3D Shape Matching with 3D Shape Contexts

Marcel Körtgen[*]
Gil-Joo Park[†]
Marcin Novotni[‡]
Reinhard Klein[§]

University of Bonn
Institute of Computer Science II.
Römerstr. 164, D-53117 Bonn
Germany

Abstract

Content based 3D shape retrieval for broad domains like the World Wide Web has recently gained considerable attention in Computer Graphics community. One of the main challenges in this context is the mapping of 3D objects into compact canonical representations referred to as descriptors or feature vector, which serve as search keys during the retrieval process. The descriptors should have certain desirable properties like invariance under scaling, rotation and translation as well as a descriptive power providing a basis for similarity measure between three-dimensional objects which is close to the human notion of resemblance.

In this paper we introduce an enhanced 3D approach of the recently introduced 2D Shape Contexts that can be used for measuring 3d shape similarity as fast, intuitive and powerful similarity model for 3D objects. The Shape Context at a point captures the distribution over relative positions of other shape points and thus summarizes global shape in a rich, local descriptor. Shape Contexts greatly simplify recovery of correspondences between points of two given shapes. Moreover, the Shape Context leads to a robust score for measuring shape similarity, once shapes are aligned.

Keywords: Quadratic Form Distances, Principal Axes Transform, Dimensionality Reduction, Histograms, Bipartite Graph Matching, Multistep Query Processing, Nearest Neighbor, Image Matching, Vector Quantization, Clustering, Multidimensional Index, Image Representation, Information Storage and Retrieval, Information Search and Retrieval

1 Introduction

It can be observed that the proliferation of a specific digital multimedia data type (e.g. text, images, sounds, video) was followed by emergence of systems facilitating their content based retrieval. With the recent advances in 3D acquisition techniques, graphics hardware and modeling methods, there is an increasing amount of 3D objects spread over various archives: general objects commonly used e.g. in games or VR environments, solid models of industrial parts, etc. On the other hand, modeling of high fidelity 3D objects is a very cost and time intensive process – a task which one can potentially get around by reusing already available models. Another important issue is the efficient exploration of scientific data represented as 3D entities. Such archives are becoming increasingly popular in the areas of Biology, Chemistry, Anthropology and Archeology to name a few. Therefore, since recently, concentrated research efforts are being spent on elaborating techniques for efficient content based retrieval of 3D objects.

One of major challenges in the context of data retrieval is to elaborate a suitable canonical characterization of the entities to be indexed. In the following, we will refer to this characterization as a descriptor. Since the descriptor serves as a key for the search process, it decisively influences the performance of the search engine in terms of computational efficiency and relevance of the results. A simple approach is to annotate the entities with keywords, however, due to the inherent complexity and multitude of possible interpretations this proved to be incomplete, insufficient and/or impractical for almost all data types, cf. [46, 18].

Guided by the fact that for a vast class of objects the shape constitutes a large portion of abstract object information, we focus in this paper on general shape based object descriptors. We now can state some requirements that a general shape based descriptor should obey:

1. **Descriptive power** - the similarity measure based on the descriptor should deliver a similarity ordering that

---

[*]e-mail: marcel@koertgen.de
[†]e-mail:parkg@zeus.informatik.uni-bonn.de
[‡]e-mail: marcin@cs.uni-bonn.de
[§]e-mail:rk@cs.uni-bonn.de

is close to the application driven notion of resemblance.

2. **Robustness** - the descriptor should be insensitive to noise and small extra features and robust against arbitrary topological degeneracies. These requirements are relevant e.g. in case of search on the World Wide Web for general objects, since such objects are likely to contain these artifacts.

3. **Invariance under transformations** - the computed descriptor values have to be invariant under an application dependent set of transformations. Usually, these are the similarity transformations Rotation, Translation and Uniform Scale.

4. **Conciseness and ease of indexing** - the descriptor should be compact in order to minimize the storage requirements and accelerate the search by reducing the dimensionality of the problem. Very importantly, it should provide some means of indexing – structuring the database in order to further accelerate the search process.

The outline of the rest of the paper is as follows: in the next section we review the relevant previous work. In Section 3 we describe the 3D Shape Contexts themselves. In Section 4 the matching is explained and reviewed in terms of accordance with the above criteria and 3D shape retrieval performance. In Section 6 we present our results and conclude in Section 7.

2 Previous Work

2.1 Systems

Up to date numerous systems for 2D image retrieval have been introduced. To gain a good overview over the state-of-the-art in this area we refer to the survey papers [46, 20, 39]. As for content based retrieval of general 3D objects, the first system was introduced in [35], which was followed by [47], a very recent result is presented in [18]. Considering systems covering narrower domains, [1] deal with anthropological data, [37, 13] facilitate the retrieval of industrial solid models, [3] explores protein databases.

2.2 Spatial domain

The spatial domain shape analysis methods yield non-numeric results, usually an attributed graph, which encodes the spatial and/or topological structure of an object. Notably, in his seminal work Blum introduced the Medial Axis Transform (MAT) [10], which was followed by a number of extensions like shock graphs, see e.g. [45], shock scaffold [30], etc. Forsyth et al. [17] represent 2D image objects by spatial relationships between stylized primitives, [36] uses a similar approach. Further technique having a long tradition is the geon based representation [9]. As for 3D industrial solid models, [12, 31] capture geometric and engineering features in a graph, which is subsequently used for similarity estimation. Hilaga et al. [23] presented a method for general 3D objects utilizing Reeb graphs based on geodesic distances between points on the mesh, which enabled a deformation invariant recognition. The methods in this class are attractive since they capture the high level structure of objects. Unfortunately though, they are computationally expensive, most of them suffer from noise sensitivity, the underlying graph representation makes the indexing and comparison of objects very difficult.

2.3 Scalar transform

The scalar transform techniques capture global properties of the objects yielding generally vectors of scalar values as descriptors.

2.3.1 Projection based techniques

Some techniques both in 2D and in 3D are based on coefficients yielded by compression transforms like the cosine [40] or wavelet transform e.g. in [26]. Fourier descriptors [51] have been applied in 2D, however, these are hard to generalize to 3D due to the difficulties in parametrization of 3D object boundaries. Moments can generally be defined as projections of the function defining the object onto a set of functions characteristic to the given moment. Since Hu [24] popularized the usage of image moments in 2D pattern recognition, they have found numerous applications. Teague [48] was first to suggest the usage of orthogonal functions to construct moments. Subsequently, several 2D moments have been elaborated and evaluated [49]: geometrical, Legendre, Fourier-Mellin, Zernike, pseudo-Zernike moments. 3D geometrical moments have been used by [14, 33], and a spherical harmonic decomposition was used by Vranic and Saupe [50]. Funkhouser et al. [18] profit from the invariance properties of spherical harmonics and present an affine invariant descriptor. The main idea behind this is to decompose the 3D space into concentric shells and define rotationally invariant representations of these subspaces. In this way a descriptor was constructed which was proven to be superior over other 3D techniques with regard to shape retrieval performance. In [49] 2D Zernike moments were found to be superior over others in terms of noise sensitivity, information redundancy and discrimination power. Guided by this, Canterakis [11] generalized the classical 2D Zernike polynomials to 3D, however, in his work Canterakis considered exclusively theoretical aspects. However, all the approaches mentioned above do not provide richness and intuitivity since moments are based on projecting to lower dimensionalities.

3 3D Shape Contexts

Our representation for a 3d shape is a set of N histograms corresponding to N points sampled from the shape boundary, also referred to as a *Shape Distribution* [34], which need not (and typically will not) refer to keypoints such as maxima of curvature or inflection points. We prefer to sample the surface of the shape with roughly uniform spacing (cf. figure 1), though this is also not critical. The sampling method for constructing used throughout this paper adapts from Osada et al. [34]. It is a fast and efficient random sampling: The complexity for taking S samples from a 3D shape with N triangles is $O(S \log(N))$.

Figure 1: Roughly uniform sampling of a 3D object. a) unsampled mesh, b) mesh sampled with 500 samples and normals.

Now consider the set of vectors originating from one sample point to all other points in the shape (cf. figure 2). These vectors express the appearance of the entire shape relative to the reference point. Obviously, this set of $N-1$ vectors is a rich description, since as N gets large, the representation of the shape becomes exact.

   The full set of vectors as a shape descriptor is inappropriate, since shapes and their sampled representation may vary from one instance to another. In contrast, we identify the *distribution* over relative positions as a robust and compact, yet discriminative descriptor. For a point P on the shape, we compute a coarse histogram of the relative coordinates of the remaining $N-1$ points. This histogram is defined to be the **Shape Context** of P. The reference orientation for this shape context can be absolute or relative. In Section 3.1.2 we describe how to derive such a relative reference frame.

3.1 3D Shape Histograms

A common approach for similarity models is based on the paradigm of feature vectors. A *feature transform* maps a complex object onto a feature vector in multidimensional space. The similarity of two objects is then defined as the vicinity of their feature vectors in the feature space.

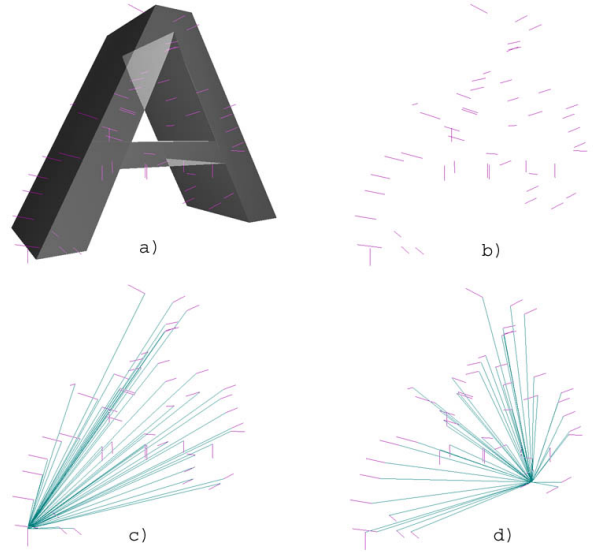   We follow this approach by introducing the 3D shape

Figure 2: a) Mesh with 50 samples, b) Just the 50 samples, c) 49 Vectors originating from one sample point, d) 49 Vectors originating from another sample point.

histograms as intuitive feature vectors. In general histograms are based on a partitioning of the space in which the object reside, i.e. a complete and disjoint decomposition into cells which correspond to the bins of the histograms. Figure 3 shows a 2D example of three types of basic space decompositions: the shell model, sector model and combined model.
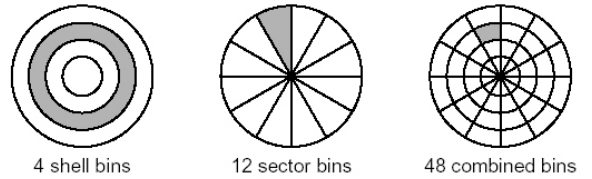
Figure 3: Shells and sectors as basic space decomposition for shape histograms. In each of the 2D examples a single bin is marked.

3.1.1 Shell Model

The 3D is decomposed into concentric shells around the center point. This representation is particularly independent from a rotation of the objects, i.e. any rotation of an object around the center point results in the same histogram. Invariance in scale is easily achieved by normalizing the shape extension and a [0,1]-parametrization of the shell-radii. With equal radii, however, the shell volumes grow quadratically with the shell index. To avoid weighting outer shells over inner shells we suggest a logarithmic parametrization of the shell radii (cf. figure 4). The radius

r of shell i then computes dependent of the log-base a and the number of shells s:

$$r_i = \frac{1}{s} \log_a(a^s \frac{i}{s}) \tag{1}$$

It is obvious that by tuning a one can easily weight shell-bins distance-dependent. Using $a = 2$ for example will result in shell-bins with equal volumes, thus equal weighted. Higher values for a will weight nearby samples exponentially more than those far away.
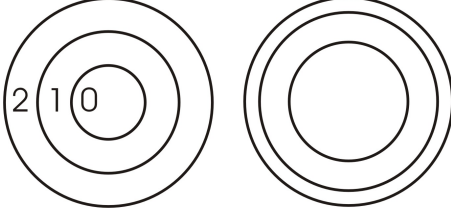

Figure 4: 2D Examples of a Histogram in Shell Model. The left one has equi-distanced shells ($a = 1, s = 3$) while the right one uses logarithmic radii ($a = 2, s = 3$). Note that in 2D the area of the shells grows linearly, while in 3D the shell volumes grows quadratically.

3.1.2 Sector Model

The 3D is decomposed into sectors that emerge from the center point of the shape. Obviously, this representation is invariant in scale but not in rotation. In a normalization step we perform translation and rotation of the object providing for rotation- and translation invariance, respectively. After the translation which maps the center of mass onto the origin we perform a Principal Axes transform on the object. The computation for a set of 3D points starts with the 3 x 3 covariance matrix where the entries are determined by an iteration over the coordinates (x, y, z) of all vertices. Here, we assume a 3D object is given as a Triangle Face set since this has become a standard representation for 3D objects. The vertices (x, y, z) then derive from the centers of mass of the respective triangle weighted by its unsigned area and normalized by the total area of the object:

1. Center of mass of triangle i:

$$f_i = \frac{1}{3} \cdot (v_1 + v_2 + v_3)$$

2. Unsigned area of triangle i:

$$\Delta f_i = \frac{1}{2} \cdot \|(v_2 - v_1) \times (v_3 - v_1)\|$$

3. Point i to contribute in the covariance matrix:

$$(x, y, z)_i = \frac{\Delta f_i f_i}{\sum_{j=1}^{n} \Delta f_j}$$

4. The covariance matrix:

$$\begin{bmatrix} \sum x^2 & \sum xy & \sum xz \\ \sum xy & \sum y^2 & \sum yz \\ \sum xz & \sum yz & \sum z^2 \end{bmatrix}$$

The eigenvectors of this covariance matrix represent the principal axes of the original 3D point set, and the eigenvalues indicate the variance of the points in the respective direction. As a result of the Principal Axes transform, all the covariances of the transformed coordinates vanish. Although this method in general leads to a unique orientation, this does not hold for the exceptional case of an object with at least two variances having the same value. An example for that would be a perfect sphere but in such a case any orientation in the sphere will result in the same histogram. Additionally, one must pay attention to the direction of the eigenvectors within the diagonalization process. Therefore we post-perform a *heaviest axis flip* similar to [15]. The basic idea behind this is to sum up positive and negative dotproducts of all vertices with the normalized eigenvectors, i.e. vertices are weighted linearly to their projected distance to the center of mass of the object. To normalize so that all objects have the same orientation we flip the axes, so that the object is "heavier" on the positive side. Additionally we sort the axes such that the new x-axis becomes the heaviest axis.
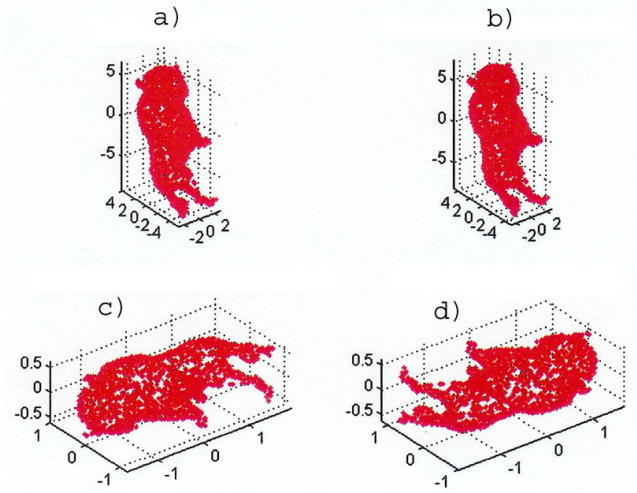

Figure 5: Normalization stages - a) Original object, b) Object after re-centering, c) Object after rotation and scaling, d) Object after flipping

Once the 3 Principal Axes of the 3D shape are computed we can easily obtain a unique orientation for each histogram. Since the center of mass of a 3D object is robust we define the first axis of the shape context to point to the center of mass of the shape. This defines a plane through the respective sample point with the normal of this first axis. Rather than computing two principal axes again in this plane we do a simple projection of the three yet computed axes onto that plane (cf. figure 6).
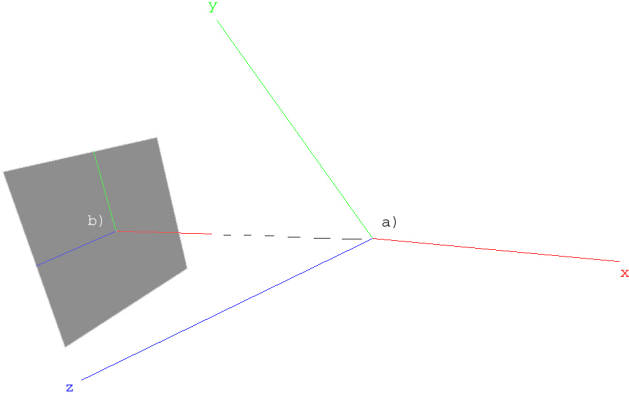
Figure 6: Unique Orientation of a sector model histogram. a) The Orientation of a 3D shape derived by PCA, b) The orientation of a histogram derived by simple projection. Note: The first axis points to the center of mass. The other two axes are obtained by projections of the principal axes.

Another simple idea for a unique orientation is to use the normal information in the sample point. Unfortunately, normals of 3D objects as retrieved from the WWW in general suffer from noise what makes them unsufficient to use for a robust descriptor. Furthermore, normals are local features and thus, cannot easily be used for global matching of two shapes.

3.1.3 Combined Model

The combined model represents more detailed information than pure shell models and pure sector models. A simple combination of two fine-grained 3D decompositions results in a high dimensionality. However, since the resolution of the space decomposition is a parameter in any case, the number of dimensions may easily be adapted to the particular application. With regard to retrieval we also mention methods to reduce dimensionality in section 5.1.4. For the combined model we suggest a log-polar coordinate system, i.e. a combined shell-sector-model with logarithmic shell radii (cf. figure 7).
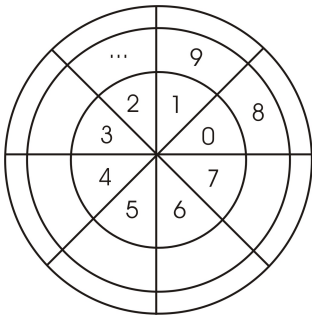

Figure 7: A 2D example of log-combined model. The numbers are the bin-indices.

4   Matching

In this section we give a detailed view on how to locally match two 3D shape contexts (cf. section 4.1) and show how 3D shape contexts can be used for the overall matching of two shapes (cf. section 4.2). For the global matching we present to methods: a 1-1 matching and a matching that is insensitive to sample count. For the latter we review some methods that enable efficient retrieval and indexing of shape with the 3D shape contexts in Section 5.

4.1   Local Matching

Concretely, for a point p_i on the shape, the corresponding histogram h_i is defined as

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\} \qquad (2)$$

As mentioned above, this histogram is said to be the *shape context* of p_i. Consider a point p_i on the first shape and a point q_j on the second shape. Let $CS_{i,j} = CS(p_i, q_j)$ denote the cost of matching these two points. We refer to *CS* as the **Shape Term**. As shape contexts are distributions represented as histograms, it is natural to use χ^2 distance:

$$CS_{i,j} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)} \qquad (3)$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histogram at p_i and q_j, respectively. This matching will result in close distributions. An example for applications with absolute reference frames are intra-industry databases where the objects are likely to have the same alignment. In an application environment where the reference frame of the shapes is not absolute, i.e. some kind of pose normalization is needed, we may take the local appearance of the shape context into account. With regard to that, we encourage the usage of a more subtle measuring that regards distances in orientations and relative positions as well as distances in distribution.

Consider two reference frames $\{u_k\}, \{v_k\}$ with $k \in \{1,2,3\}$ both pairwise orthogonal with $\|u_k\| = \|v_k\| = 1$ representing the respective orientations of h_i and h_j. The distance between these two reference frames can then be measured in terms of angle distances between the corresponding axis vectors:

$$CA_{i,j} = \frac{1}{4N} \cdot \sum_{k=0}^{3} \alpha_k \beta_k \cdot (1 - <u_k, v_k>)^2 \qquad (4)$$

where $< x, y >$ denotes the standard dot-product between vectors x and y. We refer to *CA* as the **Appearance Term**. The weights $\{\alpha_k\}, \{\beta_k\}$ can either be user-set or automatically derived from the *heaviness*-relation between the respective principal axes mentioned in section 3.1.2. For the latter assumption the following holds both for $\{\alpha_k\}$ and $\{\beta_k\}$:

1. $1 > \alpha_1 \geq \alpha_2 \geq \alpha_3 > 0$ - the weights are sorted and none is zero

2. $\sum \alpha_k = 1$ - the weights sum up to 1

Matching with the **Appearance Term** alone will result in histogram- correspondences with very similar orientations. Thus, after an applied pose normalization (section 3.1.2), this term is a compact and quickly computable orientation descriptor. However, it is obviously neither invariant to rotation as the Shape Term nor robust against distance displacements. To achieve that, we finally add a third term CP - the **Position Term** - that measures a distance of relative positions between points p_i and q_j on the two shapes being matched. Since p_i and q_j are represented relative to the *center of mass* of the respective shape and the shape extensions are both [0,1]-normalized we can simply denote this last term as a weighted quadratic form distance of the respective points:

$$CP_{i,j} = \sum_{k=1}^{3} (\alpha_k p_{i,k} - \beta_k q_{j,k})^2 \qquad (5)$$

where $p_{i,1}$ is the x-coordinate in the coordinate system of the shape and $\{\alpha_k\}, \{\beta_k\}$ are the same weights as in equation 4. A notable characteristics of the **Position Term** is the similarity to the squared euclidian distance. For symmetrical shapes like spheres, cylinder, etc. the weights $\{\alpha_k\}$ are very close resulting in symmetrical correspondences found with the position term. With regard to *clustering/vector-quantization* this can be a useful feature for grouping shape contexts together (cf. section5).

For the final local matching value $C_{i,j}$ we suggest the weighted sum of these three terms:

$$C_{i,j} = \gamma_1 \cdot CS_{i,j} + \gamma_2 \cdot CA_{i,j} + \gamma_3 \cdot CP_{i,j} \qquad (6)$$

where $\{\gamma_k\}$ are again weights in [0,1] with $\sum \gamma_k = 1$, which can be user-set or automatically derived with the same tool as for the weights $\{\alpha_k\}$. The idea behind automatic derivation of $\{\gamma_k\}$ is the observation that for symmetrical shapes the position term becomes linearly less discriminative in relative positions. Note that both the appearance term and the position term are the more discriminative the better the pose estimation was done in the preprocessing. We review the effect of tuning $\{\gamma_k\}$ in Section 6.1.

4.2 Global Matching

With regard to 3D Shape Contexts a global matching means finding correspondences between similar sample points on two shapes. Once these correspondences have been set up an affine transformation that maps the second shape onto the first shape can be estimated with standard least squares method. In the following, we briefly explain how we find these correspondences.

4.2.1 Hard Assignments

Given the set of costs $C_{i,j}$ between all pairs of points p_i on the first shape and q_j on the second shape, we want to minimize the (normalized) total matching cost,

$$H(\pi) = \frac{1}{n} \cdot \sum_{i=1}^{n} C_{i,\pi(j)} \qquad (7)$$

subject to the constraint that the matching be one-to-one, i.e. π is a permutation of $\{1, \ldots, n\}$. This is an instance of the square assignment (or weighted bipartite matching) problem, which can be solved in $O(N^3)$ time using the Hungarian method. In our experiments (cf. section 6) we used a more efficient algorithm of Joncker and Volgenant [38] The input to the assignment problem is a square cost matrix with entries $C_{i,j}$. The result is a permutation $\pi(i)$ such that $H(\pi)$ is minimized.

0.143	0.152	0.147	0.140	0.171
0.146	0.138	0.145	0.149	0.163
0.143	0.135	0.150	0.150	0.172
0.172	0.171	0.160	0.160	0.150
0.143	0.147	0.154	0.153	0.194

Figure 8: An example of a square 5×5 cost matrix

In order to devise a robust handling of outliers, one can add "dummy" nodes ([6]) to each point set with a constant matching cost of ε_d. In this case, a point will be matched to a "dummy" whenever there is no real match available at smaller cost than ε_d. Thus, ε_d can be regarded as a threshold parameter for outlier detection. Similarly, when the number of sample points on two shapes is not equal, the cost matrix can be made square by adding dummy nodes to the smaller point set. We reviewed the method above in our results (cf. section 6) but for our experiments we used a slightly different approach, which we found to be more suitable.

Having a large database of fine-sampled objects, a "one-to-one" matching between a query shape and a possibly large candidate list of high-resolution shapes would result in far too high computational costs. To improve this situation, we introduce Shape Contexts matching with *soft assignments* in contrast.

4.2.2 Soft Assignments

In the general case two shapes will have different sample counts n_1 and n_1. We assume here that $n_2 \geq n_1$ but the bidirectional is also not critical.

0.147	0.153	0.139	0.154	0.144	0.145	0.144	0.173	0.174	0.163
0.145	0.169	0.154	0.146	0.159	0.165	0.171	0.143	0.140	0.156
0.166	0.140	0.151	0.168	0.164	0.135	0.141	0.159	0.165	0.161
0.153	0.151	0.162	0.139	0.161	0.157	0.150	0.170	0.165	0.138
0.150	0.164	0.154	0.151	0.163	0.162	0.170	0.139	0.138	0.156

Figure 9: An example of a 5×10 cost matrix

Using *soft assignments* now allows assigning one sample point p_i on the first shape to match to k_i sample points $\{q_{l_1}, \ldots, q_{l_{k_i}}\}; \{l_1, \ldots, l_{k_i}\} \subseteq \{1, \ldots, n_2\}$ on the second shape with local matching values $\{C_{i,l_1}, \ldots, C_{i,l_{k_i}}\}$. To determine which of the n_2 samples on the second shape should match to p_i we set up a threshold that is determined by the cost matrix entries in row i

$$\varepsilon_i = \sigma_i \cdot |max\{C_{i,j}\} - min\{C_{i,j}\}| \qquad (8)$$

where

$$\sigma_i = \sqrt{\sum_{m=1}^{n2} (C_{i,m} - min\{C_{i,j}\})^2} \qquad (9)$$

Using this threshold we can then establish a set of k_i candidate points for each row i:

$$\{C_{i,j} : C_{i,j} \leq min\{C_{i,j}\} + \varepsilon_i\}$$

which we will denote as $\{C_{i,l_1}, \ldots, C_{i,l_{k_i}}\}$.

Having k_i matching values instead of one, the total matching cost now needs a more subtle computation:

$$H(n_1, n_2) = \frac{1}{n_1} \cdot \sum_{i=1}^{n_1} \sum_{m=1}^{k_i} w(i, l_m) \cdot C_{i,l_m} \qquad (10)$$

with weights

$$w(i, l_m) = \frac{min\{C_{i,j}\} + \varepsilon_i - C_{i,l_m}}{k_i \cdot \varepsilon_i} \qquad (11)$$

,i.e values with a larger distance to the minimum are weighted less. Note that the weights $\{w(i, l_m)\}$ are normalized by k_i. We note that one could use other filters instead, for example the Gaussian kernel. Note also that if matching a shape to itself small values for k_i imply that the sample point p_i is likely to be a *feature point* of the entire shape. Using the approach described above, the time complexity of finding the correspondences and minimization of $H(n_1, n_2)$ is $O(n_1 n_2)$

5 Future Work

Due to the enormous and still increasing size of modern databases that contains tens and hundred of thousands of 3D objects, the task of efficient query processing becomes more and more important. In the case of quadratic form distance functions, the evaluation time of a single database increases quadratically with the dimension. Thus, linearly scanning the overall database is prohibitive.

5.1 Iterated Query and Fast Pruning

Given a large set of known shapes the problem is to determine which of these shapes is most similar to a query shape. From this set of shapes, we wish to quickly construct a short list of candidate shapes which includes the best matching shapes. After completing this coarse comparison step one can then apply a more time consuming, and more accurate, comparison technique to only the shortlist. We want to leverage the descriptive power of shape contexts towards this goal of quick pruning. A few key methods we propose to use with 3D Shape Contexts and plan for the future follow below.

5.1.1 Representative Shape Contexts

Belongie et. al[32] used this method for 2D Shape Contexts on the COIL-100 database. It can easily be adapted to use with 3D Shape Contexts. Given two discriminable shapes we do not need to compare every pair of shape contexts on the objects to know that they are different. When trying to match two dissimilar shapes none of the shape contexts of the first shape have good matches on the second shape. For each of the known shapes S_i, a large s (about 100 to 500) of shape contexts $\{SC_i^j : j = 1, 2, \ldots, s\}$ is computed. But for the query shape, only a small number r (about 5 to 50) of shape contexts are computed by randomly selecting r samples on the shape. Comparisons with each of the known shapes is then done only with these r shape contexts. To compute the distance between a query shape and a known shape the best matches for each of the r shape contexts have to be found involving r-Nearest Neighbor Search. Distances are again computed using the χ^2 distance.

$$dist(S_{query}, S_i) = \sum_{j=1}^{r} \chi^2(SC_{query}^j, SC_i^*)$$

where $SC_i^* = argmin_u \chi^2(SC_{query}^j, SC_j^u)$.

5.1.2 Shapemes

The full set of shape contexts for the known shapes consists of $N \cdot s$ d-dim vectors (N: shapes in the set, s: shape contexts for each shape, d: bins in each shape context). A standard technique in compression for dealing with such a large amount of data is vector quantization. Vector quantization involves clustering the vectors and then representing each vector by the index of the cluster that it belongs to. Belongie et. al [32] call these clusters *Shapemes* - canonical shape pieces. To derive them k-means clustering is applied to all shape contexts from the known set. Each d bin shape context is quantized to its nearest shapeme, and replaced by the shapeme label (an integer in $\{1, \ldots, k\}$). By this, each collection of s shape context (d bin histograms) is reduced to a single histogram with k bins. In order to match a query shape, the same vector quantization and histogram creation is performed on the shape contexts of the query shape. Then nearest neighbor search is performed in the space of histograms of shapemes. Since the naive algorithm for doing nearest neighbor searches takes $O(ND)$ time Belongie et. al [32] suggest using recent work of the theory community on the *ε-approximate nearest*

neighbors(ε-NN) problem that can be applied here. Indyk and Motwani [25] describe an algorithm for doing ε-NN queries in $O(Dpolylog(N))$ time that uses random projections and the Johnson-Lindenstrauss lemma [27].

5.1.3 Optimal Multistep k-Nearest Neighbor

To achieve a good performance in scanning databases one can also follow the paradigm of multistep query processing: An index-based filter step produces a set of candidates, and a subsequent refinement step performs the expensive exact evaluation of the candidates [41][2]. Whereas the refinement step in a multistep query processor has to ensure the correctness, i.e. no false hits may be reported as final answers, the filter step is primarily responsible for the completeness, i.e. no actual result may be missing from the final answers and, therefore, from the set of candidates. The method of [3]fulfills this property [43] and the produced candidate list was proven to be optimal [3][2]. Thus, expensive evaluations of unnecessary candidates are avoided. Only for the exact evaluation in the refinement step, the exact object representation is retrieved from the object server.

5.1.4 Reduction of Dimensionality for Quadratic Forms

A common approach to manage objects in high-dimensional spaces is to apply techniques to reduce the dimensionality. The objects in the reduced space are then typically managed by any multidimensional index structure [19]. The typical use of common linear reduction techniques such as the Principal Components Analysis (PCA) or Karhunen-Loève Transform (KLT), the Discrete Fourier or Cosine Transform (DFT,DCT), the Similarity Matrix Decomposition [22] or the Feature Subselection [16] includes a clipping of the high-dimensional vectors such that the Euclidean distance in the reduced space is always a lower bound of the Euclidean distance in the high-dimensional space. Ankerst et. al [3] mention three important properties of the reduced distance function developed in the context of multimedia databases for color histograms [42]: First, it is a lower bound of the given high-dimensional distance function. Second, it is a quadratic form again. Third, it is the greatest of all lower-bounding distance function in the reduced space.

5.1.5 Ellipsoid Queries on Multidimensional Index Structure

Due to the geometric shape of the query range, a quadratic form-based similarity query is called an *ellipsoid query* [41]. An efficient algorithm for ellipsoid query processing on multidimensional index structures was developed in the context of approximation-based similarity search for 3D surface segments [28][29]. The method is designed for index structures that use a hierarchical directory based on

rectilinear bounding boxes such as the R-tree [21], the R^+-tree [44], the R^*-trees [5], X-tree [8][7], and Quadtrees among others. The technique is based on measuring the minimum quadratic form distance of a query point to the hyperrectangles in the directory. Recently, an improvement by using conservative approximations has been suggested [4].

6 Results

We implemented the algorithms in C++ and ran the experiments on a P3-500 MHz and a P4-2.66 GHz PC. Figure 10 shows a table of the computation times measured. In this experiments we used both computed 3D primitives generated with the software 3ds Max and a few representative 3D objects downloaded from **http://www.3dcafe.com**.

	PIII-500 MHz	P4-2.66 GHz
Histograms		
100 Smp., 288 Bins	0.0000 sec	0.0000 sec
500 Smp., 1024 Bins	0.1412 sec	0.1251 sec
1000 Smp., 4608 Bins	0.8090 sec	0.5473 sec
Local Match Matrix		
100 × 100, 288 Bins	0.0843 sec	0.0311 sec
500 × 500, 1024 Bins	4.4375 sec	1.9647 sec
1000 × 1000, 4608 Bins	102.2198 sec	34.0151 sec
1-1 Matching		
100 Smp., 288 Bins	0.0000 sec	0.0000 sec
500 Smp., 1024 Bins	0.1126 sec	0.0842 sec
1000 Smp., 4608 Bins	3.2324 sec	1.1091 sec
Soft Matching		
100/200 Smp., 288 Bins	0.0000 sec	0.0000 sec
200/500 Smp., 1024 Bins	0.0243 sec	0.0150 sec
500/1000 Smp., 4608 Bins	0.0826 sec	0.0310 sec

Figure 10: Performances measured for different parameters, like bin count, sample count, etc.

6.1 Parameter Effect

Tuning $\{\gamma_k\}$ affects global matching of two shapes in several ways. Since the involved terms - **Shape Term**, **Appearance Term** and **Position Term** - all focus on least squares they can be linearly combined (recall Equ. 6). We show now some matching results on primitive shapes to outline their characteristics. All the matching shown below have been done using - unless otherstated - 100 samples, $(6,12)$ equally spaced angle bins and 4 $\log_2$-shells. Figures 11,12,13 show examples of 1-1 correspondences found only using one of the three terms.

6.2 Hard- vs. Soft- Assignments

The main drawback using hard assignments is the constraint that the matching is one to one. That means that a possibly good matching between two sample points p_{i_2} and q_j has to be discarded (cf. figure 14) if q_j was previously assigned to another sample point p_{i_1} resulting in a penalty in minimizing the total cost $H(\pi)$ (cf. Equ. 7). Noise or irregularity in sampling would then result in a worse global matching. Soft assignments do not suffer
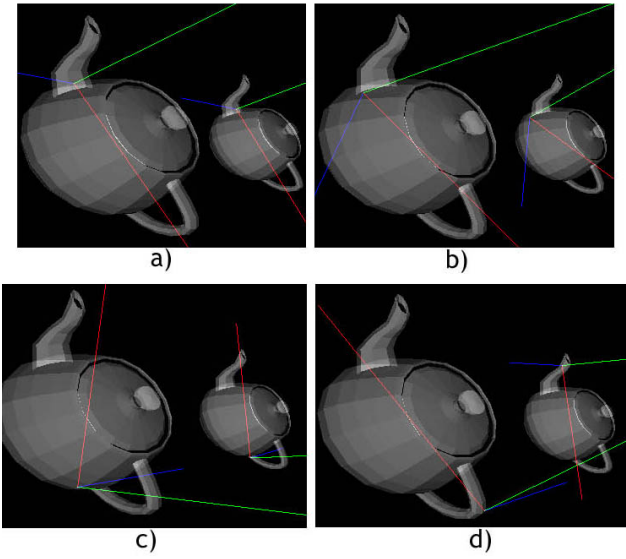
a)

b)

c)

d)

Figure 11: 1-1 correspondences with only the Shape Term. a)-c) show good matches, d) shows a *symmetrical match* subject to the constraint that one sample cannot be re-assigned (cf. section 6.2). Note that the matched histogram in the second shape regarding its orientation although has a very close *distribution* to the one in the first shape.

from this (cf. figure 15). Moreover, multiple samples in the second shape can be assigned to one sample in the first shape. However, soft assignments verify that each sample in the first shape will have an assignment but it does not guaraant that all samples in the second shape will be assigned.

6.3 Sample Count

We examine the robustness to different sample counts utilizing an efficient pruning and indexing approach (cf. Section 5). Figure 16 shows how overall matches change with the sample count. Note that the matching values with lower sample counts are higher than those with higher res-
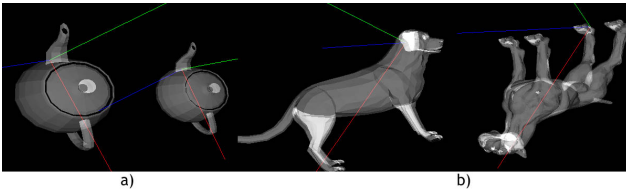

a)

b)

Figure 12: 1-1 correspondences with only the Appearance Term. a) shows a good match on equally aligned objects, b) shows a match with different alignment. Note that although not invariant to rotations the Appearance Term found a close orientation and thus a symmetrical sample.
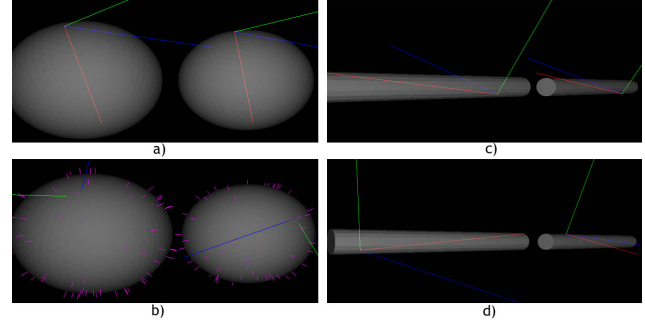

a)

c)

b)

d)

Figure 13: 1-1 correspondences with only the Position Term. a)+c) show good matches, b)+d) show symmetrical matches appearing in highly symmetrical shapes.
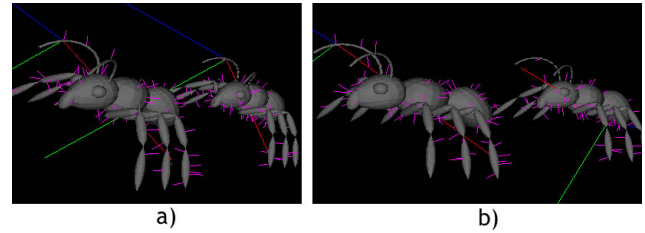

a)

b)

Figure 14: Problems with *hard assignments*: a) "succeeded" match, b) a "failed" match. The random sampling took 2 samples in the first shape but only 1 in the second shape. Since the assignment could not be reused the next most similar point was assigned.

olution - the reduced low-resolution matching is a lower bound for all higher resolution matchings.

6.4 Noise

Here we show examples on the robustness to noise of our descriptor. In Figure 17 noise was added to the 3D object. We used hard assignments there and only the shape term since there was no need for pose estimation. We matched the original object to the ones with noise applied and noted the first four match results due to the fact that sampled representations of a 3D object vary from instance to instance.
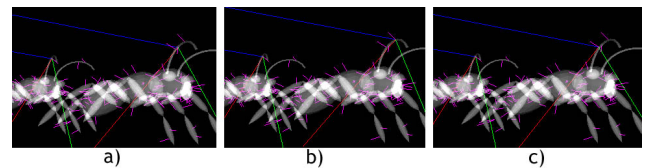

a)

b)

c)

Figure 15: The same situation as in Figure 14 with *soft assignments*. a)-c) show how 3 samples in the first shape are re-assigned to the same sample in the second shape.

a) Original Object, 100 Samples | b) 10 Samples, Match: 0.00127841
c) 100 Samples, Match: 0.00033260 | d) 200 Samples, Match: 0.00029447
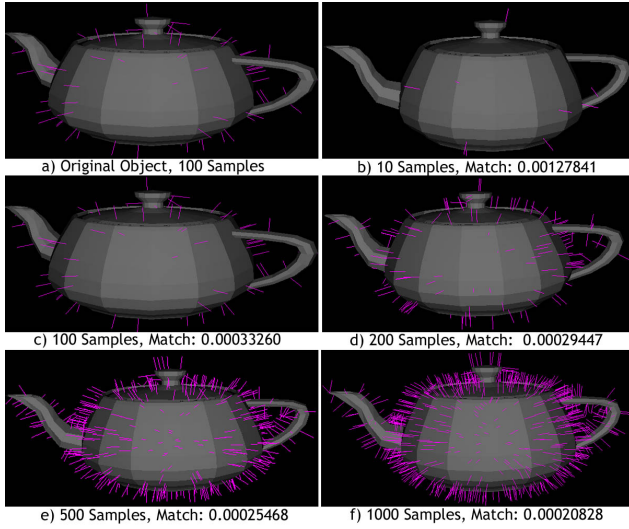e) 500 Samples, Match: 0.00025468 | f) 1000 Samples, Match: 0.00020828

Figure 16: Matching with different sample counts. The Original object was sampled with just 100 samples. b)-f) show overall matchings. In b) the sample count is lower than that of the original object in a). It shows the bidirectional case mentioned in section 4.2.2. Note that the reduced matching is a lower bound for all higher resolution matchings.

6.5   Real World Objects

In our experiments we applied 3D Shape Context matching to real world objects plain downloaded from the WWW. None of them was corrected, aligned, scaled or such. We used just 200 samples with $(6, 12)$ sector bins and 6 $\log_2$-shells each and hard assignments for the matching. Figure 18 shows the result. All objects in one row were matched to the leftmost object. The rightmost object in each row was meant to be dissimilar. Beneath the images are overall matching values

7   Conclusions

In this paper we utilized the 3D Shape Contexts for the purpose of content based retrieval of 3D objects. The quality of the descriptor regarding the retrieval performance was verified also with respect to other related recent technique. As it turns out, the 3D Shape Contexts are rich and powerful descriptors for general 3D objects in terms of retrieval performance and robustness against topological and geometrical artifacts plaguing a large amount of freely available shapes.

References

[1] 3D knowledge, 3dk.asu.edu.

a) Original Object
0.00062582 0.00062178
0.00061367 0.00063266

b) 2% Noise
0.00064002 0.00069815
0.00062219 0.00063311

c) 4% Noise
0.00070798 0.00060602
0.00062215 0.00070081

d) 16% Noise
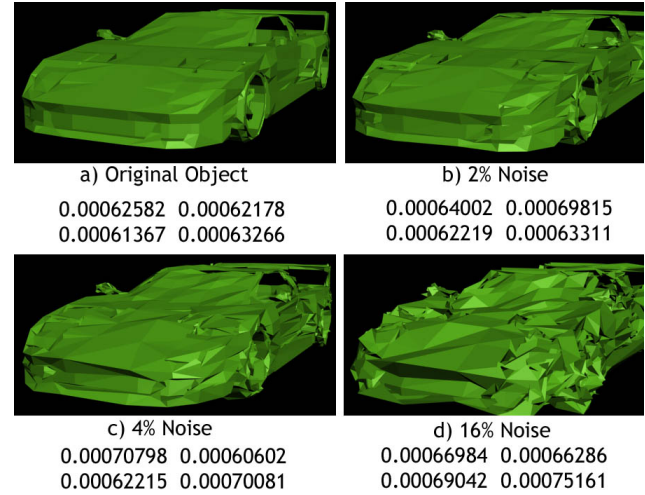0.00066984 0.00066286
0.00069042 0.00075161

Figure 17: Robustness to Noise: In b)-d) noise was added to the original object. Below are the similarity values of the first 4 matches each. Note that all values differ a bit due to the random sampling.

[2] M. Ankerst, , H.-P. Kriegel, and T. Seidl. A multi-step approach for shape similarity search in image databases. In *IEEE Transactions on Knowledge and Engineering*, volume 10, pages 996–1004, 1998.

[3] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *Sixth. Int. Symposium on Large Spatial Databases*, pages 207–226. University of Munich, Institute for Computer Science, 1999.

[4] M. Ankerst, B. Vruanmller, H.-P. Kriegel, and T. Seidl. Improving adaptable similarity query processing by using approximations. In *24th. Int. Conf. on Very Large Databases*. University of Munich, Institute for Computer Science, 1997.

[5] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The r*-tree: An efficient and robust access method for points and rectangles. In *Int. Conf. on Management of Data*, 1990.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. 2000.

[7] S. Berchtold, C. Bhm, B. Braunmller, D. Keim, and H.-P. Kriegel. Fast parallel similarity search in multimedia databases. In *Int. Conf. on Management of Data*. University of Munich, Institute for Computer Science, 1997.

[8] S. Berchtold, D. Keim, and H.-P. Kriegel. The x-tree: An index structure for high-dimensional data. In *22nd Int. Conf. on Very Large Databases*, 1996.
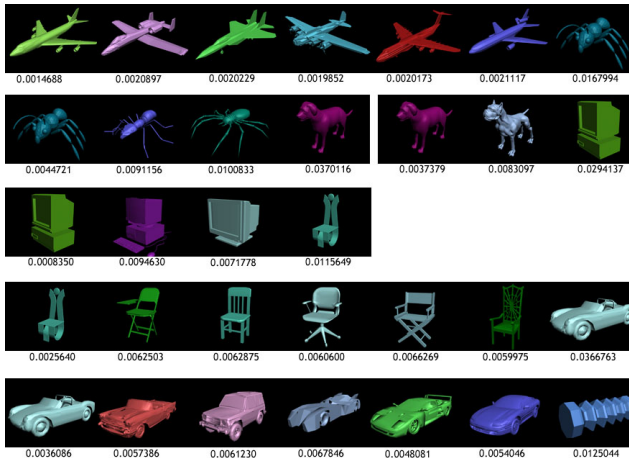
Figure 18: 3D objects from the WWW. All objects in a row are matched to the leftmost one. Beneath them is the respective Matching Value. Each rightmost object was chosen to be dissimilar to the others.

[9] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[10] H. Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205–287, 1973.

[11] N. Canterakis. 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In *11th Scandinavian Conf. on Image Analysis*, 1999.

[12] Vincent Cicirello and William C. Regli. Machining feature-based comparisons of mechanical parts. pages 176–185. Int'l Conf. on Shape Modeling and Applications, May 2001.

[13] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod. Coarse filters for shape matching. *IEEE Computer Graphics*, 22(3):65–74, 2002.

[14] M. Elad, A. Tal, , and S. Ar. Content based retrieval of vrml objects - an iterative and interactive approach. In *Eurographics Multimedia Workshop*, pages 97–108, 2001.

[15] M. Elad, A. Tal, , and S. Ar. Similarity between three-dimensional objects - an iterative and interactive approach. 2001.

[16] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3, 1994.

[17] David Forsyth, Jitendra Malik, Margaret Fleck, and Jean Ponce. Primitives, perceptual organization and object recognition. Technical report, Computer Science Division, University of California at Berkeley, Berkeley, CA 94720, 1997.

[18] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22(1), 2003.

[19] V. Gaede and O. Gnther. Multidimensional access methods. *ACM Computing Survey*, 30, 1994.

[20] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.

[21] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Int. Conf. on Management of Data*, 1984.

[22] J. Hafner, H. Shawney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17, 1995.

[23] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH*, 2001.

[24] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Information Theory*, 8(2):179–187, 1962.

[25] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613, 1998.

[26] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH '95*, pages 277–286, 1995.

[27] W. Johnson and J. Lindenstrauss. *Extensions of lipshitz mapping into Hilbert space*. Contemp. Math, 1984.

[28] H.-P. Kriegel, T. Schmidt, and T. Seidl. 3d similarity search by shape approximation. In *Fifth. Int. Symposium on Large Spatial Databases*. University of Munich, Institute for Computer Science, 1997.

[29] H.-P. Kriegel and T. Seidl. Approximation-based similarity search for 3d surface segments. *GeoInformatica Jorunal*, 1998.

[30] F. Leymarie and B. Kimia. The shock scaffold for representing 3d shape. In *Proc. of 4th International Workshop on Visual Form (IWVF4)*, 2001.

[31] David McWherter, Mitchell Peabody, William C. Regli, and Ali Shokoufandeh. Transformation invariant shape similarity comparison of solid models. ASME Design Engineering Technical Confs., 6th Design for Manufacturing Conf. (DETC 2001/DFM-21191), Sep 2001.

[32] Greg Mori, Serge Belongie, and Jitendra Malik. Shape contexts enable efficient retrieval of similar shapes. 2002.

[33] Ryutarou Ohbuchi, Tomo Otagiri, Masatoshi Ibato, and Tsuyoshi Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Pacific Graphics*, 2002.

[34] Robert Osada, Thomas Funkhouser, Bernard Chazell, and David Dobkin. Shape distributions.

[35] E. Paquet and M. Rioux. A content-based search engine for vrml databases. In *CVPR Proceedings*, pages 541–546, 1998.

[36] E. Petrakis. Design and evaluation of spatial similarity approaches for image retrieval, 2002.

[37] W. Regli. National design repository, http://edge.mcs.drexel.edu/repository/frameset.html.

[38] R.Joncker and A.Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. 1987.

[39] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Past, present, and future. In *International Symposium on Multimedia Information Processing*, 1997.

[40] M. Schneier and M. Abdel-Mottaleb. Exploiting the jpeg compression scheme for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):849–853, 1996.

[41] T. Seidl. *Adaptable Similarity Search in 3D Spatial Database Systems*. PhD thesis, University of Munich, Institute for Computer Science, 1997.

[42] T. Seidl and H.-P. Kriegel. Efficient user-adaptable similarity search in large multimedia databases. 1997.

[43] T. Seidl and H.-P. Kriegel. Optimal multi-step k-nearest neighbor search. University of Munich, Institute for Computer Science, 1998.

[44] T. Sellis, N. Roussopoulos, and C. Faloutsos. The r+ -tree: A dynamic index for multi-dimensional objects. In *Int. Conf. on Very Large Databases*, 1987.

[45] Kaleem Siddiqi, Ali Shokoufandeh, Sven J. Dickinson, and Steven W. Zucker. Shock graphs and shape matching. In *ICCV*, pages 222–229, 1998.

[46] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[47] Motofumi T. Suzuki, Toshikazu Kato, and Nobuyuki Otsu. A similarity retrieval of 3d polygonal models using rotation invariant shape descriptors. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC2000)*, pages 2946–2952, 2000.

[48] M.R. Teague. Image analysis via the general theory of moments. *Journal Optical Society of America*, 70(8):920–930, 1980.

[49] C.-H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):496–513, 1988.

[50] D. V. Vranic and D. Saupe. Description of 3d-shape using a complex function on the sphere. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2002)*, pages 177–180, 2002.

[51] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, 21:269–281, 1972.