SHREC2006
3D Shape Retrieval Contest

Remco C. Veltkamp, Remco Ruijsenaars,

Michela Spagnuolo, Roelof van Zwol,

Frank ter Haar

Department of Information and Computing Sciences,
Utrecht University

Technical Report UU-CS-2006-030

www.cs.uu.nl

ISSN: 0924-3275

SHREC2006: 3D Shape Retrieval Contest

Remco C. Veltkamp[1] Remco Ruijsenaars[1] Michela Spagnuolo[2]
Roelof van Zwol[1] Frank ter Haar[1]

[1] Department of Information and Computing Sciences, Utrecht University, the Netherlands

[2] IMATI-CNR, Genoa, Italy

1 Introduction

The Network of Excellence AIM@SHAPE [1] has taken the initiative to organize a 3D shape retrieval evaluation event: SHREC - 3D Shape Retrieval Contest [2]. The general objective is to evaluate the effectiveness of 3D-shape retrieval algorithms.

3D media retrieval is overlooked in most commercial search engines, while at the same time it is expected to represent a huge amount of traffic and data stored in the Internet. Indeed, "geometry is poised to become the fourth wave of digital-multimedia communication", where the first three waves were sound in the 70's, images in the 80's, and video in the 90's [3]. Recent advances in technology have made available cost-effective scanning devices that could not even be imagined a decade ago. It is now possible to acquire 3D data of a physical object in a few seconds and produce a digital model of its geometry that can be easily shared on the Internet. On the other hand, most PCs connected to the Internet are nowadays equipped with high-performance 3D graphics hardware, that support rendering, interaction and processing capabilities from home environments to enterprise scenarios.

Three-dimensional shape retrieval is fundamentally different from two-dimensional shape retrieval. Most 2D methods do not generalize directly to 3D. This is due to the different nature of the content: descriptors used for 2D images are concerned with color, textures, and properties that capture geometric details of the shapes segmented in the image. While one-dimensional boundaries of 2D shapes allow a direct parametrization (e.g. by arc length), the boundary of arbitrary 3D objects cannot be parametrized in a straightforward manner, especially when the shape exhibits complex topology such as through-holes or handles. Most notably, feature extraction for image retrieval is intrinsically affected by the so-called sensory gap, the gap between the physical object in a real world scene and the digital description derived from a recording of that scene. The sensory gap makes the description of objects an ill-posed problem and casts an intrinsic uncertainty on the descriptions due to the presence of information which is only accidental in the image or due to occlusion and/or perspective distortion. On the other hand, the boundary of 3D models is represented explicitly, and therefore does not need to be segmented from a background. Hence, while the understanding of the content of a 3D vector graphics remains an arduous problem, the initial conditions are different and potentially allow for more effective and reliable search results.

TREC, the Text Retrieval Conference [4], is a series of workshops on large scale evaluation of text retrieval technology organized since 1992, which has had a major impact on the text retrieval community. Following the successful example of TREC, a number of other competitions have been organized, for example:

Participant	Affiliation	Reference	Run file	Code
Chaouch et al.	INRIA, France	[11]	1	C1
			2	C2
Daras et al.	Thessaloniki, Greece	[12]	1	D1
Jayanti et al.	Purdue University, Indiana	[13]	1	J1
			2	J2
			3	J3
Laga et al.	NAIST, Japan	[14]	1	L1
			2	L2
Makadia et al.	University of Pennsylvania	[15]	1	M1
			2	M2
			3	M3
			4	M4
Papadakis et al.	Athens, Greece	[16]	1	P1
Shilane et al.	Princeton University, New Jersey	[17]	1	S1
			2	S2
			3	S3
Zaharia et al.	INT, France	[18]	1	Z1

Table 1: SHREC2006 participants.

- TRECVID, the TREC Video Retrieval Evaluation [5].
- FRGC, the Face Recognition Grand Challenge [6].
- VOC, the Visual Object Classes challenge [7].
- MIREX, the Music Information Retrieval Evaluation Exchange [8].
- INEX, the Initiative for the Evaluation of XML [9].
- FVC, the Fingerprint Verification Competition [10].

Thirteen candidates registered for SHREC2006, five withdrew after the test collection was made available. Within 48 hours after the release of the query set, each participant had to submit the retrieval results of all queries as a ranked list per query. Up to five ranked lists could be submitted. The ranked lists may be the result of runs of different algorithm, or runs of the same algorithm with different parameter settings. Table 1 lists the participants and the coding of the run files used in the rest of this paper.

2 Test collection

The collection to search in consists of all the classified models from the Princeton Shape Benchmark (PSB) Version 1, both the models from the training set and those from the test set (1814 models). The files have been renamed and put into a single directory, such that the classification of the objects cannot be derived from the path and filename. The total collection consists of 1814 models, and has been distributed via the SHREC website [2] and directly from the PSB website [19].

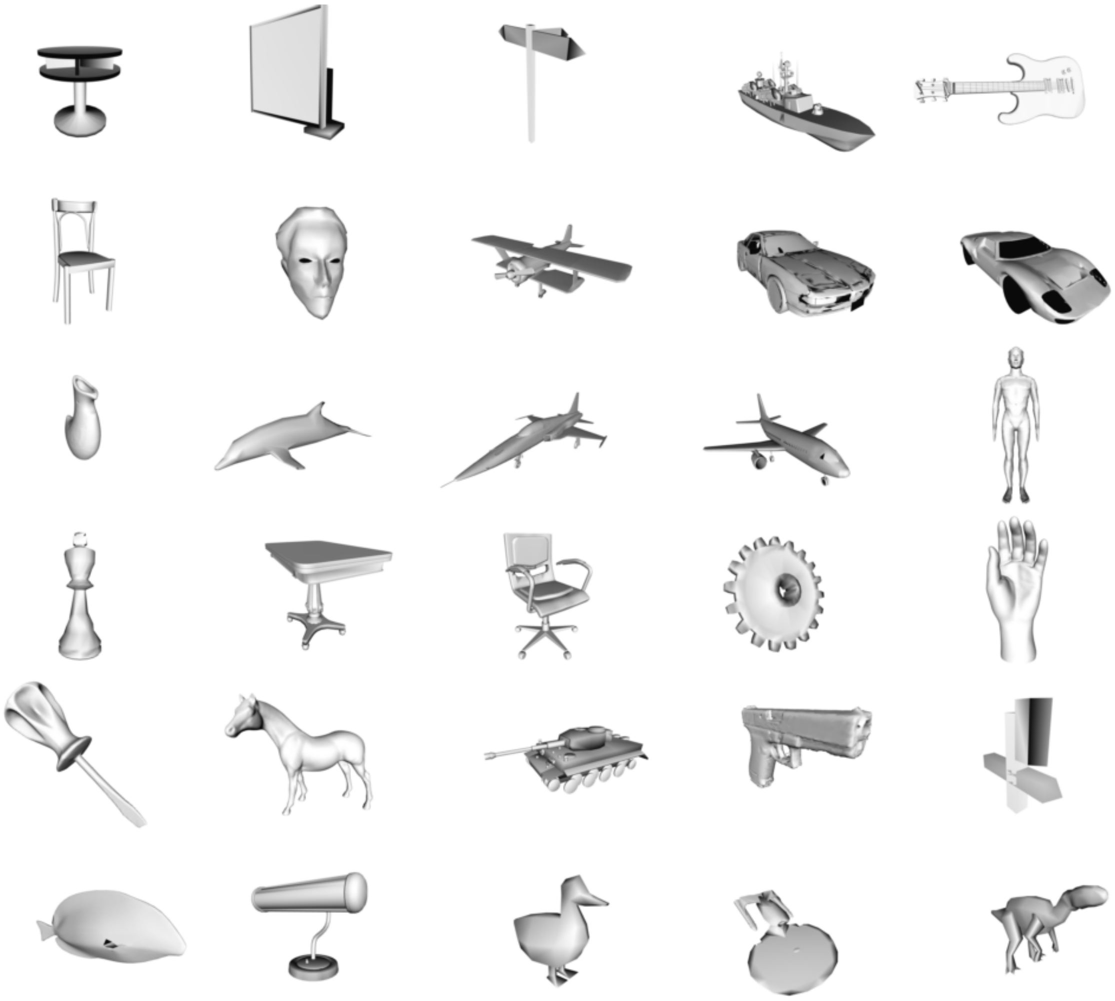


Figure 1: The 30 queries of the contest.

3 Queries

Each participant was requested to submit five candidate queries that are not already in the PSB. From all the submitted candidates, 30 queries were selected, see figure 1. Each query was classified into one of the classes of the PSB. The relevance assessments is done according to this base classification on a ternary relevance scale: highly relevant, giving a score of 2, marginally relevant giving score 1, not relevant giving score 0.

All items in the query class are considered highly relevant. Items in sister classes are considered marginally relevant. Depending on the shape, sometimes other classes are also considered (marginally) relevant. For example, query 21 is a short screwdriver. For this query, the screwdrivers in the PSB are highly relevant, and the ice creams have been labeled marginally relevant. See the website [2] for the full relevance assessment of all queries. The number of relevant items in the collection ranges from 5 for queries 21 (screwdriver) and 28 (duck), to 104 for query 13 (fighter jet). The number of marginally relevant items ranges from 0 to 218.

4 Performance measures

For each query there exists a set of highly relevant items and a set of marginally relevant items. Therefore, most of the evaluation measures have been split up as well. The evaluation measures "xxx(highly relevant)" are based on the highly relevant items only, while the evaluation measures "xxx(relevant)" are based on all relevant items (highly relevant items + marginally relevant items).

The submitted ranked lists are turned into a gain vector by replacing item IDs by their relevance scores. A highly relevant retrieved item corresponds to relevance score 2, a marginally relevant retrieved item corresponds to relevance score 1, and a non-relevant retrieved item corresponds to relevance score 0.

Throughout this document the following abbreviations are used.

- Ds: size of the dataset (1814).
- Ch: number of highly relevant classified items.
- Cm: number of marginally relevant classified items.
- Cr: number of relevant classified items ($Cm + Ch$).
- Cn: number of non-relevant classified items ($Ds - Cr$).
- Vh: number of visible highly relevant items (i.e., relevant items in the ranked list).
- Vm: number of visible marginally relevant items.
- Vr: number of visible relevant items ($Vm + Vh$).
- Va: number of visible items (length of ranked list).
- Vn: number of visible non-relevant items ($Va - Vr$).

The following evaluation measures are used.

- True Positives(highly relevant) $= Vh$.
- True Positives(relevant) $= Vr$.
- False Positives(highly relevant) $= Va - Vh$.
- False Positives(relevant) $= Va - Vr$.
- True Negatives(highly relevant) $= Ds + (Vh - Va) - Ch$.
- True Negatives(relevant) $= Ds + Vr - Va - Cr$.
- False Negatives(highly relevant) $= Ch - Vh$.
- False Negatives(relevant) $= Cr - Vr$.
- First Tier(highly relevant) = (number of visible highly relevant items within the first $d1$ items of the ranked list / $d1$) * 100% (if $Va < Ch$ then $d1 = Va$ else $d1 = Ch$).
- First Tier(relevant) = (number of visible relevant items within the first $d1$ items of the ranked list / $d1$) * 100% (if $Va < Cr$ then $d1 = Va$ else $d1 = Ch$).

- Second Tier(highly relevant) = (number of visible highly relevant items within the first $d2$ items of the ranked list / $d2$) * 100% (if $Va < 2 * Ch$ then $d2 = Va$ else $d2 = 2 * Ch$).

- Second Tier(relevant) = (number of visible relevant items within the first $d2$ items of the ranked list / $d2$) * 100% (if $Va < 2 * Cr$ then $d2 = Va$ else $d2 = 2 * Ch$).

- Precision(highly relevant) $= Vh/Va$.

- Precision(relevant) $= Vr/Va$.

- Recall(highly relevant) $= Vh/Ch$.

- Recall(relevant) $= Vr/Cr$.

- Average Precision(highly relevant) = average of the Precision(highly relevant) scores after each highly relevant retrieved item.

- Average Precision(relevant) = average of the Precision(relevant) scores after each relevant retrieved item.

- Average dynamic recall (ADR) is defined as

$$ADR = \frac{1}{q}\sum_{i=1}^{q} r_i$$

where

$$q = \begin{cases} Va & \text{if } Va < Cr \\ Cr & \text{otherwise} \end{cases}$$

and $r_i = f/i$, with

$$f = \begin{cases} \text{number of visible highly relevant items within the first } i \text{ items} & \text{if } i \leq Ch \\ \text{number of visible relevant items within the first } i \text{ items} & \text{otherwise} \end{cases}$$

- The cumulated gain vector CG is defined recursively as [21]:

$$CG[i] = \begin{cases} G[1] & \text{if } i = 1 \\ CG[i-1] + G[i] & \text{otherwise} \end{cases}$$

- The discounted cumulated gain vector DCG is defined recursively as [21]:

$$DCG[i] = \begin{cases} CG[1] & \text{if } i = 1 \\ DCG[i-1] + (G[i]/\log i) & \text{otherwise} \end{cases}$$

- The normalized cumulated gain vector NCG is obtained by dividing CG by the ideal cumulated gain vector ICG [21].

For example, let query q have $Ch = 6$ and $Cm = 5$, thus $Cr = 5 + 6 = 11$. Let the gain vector be $G' = [2, 2, 1, 2, 2, 1, 0, 1, 0, 1, 2, 0, 0, 0]$, thus $Vh = 5$, $Vm = 4$, $Va = 14$, and $Vn = 5$. Applied to the example, we get the following performance values.

- True Positives(highly relevant) $= 5$.

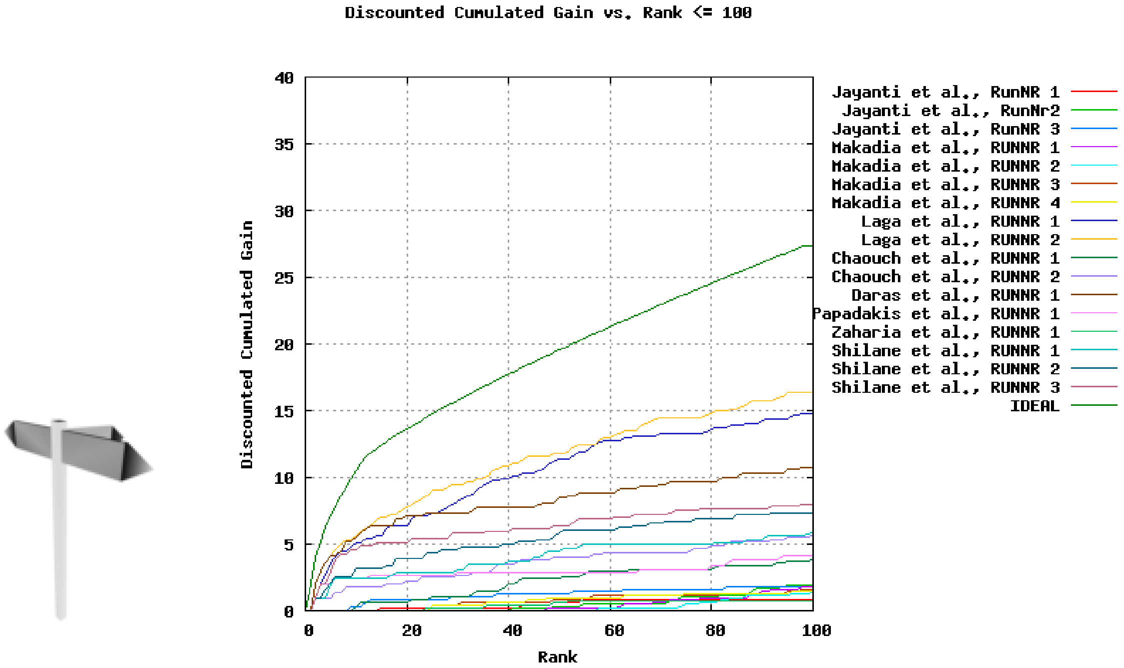


Figure 2: Query 3, a street sign, and the discounted cumulated gain vs. rank [1,100] for all submitted ranked lists.

- True Positives(relevant) = 9.
- False Positives(highly relevant) $= 14 - 5 = 9$.
- False Positives(relevant) $= 14 - 9 = 5$.
- True Negatives(highly relevant) $= 1814 + 5 - 14 - 6 = 1799$.
- True Negatives(relevant) $= 1814 + 9 - 14 - 11 = 1798$.
- False Negatives(highly relevant) $= 6 - 5 = 1$.
- False Negatives(relevant) $= 11 - 9 = 2$.
- First Tier(highly relevant) $= (4/6) * 100\% = 66.667\%$.
- First Tier(relevant) $= (9/11) * 100\% = 81.818\%$.
- Second Tier(highly relevant) $= (5/12) * 100\% = 41.667\%$.
- Second Tier(relevant) $= (9/14) * 100\% = 64.285\%$.
- Precision(highly relevant) $= 5/14 = 0.35714$.
- Precision(relevant) $= 9/14 = 0.6428$.
- Recall(highly relevant) $= 5/6 = 0.83333$.
- Recall(relevant) $= 9/11 = 0.8181$.

Rank	Query 1	Query 3	Query 6
1	Z1	L2	M2
2	M2	L1	S3
3	D1	D1	M1
4	M1	S2	D1
5	C2	S3	S1
6	C1	C2	S2
7	J2	S1	Z1
8	J1	C1	J1
9	S3	P1	P1
10	P1	J2	J2
11	S2	M1	M3
12	L1	M2	C2
13	L2	M3	M4
14	S1	J3	C1
15	J3	M4	J3
16	M3	J1	L1
17	M4	Z1	L2

Table 2: Example ranking for individual queries.

- Average Precision(highly relevant) $= (1/1 + 2/2 + 3/4 + 4/5 + 5/11)/5 = 0.80090$.

- Average Precision(relevant) $= (1/1+2/2+3/3+4/4+5/5+6/6+7/8+8/10+9/11)/9 = 0.94368$.

- ADR $= (1+1+0.667+0.75+0.8+0.667+0.857+0.875+0.777+0.8+0.818)/11 = 0.819$.

- CG $= [2, 4, 5, 7, 9, 10, 10, 11, 11, 12, 14, 14, 14, 14]$

- DCG = [2.0, 4.0, 4.63093, 5.63093, 6.492283, 6.8791356, 6.8791356, 7.212469, 7.212469, 7.5134993, 8.091629, 8.091629, 8.091629, 8.091629].

- ICG $= [2, 4, 6, 8, 10, 12, 13, 14, 15, 16, 17]$.
  NCG = [2/2, 4/4, 5/6, 7/8, 9/10, 10/12, 10/13, 11/13, 11/14, 12/15, 14/16, 14/17, 14/17, 14/17].

- IDCG = [2.0, 4.0, 5.2618594, 6.2618594, 7.123213, 7.8969183, 8.253125, 8.586458, 8.901923, 9.202953, 9.492018].
  NDCG = [2.0/2.0, 4.0/4,0, 5.2618594/5.2618594, ..., 8.091629/9.492018, 8.091629/9.492018].

In addition to the above performance measures, the average (discounted) cumulated gain versus percentage recall, evaluated at each relevant item in the ranked list is plotted.

For a whole run over all queries, the following is computed:

- Mean Average Precision(highly relevant/relevant).

- Mean First Tier(highly relevant/relevant).

- Mean Second Tier(highly relevant/relevant).

Rank	Run code	MADR value
1	M2	0.5498626
2	M1	0.54084843
3	D1	0.5242406
4	C1	0.50018275
5	P1	0.49523294
6	S3	0.4937149
7	Z1	0.49247277
8	S2	0.48770607
9	C2	0.42156762
10	S1	0.39706558
11	M3	0.39249521
12	M4	0.37667266
13	L1	0.32631385
14	L2	0.30619973
15	J2	0.26785165
16	J3	0.2370221
17	J1	0.23020707

Table 3: Mean average dynamic recall.

- Mean Average Dynamic Recall.
- Mean (Normalized) Cumulated (Discounted) Gain @ x (absolute rank).

5 Results

For each individual submitted query result list, the following performance measures are calculated: Average Precision, First Tier, Second Tier, Precision, Recall, (all these are evaluated for only highly relevant items, and all relevant items), Average Dynamic Recall, Cumulated Gain, Normalized Cumulated Gain, Discounted Cumulated Gain, Normalized Discounted Cumulated Gain (all for the first 5, 10, 25, 50, and 100 ranked items), Average (Discounted) Cumulated Gain vs. Recall %, (Discounted) Cumulated Gain vs. Rank [1,100], and Normalized (Discounted) Cumulated Gain vs. Rank [1,100].

None of the participants used an adaptive method to cut of the ranked list. Either 200 were returned (Shilane et al.), or 1802 (Jayanti et al.) or all 1814 (the rest). Therefore, the performance measures precision and recall are meaningless.

We further note that there are often no big differences between the performance measure for only the highly relevant items and all relevant items. Also between the various versions of cumulative gain there are no big differences.

Figure2 shows just one example query and its performance graph: query 3 and the graph of the discounted cumulative gain vs. rank [1,100]. In many other query results, the lines in the graphs are much closer together.

Of course, each single query says little about the overall performance of a method. Indeed, the method ranked first in terms of cumulated gain [1,100] for query 1 (Z1), has the last rank for query 3.

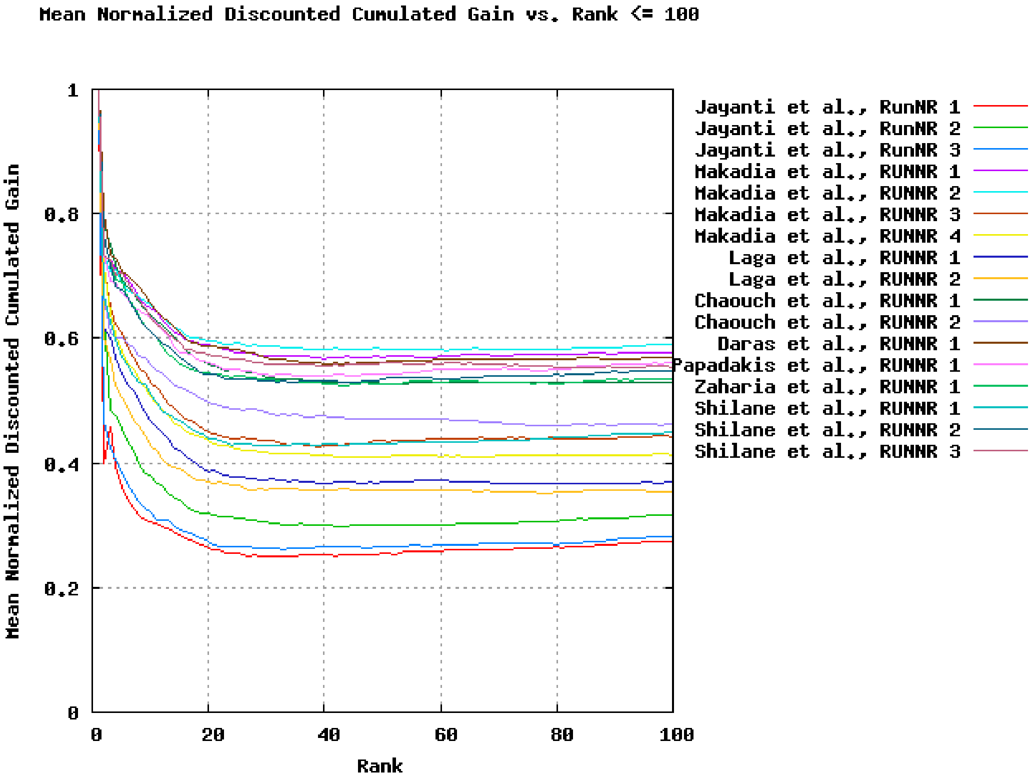


Figure 3: Mean normalized discounted cumulated gain graph over ranks [1,100]. At rank 100 the order from top to bottom is M2, M1, D1, P1, S3, S2, Z1, C1, C2, S1, M3, M4, L1, L2, J2, J3, J1.

In turn, the method ranked first for query 3 (L2), has the last rank for query 6, see table 2. However, the individual query performances may give insight into the strong and weak aspects of the methods.

The results per run file average the performance of a single method over all 30 queries, thus providing an overall impression of the performance of an algorithm. Table 3 gives a listing of all runs, ranked according their value for the Mean Average Dynamic Recall. This ranking is similar to the ranking according to their value of the Mean Normalized Discounted Cumulated Gain at rank 100 for example, in the sense that the head and the tail of the two lists are the same. Figure 3 plots the Mean Normalized Discounted Cumulated Gain over all ranks [1,100].

An overview of results is presented at a special session of SMI06, Shape Modeling International 2006 [22].

6 Concluding remarks

The contest provides a nice selection of the state of the art in the 3D shape retrieval field. We believe that the results provide an excellent opportunity to analyze the various algorithms, their strengths, as well as their weaknesses. Using a common test collection allows a direct comparison of algorithms. At the same time, a single test collection and (chosen) ground truth only shows part of the whole picture. Care should be taken not to draw too far reaching conclusions from this single evaluation.

We now used an existing test collection and classification. It would be nice to use other collections as well in the future, in order to avoid too specific engineering and over-fitting towards a particular test set, an effect that is also visible in the 2D shape classification and retrieval domain [23]. For next year it is our intention to organize a multi-track contest. Possible track themes are partial and whole matching, polygon soup and watertight model matching, mechanical part matching, molecule matching, and 3D face matching. Also for the coming years the SHREC results will be presented at the SMI conference.

Defining multiple tracks, deciding upon the test collection, queries, relevance assessment, and performance measures is a substantial amount of work. However, it works well if participants have an active role, as is shown for this edition of SHREC. We thank all the participants!

Acknowledgment

This research was supported by the FP6 IST Network of Excellence 506766 AIM@SHAPE.

References

[1] Network of Excellence AIM@SHAPE, http://www.aimatshape.net/.

[2] SHREC, 3D Shape Retrieval Contest, http://www.aimatshape.net/event/SHREC.

[3] 3-D That Could Transform The Web, News Flash - Technology, Business Week, August 10th, 2000, http://www.businessweek.com/bwdaily/dnflash/aug2000/nf20000810_036.htm.

[4] TREC, Text Retrieval Conference, http://trec.nist.gov/.

[5] TRECVID, TREC Video Retrieval Evaluation, http://www-nlpir.nist.gov/projects/trecvid/.

[6] FRGC, Face Recognition Grand Challenge, http://www.frvt.org/FRGC/.

[7] VOC, Visual Object Classes challenge, http://www.pascal-network.org/challenges/VOC/.

[8] MIREX, Music Information Retrieval Evaluation Exchange, http://www.music-ir.org/mirex2006/.

[9] Initiative for the Evaluation of XML, http://www.music-ir.org/mirex2006/.

[10] FVC, Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2006/.

[11] M. Chaouch, A. Verroust-Blondet. Enhanced Silhouette and Depth-buffer based approaches for 3D shape retrieval. These proceedings.

[12] P. Daras, D. Tzovaras, A. Mademlis, A. Axenopoulos, D. Zarpalas, M. G. Strintzis. 3D search and retrieval using Krawtchouk moments. These proceedings.

[13] S. Jayanti, Y. Kalyanaraman, J. Pu, K. Ramani. An engineering drawing approach to 3D shape retrieval. These proceedings.

[14] H. Laga, K. Chiara, M. Nakajima. 3D model retrieval using spherical extent functions and wavelet descriptors. These proceedings.

[15] A. Makadia, K. Daniilidis. Light Field Similarity for Model Retrieval These proceedings.

[16] P. Papadakis, I. Pratikakis, S. Perantonis, T. Theoharis. A concrete radialized spherical projection descriptor for 3D shape retrieval. These proceedings.

[17] P. Silane, T. Funkhouser. Partial Matching of 3D Shapes with Priority-Driven Search.

[18] T. Zaharia, F. Prêteux. The canonical 3D Hough transform descriptor. These proceedings.

[19] Princeton Shape Benchmark, `http://shape.cs.princeton.edu/benchmark/`.

[20] R. Typke, R. C. Veltkamp, F. Wiering. Evaluating retrieval techniques based on partially ordered ground truth lists. In: Proceedings International Conference on Multimedia & Expo (ICME) 2006.

[21] K. Järvelin, J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. In: ACM Transactions on Informations Systems, 20(4), p. 422-446, 2002.

[22] Shape Modeling International 2006, `http://www.ifs.tohoku.ac.jp/SMI06/`.

[23] R. C. Veltkamp, L. J. Latecki. Properties and Performances of Shape Similarity Measures. In: Proceedings of the IFCS06 Conference on Data Science and Classification, 2006.

Enhanced Silhouette and Depth-buffer based approaches for 3D-Shape Retrieval

Mohamed Chaouch and Anne Verroust-Blondet
INRIA Rocquencourt
Domaine de Voluceau, B.P. 105
78153 Le Chesnay Cedex, FRANCE
{Mohamed.Chaouch,Anne.Verroust}@inria.fr

Abstract

We describe here two methods for 3D model indexing and retrieval using 2D/3D shape descriptors based on silhouettes or depth-buffer images we used in the SHREC - 3D Shape Retrieval Contest organized by the Network of Excellence AIM@SHAPE. Considering that the views characterizing the 3D model don't have the same value of relevance in the 3D-shape description, we associate to each view a relevance index which will be afterward used in the estimation of the degree of similarity between the 3D objects.

1 Introduction

The 3D retrieval methods we propose in the SHREC - 3D Shape Retrieval Contest organized by the Network of Excellence AIM@SHAPE is presented in [1]. It is based on the computation of the 2D views of the 3D models. We take into account the relevance associated to the 2D views in the shape matching algorithm. Several relevance index models measuring the amount of significant information associated to a 2D view are introduced for silhouettes and depth-buffer images. In this paper, we describe our method and present its evaluation on the contest collection which consists of the unclassified and renamed set of models from the Princeton Shape Benchmark [3]. Our paper [1] contains more details and discussions on the choice of relevance index methods.

2 Similarity search of 3D models

To respect the diversity of information among various views, we will associate to each silhouette or depth-buffer image a relevance value which will be used in the estimation of the degree of similarity between two 3D objects. This approach can be applied to any type of 2D/3D descriptors. Here, we consider two 2D/3D shape descriptors based on the Fourier transform of rendered silhouettes and depth-buffer images of the 3D models, presented in [2]. These descriptors require pose normalization to provide invariance under similarity transformations. Thus, there are four major steps for measuring the similarity between 3D models:

1. Normalize pose of 3D model. The Principal Component Analysis (PCA) generates an alignment of a 3D-mesh model into a canonical coordinate system frame by translating, rotating, reflecting, and scaling the set of vertices. We have retained the ”Continuous” PCA [5, 6] because it appears to be more complete and the most stable of all the approaches we have studied.

2. Extract feature vectors (2D/3D descriptors).

3. Determine relevance indices of projection images.

4. Compute the dissimilarity between 3D models.

2.1 2D/3D Descriptors

Two 2D/3D shape descriptors of [2] have been retained in our 3D model retrieval process. They both need a CPA normalization process.

The first one uses silhouettes. Each 3D model is projected perpendicularly on the planes of his own canonical bounding cube, in order to generate three silhouette images. Each silhouette $\mathbf{s_i}$ = $\{s_i(a,b);\ s_i(a,b) \in\ \{0,1\}$, $a = 0,...,N-1, b = 0,...,N-1\}$, $i \in \{1,2,3\}$, is defined as an outer contour approximated by K equally-spaced sequential points $\mathbf{c_i} = \{c_i(t); t = 0,...,K\}$, $i \in \{1,2,3\}$. The c_i are selected, computing the intersection of the contour with a set of rays emanating from the image center O and having a uniform angular distribution. They form the input to the one-dimensional discrete Fourier transform (1D-DFT). The shape feature vector of a 3D model is formed by the low frequency part coefficients of the Fourier spectrums, $\mathbf{F}^i$, of the three contour images.

The second 2D/3D shape descriptor is based on depth-buffer images. Six depth-buffer images are associated to the faces of the extended bounding box. The 6 NxN pixels images $\mathbf{v_i}$ = $\{v_i(a,b);\ v_i(a,b) \in [0,1]$, $a = 0,...,N-1$, $b = 0,...,N-1\}$, $i \in \{1,...,6\}$, with $N = 2^l$, are transformed using the two-dimensional discrete Fourier transform (2D-DFT) to represent the feature in the spectral domain. Finally, each depth-buffer image $\mathbf{v_i}$, $i \in \{1, ..., 6\}$, is represented by the absolute values of k low-frequency Fourier coefficients, $\mathbf{F}^i$, and constitute the feature vector of dimension $6k$.

2.2 Relevance index models

The relevance index indicates the density of information contained in the 2D image and is calculated for each view characterizing the 3D model. Since the optimal way of measuring the relevant information about 2D-shape is not prescribed, we can consider a variety of different methods to define relevance index. It depends on the nature of information extracted from the 2D-shape such as area, contour, curvature, depth, structure, connectivity, compactness... In this section, for only one information type, we define one relevance index. However, to capture another relevance over different aspects and characteristics of a 2D-shape, we can combine different relevance indices described in what follows. To compute the relevance indices, we can use several statistical measures such as average or variance. For the SHREC - 3D Shape Retrieval Contest, we have chosen one relevance index model for each type of 2D views, considering the results presented in [1].

2.2.1 Silhouette relevance index model

The relevance index associated to a silhouette image is based on the computation of the area of the projected surface of the 3D model on the corresponding face of the bounding box.

$$R_a = card\{s_{ab} | s_{ab} = 1, 0 \leq a, b \leq N-1\}, \qquad (1)$$

where s_{ab} is the pixel value of the image at position (a, b). We can also retain this relevance value for depth-buffer images, keeping only depth pixels with positive values.

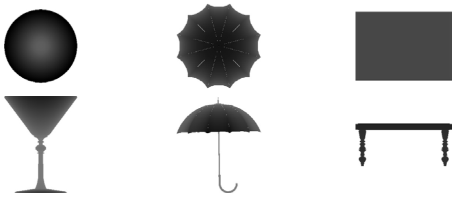


Figure 1. Limitation of the relevance interpretation (using R_a).

Using the area of the projected surface, the results will give the smallest relevance values for the images of the second row of Figure 1, but it is obvious that the corresponding views are more relevant than on the first row. Then the R_a is not suitable for measuring the relevance index in some cases. To moderate the influence of the area, we take the square root of the relevance defined in Equation (1) as relevance index (see the values of R_{sa} for the human biped model in Figure 2):

$$R_{sa} = \sqrt{card\{s_{ab} | s_{ab} = 1, 0 \leq a, b \leq N-1\}} \qquad (2)$$

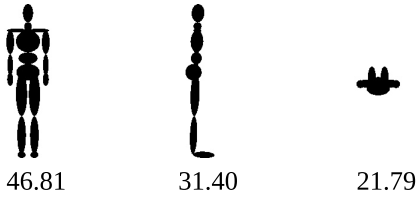


Figure 2. Three silhouettes images of Human-biped model and their R_{sa} (%) relevance values.

2.2.2 Depth-buffer relevance index model

The relevance index associated to a depth-buffer image is the sum of the distances between the center of mass of the 3D model and all visible points of the 3D model:

$$R_g = \frac{1}{2w} \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} d_{ab}, \qquad (3)$$

$$d_{ab} = \sqrt{|a - N/2|^2 + |b - N/2|^2 + 2w|v_{ab} - 1/2|^2},$$

where $2w$ is the length of the sides of the extended enclosing bounding box. The values of R_g for the human-biped model are given in Figure 3.

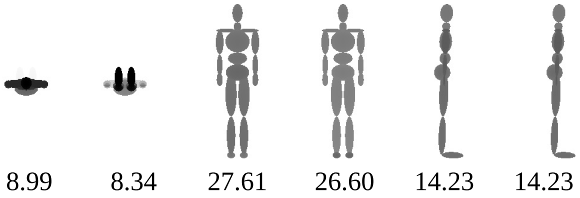


Figure 3. R_g (%) relevance values for the depth-buffer images of the human-biped model.

2.2.3 Normalized relevance indices

Let R^i be the relevance indices associated respectively to the views $1 \leq i \leq N_v$. We use the normalized relevance indices $\bar{R}^i$:

$$\bar{R}^i = R^i / \sum_{i=1}^{N_v} R^i. \qquad (4)$$

2.3 Matching two 3D models

To compare two 3D models $O1$ and $O2$, we generate the feature vectors $\mathbf{F}_1^i$ and $\mathbf{F}_2^i$, $1 \leq i \leq N_v$ of the silhouettes or depth-buffer images. Then, we compute the relevance indices of the images associated to each object, $\bar{R}_1^i$ and $\bar{R}_2^i$,

$1 \leq i \leq N_v$, where N_v is the number of views. Finally, the dissimilarity between a pair of 3D models $O1$ and $O2$ is:

$$\Delta(O1, O2) = \sum_{i=1}^{N_v} d(\bar{R}_1^i \mathbf{F}_1^i, \bar{R}_2^i \mathbf{F}_2^i). \qquad (5)$$

We take $N_v = 3$ silhouettes and $N_v = 6$ depth-buffer images of each 3D object. In what follows, ESA and EDBA denote our enhanced approach based respectively on silhouettes and depth buffer images.

3 Experimental results

The experimental results presented here were obtained with the following parameters: 256x256 size for the projection images; 100 (resp. 73) low-frequency Fourier coefficients for one silhouette (resp. depth-buffer) image; l_1 distance for shape similarity computation. The runfiles number 1 and 2 were obtained using respectively the EDBA and the ESA approaches. In figures 4 and 5, the depth-buffer images of the 30 query models of the SHREC - 3D Shape Retrieval Contest and their relevance index values are shown. The classification and the relevance assessments computed by the SHREC organizers (cf. table 1) confirm the measures we have made on the Princeton Shape Benchmark in [1]: the EDBA is more efficient than the ESA. Our descriptors are better in precision than in recall: in most of the cases, they return a good proportion of models belonging to the query's class in the beginning, but, when there is a great variety of shapes inside a same class, they are not efficient to retrieve all the models of the query's class. Its is why the mean normalized discounted cumulated gain is often very good for 5 and 10 and looses its efficiency for 25, 50 and 100, in comparison with the other descriptors. Moreover, introducing relevance index values in the similarity computation amplifies the precision vs the recall performance of our descriptors.

Let us examine now how the EDBA and ESA methods work on the SHREC queries:

- the queries 2 (computer monitor), 11 (vase), 21 (screwdriver) and 25 (mailbox) seem to be not significant due to the poor results given by all the SHREC participants.

- EDBA obtained good results for queries 1 (table single leg round), 7 (head body part), 9 (sedan car), 10 (sports car), 15 (human biped), 16 (chess piece) and 30 (trex biped). We notice that in these cases, the queries'classes are relatively homogeneous in shape and have relevant 2D views.

- ESA obtained poor results for query 26 (fish). When examining the models present in the fish class, we see that ESA retrieves with difficulty fishes from which the silhouette postpones the query's one (more stretched out, presence of fins, etc...).

In conclusion, EDBA has obtained good retrieval results in the SHREC classification but is more efficient in precision than in recall. As it was positioned in second (cf. [1]) computing mean NN, FT, ST and DCG on the Princeton Shape Benchmark w.r.t. the descriptors presented in [4], it would be interesting to compute these measures for all the descriptors given by the SHREC participants to see if these two classifications give similar results.

	EDBA	**ESA**	**Best value**
MFT %	38.1274	32.1507	44.7698
MST %	22.8599	21.0847	27.8649
MADR	0.5001	0.4215	0.5498
MNDCG 5	0.6825	0.5936	0.7015
MNDCG 10	0.6286	0.5490	0.64470
MNDCG 25	0.5469	0.4852	0.5906
MNDCG 50	0.52748	0.47129	0.5831
MNDCG 100	0.5293	0.4615	0.5914

Table 1. The mean first tier (MFT), the mean second tier (MST), the mean average dynamic recall (MADR) and the mean normalized discounted cumulated gain (MNDCG) for all the queries

Acknowledgments

We would like to thank Arghyro Paouri who helped us in building the five 3D models we gave to the SHREC - 3D Shape Retrieval Contest organizers. This work has been supported in part by the DELOS Network of Excellence on Digital Libraries (EU IST Network of Excellence G038-507618).

References

[1] M. Chaouch and A. Verroust-Blondet. Enhanced 2D/3D approaches based on relevance index for 3D-shape retrieval. In *International Conference on Shape Modeling and Applications 2006 (SMI 2006)*, June 2006.

[2] M. Heczko, D. A. Keim, D. Saupe, and D. V. Vranic. Verfahren zur Ähnlichkeitssuche auf 3D-objekten (methods for similarity search on 3D databases). *Datenbank-Spektrum*, 2(2):54–63, 2002.

[3] U. Princeton 3D models search engine. http://shape.cs.princeton.edu/search.html.

[4] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling and Applications Conference, SMI'2004*, pages 167–178, Genova, Italy, June 2004. IEEE.

[5] D. Vranic, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhune-Loeve transform and spherical harmonics. In *2001 Workshop Multimedia Signal Processing*, Cannes, France, Oct. 2001.

[6] D. V. Vranic and D. Saupe. 3D shape descriptor based on 3D Fourier transform. In K. Fazekas, editor, *EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001)*, pages 271–274, Budapest, Sept. 2001.

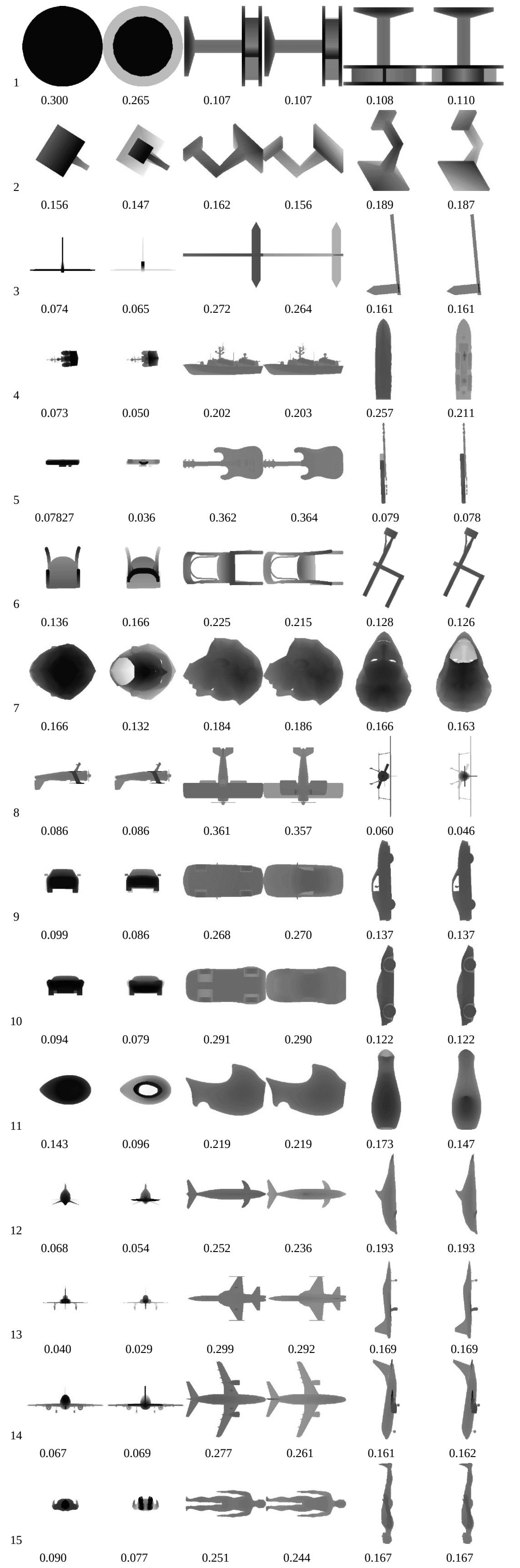


Figure 4. The relevance index values of the query objects (1-15) for depth-buffer images

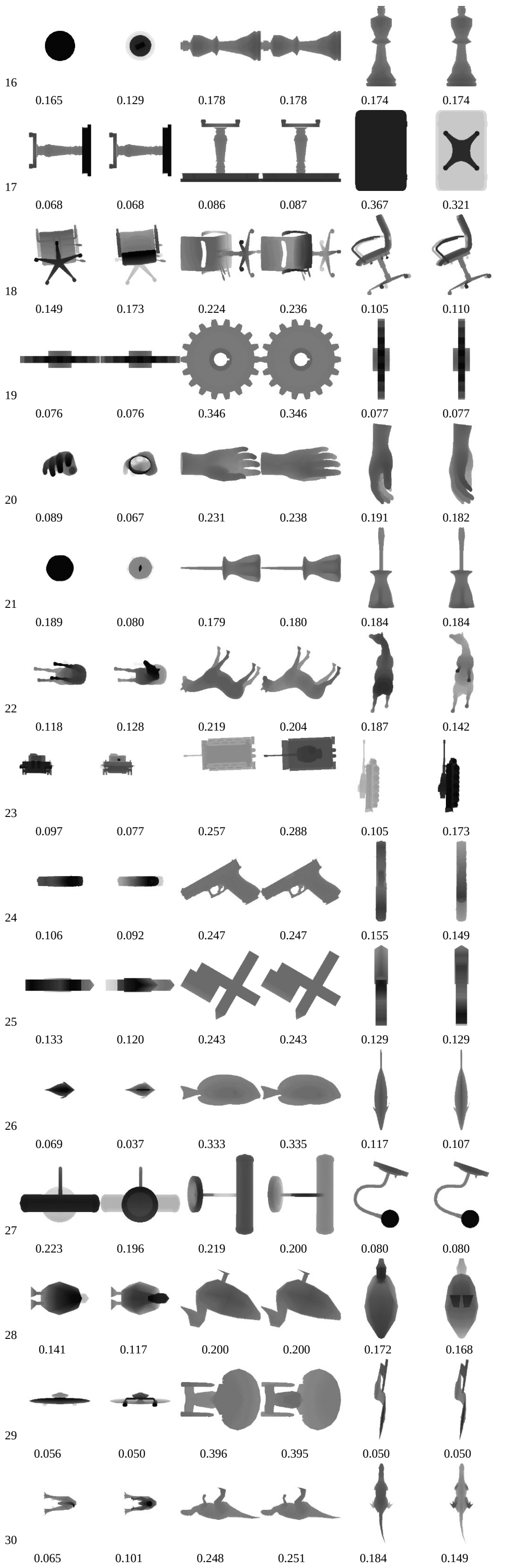


Figure 5. The relevance index values of the query objects (16-30) for depth-buffer images

3D Search and Retrieval using Krawtchouk Moments

Petros Daras [1], Dimitrios Tzovaras [1], Athanasios Mademlis [2],
Apostolos Axenopoulos [1], Dimitrios Zarpalas [1] and Michael. G. Strintzis [1,2]

[1]Informatics and Telematics Institute
1st Km Thermi-Panorama Road,
Thessaloniki 57001, Greece
Email: daras@iti.gr

[2]Information Processing Laboratory
Electrical and Computer Engineering Department
Aristotle University of Thessaloniki, Greece
540 06 Thessaloniki, Greece

Abstract

In this paper a combination of a rotation invariant method with a method which utilizes rotation normalization is proposed. Both methods used are based on 2D/3D Krawtchouk moments. The first method is an extension of that which was originally introduced in [1] and utilizes 2D Krawtchouk moments while the second method was originally introduced in [2] and utilizes 3D Krawtchouk moments.[1]

1 Introduction

The huge amount of 3D models available and the increasingly important role of multimedia in many areas such as medicine, engineering, architecture, graphics design etc, showed up the need for efficient data access in 3D model databases. An important question arises, is how to search efficiently for 3D objects into many freely available 3D model databases. A query by content approach seems to be the simpler and more efficient way.

2 Spherical 3D Trace Transform Approach

Given a 3D object, its volumetric binary function is calculated $f_b(\boldsymbol{x})$, where $\boldsymbol{x} = (x, y, z)$ and $x, y, z \in [0, 2N)$, by voxelizing the whole object, which is defined as:

$$f_b(\boldsymbol{x}) = \begin{cases} 1, & \text{when } \boldsymbol{x} \text{ lies within the 3D model's volume,} \\ 0, & \text{otherwise.} \end{cases}$$

In order to achieve translation invariance the 3D object's center of mass is calculated and the model is translated so as its center of mass coincides with the coordinates system origin. Afterwards, the maximum distance $maxD$ of the most distant voxel from the origin is found and the model is scaled by the factor $1/maxD$, hence scaling invariance is also accomplished.

Then, every eight neighboring voxels are grouped, forming a bigger one and $f_b(\boldsymbol{x})$ is transformed to the integer volumetric function of the model $f(\boldsymbol{x})$, which takes values from 0, none of the eight voxels lying inside the object's volume, to 8, all of them lie inside, and $\boldsymbol{x} = (x, y, z)$, $x, y, z \in [0, N)$. This transformation denotes that more attention is given to the voxels lying inside the object's volume, which characterize more reliable the 3D object.

2.1 Decomposition of $f(\boldsymbol{x})$

The next step involves the decomposition of $f(\boldsymbol{x})$ into planes. Each plane in the 3D space can be fully described by the spherical coordinates (ρ, θ, ϕ) of the point on which the plane is tangential to a sphere originated from the center of the coordinate system. Imagine concentric spheres, simulated by icosahedra whose triangles are subdivided in many smaller equilateral triangles. The barycenters of these triangle are considered to be the characteristic (tangential) points for the planes.

The intersection of each plane with the object's volume provides a spline of the object, which can be treated as a 2D image with dimensions $N \times N$. Consider a 2D functional F, which is applied to this 2D image, producing a single value. Let us assume that the result of that functional when applied to all splines, will be a function whose domain is the set of the aforementioned points, and its range is the results of the functional. The mathematical expression of that transformation can be written as:

$$F\{f(\boldsymbol{x})\} = g(\rho, \theta, \phi)$$

Restricting to different values of ρ, $g(\rho, \theta, \phi)$ can be considered as a set of functions $g_\rho(\theta, \phi)$ whose domain is con-

[1] This work was supported by the PENED2003 project of the Greek Secretariat of Research and Technology

centric spheres. Now, let T be a functional which can be applied to a function defined on a sphere, producing a single value. Then, the result of the T functional to every $g_\rho(\theta, \phi)$ is a vector with length equal to the number of the spherical functions. The Krawtchouk moments were used as F functionals and the Spherical Fourier Transform as T functional.

2.2 Rotation Invariance Requirements

In order to produce rotation invariant descriptor vectors two requirements should be met. Imagine that the model is rotated, hence $f(\boldsymbol{x})$ is rotated, producing $f'(\boldsymbol{x}) = f(\boldsymbol{Ax})$, where $\boldsymbol{A}$ a 3D rotation matrix. The splines that will be derived from $f'(\boldsymbol{x})$ will be the same with the ones derived from $f(\boldsymbol{x})$, but the characteristic points of the planes will also be rotated by the same rotation matrix. This transformation can be translated to a rotation of $g_\rho(\theta, \phi)$ by $\boldsymbol{A}$. This problem can be settled by using a rotation invariant T functional. Moreover, the planes that are perpendicular to the axes of rotation, will be rotated around their characteristic point, resulting to a rotated version of their 2D image. Thus, the F functionals should also be rotation invariant.

2.3 2D Krawtchouk Moments

Krawtchouk moments [3] is a set of moments formed by using Krawtchouk polynomials as the basis function set. The n^{th} order classical Krawtchouk polynomials are defined as:

$$K_n(x; p, N) = \sum_{\kappa=0}^{N} a_{\kappa,n,p} x^\kappa \Rightarrow$$

$$\Rightarrow K_n(x; p, N) =_2 F_1(-n, -x; -N; \frac{1}{p}) \qquad (1)$$

where $x, n - 0, 1, 2, \ldots, N$, $N > 0$, $p \in (0, 1)$, ${}_2F_1$ is the hypergeometric function defined as:

$${}_2F_1(a, b; c; z) = \sum_{\kappa=0}^{\infty} \frac{(a)_\kappa (b)_\kappa}{(c)_k} \frac{z^\kappa}{\kappa!}$$

and $(a)_\kappa$ is the Pochhammer symbol given by:

$$(a)_\kappa = a(a+1)\ldots(a+\kappa-1) = \frac{\Gamma(a+\kappa)}{\Gamma(a)}$$

where $\Gamma(.)$ is the gamma function.

For each $\hat{f}(i, j)$ with spatial dimension $N \times N$, the Krawtchouk moment invariants can be defined using the classical geometric moments:

$$M_{km} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} i^k j^m f(i, j)$$

The standard set of geometric moment invariants, which are independent of rotation [4] can be written as:

$$\nu_{km} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [icos\xi + jsin\xi]^k [jcos\xi - isin\xi)]^m f(i, j)$$

where $\xi = (1/2)tan^{-1} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}$ and μ are the central moments:

$$\mu_{pq} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - \bar{x})^p (j - \bar{y})^q \hat{f}_t(i, j), \quad p, q = 0, 1, 2, \ldots$$

The value of ξ is limited to $-45^o \leq \xi \leq 45^o$. In order to obtain the exact angle ξ in the range of 0^o to 360^o modifications described in detail in [5] are required.

Following the analysis described in [3], the rotation invariant Krawtchouk moments are computed by:

$$\tilde{Q}_{km} = [\rho(k)\rho(m)]^{-(1/2)} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i,k,p_1} a_{j,m,p_2} \nu_{ij} \quad (2)$$

where the coefficients $a_{\kappa,n,p}$ can be determined by (1), and

$$\rho(n) = \rho(n; p, N) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n} \qquad (3)$$

In this paper, parameter p of Krawtchouk polynomials has been selected to be $p = 0.5$

2.4 Spherical Fourier Transform

Spherical harmonics [6] are special functions on the unit sphere, generally denoted by $Y_{lm}(\boldsymbol{\eta})$, where $l \geq 0$, $|m| \leq l$ and $\boldsymbol{\eta}$ is the unit vector in $\mathcal{R}^3$, $\boldsymbol{\eta} = [cos\phi sin\theta, sin\phi sin\theta, cos\theta]$. Using this notation, $g_\rho(\theta, \phi)$ can be rewritten as $g_\rho(\boldsymbol{\eta})$.

These functions form a complete orthonormal set on the unit sphere:

$$\sum_{i=1}^{N_s} Y_{lm}(\boldsymbol{\eta}_i) Y_{kj}(\boldsymbol{\eta}_i) = \delta_{lk}\delta_{mj} \qquad (4)$$

where N_s is the total number of sampled points on the unit sphere (in our case the number of the equilateral triangle barycenters of the icosahedron). Hence, each function $g_\rho(\boldsymbol{\eta})$ can be expanded as an infinite Fourier series of spherical harmonics:

$$g_\rho(\boldsymbol{\eta}_i) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \alpha_{lm} Y_{lm}(\boldsymbol{\eta}_i), \qquad i = 1, \ldots, N_s \quad (5)$$

where the expansion coefficients α_{lm} are determined by:

$$\alpha_{lm} = \sum_{i=1}^{N_s} g_\rho(\boldsymbol{\eta}_i) Y_{lm}(\boldsymbol{\eta}_i) \Delta\boldsymbol{\eta} \qquad (6)$$

where $\Delta\boldsymbol{\eta}$ is the area of each triangle, hence $\Delta\boldsymbol{\eta} = \frac{4\pi}{N_s}$, since all the equilateral triangles have the same area and each icosahedron is assumed to be of unit radius. The overall vector length of α_{lm} coefficients with the same l:

$$A_l^2 = \sum_m \alpha_{lm} \tag{7}$$

is preserved under rotation and this is the reason why the quantities A_l are known as the rotationally invariant shape descriptors. In the proposed method, for each l the corresponding A_l is a spherical functional T.

2.5 Descriptor Extraction

For each F functional, a descriptor vector with length $L{\cdot}N_\rho$, where $N_\rho = 20$ and $L = 26$, is produced. In our experiments the first four Krawtchouk moments ($\tilde{Q}_{00}$, $\tilde{Q}_{10}$, $\tilde{Q}_{11}$, $\tilde{Q}_{20}$) were used as F functionals and four descriptor vectors were formed ($\mathbf{D}_{3DTrace00}$, $\mathbf{D}_{3DTrace10}$, $\mathbf{D}_{3DTrace20}$ and $\mathbf{D}_{3DTrace11}$, respectively).

3 3D Krawtchouk moments Approach

In this section the necessary steps followed so as to obtain descriptor vectors based on the 3D Krawtchouk moments are given.

3.1 Rotation Estimation

An essential part of the approach contains a novel combination of the two dominant rotation estimation methods, PCA and VCA [7], which have been proposed so far in the literature.

The VCA method achieves more accurate rotation estimation results than PCA when the 3D objects are composed of large flat areas. Otherwise, PCA produces better results than VCA. The proposed fully automatic approach tracks wrong rotation estimated objects produced either from PCA, or from VCA, and selects the most appropriate one.

In this paper for every model, rotation normalization is estimated using both PCA and VCA. Then, the volume of the bounding boxes parallel to principal axes are computed and the method which leads to minimum volume is chosen.

3.2 Extraction Of Krawtchouk Descriptors

In [3], Yap *et al.* introduced Krawtchouk moments and Krawtchouk moment invariants for image analysis, 2D object recognition and region based feature extraction (2D case), based on Krawtchouk polynomials. Their work was extended in 3D case [2] and the discrete Weighted 3D Krawtchouk moments were defined. A short description of this extension is presented in the sequel.

As it was mentioned earlier (Section 2), $f(\boldsymbol{x})$ is the volumetric representation of the 3D object. Then, the 3D Krawtchouk moments of order (n+m+l) of f, are defined as:

$$\begin{aligned}\bar{Q}_{nml} \quad = \quad & \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} \bar{K}_n(x;p_x,N-1)\times \\ & \times\bar{K}_m(y;p_y,N-1)\bar{K}_l(z;p_z,N-1)\times \\ & \times f(x,y,z) \end{aligned} \tag{8}$$

where

$$\bar{K}(x;p,N) = K_n(x;p,N)\sqrt{\frac{w(x;p,N)}{\rho(n;p,N)}} \tag{9}$$

and

$$w(x;p,N) = \left(\begin{array}{c} N \\ x \end{array}\right) p^x(1-p)^{N-x} \tag{10}$$

The 3D Krawtchouk moments can then be used to form the descriptor vector of every object. Specifically, the descriptor vector is composed of 3D Krawtchouk Moments up to order s, where s is experimentally selected to be $s = 18$.

$$\mathbf{D}_{3DKraw} = \left[\bar{Q}_{nml}\right], n+m+l = 0\ldots s \tag{11}$$

4 Matching

The first step is the normalization of each descriptor according to:

$$\tilde{D}(i) = \frac{1}{\sum_{i=1}^{T}|D(i)|}D(i)$$

where T is the number of descriptors in the descriptor vector $\mathbf{D}$, $D(i)$ is the $i-th$ element of $\mathbf{D}$, and $\tilde{D}(i)$ is $i-th$ element of the normalized vector $\tilde{\mathbf{D}}$.

Then, the well-known L1-norm defined as:

$$L_1(A,B) = \sqrt{\sum_{i=1}^{T}|\tilde{D}^A(i) - \tilde{D}^B(i)|}$$

is used for every normalized descriptor vector $\tilde{\mathbf{D}}_{3DKraw}$, $\tilde{\mathbf{D}}_{3DTrace00}$, $\tilde{\mathbf{D}}_{3DTrace10}$, $\tilde{\mathbf{D}}_{3DTrace20}$, $\tilde{\mathbf{D}}_{3DTrace11}$ and five distances are computed: $L_{3DTrace00}$, $L_{3DTrace10}$, $L_{3DTrace20}$, $L_{3DTrace11}$ and L_{3DKraw}, each one for every normalized descriptor vector. It has to be mentioned that due to the ambiguity of axis orientation after the rotation estimation that takes place for the 3D Krawtchouk approach, the distance is selected to be the minimum for every possible orientation.

The total distance is computed as follows:

$$\begin{aligned} L_{tot} &= a_1 L_{3DTrace00} + a_2 L_{3DTrace10} + a_3 L_{3DTrace20} \\ &\quad + a_4 L_{3DTrace11} + a_5 L_{3DKraw} \end{aligned} \tag{12}$$

where $a_1 = a_4 = 0.15$, $a_2 = a_3 = 0.25$ and $a_5 = 0.2$. These values were experimentally selected.

5 Results

The proposed method was evaluated in terms of retrieval accuracy, using the Princeton Shape benchmark (PSB) database which consists of 1814 3D objects. The performance of the proposed method against the other 16 competitive ones which took part in the SHREC contest, was proved to be among the best. The results published by the contest organizers have shown that the proposed method clearly outperforms the other methods if we take into account the first 10% of the retrieved results and it is among the first 3-4 methods concerning the overall performance. Also, it should be clearly stated that the proposed method is based on a native 3D descriptor extraction algorithm.

Very useful conclusions can be derived by examining the Mean Normalized Cumulated Gain (MNCG) and the Mean Normalized Discounted Cumulated Gain (MNDCG) graphs. Both graphs visualize the performance of the retrieval methods as a function of the retrieved results. However, MNDCG applies a discount factor to devaluate late-retrieved results and, thus, it is an appropriate user-oriented evaluation metric for retrieval applications. Our method is always in the first three positions based on MNDCG and in the first four based on MNCG. It should be noticed that our method is ranked first using both MNCG and MNDCG, considering the first 10% of the retrieved results. That means that the proposed method first retrieves the more relevant to the query 3D objects.

Specifically, based on MNDCG, the proposed method is ranked first for 5% and 10% of the retrieved results, second for 25% and third for 50% and 100%. Based on MNCG, the proposed method is first at 5%, third at 25% and forth at 50% and 100%. However, by examining the MNCG graph, methods ranked in the second, third and forth position after 50% of the retrieved results change consecutively.

Rank	Participant	RunNr	Mean ADR
1	Makadia et al.	2	0.54986260
2	Makadia et al.	1	0.54084843
3	*Daras et al.*	*1*	*0.52424060*
4	Chaouch et al.	1	0.50018275
5	Pratikakis et al.	1	0.49523294
6	Shilane et al.	3	0.49371490
7	Zaharia et al.	1	0.49247277
8	Shilane et al.	2	0.48770607
9	Chaouch et al.	2	0.42156762
10	Shilane et al.	1	0.39706558
11	Makadia et al.	3	0.39249521
12	Makadia et al.	4	0.37667266
13	Laga et al.	1	0.32631385
14	Laga et al.	2	0.30619973
15	Jayanti et al.	2	0.26785165
16	Jayanti et al.	3	0.23702210
17	Jayanti et al.	1	0.23020707

Table 1: Ranking Results based on Mean Average Dynamic Recall (ADR) value. The Second and Third column show the name of the first author and the serial number of the method respectively

The proposed method is ranked third with respect to Mean First and Second Tier measures, if only the highly relevant objects considered as similar. However, if marginally relevant objects are considered as similar too, it is ranked 6-th and 7-th respectively, although the differences between the methods from 3rd to 7th position are close to 0.5%. The main reason for these results derives from the fact that marginally relevant objects are usually semantically and functionally similar rather than geometrically similar, while the proposed method does take into account only geometrical information. In addition, the proposed method is ranked third based on Mean Dynamic Average Recall metric, which measures the overall performance of the retrieval method. The Mean ADR takes into account both highly and marginally relevant objects with different weights. The results based on ADR are presented in Table 1. These results clearly rank the proposed method among the best in native 3D content-based search.

References

[1] P.Daras, D.Zarpalas, D.Tzovaras and M.G.Strintzis: “3D Model Search and Retrieval Based on the Spherical Trace Transform”,*IEEE International Workshop on Multimedia Signal Processing (MMSP 2004)*, Siena, Italy, October 2004.

[2] A. Mademlis, A. Axenopoulos, P. Daras, D. Tzovaras and M. G. Strintzis, “3D Content-based Search Based on 3D Krawtchouk Moments”, *In Proc. of Third International Symposium on 3D Data Processing, Visualization and Transmission* (3DPVT 2006), accepted for publication.

[3] P-T. Yap, R. Paramesran, “Image Analysis by Krawtcouk Moments”, *IEEE Trans. On Image Processing*, Vol. 12, No.11 pp. 1367-1377, November 2003.

[4] M.K.Hu, “Visual Pattern Recognition by Moment Invariants”, *IRE Trans. on Information Theory*, Vol 8, pp. 179-197, 1962.

[5] M. R. Teague, “Image analysis via the general theory of moments”, *Journal of Optical Society of America*, vol. 70, pp. 920930, 1979.

[6] D.W.Ritchie, “Parametric Protein Shale Recognition”, *PhD Thesis*, University of Aberdeen, 1998.

[7] J. Pu, and K. Ramani “ An Approach to Drawing-Like View Generation From 3D Models”, *In Proc. of IDETC/CIE 2005, ASME 2005*

An engineering drawing approach to 3D shape retrieval

Subramaniam Jayanti, Yagna Kalyanaraman, Jiantao Pu, Karthik Ramani
PRECISE, School of Mechanical Engineering
Purdue University, West Lafayette, IN 47906 USA

Abstract

This paper presents a 3D model retrieval approach by representing a 3D shape at three distinct levels of detail: contour level, silhouette level, and drawing level. This 3D-to-2D representation is supported by a 3D pose determination algorithm; to compute the similarity between 2D shapes at these levels, a combination of two different 2D shape descriptors is used to achieve a better performance. In addition, the results from the three levels were combined in for obtaining the final ranked retrieval.

1. Introduction

In the past decades, a large amount of 3D models have been accumulated in the engineering domain [1, 2]. Reusing and sharing the knowledge embedded in these models is becoming an important way to accelerate the design process, improve product quality, and reduce costs. Generally, the term 'similar models' is meant for objects that are globally similar, but still have some differences in specific features. Therefore, a good search strategy for a 3D retrieval system should not only consider the global but also the local shape comparison between 3D models. Meanwhile, efficient query interfaces are also critical for engineering applications.

In our research, we use three orthogonal views (front view, side view and top view) to represent a 3D model. The query can be a 2D sketch, legacy drawing or a 3D model [3, 4, 5]. In a benchmark study for engineering shapes, which evaluated 13 different shape representation methods, Iyer et al. [6] found that the view-based methods showed better retrieval performance. In a similar study Shilane et al. [7] showed that light field descriptor (LFD), another view-based approach, achieved the best performance.

In this study, we apply the Multiple Level of Detail (MLD) method, an extension of the orthogonal view-based 3D similarity approach presented in [5], by splitting the information in each view into three distinct levels of detail – silhouette, contour and drawing level. Generally, engineering shapes have high genus with several internal details and features, such as through and blind holes, which greatly affect the perceived similarity, as opposed to the global external shape. Through the SHREC experiments we have tried to explore an engineering perspective of shape applied to multimedia databases.

We evaluated different schemes for combining the similarity obtained from the three levels of detail, by combining distances. The proposed approach has many advantages: (1) it requires only three orthogonal viewing directions, which is concurrent with engineering view of shape and design; (2) it allows users to sketch queries at any level of detail, thus supporting a coarse-to-fine search as well as searching with legacy 2D drawings; and (3) it has better retrieval performance (i.e., accuracy and efficiency) compared to several shape descriptors and comparable performance to the LFD.

2. Multiple Levels of Detail representation

The multi-view, multiple level of detail approach described in this paper falls among the view-based methods, and was inspired by several factors arising from traditional engineering knowledge and from a well-known fact that the human visual perception of a shape is organized from coarse-to-fine details [9, 10, 11]. Three-view drawings are widely used and have historically played an important role in engineering and technology, because a single view such as an isometric or oblique view is not enough to reconstruct or manufacture a 3D model which holds rich topologic and geometric information [12]. Recently, Guan et al. [13] also proposed generating six orthogonal views using visual hulls for matching multimedia objects although other non-orthogonal, single-view approaches have been explored in [14].

From an engineering graphics viewpoint, deciding the best view is an important step in obtaining multi-view drawings. Generally in order to get the best views, the object must be positioned within an imaginary glass box such that the surfaces of major features are either perpendicular or parallel to the glass planes. The goal of this step is to create views with a minimum number of hidden lines and to draw the view of an object in its most natural position. Multi-view drawings provide the most accurate description of three-dimensional objects and structures for engineering and manufacturing requirements [15]. Hence, an orthogonal view-based approach to shape representation lends itself to easy interpretation, understanding, and interaction for engineering applications. In addition, the proposed method enables searching with freehand sketches and 2D engineering drawings as well as 3D models.

To obtain such 2D orthographic views, the following steps were involved: (1) a pose determination method to compute the pose of a 3D model by finding three orthogonal orientations and (2) a drawing-like view generation method [3]. Along the three orientations, the projected views have a good accordance with the three main views concept (i.e., top view, front view and side view) in engineering graphics.

2.1. 3D pose determination

Table 1: Examples of pose estimation

In [3], Pu and Ramani proposed a method to compute the pose of a 3D model by finding the orthogonal orientations with the maximum virtual contact area (VCA). VCA is defined as the bounding area formed by polygons that have the same distance from a predefined point and have the same normal. The key step in obtaining the principal axes is to determine the polygons of a 3D object that have the same normal and lie in the same plane. To obtain the direction along which the VCA is the maximum, we need to find all polygons that have the same normal direction and the same distance to the mass center. The direction that gives the maximum VCA is the first principal axis $\mathbf{b}^u$ of the 3D object orientation. To get the next principal axis $\mathbf{b}^v$ of an object orientation, we find the normal that satisfies two conditions: (a) has maximum area; and (b) is orthogonal to the first principal axis. The third axis can then be obtained by performing the cross product between $\mathbf{b}^u$ and $\mathbf{b}^v$.

A detailed comparison between the pose estimation methods based on VCA and existing pose estimation approaches such as Principal Component Analysis (PCA) and EGI/CEGI is available in [3]. In Table 1 a few examples from the multimedia and engineering domain are presented to show the generality of the proposed method.

2.2. MLD computation

Table 2: MLD based Representation

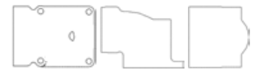
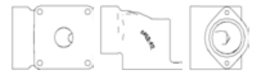
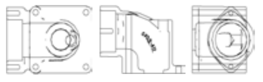
3D Model	
Contour Level	
Silhouette Level	
Drawing Level	

Table 2 shows an example of the proposed MLD representation of a 3D shape. The contour level reflects the global shape by which a user can “guess” the true object to some extent. The silhouette level conveys additional shape details using a few more additions to the contour level view. When the detailed shape information is not important, the silhouettes are enough to differentiate two similar objects with a higher confidence than the contour level. The third level contains the complete information, including the visual appearance and the occluded structure, using which a user can determine its shape precisely. Because of the intricate shape of engineering objects, we consider complete details in the third level. At the contour level, there are three different views along the principal axes; at the silhouette level, there are six different views; and at the full level, we use the three traditional drawing-like views along the principal axes to represent the drawing level. Below we will describe the algorithms to generate these views.

2.2.1 Contour level. To obtain the shape at the contour level, the 3D model is projected along the intended directions by setting the z-value of each polygon to zero and rendering it in image space. In implementation, this operation can be efficiently finished with the help of any classic algorithms such as Painter's Algorithm or the Z-buffer Algorithm.

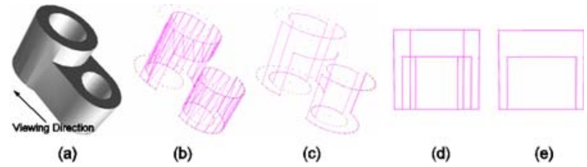


Figure 1: Four steps for silhouette generation: (a) an original 3D model; (b) the result after backface culling operation; (c) the result after inside face culling operation; (d) the view without occlusion culling (Drawing Level); and (e) the silhouette view of (a).

2.2.2. Silhouette Level. A silhouette edge connects two polygons, one of which faces toward the viewer and the other facing away from the viewer. When a 3D object is projected onto a 2D viewing plane, its visual appearance is formed by all its silhouette edges. The efficient extraction of silhouettes has been widely studied in many applications ranging from computer vision to non-photorealistic rendering. Earlier hidden line removal methods [20] have been used to get the silhouette edges. Recent methods render the silhouettes of 3D models in image space include [21, 22]. In our application, we propose a method by which the silhouettes of a 3D model are efficiently generated in object space instead of image space. Figure 1 shows the process of the silhouette generation.

The first step is culling backward polygons [16]. Our contribution in this step is in improving the efficiency of the second step of discarding the inside-edges. The inside-edge has a distinguishing property: *it is shared by two polygons*. With this definition, we can cull the inside-edges completely by traversing all the triangles with the help of a look-up table. The look-up table is an $m \times m$ matrix, where m is the number of remaining vertices after backface culling. The element (i, j) represents the edge formed by vertices i and j. First, we can code all vertices from 0 to $m - 1$. Then by traversing the 3D model after backface culling, we can fill this table according to the connecting relationship of the edges. Finally, the inside-edges are determined by checking whether the element at (i, j) is filled more than twice. In practice, to save memory, we can use bits to represent the table. Figure 1 (c) shows the inside-edge culled result. The computational complexity of the look-up table method is $O(n+m)$.

The third step is projecting the result after the inside-edge culling operation along the respective orientation. In order to render the silhouette the occluded structure is culled by executing a series of fast ray-triangle intersection tests. The final 2D view is obtained by projecting the remaining polygons onto the corresponding projection planes.

2.2.3. Drawing level. We perform all the steps from silhouette extraction (Section 2.2.2), except occluded edge culling, to obtain the drawing level views, which contain the complete shape information.

2.3. View Similarity

While pose determination is the process of registering the viewing direction of similar objects, we still need a shape descriptor with rotation invariance for registering the 2D geometric shapes no matter how they are rotated in object space. After view-generation, the 2D views thus obtained are represented using two different shape descriptors – 2.5D Spherical Harmonics and 2D Shape Histogram. Dissimilarity between a pair of 2D views is obtained as a linear sum of dissimilarities from these two shape descriptors. Given a pair of 3D objects (i.e. two sets of 2D views), the pair of 2D views which provide the best match (i.e. with the least dissimilarity) are the principal matching view pair. The other two matching view pairs are determined likewise from the remaining views. The total dissimilarity between two 3D models is obtained through the summation of dissimilarity between the three sets of matching view pairs.

2.3.1 2D Shape Histograms (2DSH). To measure the similarity between 3D shapes, Osada et al. [17] represented a 3D shape signature named shape distribution that is formed by random points sampled uniformly from the shape surface. We used the 2D analog of this method and call it 2D Shape Histogram (2D SH). For this method we sampled 500 points from the image and calculated the histogram of all the pair-wise distances among those points. For two histograms, i.e., **H**$_1$ and **H**$_2$, their similarity W is given by: $W(H_1, H_2) = L_n(H_1, H_2) = \sqrt[n]{\sum_{i=0}^{h} (H_1(i) - H_2(i))^{1/n}}$ (1)

where h is the number of bins in the histogram.

2.3.2 2.5D Spherical Harmonics (2.5DSH). As a robust rotation invariant descriptor, spherical harmonics representation has been successfully applied to 3D shape matching [18]. It arises on the sphere in the same way that the Fourier exponential function arises on the circle. Funkhouser et al. [1] used a 2D analog of the spherical harmonics to extract a series of rotation invariant signatures by dividing a 2D silhouette shape into multiple circular regions. However, this method has two major limitations: one angle-to-multiple distances characteristic of a spherical function and instability caused by shape perturbation. In order to overcome these limitations and thus obtain a set of rotation invariant signatures for a 2D drawing, the 2.5D Spherical Harmonics (2.5D SH) representation was developed in [11]. First a bounding sphere for a given 2D drawing is calculated and rays are cast in different directions from the centroid. The intersection points of the rays with the edges of the drawing are represented in a 3-dimensional coordinate system where the z-value is the distance from the centroid. By uniquely projecting a 2D view into the 3D space, as shown in Figure 2 (b), a representation of the 2D view is generated [11].

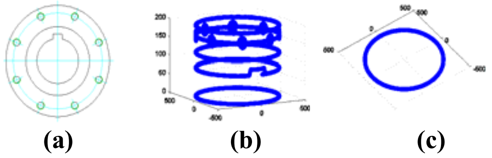


Figure 2: An example of 2.5D Spherical Harmonics representation: (a) is a 2D drawing; and (b), (c), and (d) show the 3D representation of the drawing.

Finally, to obtain the rotation-invariant descriptor, we use the fast spherical harmonics transformation method [19] in which a spherical function of bandwidth B is sampled on the 2B-many Chebyshev points and not the B-many Gaussian points. These sampled points form a 2B×2B equiangular grid along the longitude and latitude of a sphere.

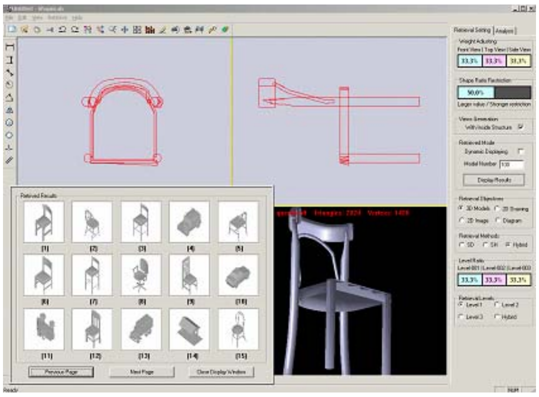


Figure 3: Retrieval results for a Chair model with Level 1 (Method 1)

3. Experiments with MLD

In this study we present results from three promising search strategies, each utilizing a different combination of search parameters. In the first strategy (result file: RUNNR 1) all the three levels of detail (Contour, Silhouette and Drawing) were employed in a linear sum combination while all the views were also given equal weight (0.333). In the second strategy (result file: RUNNR 2), we only used the Contour Level (Level 1) with equal weights on all the three views (0.333), since the Contour Level captures objects with low genus and is especially suited for retrieving multimedia objects such as humans, dinosaurs, etc. In the third strategy (result file: RUNNR 3), we increased the weight on the Front view, leading to a view weight combination of (75%, 12.5% and 12.5%) with Level 1. This strategy provides high retrieval performance for objects that can be distinguished from other objects in one main view.

4. Remarks

From the SHREC experiments we observed that the large number of internal lines generated for multimedia objects pose a challenge for the 2.5D Spherical Harmonics method leading to significant number of false positives (see Table 3). To gain a deeper understanding, we experimented with the popular image matching algorithm based on Fourier Descriptors and Zernike moments in conjunction with our pose estimation method. Significant improvement in retrieval performance was achieved with as can be seen from Table 1. The distance function is a sum of the normalized distances obtained from the two descriptors. The normalized weights for the Zernike moments and Fourier descriptors were set to equal value. Due to space constraints, we only present a subset of our results along with the best descriptor (Shilane et al., Method 3) from the SHREC competition for comparison.

Table 3: Comparison of retrieval performance

Run File	**Method**	**First tier (highly relevant)**	**Second tier (highly relevant)**	**Mean Average Precision**
Jayanti et al.	2	17.28%	12.22%	0.17084
Jayanti et al.	3	15.95%	11.64%	0.15204
Jayanti et al.	1	15.06%	11.07%	0.14377
Shilane et al.	3	40.87%	25.64%	0.53659
Zernike + Fourier	New	24.64%	8.75%	0.24178

5. Conclusions

From the SHREC experiments, we find that the MLD method is not readily suitable for searching in multimedia databases. However, this method provides the advantage of searching with freehand sketches and 2D drawings as well as 3D models [3D PVT]. Other image-based descriptors like Fourier Descriptors and Zernike moments can be used to improve the retrieval performance for multimedia databases.

6. References

[1] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D.,“A Search Engine for 3D Models,” ACM Transactions on Graphics, 22(1): 83-105, 2003.

[2] Chen, D.Y., Tian, X.P., Shen, Y.T., and Ouhyoung, M., “On Visual Similarity Based 3D Model Retrieval,” Computer Graphics Forum (Eurographics’2003), 22(3):223--232, 2003.

[3] Pu, J.T. and Ramani, K., “An Automatic Drawing-like View Generation Method from 3D Models”, Proc. ASME IDETC/CIE Conference, Long Beach, CA, September 24-28, 2005.

[4] Pu, J.T. and Ramani, K., “A 2D Sketch User Interface for 3D CAD Model Retrieval,” Journal of Computer Aided Design and Application, 2(6):717-727, 2005.

[5] Pu, J.T. and Ramani, K., “On Visual Similarity based 2D Drawing Retrieval”, Computer-Aided Design 38 (2006) 249–259.

[6] Iyer, N., Jayanti, S., and Ramani, K., “An Engineering Shape Benchmark for 3D models,” Proc. of ASME IDETC/CIE 2005, Long Beach, CA, September 24-28, 2005.

[7] Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T., “The Princeton Shape Benchmark,” Proc. Shape Modeling International 2004, pp.167-178, Genova (Italy), 2004

[8] Pu, J.T. and Ramani, K., “On Visual Similarity based 2D Drawing Retrieval”, Computer-Aided Design 38 (2006) 249–259.

[9] Schyns. P. G., “Diagnostic Recognition: Task Constraints, Object Information, and Their Interactions,” Cognition, 67: 147-179, 1998.

[10] Parker, D. M., Lishman, J. R., and Hughes, J., “Role of Course and Fine Spatial Information in Face and Object Processing,” Journal of Experimental Psychology, 22: 1448-1466, 1996.

[11] Sanocki, T., “Interaction of Scale and Time during Object Identification,” Journal of Experimental Psychology: Human Perception and Performance, 27, 290-302, 2001.

[12] Geng, W., Wang, J., and Zhang, Y., Embedding visual cognition in 3D reconstruction from multi-view engineering drawings Computer-Aided Design, Volume 34, Issue 4, 1 April 2002, Pages 321-336

[13] Guan, S.-H., Hsieh, M.K, Yeh, C.C., and Chen, B.-Y., “Enhanced 3D Model Retrieval System through Characteristic Views using Orthogonal Visual Hull.” ACM SIGGRAPH 2004 Conference, Los Angeles, California, USA., 2004.

[14] Polonsky, O., Patanè, G., Biasotti, S., Gotsman, C., and Spagnuolo, M., “What's in an image?” The Visual Computer 21(8-10): 840-847 (2005)

[15] Bertoline, G.R., Wiebe, E.N., (2005), Fundamentals of Graphics Communications, Fourth Edition, McGraw Hill, pp 191-220

[16] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., and Ramani, K., “Three Dimensional Shape Searching: State-of-the-art Review and Future Trends,” Computer-Aided Design, Volume 37, Issue 5, 15 April 2005, Pages 509-530.

[17] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D., “Shape Distribution,” ACM Transactions on Graphics, 21(4): 807-832, 2002.

[18] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S., “Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors,” In Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, Aachen (Germany), pp.156-164, 2003.

[19] Healy, D., Kostelec, P., Moore, S., “FFTs for the 2-Sphere--Improvements and Variations,” Journal of Fourier Analysis and Applications, 9(4): 341 – 385, 2003.

[20] Appel, A., “The Notion of Quantitative Invisibility and the Machine Rendering of Solids,” Proc. ACM National Conference 1967, pp.387-393, Washington D.C., 1967.

[21] Markosian, L., Kowalski, M.A., Trychin, S.J., Bourdev, L.D., Goldstein, D., and Hughes, J.F., “Real-time Non-photorealistic Rendering,” Proc. SIGGRAPH, pp.415-420, 1997.

[22] Raskar, R. and Cohen, M., “Image Precision Silhouette Edges,” Proc. of Symposium on Interactive 3D graphics, pp.415-420, Atlanta (USA), 1997.

[23] Khotanzad, A. and Hong, Y., “Invariant Image Recognition by Zernike Moments,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 12, No. 5, 1990, pp. 489-497.

[24] Zhang D. S. and Lu, G. J., (2001), “Shape retrieval using Fourier descriptors.” IEEE International Conference on Multimedia and Expo (ICME), 1139- 1142.

3D Model Retrieval using Spherical Extent Functions and Wavelet Descriptors

Hamid Laga Kunihiro Chihara
Graduate School of Information Science
Nara Institute of Science and Technology, Japan
{hamid, chihara}@is.naist.jp

Masayuki Nakajima
Computer Science Department
Tokyo Institute of Technology, Japan
nakajima@img.cs.titech.ac.jp

Abstract

In this paper we propose the use of spherical wavelet descriptors for content-based 3D model retrieval. Each model is represented with its spherical extent function (EXT). We consider both the latitude-longitude parameterization and the flat-octahedron-based parameterization. For each case, spherical wavelet descriptors are extracted and used as a mean for shape comparison. Experiments are performed on the Princeton Shape Benchmark.

1 Introduction

A challenging issue in content-based 3D model retrieval is the description of shapes with suitable numerical representations called *shape descriptors*. In general a shape descriptor should be discriminative, compact, easy to compute, and invariant under similarity transformations [7, 16].

In this paper we present a new content-based 3D model retrieval method relying on spherical wavelet transform (SWT) of the shape function. Spherical Wavelets have been introduced by Schröder et. al. [12] and since, they have been used to solve many geometry processing problems including 3D model compression [4]. Similar to first generation wavelets, SWT is an effective tool to analyze shape functions defined on the sphere as they provide a natural partition of the function spectrum into multiscale and oriented sub-bands. SWT is a natural extension of spherical harmonics [2] and 3D Zernike moments [9]. It offers better feature localization and takes all the advantages of wavelets over Fourier analysis.

Most of 3D shape retrieval techniques proposed in the literature aim to extract from the 3D model meaningful descriptors based on the geometric and topological characteristics of the object. Survey papers to the related literature have been provided by Tangelder et. al [15] and Iyer et. al [5]. They fall into three broad categories; feature-based including global and local features, graph-based and view-based similarity.

View-based techniques compare 3D objects by comparing their 2D projections. The Lightfields [1] are reported to be the most effective descriptor [13]. View-based techniques are suitable for implementing query interfaces.

Graph-based techniques compare 3D shapes by comparing their graph representation. The Reeb graphes [3], and skeletons [14] are among the most popular. Other techniques include methods based on the distribution of features such as shape distributions [10] and local features such as spin images [6].

Feature-based methods aim to extract compact descriptors from the 3D object. A popular approach is to represent the shape using functions defined on the unit sphere. Funkhouser et al.[2] uses spherical harmonics (SH) to analyze the shape function. They demonstrated later that spherical harmonics can be used to achieve rotation invariance provided that the shape function is defined on the sphere [7]. Novotni et. al. [9] uses 3D Zernike moments (ZD) as a natural extension of SH. Representing 3D shapes as functions on concentric spheres has been extensively used. Our developed descriptors fall into this category and are a natural extension of SH and ZD.

The issue of extracting invariant shape features is always an important problem in content-based 3D model retrieval. While translation and scale invariance can be easily achieved [11, 2, 7], rotation invariance is still a challenging issue. Methods that require pose normalization are based on PCA, or continuous PCA [18]. However, PCA-based alignment is known to misbehave and therefore, it hampers significantly the retrieval performance [7]. *Invariant* methods, describe shapes in a transformation invariant manner by discarding alignment-dependent shape information. This include the Spherical harmonics [17] and power spectrum-based [16, 7] methods. These approaches rely on the sampling of the shape function in the latitude and longitude directions, and therefore, the rotation invariance is affected by the sensitivity of the sampling stage [8].

In this paper we propose the evaluation of spherical extent function and wavelet descriptors for 3D shape retrieval. We discuss two parameterization methods: (1) the latitude-longitude parameterization, and (2) the flat octahedron-based parameterization (section 2). Then we propose the spherical wavelet descriptors which can be computed from the two parameterizations (section 3). Section 4 discusses the performance evaluation results. Section 5 concludes the paper and presents some issues for future research.

2 Shape parameterization

The steps commonly used to compare 3D shapes are: **(1) Normalization:** the center of mass of the object is translated to the origin $(0, 0, 0)$, and the object is scaled to fit within a unit ball. **(2) Parameterization:** compute a shape function at discrete locations $P = \{p_1, \ldots, p_k\}$ sampled from the parameterization domain. **(3) Shape description:** feature vectors are extracted from the parameterization and used as a mean for shape comparison.

Spherical parameterization is a popular method where a 3D shape is represented with functions on the unit sphere. For simplicity, we consider the Spherical Extent Function (EXT) [11], but our approach extends for any spherical function. Spherical Extent Function f represents the extent of the shape in the radial directions.

2.1 Latitude-longitude parameterization

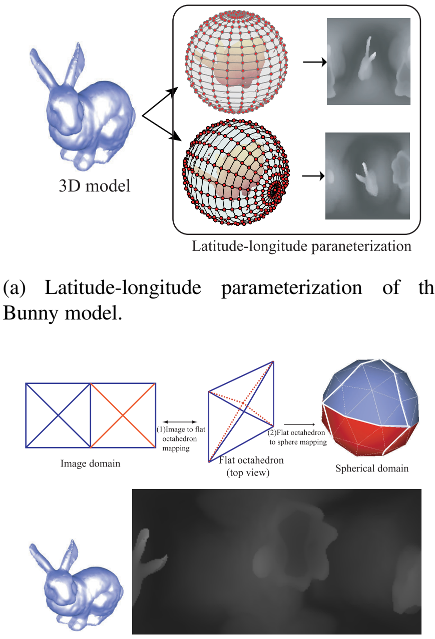
This popular parameterization is obtained by casting rays in different radial directions (θ, ϕ), where θ and ϕ correspond to the elevation and azimuth angles. Then the extent of the shape is measured along each direction. Figure 1a shows the parameterization of the Stanford Bunny model.

This representation, along with spherical harmonic transforms (SHT), has been previously used to extract efficient shape descriptors [7, 16]. The main observation, however, is that the obtained discrete shape function has singularities near the two poles. Spherical harmonic-based descriptors extracted from the latitude-longitude parameterization are also very sensitive to the sampling direction and resolution. Therefore, the rotation invariance of the power spectrum [7] is affected [8].

2.2 Flat octahedron-based parameterization

We aim to find a parameterization Φ that samples the shape uniformly in all directions. To achieve this in practice, we investigated two approaches originally proposed for spherical parameterization and geometry image compression [4]:

1. **Geodesic sphere.** we sample the shape function by casting rays from the shape's center of mass to the vertices of a geodesic sphere. The coarsest (level-0) representation of the shape function is obtained using a basic octahedron of 20 vertices. Finer levels are obtained by recursive subdivisions. Shape descriptors can be extracted using the second generation Spherical Wavelets [12]. In practice, however, this approach achieved very low performance.


3D model
Latitude-longitude paraneterization
Image domain
(1)Image to flat octahedron mapping
Flat octahedron (top view)
(2)Flat octahedron to sphere mapping
Spherical domain

(a) Latitude-longitude parameterization of the Bunny model.

(b) Flat octahedron parameterization procedure.

Figure 1: Parameterization procedures

2. **Flat octahedron parameterization.** Hoppe et. al [4] maps the sphere onto a rectangular domain using spherical parameterization of a *flattened octahedron* domain. As shown in Figure 1b, the left and right halves of the image domain are mapped respectively to the north and south hemispheres of the unit sphere. Note that the flattened octahedron unfolds isometrically onto a square image. Hence, image processing tools can be used with simple boundary extension rules.

The benefits of the flat octahedron-based parameterization is two fold; (1) the shape is sampled uniformly in all directions. This eliminates the singularities that appear at each pole in the latitude-longitude parameterization, and (2) the discrete shape function is invariant to rotations along the edges of the spherical triangles. We make use of these properties to build efficient shape descriptors.

3 Wavelet descriptors

To describe 3D models with efficient numerical representation we propose to analyze the shape function using wavelets tools which are known to be powerful for texture description. During the off-line database indexation, each 3D model is described with a spherical wavelet-based descriptor computed as follows:

1. **Initialization:**
   (a) Normalize the 3D model for translation, scale, rotation and axis flip.
   (b) Parameterize the normalized 3D model into a geometry image I (the shape function f) of size $w \times h = 2^{n+1} \times 2^n$.
   (c) $A^{(n)} \leftarrow I$, $l \leftarrow$ desired decomposition levels l_{max}.
2. **Forward transform:** repeat until $l = 0$:
   (a) Perform a forward spherical wavelet transform on $A^{(l)}$. We get an approximation $A^{(l-1)}$, and detail coefficients $C^{(l-1)} = \{LH^{l-1}, HL^{l-1}, HH^{l-1}\}$ of size $2^{n-(l_{max}-l)} \times 2^{n-(l_{max}-l)-1}$.
   (b) $l \leftarrow l - 1$.
3. **Shape description:** the approximation $A^{(0)}$ and the coefficients $C^{(0)}, \dots, C^{(n-1)}$ are collected into a feature vector F.

The same process is also applied on-line to the query model. The image wavelet transform uses separable filters, so at each step it produces an approximation image A and three detail images HL, LH, and HH. In this paper, we experimented with the Haar wavelets. Other wavelet basis can be used and a further investigation is required to select efficient wavelet basis.

A special care should be taken when processing the boundaries of the geometry images. We use the spherical extension rules as proposed in [8].

In the final step, we build a shape descriptor by collecting a subset of the wavelet coefficients into a vector F. We use the first $d = 256$ coefficients. The dissimilarity between two 3D models O_i and O_j is the Euclidean distance between their feature vectors F_i and F_j.

Finally, we experimented with the flat octahedron-based parameterization, referred as **SWCd_RUNNR_1**, and the latitude-longitude parameterization, referred as **SWCd_RUNNR_2**. The implementations of the two methods differ only in the rules used for boundary extension [8]. Notice that, both the two descriptors require pose normalization.

4 Experimental results

We have implemented the algorithms described in this paper and evaluated their performance on the Princeton Shape Benchmark (PSB)[13]. The program requires the setting of three parameters: (1) the size of the geometry images $w \times h = 128 \times 64$, (2) number of decomposition levels $l_{max} = 5$, and (3) the descriptor length $d = 256$.

Table 1: Mean performance measures (DR: Dynamic Recall), 1: SWCd_RUNNR_1, 2: SWCd_RUNNR_1).

	Precision	1^{st}	2^{nd}	DR	NCG	NDCG
		-Tier	-Tier		@10	@10
1	0.232	24.1%	15.25%	0.32	0.39	0.45
2	0.222	23.12%	16.23%	0.30	0.36	0.42

We query the database with 30 polygon soup models and return for each query a ranked list of 3D models sorted according to their dissimilarity to the query. Figures 2b and 2c show some retrieval results using the two descriptors SWCd_RUNNR_1 and SWCd_RUNNR_2, respectively. The query models, provided in SHREC - 3D Shape Retrieval Context shown in Figure 2a, are not part of the PSB. The top five shapes of the ranked list of each run are displayed.

Table 1 summarizes the average performance of the two descriptors using different performance measures, computed over the 30 queries. The first result is that the flat-octahedron parameterization performs slightly better than the latitude-longitude parameterization on all the performance measures. Our lowest performance has been observed on Query17, Query21 and Query28 where the two algorithms fail to retrieve relevant shapes near the top of the ranked list. Our algorithms however perform very well on car and airplane classes.

Finally, the developed descriptors (SWCd_RUNNR_1 and SWCd_RUNNR_2) rank at the thirteenth (respectively fourteenth) position (over 17 runs) in the Shape Retrieval Context (SHREC-3D). This low performance is expectable since the spherical extent functions on which we apply the wavelet analysis ignore the interior details of the shape.

5 Conclusion and future work

We proposed in this paper the use of spherical wavelet analysis for the indexation and retrieval of 3D models. Experimental results on the Princeton Shape Benchmark using SHREC queries show that the new descriptors perform well on some shape classes such as planes, cars and chairs, but are not efficient on classes of shapes that contain many interior details, and on stick-like shapes.

In future we would like to experiment with other spherical shape functions to take into account the interior details of the shapes. We plan also to test different wavelet basis, and investigate what could be a best basis for different shape classes.

Acknowledgement. This work is partially supported by the Japan Society for the Promotion of Science (JSPS).

References

[1] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.

[2] T. A. Funkhouser, P. Min, M. M. Kazhdan, J. Chen, J. A. Halderman, D. P. Dobkin, and D. P. Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 22(1):83–105, 2003.

[3] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *In Proceedings of ACM SIGGRAPH*, pages 203–212. ACM Press, 2001.

[4] H. Hoppe and E. Praun. Shape compression using spherical geometry images. In *MINGLE 2003 Workshop.*, number 2, pages 27–46, 2003.

[5] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[6] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[7] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP '03: Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164. Eurographics Association, 2003.

[8] H. Laga, H. Takahashi, and M. Nakajima. Spherical wavelet descriptors for content-based 3d model retrieval. In *Shape Modeling International*, Japan, June 2006. to appear.

[9] M. Novotni and R. Klein. Shape retrieval using 3d zernike descriptors. *Computer Aided Design*, 36(11):1047–1062, 2004.

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *Shape Modeling International*, pages 154–166, Italy, May 2001.

[11] D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. In B. Radig and S. Florczyk, editors, *DAGM-Symposium*, volume 2191 of *Lecture Notes in Computer Science*, pages 392–397. Springer, 2001.

[12] P. Schroder and W. Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *In Proceedings of SIGGRAPH 1995*, pages 161–172. ACM Press, 1995.

[13] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *In Proceedings of the Shape Modeling International 2004 (SMI'04)*, pages 167–178. IEEE Computer Society, june 2004.

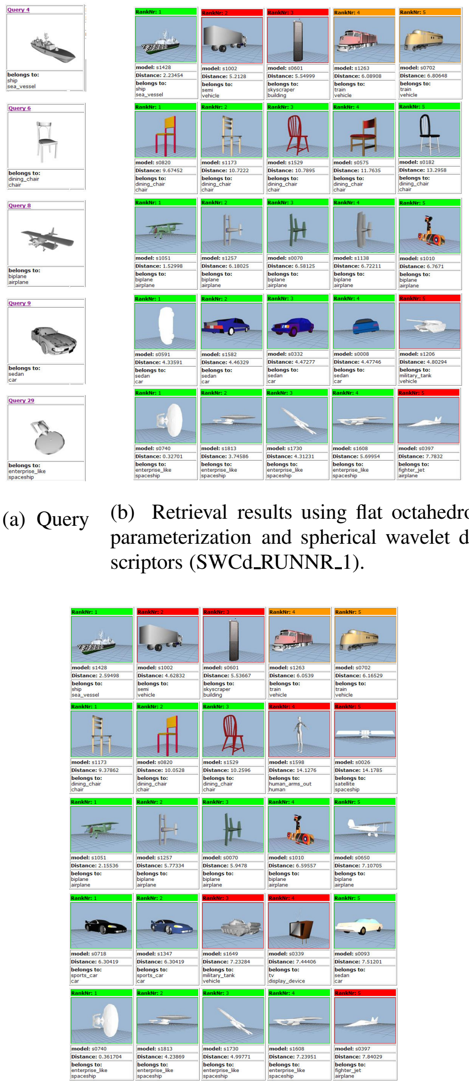
[14] H. Sundar, D. Silver, N. Gagvani, and S. J. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International*, pages 130–142, 2003.

[15] J. W. Tangelder and R. C. Veltkamp. A survey of content based 3d shape retrieval. In *Shape Modeling International 2004, Genova, Italy*, pages 145–156, June 2004.

[16] D. V. Vranic. An improvement of rotation invariant 3d-shape based on functions on concentric spheres. In *ICIP2003*, pages 757–760, 2003.

[17] 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*, pages 177–180, August 2002.

[18] D. V.Vranic. *3D Model Retrieval*. Phd dissertation, Universitat Leipzig, Institut Fur Informatik, 2003.



(a) Query

(b) Retrieval results using flat octahedron parameterization and spherical wavelet descriptors (SWCd_RUNNR_1).

(c) Retrieval results using latitude-longitude parameterization and spherical wavelet descriptors (SWCd_RUNNR_2).

Figure 2: Examples of retrieval results using the two shape descriptors.

Light Field Similarity for Model Retrieval

Ameesh Makadia and Kostas Daniilidis
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA, 19104, USA
{makadia, kostas}@cis.upenn.edu

Abstract

The ability to perform a Google-like search on a database of 3D models is becoming a growing necessity as the number of 3D models in circulation is rapidly increasing. One of the many existing methods dealing with this problem considers the matching problem as one of visual similarity. This idea of measuring the distance between two models as the distance between their corresponding silhouette images performs well on a number of benchmarks, but comes with a few inherent limitations, the biggest of which is that at the time of comparison all possible rotational alignments between two models need to be considered. In this paper we present an approach based on the silhouette representation of a model which deals with these issues and reduces the problem of similarity estimation to that of computing the Euclidean distance between small feature vectors.

1. Introduction and related work

Laser-scanned objects, CAD models, and even image-based reconstructions are just a few of the sources contributing to the rapidly growing number of publicly available 3D models. Along with these vast resources of 3D collections comes the need for a fast, large-scale model retrieval and matching system. Although the availability of 3D information has sparked a number of methods which take advantage of the complex geometric information for each model, one of the consistently best performing methods has been the retrieval method based only on visual similarity [3, 12]. Surprisingly it is shown model silhouettes provide enough information for recovery, eliminating the need to process often complex 3D structural or surface information. One explanation for this result is that methods based on shape or geometry often rely on the computation of descriptive, local features. This task is made difficult by the fact that local shape representation may vary widely among objects from the same class (one reason may be the level of complexity used when generating the original model).

Another popular class of methods considers global representations of the models ([14, 2, 5, 6]) and in particular cases the representations of the models analyzed using the spherical harmonic transform ([14, 7], among others). We will also use the spherical harmonic representation and its rotational invariants in a similar manner, as will be explained in later sections.

The method we present in this paper is motivated by the visual similarity method of [3]. The basic idea is based on approximating the light field [9] of a 3D model by capturing silhouettes from a fixed set of positions on the sphere. While their method proves promising, it has a few inherent limitations. The spherical positions of the silhouettes are restricted by the fact that comparisons can only be made for rotations which map the samples onto themselves. There is no natural way to perform approximate comparisons, which is a necessity considering very large databases may be queried, and there is no flexibility in the rotations that can be tested. Finally, for any pair of models, a brute force traversal through all possible rotational alignments must be made (although a hierarchical approach can help speed up the comparisons). We will present in this paper a method for model retrieval which addresses these three concerns. Our method takes advantage of the fact that the light field representation of a 3D model can be considered a function on the tangent bundle of the sphere (where each tangent plane represents a different model silhouette). The comparison of two light fields can then be written as the correlation of two tangent bundles over the space of 3D rotations. We show how such a comparison can be written as a correlation integral, which can be computed as a multiplication in the Fourier domain. We also present an alternate method which uses the same representation, but instead utilizes rotational invariants in the spherical Fourier domain to represent the light field with one small feature vector. In this case the similarity between two models is just the Euclidean distance between two such vectors.

2. Light field representation

The model retrieval method in [3] considers silhouettes taken at 20 (in practice only 10 are needed) different locations on the sphere, and there are only 60 rotations which map these points onto themselves. In order to create a denser sampling of silhouette positions and to consider more rotations, the only solution is to recreate the configuration of 10 vertices at a different reference orientation. Repeating this configuration 9 times, a total of 100 silhouettes are generated, and the number of rotations which need to be traversed before a distance measure is obtained rises to 5,460. We would like to develop a method that is more flexible to the number of silhouettes that can be used, a method which has natural approximation capabilities for speed considerations, and one that does not require the brute-force traversal of all possible rotational alignments.

We begin by describing a modified representation of the light field for a 3D model. Instead of capturing silhouettes from pre-determined positions, we can specify the locations given the desired resolution. Given a spherical bandwidth B such that $(2B)^2$ samples will be uniformly placed on the sphere, in spherical coordinates (see Figure 1 to see what this sampling looks like on the sphere). Only the value of B needs to increase until the desired spacing is achieved. A silhouette will then be generated from each of these samples, and this collection of silhouettes will be the model's light field representation. The silhouette at sphere point $p(\theta, \phi)$ is a binary function obtained by orthographically projecting the 3D model onto the tangent plane at p. The orientation of the tangent plane is given by the rotation $R = R_z(\phi)R_y(\theta)$ (i.e. the tangent plane at the north pole maps onto the tangent plane at p via the rotation R).

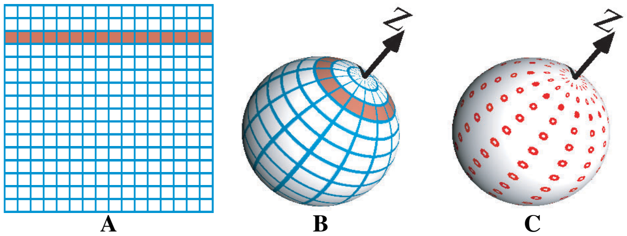


Figure 1: On the left (**A**) is a spherical grid with 256 samples. The sphere $\mathbb{S}^2$ is sampled uniformly in spherical coordinates, thus creating a square grid. (**B**) depicts the corresponding regions mapped onto the sphere. The highlighted samples correspond to the highlighted row in (**A**). The sample centers (origins of the tangent planes) are displayed in (**C**).

3. Model similarity as correlation

We now present the first method to measure similarity of two light fields. For the moment, consider the continuous light field function $L(p, v)$ which gives the binary value of the point v on the silhouette taken from spherical location p. If we process the silhouettes to generate smaller features which may encode some translational or rotational invariants, then our light field representation can be stored as a vector-valued function on the sphere, given as $L(p, x)$. In other words, $L([0\ \ 0\ \ 1]^T, x)$ is some feature vector computed for the silhouette obtained from the north pole. For simplicity we will use the centroid-distance functions and Zernike moments used in [15, 3]. If we define the similarity of two feature vectors to be their correlation coefficient, then we can claim the similarity of two 3D models as the maximum correlation coefficient of their light field representations over all possible rotational alignments. In other words, model similarity is given by the maximum of the following rotational function:

$$G_c(R) = \int_x \int_p L_1(p, x) L_2(R^T p, x) dp dx \qquad (1)$$

The key here is in recognizing the underlying spherical integration as a correlation of two spherical signals. It has been shown that the correlation integral $G(R) = \int L_1(p)L_2(R^T p)dp$ can be estimated efficiently as a multiplication in the spherical Fourier domain. We refer readers to [8, 11] for the details, and for other applications of the spherical correlation alignment. We will write $\hat{f}^l_m$ for the spherical Fourier coefficients of degree l and order m, $\hat{f}^l$ for the vector of $(2l + 1)$ coefficients $\hat{f}^l = [\hat{f}^l_l \hat{f}^l_{l-1} \cdots \hat{f}^l_{-l+1} \hat{f}^l_{-l}]^T$, and $\hat{G}^l_{mp}$ for the coefficients of the Fourier transform defined on the rotation group $SO(3)$.

The spherical correlation theorem states that the Fourier transform of the spherical correlation function G can be obtained as

$$\hat{G}^l_{mp} = \hat{L_1}^l_m \hat{L_2}^l_p$$

and so the samples of the correlation function are recovered with $G(R) = ISOFT(\hat{G})$ where $ISOFT$ is the inverse SO(3) Fourier transform. Our light-field correlation can now be written as

$$G_c(R) = \int_x ISOFT(\hat{G})(x) dx \qquad (2)$$

If we let B represent the bandwidth of the spherical functions (meaning only coefficients up to degree $B - 1$ are computed), then the inverse SO(3) Fourier transform will leave us with with $2B$ samples in each of the three Euler angles, giving us an accuracy up to $\pm \left(\frac{180}{2B}\right)^\circ$ in α and γ and $\pm \left(\frac{90}{2B}\right)^\circ$ in β. Here α, β, γ form the traditional ZYZ Euler angle parameterization of $SO(3)$. To estimate $G_c(R)$ for a finer sampling using interpolation, we only need to compute equation (2) assuming the coefficients for degrees above the bandwidth are equal to zero. To compute the SFT of a discrete spherical function we can use a fast $O(B^2 log^2 B)$ algorithm developed by Driscoll and Healy [4], A similar separation-of-variables approach exists for a fast $SO(3)$ Fourier transform in $O(B^3 log^2 B)$ [8].

3.1. Computational considerations

We can take advantage of having structural information to reduce the computational and storage burden of our light-field features. Knowing that the silhouette generated at any point p on the sphere is just a projection of the model onto the silhouette plane, it is clear that the silhouette generated from the antipodal point $-p$ will just be a reverse image. Since the contour-distance features and Zernike moment features of a silhouette are invariant to rotations, we only need to generate silhouettes from one hemisphere. This means that the underlying spherical function of $L(p, x)$ is has the even property that $L(p, x) = L(-p, x)$. For such functions, it is known that all spherical Fourier coefficients of an odd degree are zero. Also, since the spherical function is real-valued, the coefficient vectors $\hat{f}^l$ exhibit the hermitian property that opposite orders are related by conjugation. These two facts mean we only need to compute $\hat{f}^l_m$ for l even and $m \geq 0$. Along with the fast correlation theorem, these speedups allow us to find the global maximum of the rotation function $G_c(R)$ faster than doing a brute-force evaluation for each sample of the rotation space.

4. Rotational invariants

For some applications such as database model retrieval, where the goal is to create a ranked list of similar models, the correlation method may not be fast enough to do pairwise comparisons against a database of thousands of models. In this setting, we can use invariant properties of the spherical harmonic coefficients (see [7, 10] for other uses of these invariants) to encode a feature vector which does not depend on the orientation of the original polygonal model. Once it is understood how the individual coefficients change under rotation, it can be easily verified that the vector norm of the coefficients of a particular degree remains unchanged under rotations. Intuitively, we would expect a rotation to manifest itself as a modulation of the Fourier coefficients as is the case in traditional Fourier analysis. This is, in fact, the observed effect. As spherical functions are rotated by elements of the rotation group $SO(3)$, the Fourier coefficients are “modulated” by the irreducible representations of $SO(3)$:

$$f(\eta) \mapsto f(R^T\eta) \Longleftrightarrow \hat{f}^l \mapsto U^l(R)^T\hat{f}^l \qquad (3)$$

The U^l matrix representations of $SO(3)$ are the spectral analogue to 3D rotations. The unitarity of these representations ensures that the rotation of a function does not alter the distribution of spectral energy among degrees:

$$||U^l(R)\hat{f}^l|| = ||\hat{f}^l||, \forall R \in SO(3) \qquad (4)$$

By considering only the magnitudes of the coefficient vectors, the light-field feature size is further reduced. The total size is equal to $\lfloor \frac{B}{2} \rfloor N$, where B is the spherical bandwidth and N is the size of the individual silhouette features. For example, consider a model for which we render a very large number of silhouettes (a bandwidth of $B = 17$ means we must render 578 silhouettes in one hemisphere). Assuming we keep 35 Zernike coefficients and 10 contour distance coefficients for each silhouette (as in [3]), we can represent the entire light-field with one feature vector with just $(35 + 10) * 8 = 360$ elements. The distance between two models is defined as the Euclidean distance between their respective feature vectors, and this computation will be very fast even for an entire database of models.

5. Discussion

Two versions of the correlation method and two versions of the rotational invariants comparison method were entered in SHREC, the 3D Shape Retrieval Contest [1, 13]. The four entries were entered under the name ”Makadia.” The first run compared models using the correlation method, but only Zernike features were used. The second run was again the correlation method, but both Zernike and contour-distance features were used. The third run compared models with the faster invariant vector comparison, using only Zernike features, while the fourth run performed the vector comparisons using Zernike and contour-distance features. Of the 17 main categories of analysis, the correlation methods finished first in 11. The faster vector-invariant comparisons, however, finished in the middle of the pack in all categories. The obvious tradeoff is in accuracy versus comparison times. The time to compare two models using the vector comparison is less than 0.0001 seconds, while the time to compare two models using the correlation method is just less than 0.1 seconds (for a bandwidth of $B = 17$). While the current implementation is a naive, preliminary effort, the difference in times is indicative of the differences in complexity between correlations and vector comparisons. Figure 2 shows that the accuracy of the correlation method may be achieved without performing pairwise correlations against an entire database. The faster invariant vector comparisons can be used to generate a much smaller set of possible matching objects, and the correlations can be used to provide an accurate ranking within this pruned set.

5.1. Future work

The algorithms proposed in this document can be extended in many ways. One way to speed up the model comparisons would be to perform the correlations at smaller resolutions. This is a natural approximation obtained by varying the number of Fourier coefficients used in the correlation. By reducing the number of coefficients, a faster correlation estimate will be obtained for a sparser sampling of the rotation space. This property leads to a hierarchical model re-

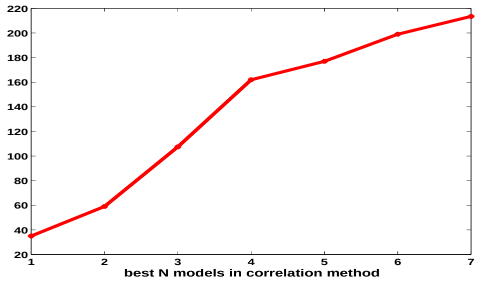


Figure 2: This plot shows how many models in the ranked list (obtained with vector comparisons) you need to traverse before finding 50% of the best N matches in the ranked list obtained with correlations. The plot shows the median over all queries. For example, 50% of the best 100 matches from the correlation method will appear in the first 213 matches from the ranked list obtained with fast vector comparisons.

covery approach, where in each successive pass the model database can be pruned with correlation comparisons using increasing numbers of coefficients. On another note, it may be beneficial to retain more information beyond a binary silhouette image. One could capture a depth map, where each pixel marks the distance from the model to the silhouette plane, to help disambiguate between very different objects which generate similar silhouettes.

6. Conclusion

In this paper we presented two new approaches for comparing 3D models. The first method considers the best possible correlation alignment between the models' light field representations. The benefit of this approach is in the fast correlation estimation using the Spherical Fourier Transform, and the flexibility and approximation allowed by varying the number of coefficients used. The second method utilizes the rotational invariants of the spherical transform to encode an entire model light field with just one small feature vector. This allows us to compute model distance with fast Euclidean distance measurements.

References

[1] AIM@SHAPE. `http://give-lab.cs.uu.nl/shrec/shrec2006/`.

[2] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In R. Güting, D. Papadias, and F. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99*, volume 1651, pages 207–228, Hong Kong, China, 1999. Springer.

[3] Y.-T. Shen D.-Y. Chen, X.-P. Tian and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Eurographics*, 2003.

[4] J.R. Driscoll and D.M. Healy. Computing fourier transforms and convolutions on the 2-sphere. *Advances in Applied Mathematics*, 15:202–250, 1994.

[5] B. K. P. Horn. Extended gaussian images. *IEEE*, 72:1671–1686, 1984.

[6] M. Kazhdan. *Shape Representations and Algorithms for 3D Model Retrieval*. PhD thesis, Princeton University, 2004.

[7] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, June 2003.

[8] P. J. Kostelec and D. N. Rockmore. FFTs on the rotation group. In *Working Paper Series, Santa Fe Institute*, 2003.

[9] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. of ACM SIGGRAPH*, pages 31–42, 1996.

[10] A. Makadia and K. Daniilidis. Direct 3D-rotation estimation from spherical images via a generalized shift theorem. In *IEEE Conf. Computer Vision and Pattern Recognition*, Wisconsin, June 16-22, 2003.

[11] A. Makadia and K. Daniilidis. Rotation recovery from spherical images without correspondences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28, 2006.

[12] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International*, Genova, Italy, June 2004.

[13] R. Typke, R. C. Veltkamp, and F. Wiering. Evaluating retrieval techniques based on partially ordered ground truth lists. In *Proceedings International Conference on Multimedia & Expo*, 2006.

[14] D. V. Vranic and D. Saupe. 3d model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 392–397, London, UK, 2001. Springer-Verlag.

[15] D. S. Zhang and G. Lu. An integrated approach to shape based image retrieval. In *Proc. of 5th Asian Conference on Computer Vision (ACCV)*, pages 652–657, Melbourne, 2002.

A Concrete Radialized Spherical Projection Descriptor for 3D Shape Retrieval

Panagiotis Papadakis[1, 2], Ioannis Pratikakis[1], Stavros Perantonis[1] and Theoharis Theoharis[2]

[1]Computational Intelligence Laboratory
Institute of Informatics and Telecommunications
National Center for Scientific Research “Demokritos”
GR-153 10 Agia Paraskevi, Athens, Greece

[2]Computer Graphics group
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
GR-15784 Panepistimiopolis, Ilissia, Athens, Greece

Abstract. In this paper, we present a 3D shape retrieval methodology that relies upon a shape representation based on the theory of spherical harmonics. To alleviate the problem of rotation invariance, we invoke in a parallel fashion, the Continuous Principal Component Analysis along with a novel approach which applies PCA on the mesh normals. After translating to the model’s center of mass, the 3D model is decomposed into a set of spherical functions with increasing radius, which represent the intersections of the model’s surface with rays emanating from the origin and parts closer to the origin than the furthest intersection point at each ray. Properties of spherical harmonics are used to achieve scaling invariance and axial flipping invariance.

1. Analysis of shape descriptor extraction steps

The distinct stages followed during the proposed 3D shape descriptor extraction are shown in Fig. 1.

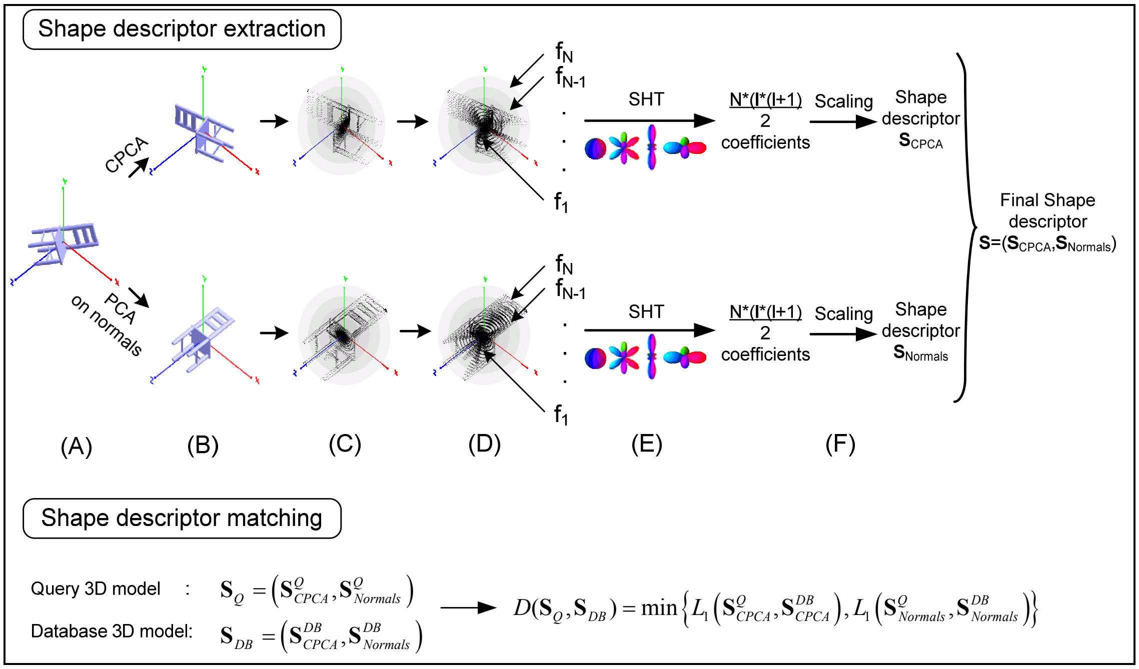

Shape descriptor extraction
CPCA
PCA on normals
f_N
f_{N-1}
f_1
SHT
$\frac{N^*(l^*(l+1)}{2}$ coefficients
Scaling
Shape descriptor S_{CPCA}
Shape descriptor $S_{Normals}$
Final Shape descriptor $S=(S_{CPCA},S_{Normals})$
(A)
(B)
(C)
(D)
(E)
(F)
Shape descriptor matching
Query 3D model : $\mathbf{S}_Q=\left(\mathbf{S}^Q_{CPCA},\mathbf{S}^Q_{Normals}\right)$
Database 3D model: $\mathbf{S}_{DB}=\left(\mathbf{S}^{DB}_{CPCA},\mathbf{S}^{DB}_{Normals}\right)$
$D(\mathbf{S}_Q,\mathbf{S}_{DB})=\min\left\{L_1\left(\mathbf{S}^Q_{CPCA},\mathbf{S}^{DB}_{CPCA}\right),L_1\left(\mathbf{S}^Q_{Normals},\mathbf{S}^{DB}_{Normals}\right)\right\}$

Figure 1. The consecutive stages (A)-(F) of the proposed 3D shape matching scheme for retrieval purposes.

In this section, we detail the stages depicted in Figure 1.
- **Stage A**: The model is translated to its center of mass **m** which is computed by using the Continuous Principal Component Analysis (CPCA) [1]. Thus **m** equals to:

$$\mathbf{m}=\frac{1}{S}\sum_{i=1}^{N}\iint_{\mathbf{v}\in T_i}\mathbf{v}ds \Rightarrow \mathbf{m}=\frac{1}{S}\sum_{i=1}^{N}S_i\cdot\left(\mathbf{A}_i+\mathbf{B}_i+\mathbf{C}_i\right)/3 \tag{1}$$

where S is the total surface area of the model, N is the number of triangles of the model, **v** is a vertex of the model, S_i is the surface area of triangle T_i and A_i, B_i and C_i are the vertices of T_i.

- <u>Stage B</u>: We apply two alternative alignment procedures in a parallel fashion. In the first procedure, we apply the CPCA, where the covariance matrix of the model's surface is given in Eq. 2.

$$\begin{gathered}\mathbf{C}=\frac{1}{S}\iint_{\mathbf{v}\in M}(\mathbf{v}-\mathbf{m})\cdot(\mathbf{v}-\mathbf{m})^T ds \Rightarrow \\ \mathbf{C}=\frac{1}{12\cdot S}\sum_{i=1}^{N}S_i\cdot\left[f(\mathbf{A}_i)+f(\mathbf{B}_i)+f(\mathbf{C}_i)+9\cdot f\left(\left(\mathbf{A}_i+\mathbf{B}_i+\mathbf{C}_i\right)/3\right)\right]\end{gathered} \tag{2}$$

where **C** denotes the model's covariance matrix, f(**v**) = $(\mathbf{v}-\mathbf{m})\cdot(\mathbf{v}-\mathbf{m})^T$ and M is the set of vertices of the polygonal model. The produced aligned model is shown in Figure 1.(B).
In the second alignment procedure, we use the model's mesh normals as input for the computation of the covariance matrix in PCA[2]. Specifically, first we compute the mean normal $\mathbf{m}_{NI}$:

$$\mathbf{m}_{NI}=\frac{1}{2S}\sum_{i=1}^{N}\left(S_i\cdot\mathbf{n}_i+S_i\cdot(-\mathbf{n}_i)\right)=\mathbf{0} \tag{3}$$

where $\mathbf{n}_i$ is the normal vector of the triangle i. The mean normal is always the zero vector because we treat each triangle twice, first with its normal direction and second with its opposite direction. This is done to avoid possible incorrect ordering of the triangle vertices which affects its direction sign. Thereafter, the covariance matrix is computed as follows:

$$\begin{aligned}\mathbf{C}&=\frac{1}{2S}\sum_{i=1}^{N}S_i\left(\left(\mathbf{n}_i-\mathbf{m}_{NI}\right)\cdot\left(\mathbf{n}_i-\mathbf{m}_{NI}\right)^T+S_i\left(-\mathbf{n}_i-\mathbf{m}_{NI}\right)\cdot\left(-\mathbf{n}_i-\mathbf{m}_{NI}\right)^T\right)=\\&=\frac{1}{2S}\sum_{i=1}^{N}S_i\cdot\left((\mathbf{n}_i\cdot\mathbf{n}_i^T)+(-\mathbf{n}_i)\cdot(-\mathbf{n}_i)^T\right)=\frac{1}{S}\sum_{i=1}^{N}S_i\cdot(\mathbf{n}_i\cdot\mathbf{n}_i^T)\end{aligned} \tag{4}$$

The produced aligned model is shown in Figure 1.(B). After the completion of Stage B we have two alignment variants of the initial model. This implies that the following stages will finally lead to two 3D shape signatures.

- <u>Stage C</u>: In this stage, we compute the intersections of the triangles of the aligned model with rays emanating from the origin and with directions (θ_j, φ_k):

$$\theta_j=\frac{(2j+1)\pi}{4B},\ \varphi_k=\frac{2\pi k}{2B} \tag{5}$$

where B is the sampling bandwidth and j, k = 0,1,…,2B-1. Let S^2 denote the (θ_j, φ_k) sample points on unit sphere and center at the origin and $Inters_i(\theta_j, \varphi_k)$ be the distance from the origin of the i-th intersection of the model's surface with ray (θ_j, φ_k) where i=1,2,… . If there is not any intersection at ray (θ_j,φ_k) then Inters(θ_j, φ_k) = -1. The 3D model is represented as a set of N spherical functions $f_r : S^2 \rightarrow [l_r, l_{r+1})\bigcup\{0\}$ with radius r = 1,2,…,N where f_r is denoted as:

$$f_r(\theta_j,\varphi_k)=\max\left\{\{0\}\bigcup\left\{Inters_i(\theta_j,\varphi_k)\ \mid\ l_r\le Inters_i(\theta_j,\varphi_k)<l_{r+1}\right\}\right\} \tag{6}$$

where $[l_r, l_{r+1})$ are the spherical function's boundaries given by:

$$l_r = \begin{cases} 0 & , \quad r = 1 \\ \dfrac{(r-0.5)\cdot M \cdot d_{avg}}{N} & , \quad 1 < r \leq N+1 \end{cases} \tag{7}$$

where d_{avg} is the average distance of the surface from the center of mass and M is the radius of the largest sphere. The set of N spherical functions representing the 3D model is shown in Figure 1 (C).

- **<u>Stage D</u>**: Let $mdist_{j,k} = \max\{Inters_i(\theta_j, \varphi_k)\}$ be assigned to the spherical function point $f_D(\theta_j, \varphi_\kappa)$. Then all $f_r(\theta_j, \varphi_\kappa)$ with r < D are assigned values as if they belonged to the model by setting:

$$f_r(\theta_j, \phi_k) = \frac{r \cdot M \cdot d_{avg}}{N}, \quad \forall r < D. \tag{8}$$

Practically, if the model is viewed from the (θ_j,φ_k) direction then the part of the model along ray (θ_j,φ_k) closer to the origin than $mdist_{j,k}$ is occluded by the intersection point corresponding to $mdist_{j,k}$. According to the human perception we may assume that the occluded part actually belongs to the model. Therefore, taking into account Eq. 6, 8, the functions f_r are denoted as:

$$f_r(\theta_j, \varphi_k) = \begin{cases} mdist_{j,k} & , \text{ if } mdist_{j,k} \neq -1 \text{ and } l_r \leq mdist_{j,k} < l_{r+1} \\ \dfrac{r \cdot M \cdot d_{avg}}{N}, & \text{ if } mdist_{j,k} \neq -1 \text{ and } r < D \\ 0 & , \text{ otherwise} \end{cases} \tag{9}$$

These spherical functions lead to a model representation as shown in Figure 1 (D).

- **<u>Stage E</u>**: A spherical function f(θ, φ) can be expressed as a sum of spherical harmonic functions [3] as:

$$f(\theta, \phi) = \sum_{l=0}^{B-1} \sum_{m=-l}^{l} \hat{f}(l,m) Y_l^m(\theta, \phi) \tag{10}$$

where B is the sampling bandwidth, $Y_l^m(\theta, \phi)$ is the spherical harmonic of degree l and order m and $\hat{f}(l,m)$ is the equivalent spherical harmonic coefficient. The $\hat{f}(l,m)$ coefficients are computed as in the following:

$$\hat{f}(l,m) = \frac{\sqrt{2\pi}}{2\text{B}} \sum_{j=0}^{2B-1} \sum_{k=0}^{2B-1} a_j^{(B)} \cdot f(\theta_j, \phi_k) \cdot e^{-im\phi_k} P_l^m(\cos\theta_j), \tag{11}$$

where $a_k^{(B)}$ are appropriate weights. The spherical harmonics transform (SHT) [3] is applied to each of the N spherical functions, producing B^2 coefficients. Since the input of the SHT is a real-valued function, the symmetry $\hat{f}(l,m) = \overline{(-1)^m \hat{f}(l,-m)}$ exists between the coefficients. This enables us to store only the positive order coefficients, if we use them accordingly at the matching stage (Section 2). The number of these coefficients is:

$$N \cdot (L_{bands} \cdot (L_{bands} + 1))/2, \; L_{bands} \leq B. \tag{12}$$

Stage F: The obtained coefficients are scaled to have unit L_1 norm. By definition, spherical harmonic coefficients are exactly proportional to the model's scale. Thus scaling invariance is obtained by scaling the coefficients which is done in constant complexity, because we sample all models with the same bandwidth. Axial flipping invariance is achieved by using the following properties:

i. If the model is reflected with respect to the x-axis (yz plane): $\hat{f}(l,m) \Rightarrow \overline{\hat{f}(l,m)}$.

ii. If the model is reflected with respect to the y-axis (xz plane): $\hat{f}(l,m) \Rightarrow (-1)^{l+m}\hat{f}(l,m)$.

iii. If the model is reflected with respect to the z-axis (xy plane): $\hat{f}(l,m) \Rightarrow (-1)^{m}\overline{\hat{f}(l,m)}$.

Only the sign of the real and imaginary component of a coefficient may change, so we store the absolute values of the imaginary components for any *l*, *m* and the absolute values of the real components for the case where *m* or *l* is odd. If *l, m* are both even, then the sign of the real component does not change and in this case we store the actual values.

2. Similarity measure

Each spherical harmonic coefficient is a complex number. To compare two shape descriptors, the L_1 distance is computed. Let $\hat{f}(l,m) = \mathrm{Re} + j\,\mathrm{Im}$, $\widehat{f'}(l,m) = \mathrm{Re}' + j\,\mathrm{Im}'$ be two coefficients corresponding to two different shape descriptors. The L_1 distance between them is:

$$\left|\hat{f}(l,m) - \widehat{f'}(l,m)\right| = \left|\mathrm{Re} - \mathrm{Re}' + j(\mathrm{Im} - \mathrm{Im}')\right| = \sqrt{(\mathrm{Re} - \mathrm{Re}')^2 + (\mathrm{Im} - \mathrm{Im}')^2} \qquad (13)$$

To compute the overall difference between two shape descriptors, we compute the above for all degrees $l \le L_{bands}$, but only for the positive order coefficients. Using the symmetry between the coefficients of positive and negative order, we take:

$$\begin{aligned} &\left|\hat{f}(l,m) - \widehat{f'}(l,m)\right| = \left|(-1)^m\overline{\hat{f}(l,-m)} - (-1)^m\overline{\widehat{f'}(l,-m)}\right| = \\ &\left|\overline{\hat{f}(l,-m)} - \overline{\widehat{f'}(l,-m)}\right| = \left|\overline{\hat{f}(l,-m) - \widehat{f'}(l,-m)}\right| = \left|\hat{f}(l,-m) - \widehat{f'}(l,-m)\right|. \end{aligned} \qquad (14)$$

This enables us to only store the positive order coefficients in the shape descriptor, but we multiply each $\left|\hat{f}(l,m) - \widehat{f'}(l,m)\right|$, $0 < m \le l$ by two, to take into account the negative order coefficients.

As explained in the previous section the shape descriptor consists of two sums of coefficients. The first corresponds to the model aligned with CPCA and the second to the model aligned by applying the PCA to the mesh normals. When comparing two shape descriptors, the comparison is done between the corresponding sums. Specifically, the CPCA aligned version of the first model is compared with the CPCA aligned version of the second, and the same is done for the PCA using mesh normals aligned version. The comparison which gives the minimum distance is taken to be the similarity measure between the two models.

References

[1]. D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics," in *Proc. IEEE Workshop Multimedia Signal Processing (MMSP2001)*, J.-L. Dugelay and K. Rose, Eds., Cannes, France, October 2001, pp. 293-298.

[2] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Academic Press, 1999.

[3]. D. Healy Jr., D. Rockmore, P. Kostelec and S. Moore, "FFTs for the 2-Sphere - Improvements and Variations". J*ournal of Fourier Analysis and Applications,* **9,** 4: 341-385, July 2003.

Partial Matching of 3D Shapes with Priority-Driven Search

Philip Shilane Thomas Funkhouser
Princeton University

1 Method

This project represents the results of our ongoing work to efficiently use local shape features and is more thoroughly described in the article “Partial Matching of 3D Shapes with Priority-Driven Search” [Funkhouser and Shilane 2006] with the following abstract.

“Priority-driven search is an algorithm for retrieving similar shapes from a large database of 3D objects. Given a query object and a database of target objects, all represented by sets of local 3D shape features, the algorithm produces a ranked list of the *c* best target objects sorted by how well any subset of *k* features on the query match features on the target object. To achieve this goal, the system maintains a priority queue of potential sets of feature correspondences (partial matches) sorted by a cost function accounting for both feature dissimilarity and the geometric deformation. Only partial matches that can possibly lead to the best full match are popped off the queue, and thus the system is able to find a provably optimal match while investigating only a small subset of potential matches. New methods based on feature distinction, feature correspondences at multiple scales, and feature difference ranking further improve search time and retrieval performance. In experiments with the Princeton Shape Benchmark, the algorithm provides significantly better classification rates than previously tested shape matching methods while returning the best matches in a few seconds per query.“

Please see the paper for further details.

2 Submission

We submitted three retrieval runs to highlight various aspects of our technique. The first is the **Global Harmonic Shape Descriptor** (GHSD). This is the Harmonic Shape Descriptor [Funkhouser et al. 2003, Kazhdan et al. 2003] positioned at the center of mass of each shape and scaled to describe the entire shape. We included the GHSD in this contest since it is often used for comparison in the literature. Our two main techniques use local versions of the Harmonic Shape Descriptor in a multi-point matching system.

- **Priority-Driven Search with Ranks and Multi-Scale** Local features are used for matching with a cost function that incorporates differences between the ranks of corresponding descriptors, positions, and normals. The best combination of features was found for each of four descriptor scales independently and their costs were combined.
- **Priority-Driven Search with Ranks, Multi-Scale, and Distinction** The previous technique is augmented by selecting a small set of distinctive features for each target in the database. Then, correspondences from the query are only considered against the distinctive subset of features among the targets. In order to calculate distinction, we needed a classification for the target shapes. We used the GHSD to map from the target shapes to their closest matches in a classified database. Then, distinction scores for each feature were calculated based on how well they matched to shapes of the same class as described in Shilane et al. [2006]. While mapping the unclassified shapes to a classified

database is perhaps beyond the scope of this contest, it shows how well multi-point matching can work when focused on distinctive features.

References

[Funkhouser and Shilane 2006] FUNKHOUSER, T., AND SHILANE, P. 2006. Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing*.

[Funkhouser et al. 2003] FUNKHOUSER, T., MIN, P., KAZHDAN, M., CHEN, J., HALDERMAN, A., DOBKIN, D., AND JACOBS, D. 2003. A search engine for 3D models. *Transactions on Graphics 22*, 1, 83–105.

[Kazhdan et al. 2003] KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, 167–175.

[Shilane and Funkhouser 2006] SHILANE, P., AND FUNKHOUSER, T. 2006. Selecting distinctive 3D shape descriptors for similarity retrieval. In *Shape Modeling International*.

The Canonical 3D Hough Transform Descriptor

Titus Zaharia and Françoise Prêteux

Groupe des Ecoles des Télécommunications
Institut National des Télécommunications
Unité de Projets ARTEMIS
9, rue Charles Fourier 91011 Evry Cedex, France
Email: titus.zaharia@int-evry.fr, francoise.preteux@int-evry.fr

Abstract

This paper tackles the issue of 3D mesh indexation by using shape descriptors (SDs) under constraints of geometric and topological invariance. A 3D shape descriptor, so-called the *Canonical 3D Hough Transform Descriptor* (C3DHTD) is here described. Intrinsically topologically stable, the C3DHTD is not invariant to geometric transformations. Nevertheless, we show mathematically how the C3DHTD can be optimally associated (in terms of compactness of representation and computational complexity) with a spatial alignment procedure which leads to a geometric invariant behavior.

1. Introduction

The shape of an object is conceptually independent of its spatial position and size. A shape descriptor (SD) should therefore satisfy invariance properties with respect to geometric transforms such as isometries and isotropic scaling. In addition, since multiple topological representations can be associated with the same 3D mesh object, a SD should exhibit a stable behavior with respect to the mesh topology.

This paper describes the *Canonical 3D Hough Transform Descriptor* (C3DHTD), introduced in [1], [2], [3]. Intrinsically topological invariant, we show how the C3DHTD can be optimally associated with a spatial alignment procedure ensuring a geometric invariant behavior, while satisfying the storage and computational complexity requirements specific to similarity-based retrieval applications.

The paper is organized as follows. Section 2 recalls the principle of the 3D Hough transform (3D HT), and points out its limitations when applied to 3D mesh similarity retrieval. Section 3 introduces the C3DHTD and shows how it overcomes such limitations as soon as a canonically invariant unit sphere partition is considered. Concluding remarks are presented in Section 4.

2. The 3D Hough Transform

Extensively used for 2D/3D primitive detection and recognition, the 3D HT provides a generic and highly detailed representation that makes it attractive for 3D content description purposes. However, in the case of shape-based similarity retrieval applications, it is of crucial importance to consider the specific requirements of geometric invariance, storage and computational complexity, and support of appropriate similarity measures.

Is the 3D HT well-adapted to such strong requirements?

Before answering this question, let us first briefly recall the 3D HT definition.

2.1. Definition of the 3D HT

The 3D HT is based upon the principle of accumulating points in $\mathbb{R}^3$ within a set of planes.

Let $E \subset \mathbb{R}^3$ a finite set of points, with coordinate specified within a Cartesian coordinate system (O, x, y, z). Let us recall that a plane $\Pi \subset \mathbb{R}^3$ is uniquely defined by a triplet (s, θ, φ), where $s \geq 0$ denotes the distance from the origin of the coordinate system to plane Π, and $\theta \in [0, 2\pi)$ and $\varphi \in [-\pi/2, \pi/2)$ resp. denote the two angles (azimuth and elevation) associated with the spherical representation of the plane's unit length normal vector $\boldsymbol{n}$.

By uniformly sampling each axis of the parameter space (s, θ, φ), we respectively obtain N_s, N_θ et N_φ prototype elements, regrouped within the following sets:

- $\Xi = \{s_i = i\Delta_s\}_{i=0}^{N_s-1}$, where $\Delta_s = S_{\max} / N_s$, and S_{max} stands for a sufficient large value defining the mesh size,
- $\Theta = \{\theta_j = (j+0.5)\,\Delta_\theta\}_{j=0}^{N_\theta-1}$, where $\Delta_\theta = 2\pi / N_\theta$, and
- $\Psi = \{\varphi_k = (k - N_\varphi / 2 + 0.5)\,\Delta_\varphi\}_{k=0}^{N_\varphi-1}$ },where $\Delta_\varphi = \pi / N_\varphi$.

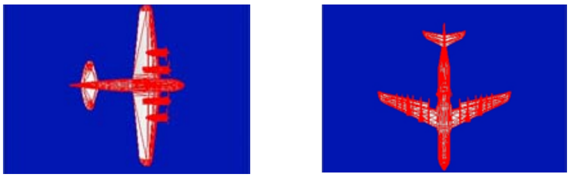
The 3D HT is then defined as an application $h : (\Xi \times \Theta \times \Psi) \rightarrow$ IR, each point $p \in E$ giving an additive contribution, w_{jk}^p, to each element $h(\hat{s}_{jk}^p, \theta_i, \varphi_k)$, where $\hat{s}_{jk}^p$ is the element in Ξ closest to the distance s_{jk}^p to the system's origin of the plane passing through *p* and with orientation (θ_j, φ_k).

In the case of polygonal models, *E* stands for the set of gravity centers of the mesh faces. The choice strategy of the weights w_{jk}^p, is extensively discussed in Section 3.2.

2.2. On the geometric invariance of the 3D HT

The previous considerations are developed under the assumption of a given Cartesian coordinate system. In order to ensure an invariant behavior with respect to geometric transforms such as isometries and isotropic scaling, an object-depended coordinate system should be first defined. The construction of such a coordinate system is based upon principal component analysis (PCA), for determining the object's principal axes. The object's gravity center is selected as the system's origin.

For achieving the spatial alignment of the 3D objects, most of the methods reported in the literature perform a (*x*, *y*, *z*) labeling of the principal axis in ascending or descending order of the eigenvalues $\lambda_{1,}$, λ_2 and λ_3. This methodology is unable to take into account the diversity of meshes encountered in practice. Figure 1 shows an example of a miss-alignment of two airplanes due to such a crisp labeling of the principal axes. In addition, performing just a PCA cannot provide the axis orientation, which results in an ambiguity in the definition of the coordinate system.



1.a. 1.b.

Figure 1. Miss-alignment resulting from the PCA approach. The major principal axis corresponds to the vertical direction.

To ensure a complete and non ambiguous representation during the matching stage, the set of 3D HTs corresponding to all the possible PCA-based coordinate systems (PCA CS) has to be available. Considering the (*x*, *y*, *z*) labeling of the three principal axes, and two possible orientations for each axis, results in 48 possible PCA CSs. Consequently, a complete descriptor needs to integrate the corresponding 48 HTs. However, some nice properties on the HT coefficients (circular shift, mirror reflections) can be exploited when changing the PCA CS. Mathematically, we have established [2] that 3 HTs, so-called *generating configurations* (GCs), are sufficient for deriving a complete representation. The three GCs are defined as the HTs corresponding to three distinct PCA CSs, obtained by letting each axis of inertia successively become the system's *z* axis, with an arbitrary orientation.

Let us now consider a given domain *D* (grey cells in Figure 2.a) on the uniformly sampled unit sphere. When changing the PCA CS, *D* does not fit anymore the resulting unit sphere partition (Figure 2.b).

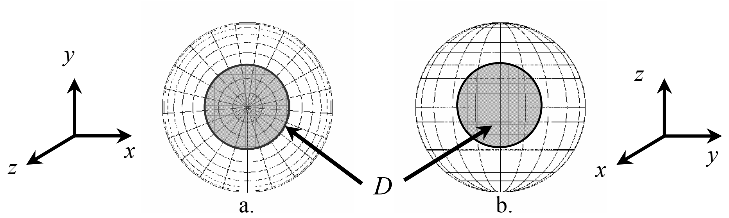


Figure 2. The effect of uniformly sampling the spherical coordinates.

This shows that the uniform sampling of the spherical angle coordinates θ and φ, which leads to partitions of the unit sphere into "meridians" and "parallels", which are non-equivalent within the three PCA CSs. Consequently, the 3 GCs represent a *minimal* generating set ensuring the completeness of the Hough representation. Nevertheless, keeping 3 distinct 3D HTs would lead to a prohibitively complex SD, since even for a coarse quantization with N_s=10, N_θ=20, and N_φ=10, the descriptor size will be of 3 x 2000 = 6000 floating point numbers. In addition, since the 3 GC are not equivalent, defining symmetric similarity measures would require highly time-consuming procedures.

To conclude, a geometric invariant 3D HT-based descriptor requires a high complexity in terms of descriptor size and matching computation time, which is untractable in practice.

Let us show how it is possible to overcome such drawbacks by abandoning the principle of uniform sampling of the unit sphere.

3. The canonical 3D HT descriptor

Our goal is to define partitions, for which the three GCs become equivalent, in the sense that there exists a one-to-one mapping between them.

3.1. Canonically invariant unit sphere partition

By construction, a partition obtained by projecting the vertices of any regular polyhedron onto the unit sphere is invariant to changes between the 48 PCA CSs

In addition, multiple granularity levels can be obtained for such partitions, by recursively subdividing each of the polyhedral faces (Figure 3).

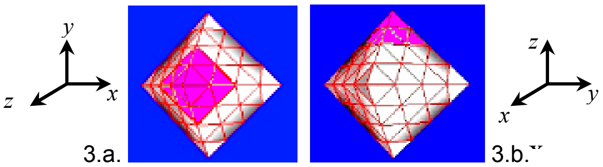


Figure 3. Canonically invariant partition obtained by two levels of subdivision of an octahedron.

Figure 3 illustrates this nice behavior, the partition cells in grey (Figure 3.a) being one-to-one mapped into corresponding cells in Figure 3.b, associated with a different PCA CS. Consequently, the 3 GCs can be derived one from each other by using appropriate permutations of the HT coefficients.

Let us note that the construction of other invariant partitions is also possible, but for reasons of simplicity we will retain here uniquely the octahedron-based partition.

In this way, we construct the *canonical 3D HTD* (C3DHTD), completely specified by an unique 3D HT, and associated with a canonical coordinate system arbitrarily selected from the 48 possible PCA CSs.

We can define now the similarity measure between two C3DHT as follows:

$$d(h_Q, h_R) = \min_{i=1,\dots 48} \left\{ \left\| h_Q - h_R^i \right\| \right\}, \qquad (1)$$

where h_Q (resp. h_R) denotes the query (resp. reference) C3DHTD, $\{ h_R^i \}_i$ are the 48 HTs of the reference model, corresponding to the complete set of PCA CSs, and $\| \cdot \|$ denotes the L$_1$ or L$_2$ norm.

Let us demonstrate that the above-defined similarity measure is symmetric. Let $\{T_i\}_{i=1}^{48}$ denote the set of the mappings between the 3D HTs associated with the 48 PCA CSs changes. Each T_i is expressed as a permutation, which is an *isometric* application. By observing that $\{T_i\}_{i=1}^{48} = \{T_i^{-1}\}_{i=1}^{48}$, we then obtain:

$$\begin{aligned} d(h_Q, h_R) &= \min_{i=1,\ldots 48}\left\{\left\|h_Q - h_R^i\right\|\right\} = \min_{i=1,\ldots 48}\left\{\left\|h_Q - T_i h_R\right\|\right\} = \\ &= \min_{i=1,\ldots 48}\left\{\left\|Ti(T_i^{-1} h_Q - h_R)\right\|\right\} = \min_{i=1,\ldots 48}\left\{\left\|T_i^{-1} h_Q - h_R\right\|\right\} = \\ &= \min_{i=1,\ldots 48}\left\{\left\|h_R - T_i\ h_Q\right\|\right\} = d(h_R, h_Q). \end{aligned}$$

Let us observe that applying such a naturally symmetric similarity measure results in a gain factor of 3 in computational time with respect to symmetric similarity measure associated with non-canonical 3D HT-based representations.

3.2. Implementation issues

The first implementation issue concerns the definition of weights w_{jk}^p, expressing the contribution of each face p to the 3D HT, with respect to the orientation given by the spherical angles (θ_j, φ_k).
The simplest definition consists of considering w_{jk}^p equal to the relative area A_p of face p, with respect to the total mesh area, for each value of j and k. However, the face orientation information is here completely eliminated. Coarse quantization of the parameter space (s, θ, φ), necessary for obtaining representations of reasonable complexities, will lead to an effect of over-accumulation of "parasite" areas that would seriously compromise the quality of the representation. Therefore, we have integrated the face orientation information within the weight as follows:

$$w_{jk}^p = \begin{cases} A_p\left|\left\langle n^p, n_{jk}\right\rangle\right|, & \text{if } \left|\left\langle n^p, n_{jk}\right\rangle\right| \geq T \\ 0, & \text{otherwise} \end{cases}, \qquad (2)$$

where n^p denotes the unit length normal vector of face p, $n_{jk} = (\cos(\varphi_k)\cos(\theta_j),\ \cos(\varphi_k)\sin(\theta_j),\ \sin(\varphi_k))^t$, $T \in [0,1]$ is a pre-defined threshold and $\langle\cdot,\cdot\rangle$ stands for the scalar product of two vectors in $\Re^3$.Within this formulation, the weight w_{jk}^p becomes equal to the thresholded projection area of the considered face p on a plane of orientation (θ_j, φ_k). Parameter T controls the influence of the orientation information, which is reinforced when $T \to 1$.
The second implementation issue is related to the definition of the mesh size S_{max}. Here, we have adopted a statistical approach derived from PCA, and set $S_{\max} = 1.5\sqrt{\lambda_1 + \lambda_2 + \lambda_3}$, where λ_1,, λ_2 and λ_3 are the lengths of the principal axes. Such an approach provides a more robust measure of scale than classic bounding-box-based procedures.
Finally, let us note that the underlying principle of accumulation holds only if the size of the mesh polygons is well-adapted to the HT specific granularity, controlled by the step Δ_s. The polygonal models are therefore initially adapted to Δ_s, by recursively subdividing each face of the mesh with at least an edge longer than Δ_s (Figure 4).

4.a. Initial model.

4.b. Remeshed model.

Figure 4. Adaptive remeshing of polygonal models.

The connectivity information is in this case irreversibly degraded, but this is without any consequence, since the C3DHTD is completely independent of the mesh topology.

4. Conclusion

This paper presented a 3D Hough transform-based shape descriptor, for shape representation and similarity retrieval of 3D polygonal mesh models. The proposed C3DHTD is intrinsically invariant with respect to topological representations and provided with a behavior independent of geometric transforms.

5. References

[1] T. Zaharia, F. Prêteux, "Shape-based retrieval of 3D mesh models", *Proc. IEEE International Conference on Multimedia and Expo* (ICME 2002), Lausanne, Switzerland, August 2002.

[2] T. Zaharia, F. Prêteux, "Hough transform-based 3D mesh retrieval", *Proc. SPIE Conference on Vision Geometry*, Vol. 4476, pp. 175-185, San Diego, CA, United States, August 2001.

[3] T.Zaharia, F.Preteux, "Descripteurs de forme pour l'indexation de maillages 3D", *Technique et Science Informatiques (TSI) – Special issue on Multimedia Indexation*, Vol. 22(9), pp. 1077-1105, 2003.