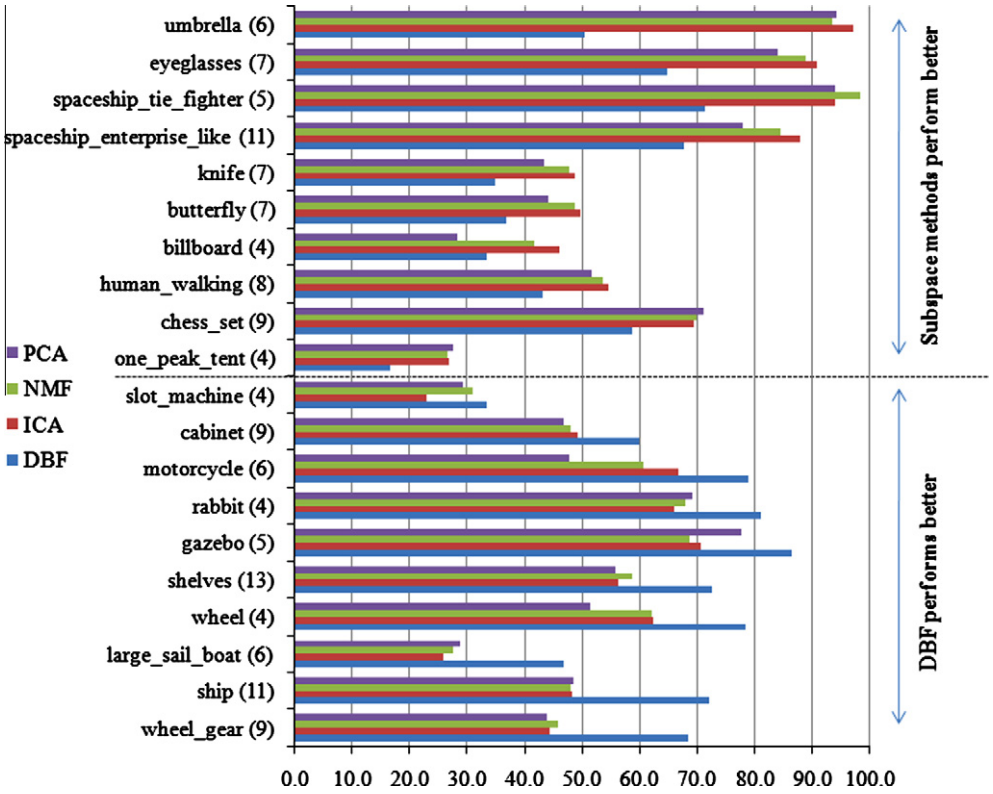


Fig. 16. Class-based DCG values for DBF method and the subspace-based methods.

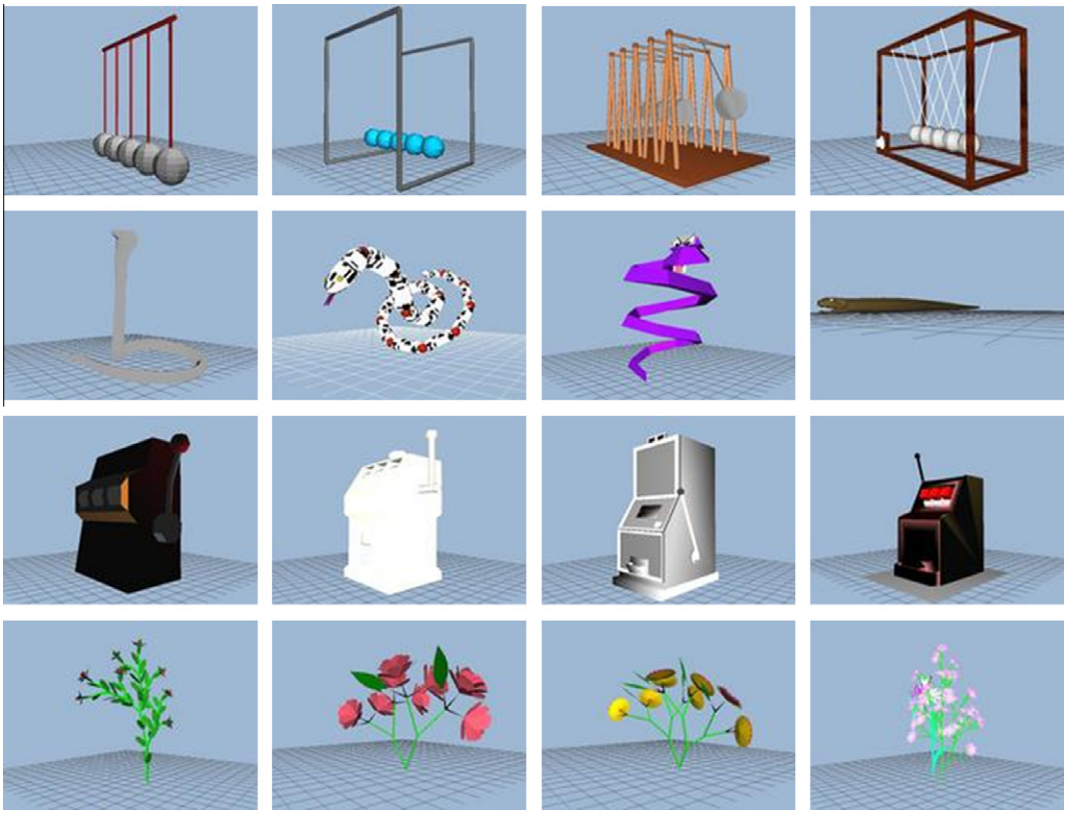
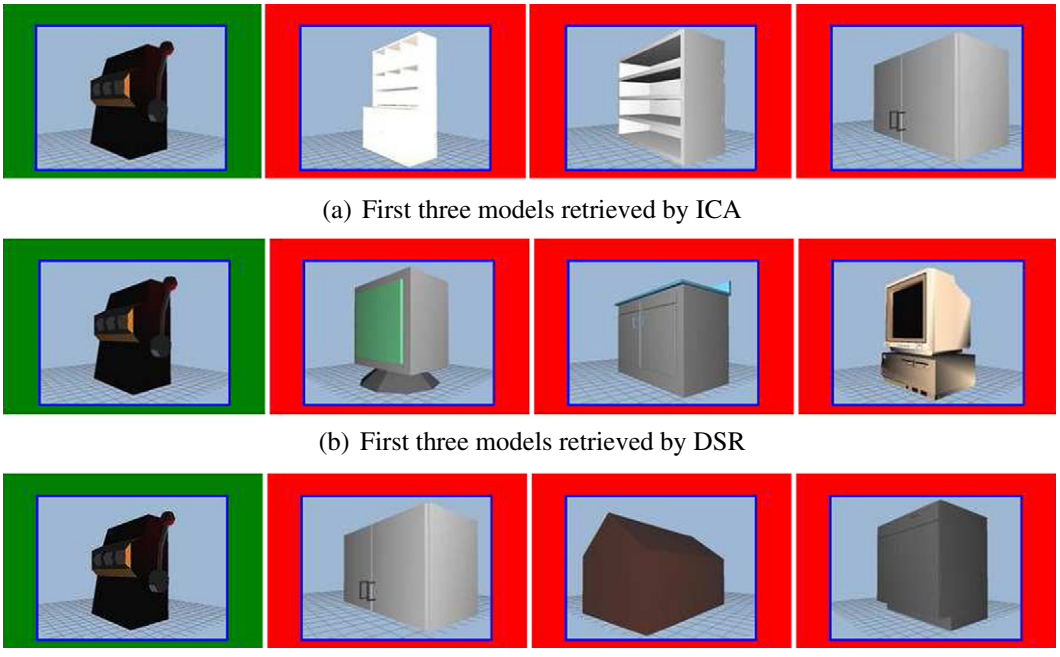


Fig. 17. Objects from classes for which both methods give poor results.



(a) First three models retrieved by ICA

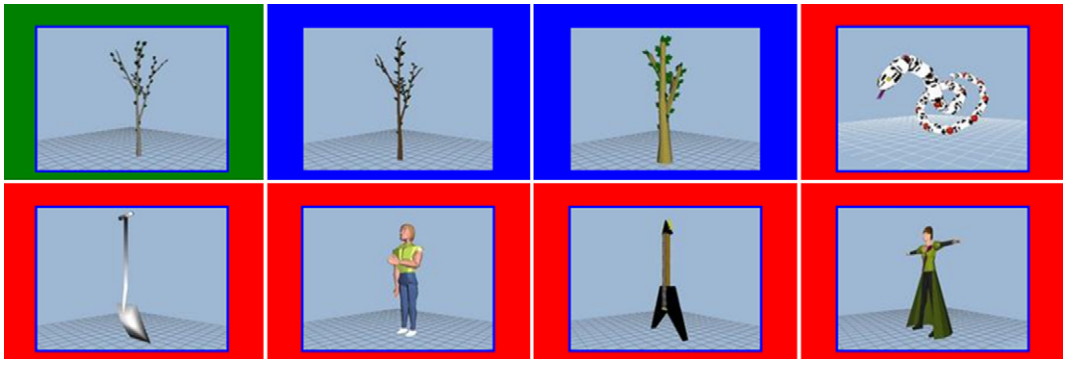
(b) First three models retrieved by DSR

(c) First three models retrieved by DBF

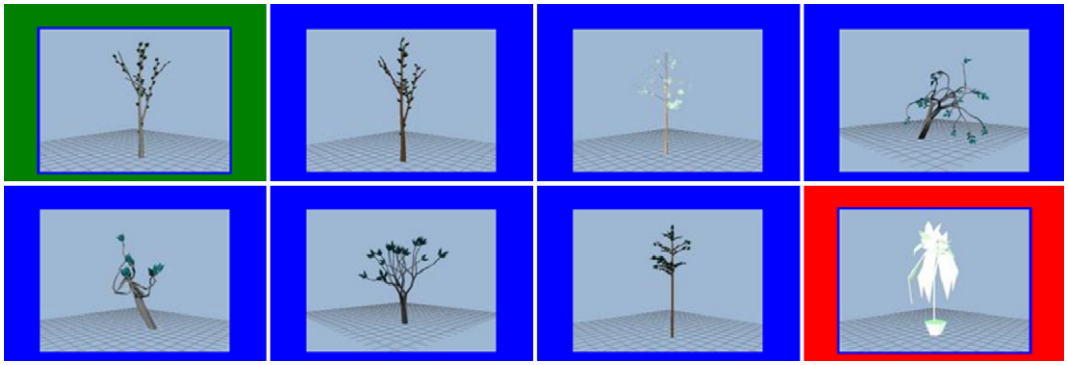
Fig. 18. First three models retrieved by ICA (a), DSR (b), and DBF (c), in response to a "slot_machine" query.

models retrieved by ICA shows that the branches of some of the tree models are not resolved by our voxel-based method. One reason is that we use $32 \times 32 \times 32$ voxel resolution; whereas the DSR method uses depth buffer data of resolution 256×256. Another reason is that the distance transform fills in the small gaps.

- In Fig. 16, where we compare our subspace methods with the DBF descriptor, we see that the highest DCG differences occur in classes "ship" and "wheel_gear". From Fig. 20, we can observe that ICA matched the gear with other disk-like shapes regardless of its cogs. Similarly in Fig. 21, ICA could not resolve the small details between the ships and the submarines.

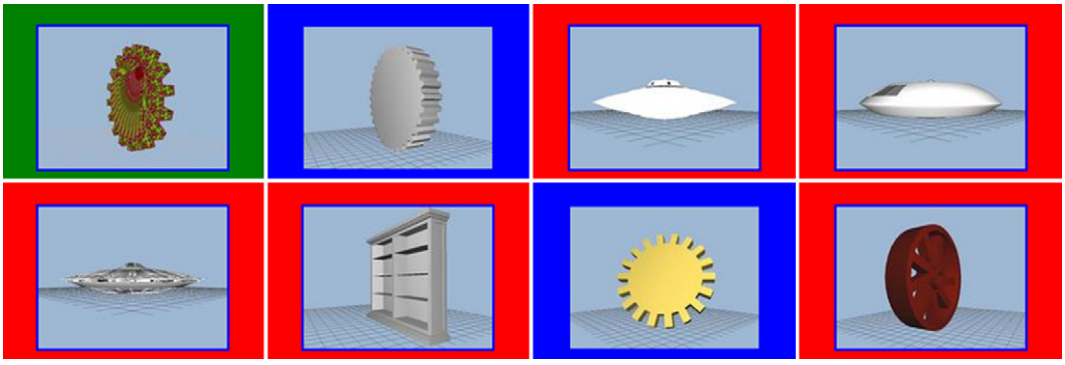


(a) First seven models retrieved by ICA

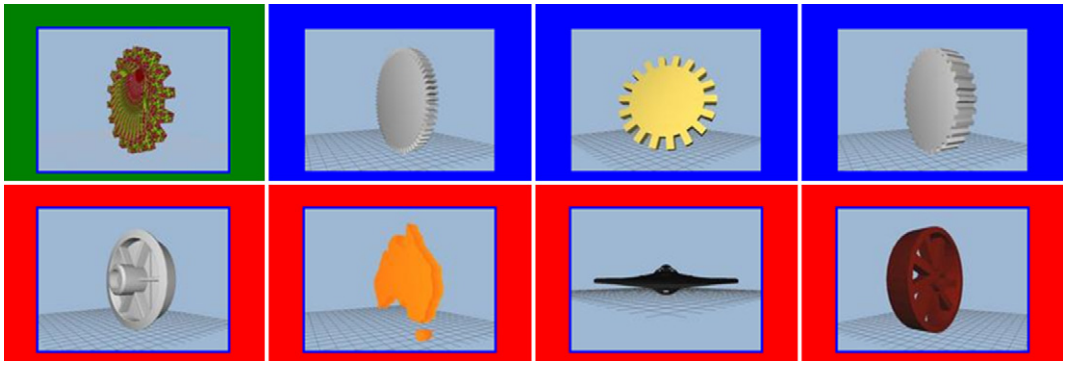


(b) First seven models retrieved by DSR

Fig. 19. First seven models retrieved by ICA (a), and DSR (b), in response to a "tree_barren" query.

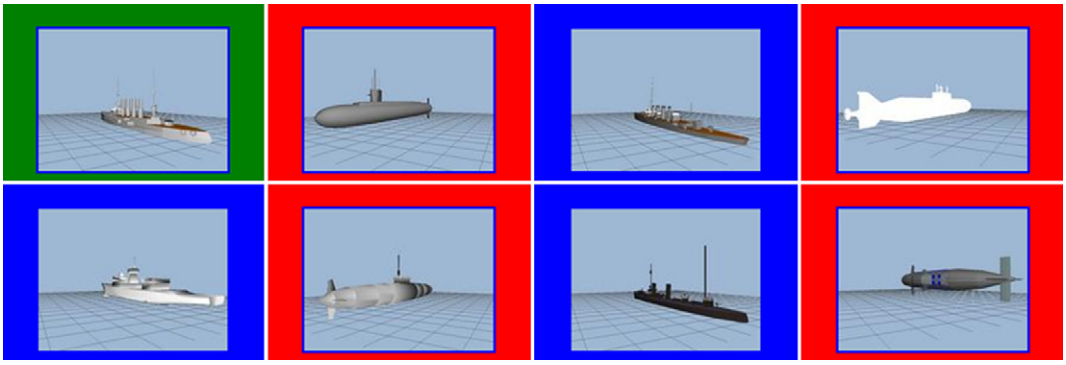


(a) First seven models retrieved by ICA

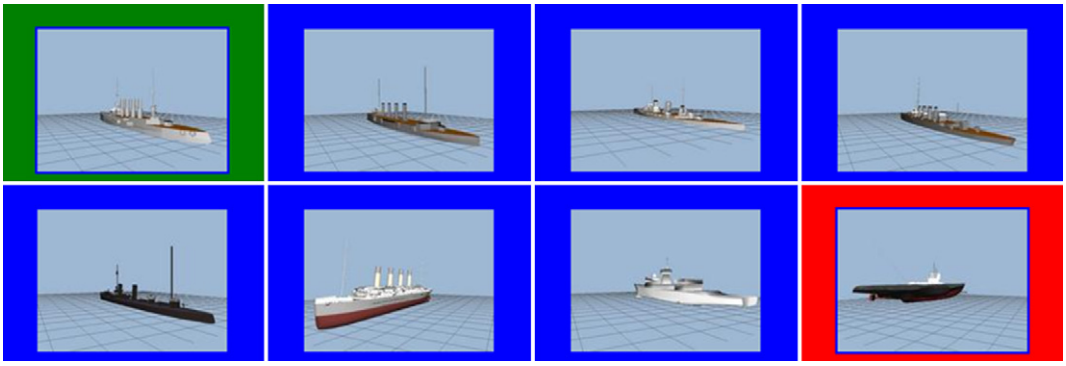


(b) First seven models retrieved by DBF

Fig. 20. First seven models retrieved by ICA (a), and DBF (b), in response to a "wheel_gear" query.



(a) First seven models retrieved by ICA



(b) First seven models retrieved by DBF

Fig. 21. First seven models retrieved by ICA (a), and DBF (b), in response to a "ship" query.

- Despite these shortcomings of our subspace methods; they perform better than DSR or DBF methods on nearly half of the classes. The DCG values obtained by ICA are higher than those of DSR for 44 classes out of the 93 classes of PSB test set. Similarly ICA gives higher DCG results than DBF for 41 classes.

7.6.2. Fusion of methods

Since the subspace methods and the DSR and DBF descriptors respond differently to different classes, we found it useful to report the gain obtained from the their fusion. Table 8 gives performance results obtained with fusion of the subspace methods with each other and with the DSR and DBF descriptors. The fusion is performed via the summation of the scores obtained from each descriptor. The fusion of subspace methods with each other does not bring much gain, except for a modest gain in the NN measure for the fusion of ICA and NMF. However, when we combine the ICA and NMF-based descriptors with the DSR or DBF, we get a significant improvement in retrieval performance. These results show that the subspace methods can be even more beneficial when used in combination with other methods. Indeed, when we combine the DSR, DBF, ICA and NMF methods, we achieve the highest performance reported so far on the PSB test set, among the unsupervised retrieval methods in the literature.

Table 8
Fusion of subspace methods and DSR and DBF descriptors.

	NN	FT	ST	DCG
Fusion				
PCA + ICA	67.0	39.7	51.3	65.6
PCA + NMF	65.9	38.8	50.6	64.9
ICA + NMF	67.3	39.4	51.3	65.5
PCA + ICA + NMF	66.7	39.6	51.2	65.5
Fusion with DSR				
DSR + PCA	67.7	41.6	53.1	67.4
DSR + ICA	69.3	44.2	55.4	69.1
DSR + NMF	69.1	43.8	55.1	68.8
DSR + ICA + NMF	71.0	44.7	56.1	69.6
Fusion with DBF				
DBF + PCA	69.2	40.5	51.4	66.8
DBF + ICA	70.5	42.5	53.7	68.2
DBF + NMF	70.2	41.9	53.0	67.8
DBF + NMF + ICA	70.1	43.3	54.6	68.7
Fusion with DBF and DSR				
DBF + DSR	73.4	45.0	56.2	70.2
DBF + DSR + NMF + ICA	73.6	46.2	57.7	71.1

8. Conclusion

In this work, we have developed 3D model retrieval schemes using various subspaces of object shapes. We have investigated the potential of three popular subspace techniques, namely PCA, ICA and NMF, since each of them describes somewhat different statistical characteristics of the data. The strength of subspace methods is that they can capture succinctly the shape characteristics and being generative methods, they can easily gloss over minor differences and defects. The downside of them is that they are affected by gross pose uncertainties. However, the pose dependency of the subspace methods is solved for by the use of CPCA-based pose normalization, followed by voxelization, inverse distance transform and exhaustive pose optimization.

The two main results of our research is, first that ICA and NMF-based schemes provide retrieval performances on a par with the alternative state-of-the-art methods, and second that decision fusion of these schemes advances the performance on Princeton Shape Benchmark beyond that of any one method in the literature.

We conjecture that there is still room for performance improvement. Our future research effort will concentrate on the following:

- The subspace methods can be applied on alternative representations of the data, for example on the point cloud or the depth image representations instead of the voxel data.
- Robust versions of subspace building can offer enhanced solutions, especially when the data is corrupted by outliers, which can potentially adversely affect the performance. In this respect, kernel PCA, sparse PCA [44,45], robust PCA [46] or other variants of ICA and NMF can be adopted and compared.
- The matching strategy can be improved by considering finer pair-wise alignment of models. One can consider matching manifolds of projections, obtained by fine sampling of the rotation space instead of simply using the possible mirror reflections and axis relabelings.

Acknowledgment

This research was supported by Boğaziçi University fund 03A203 and by TUBITAK project 103E038.

References

[1] T. Mcinerney, D. Terzopoulos, Deformable models in medical image analysis: a survey, Medical Image Analysis 1 (1996) 91–108.
[2] M. Styner, G. Gerig, Medial models incorporating shape variability, in: Inf. Proc. in Med. Imaging, 2001, Springer, pp. 502–516.
[3] S.C. Joshi, M.I. Miller, U. Grenander, On the geometry and shape of brain submanifolds, International Journal of Pattern Recognition and Artificial Intelligence 11 (1997) 1317–1343.
[4] K.W. Bowyer, K. Chang, P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition, Computer Vision and Image Understanding 101 (1) (2006) 1–15.
[5] B. Gokberk, H. Dutagaci, A. Ulas, L. Akarun, B. Sankur, Representation plurality and fusion for 3-D face recognition, IEEE Transactions on Systems Man and Cybernetics Part B 38 (1) (2008) 155–173.
[6] A. Godil, Y. Ressler, P. Grother, Face recognition using 3D facial shape and color map information: comparison and combination, in: Proceedings of the SPIE – The International Society for Optical Engineering, 2006, pp. 351–361.
[7] D.V. Vranic, D. Saupe, J. Richter, Tools for 3d-object retrieval: Karhunen–Loeve transform and spherical harmonics, in: Proceedings of IEEE 2001 Workshop on Multimedia Signal Processing, 2001, pp. 293–298.
[8] B. Bustos, D.A. Keim, D. Saupe, T. Schreck, D.V. Vranić, Feature-based similarity search in 3d object databases, ACM Computing Surveys 37 (4) (2005) 345–387.
[9] J.W. Tangelder, R.C. Veltkamp, A survey of content based 3d shape retrieval methods, in: SMI '04: Proceedings of the Shape Modeling International 2004, IEEE Computer Society, 2004, pp. 145–156.
[10] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Computer-Aided Design 37 (5) (2005) 509–530.
[11] D.V. Vranic, 3D Model Retrieval, Ph.D. Thesis, University of Leipzig, 2004.
[12] M. Kazhdan, Shape Representations and Algorithms for 3D Model Retrieval, Ph.D. Thesis, Princeton University, 2004. <http://www.cs.princeton.edu/gfx/pubs/_2004_SRA>.
[13] C.B. Akgül, Density-based Shape Descriptors and Similarity Learning for 3D Object Retrieval, Ph.D. Thesis, Bogazici University, March 2007. <http://www.tsi.enst.fr/akgul/pdfs/thesis-cba-web.pdf>.
[14] S. Goodall, 3-d Content-based Retrieval and Classification with Applications to Museum Data, Ph.D. Thesis, University of Southampton, March 2007. <http://eprints.ecs.soton.ac.uk/13812/>.
[15] H.-P. Kriegel, P. Kröger, Z. Mashael, M. Pfeifle, M. Pötke, T. Seidl, Effective similarity search on voxelized cad objects, in: DASFAA '03: Proceedings of the Eighth International Conference on Database Systems for Advanced Applications, 2003, pp. 27.
[16] H.-P. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle, M. Schubert, Using sets of feature vectors for similarity search on voxelized cad objects, in: SIGMOD '03: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, 2003, pp. 587–598.
[17] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, T. Funkhouser, A reflective symmetry descriptor, in: Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), 2002, pp. 642–656.
[18] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3D models, ACM Transactions on Graphics 22 (1) (2003) 83–105.
[19] J. Ricard, D. Coeurjolly, A. Baskurt, Generalizations of angular radial transform for 2D and 3D shape retrieval, Pattern Recognition Letters 26 (14) (2005) 2174–2186.
[20] M. Novotni, R. Klein, A geometric approach to 3D object comparison, in: SMI '01: Proceedings of the International Conference on Shape Modeling & Applications, 2001, p. 167.
[21] M. Novotni, R. Klein, 3D zernike descriptors for content based shape retrieval, in: SM '03: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, 2003, pp. 216–225.
[22] M.T. Suzuki, T. Kato, N. Otsu, A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors, IEEE International Conference on Systems, Man, and Cybernetics 4 (2000) 2946–2952.
[23] H. Sánchez-Cruz, E. Bribiesca, A method of optimum transformation of 3D objects used as a measure of shape dissimilarity, Image and Vision Computing 21 (12) (2003) 1027–1036.
[24] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, S. Rusinkiewicz, A reflective symmetry descriptor for 3d models, Algorithmica 38 (1) (2003) 201–225.
[25] R. Gal, D. Cohen-Or, Salient geometric features for partial shape matching and similarity, ACM Transactions on Graphics 25 (1) (2006) 130–150.
[26] E. Paquet, Description of shape information for 2-d and 3-d objects, Signal Processing: Image Communication 16 (2000) 103–122.
[27] C.B. Akgül, B. Sankur, Y. Yemez, F. Schmitt, Density-based 3D shape descriptors, EURASIP Journal on Advances in Signal Processing 2007 (1) (2007) 209.
[28] C.B. Akgül, B. Sankur, Y. Yemez, F. Schmitt, 3D model retrieval using probability density-based shape descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (6) (2009) 1117–1133.
[29] P. Papadakis, I. Pratikakis, S. Perantonis, T. Theoharis, Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation, Pattern Recognition 40 (9) (2007) 2437–2452.
[30] D.V. Vranic, D. Saupe, 3d model retrieval, in: Spring Conference on Computer Graphics and its Applications, 2000, pp. 89–93.
[31] D.V. Vranic, An improvement of rotation invariant 3d shape descriptor based on functions on concentric spheres, in: Proceedings of IEEE International Conference on Image Processing, vol. 3, 2003, pp. 757–760.
[32] Y. Liu, J. Pu, H. Zha, W. Liu, Y. Uehara, Thickness histogram and statistical harmonic representation for 3D model retrieval, in: 3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium, 2004, pp. 896–903.
[33] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3d shape descriptors, in: SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2003, pp. 156–164.
[34] J. Ricard, D. Coeurjolly, A. Baskurt, Art extension for description, indexing and retrieval of 3D objects, in: ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3, 2004, pp. 79–82.
[35] Z.B. Azouz, C. Shu, R. Lepage, M. Rioux, Extracting main modes of human body shape variation from 3-D anthropometric data, in: 3DIM '05: Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling, 2005, pp. 335–342.
[36] A. Ruto, M. Lee, B. Buxton, Comparing principal and independent modes of variation in 3D human torso shape using PCA and ICA, in: ICArn 2006, ICA Research Network International Workshop, 2006.
[37] C.C. Aggarwal, A. Hinneburg, D.A. Keim, On the surprising behavior of distance metrics in high dimensional space, in: Lecture Notes in Computer Science, Springer, 2001, pp. 420–434.
[38] I. Noy-Meir, Data transformations in ecological ordination: I. Some advantages of non-centering, The Journal of Ecology 61 (2) (1973) 329–341.
[39] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications., Neural Networks 13 (4–5) (2000) 411–430.
[40] D.D. Lee, H.S. Seung, Learning the parts of objects by nonnegative matrix factorization, Nature 401 (1999) 788–791.
[41] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: SMI '04: Proceedings of the IEEE International Conference on Shape Modeling and Applications, 2004, pp. 167–178.
[42] H. Dutagaci, A. Godil, B. Sankur, Y. Yemez, View subspaces for indexing and retrieval of 3d models, in: Proc. SPIE, Three-Dimensional Image Processing (3DIP) and Applications, vol. 7526, 2010.
[43] D.-Y. Chen, M. Ouhyoung, A 3D object retrieval system based on multi-resolution reeb graph, in: Proceedings of Computer Graphics Workshop, 2002, pp. 16–20.
[44] R. Zass, A. Shashua, Nonnegative sparse PCA, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems, vol. 19, MIT Press, Cambridge, MA, 2007, pp. 1561–1568.
[45] K. Huang, S. Aviyente, Sparse representation for signal classification, in: NIPS, 2006, pp. 609–616.
[46] F.D.L. Torre, M.J. Black, A framework for robust subspace learning, International Journal of Computer Vision 54 (1–3) (2003) 117–142.