SHREC2007
3D Shape Retrieval Contest

Remco C. Veltkamp

Frank B. ter Haar

Department of Information and Computing Sciences, Utrecht University

Technical Report UU-CS-2007-015

www.cs.uu.nl

ISSN: 0924-3275

SHREC2007: 3D Shape Retrieval Contest

Remco C. Veltkamp   Frank B. ter Haar

June 2007

1 Introduction

The Network of Excellence AIM@SHAPE organized SHREC, the 3D Shape Retrieval Contest for the first time in 2006. The general objective is to evaluate the effectiveness of 3D-shape retrieval algorithms. There was only one track, the retrieval of polygonal models that are not necessarily watertight (polygon soups), and there were eight participants. For more information about the organization, see the SHREC home page at `http://www.aimatshape.net/event/SHREC/`. For the preceedings of that event, see [1]. The results were presented at the Shape Modeling International conference SMI'06.

After the success of SHREC2006, AIM@SHAPE has taken the initiative to organize a sequel. The contest is organized in conjunction with the SMI'07 conference (Shape Modeling International), where the evaluation results are presented.

In contrast with the previous contest, this year the contest involves multiple tracks. Track organizers take care of the following aspects:

- The particular task. One might ask for a complete or limited ranking, a classification, etc.
- The collection. Is the test set collected or generated. Are there copyright issues? Will it be made public? Is there a classification of the models?
- The queries. How are the queries determined? Are the query models from the collection or new models, or is it just a verbal description, etc.
- The ground truth. Is there a ground truth, is there a relevance scale (highly relevant, marginally relevant, ...), how and when is it determined (based on classification, visual inspection, ...), etc.
- The evaluation method. Which performance measure will be used for the evaluation (precision, recall, nearest neighbor, k-th tier, average dynamic recall, normalized cumulated gain, etc.).
- The procedural aspects. Does every participant perform the queries, or is that done in a central place? Who does the performance assessment? When are test set and queries made available?

2 Tracks

Seven tracks have been proposed. Two tracks, one on relevance feedback and one on the evaluation of similarity measures, received too few registrations of participants, and were cancelled. The following five tracks have been organized and were run:

- Watertight models. This track was organized by Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi (CNR - IMATI). Eight groups initially registered, five groups actually participated [7, 8, 9, 10, 11]. For the track report, see [2].
- CAD models. This track was organized by Yagnanarayanan Kalyanaraman and Karthik Ramani (Purdue University). Nine groups initially registered, four groups actually participated [9, 10, 11, 12]. For the track report, see [3].
- Partial matching. This track was organized by Simone Marini, Laura Paraboschi, and Silvia Biasotti (CNR - IMATI). Five groups initially registered, only two actually participated [13, 14]. For the track report, see [4].
- Protein models. This track was organized by Maja Temerinac, Marco Reisert and Hans Burkhardt (Albert-Ludwig University Freiburg). Three groups participated [5, 15, 16]. For the track report see [5].
- 3D face models. This track was organized by Frank ter Haar and Remco Veltkamp (Utrecht University). Seven groups initially registered, three actually participated [17, 18, 19]. For the track report see [6].

While last year there were eight participants in a single track, this year there are 13 participants, spread over five tracks. So the total number of participants has increased, but the participation per track has decreased, although some tracks received a higher number of initial registrations. We conjecture that participants generally need more time for preparation in order to inspect data formats etc.

Some groups participated in more than one track with the same method. In particular, [9, 10, 11] participate in the watertight models track as wel as in the CAD models track. It is interesting to see their relative performance.

Acknowledgment

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

References

[1] Remco C. Veltkamp, Remco Ruijsenaars, Michela Spagnuolo, Roelof van Zwol, Frank ter Haar. SHREC2006: 3D Shape Retrieval Contest. Technical Report UU-CS-2006-030, 2006.

[2] Daniela Giorgi, Silvia Biasotti, Laura Paraboschi. SHape REtrieval Contest 2007: Watertight Models Track. These proceedings, p. 5-10.

[3] Yagnanarayanan Kalyanaraman, Karthik Ramani. SHape REtrieval Contest 2007: CAD Models Track. These proceedings, p. 11-12.

[4] Simone Marini, Laura Paraboschi, Silvia Biasotti. SHape REtrieval Contest 2007: Partial Matching Track. These proceedings, p. 13-16.

[5] Maja Temerinac, Marco Reisert, Hans Burkhardt. SHape REtrieval Contest 2007: Protein Retrieval Track. These proceedings, p. 17-21.

[6] Frank B. ter Haar, Remco C. Veltkamp. SHape REtrieval Contest 2007: 3D Face Models Track. These proceedings, p. 22-26.

[7] Ceyhun Burak Akgül, Francis Schmitt, Bülent Sankur, Yücel Yemez. Multivariate Density-Based 3D Shape Descriptors. These proceedings, p. 27-30.

[8] Mohamed Chaouch, Anne Verroust-Blondet. 2D/3D Descriptor based on Depth Line Encoding. These proceedings, p.31-32.

[9] Thibault Napoléon, Tomasz Adamek, Francis Schmitt, Noel E. O'Connor. Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation. These proceedings, p. 33-37.

[10] P. Daras, D. Tzovaras, A. Axenopoulos, D. Zarpalas, A. Mademlis and M. G. Strintzis. The Spherical Trace Transform. These proceedings, p. 38.

[11] Tony Tung and Francis Schmitt. Shape retrieval of watertight models and CAD models using aMRG These proceedings, p. 39-45.

[12] Jun Kobayashi, Akihiro Yamamoto, Toshiya Shimizu, and Ryutarou Ohbuchi. A database-adaptive distance measure for 3D model retrieval. These proceedings, p. 46-49.

[13] Nicu D. Cornea and M. Fatih Demirci. 3D Object Retrieval using Many-to-many Matching of Reconstruction-based Curve Skeletons. These proceedings, p. 50-52.

[14] S. Marini, S. Biasotti, M. Spagnuolo and B. Falcidieno. Sub-part Correspondence by Structural Descriptors of 3D Shapes. These proceedings, p. 53-55.

[15] Bin Li, Yi Fang, Karthik Ramani, Daisuke Kihara. Auto classification of SHREC 2007-Protein Challenge. These proceedings, p. 56-58.

[16] P. Daras, V.Tsatsaias, M. G. Strintzis. 3D Protein Classification Using Topological and Geometrical Information. These proceedings, p. 59.

[17] Stefano Berretti. 3D Face Recognition by Matching Facial Regions. These proceedings, p. 60-64.

[18] Tung-Ying Lee and Shang-Hong Lai. An ICP-Based Approach to Retrieving Similar 3D Face Models. These proceedings, p. 65-68.

[19] Frank B. ter Haar and Remco C. Veltkamp. 3D Face Retrieval using Advanced Profile Matching These proceedings, p. 69-72.

SHape REtrieval Contest 2007: Watertight Models Track

Daniela Giorgi, Silvia Biasotti, Laura Paraboschi

CNR - IMATI
Via De Marini 6, 16149, Genova, Italy
{daniela.giorgi,silvia.biasotti,laura.paraboschi}@ge.imati.cnr.it

1. Introduction

Despite the availability of many effective 3D retrieval methodologies, only a handful of commercial products currently incorporate such techniques. A major barrier to the adoption of these techniques in commercial services is the lack of standardized evaluation: it is almost never obvious what is the best shape characterization or the best similarity measure for a given domain. Having a common understanding on 3D shape retrieval would help users to orient themselves to select the retrieved technique most suitable for their own specific needs. In this context, the aim of SHREC is to evaluate the performance of existing 3D-shape retrieval algorithms, in terms of their strengths as well as their weaknesses, using a common test collection that allows for a direct comparison of methods.

The pecularity of 3D media retrieval is given by the existence of many different representations for 3D shapes, from point-set models, to the many types of boundary representations and decomposition models [3]. Each particular representation is suitable to cope with particular application needs. For this reason, after the first successful experience of SHREC 2006, the contest has moved towards a multi-track organization.

In this report we present the results of the Watertight Models Track. Watertight models are object models represented by seamless surfaces, meaning that there are no defective holes or gaps. They turn out to be useful for many applications, such as rapid prototyping or digital manifacturing.

2. Data collection and queries

The collection to search in was made of 400 watertight mesh models, subdivided into 20 classes of 20 elements each. The experiment was designed so that each model was used in turn as a query against the remaining part of the database, for a total number of 400 queries. For a given query, the goal of the track is to retrieve the most similar objects, that is, the whole set of objects of the class it belongs to.

The type and categorization of the models in a database are crucial when testing a retrieval method, and it is difficult to separate out the influence of the dataset in the performance [6]. On the basis of these observations, we built our benchmark, shown in Figure 1. We manually established the ground truth, so that the classes exhibit sufficient and diverse variation, from pose change (e.g. the "armadillo" class) to shape variability in the same semantic group (e.g. the class of vases). All classes are made up of the same number of objects (20), so that *generality* is kept constant for each query [2], thus preventing from giving a different level of importance to different queries. Generality represents the fraction of relevant items to a given query with respect to the irrelevant embedding, i.e. the whole set of non-relevant models in the database; as observed in [2], it is a major parameter in influencing the retrieval performance.

The original models of our database were collected from several web repositories, namely the National Design Repository at Drexel University [1], the AIM@SHAPE repository [2], the Princeton Shape Benchmark [5], the CAESAR Data Samples [3], the McGill 3D Shape Benchmark [4], the 3D Meshes Research Database by INRIA GAMMA Group [5], the Image-based 3D Models Archive.

3 Participants

Each participant was asked to submit up to 3 runs of his/her algorithm, in the form of 400×400 dissimilarity matrices; each run could be for example the result of a different setting of parameters or the use of a different similarity metric. We remind that the entry (i, j) of a

[1] http://www.designrepository.org

[2] http://shapes.aim-at-shape.net

[3] http://www.hec.afrl.af.mil/HECP/Card1b.shtml

[4] http://www.cim.mcgill.ca/ shape/benchMark/

[5] http://www-c.inria.fr/gamma/download/

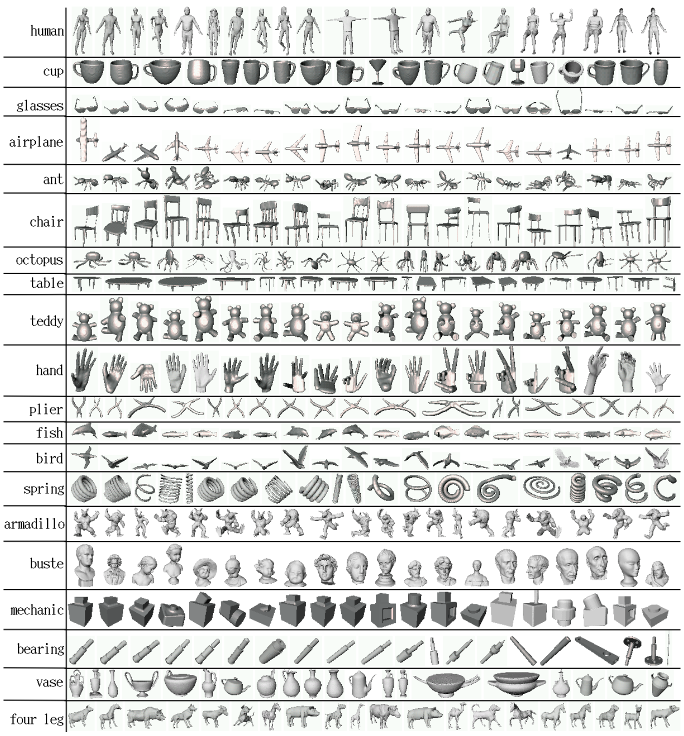


Figure 1. The database that has been used, divided into classes

dissimilarity matrix represent the distance between models i and j.

In this track five groups participated:

1. Ceyhun Burak Akgül, Francis Schmitt, Bülent Sankur and Yücel Yemez [7], who sent 3 matrices;
2. Mohamed Chaouch and Anne Verroust-Blondet [8] with 2 matrices;
3. Thibault Napoléon, Tomasz Adamek, Francis Schmitt and Noel E. O'Connor [9] with 3 matrices;
4. Petros Daras, Dimitrios Tzovaras, Apostolos Axenopoulos, Dimitrios Zarpalas, Athanasios Mademlis and Michael G. Strintzis [10] sent 1 matrix;
5. Tony Tung and Francis Schmitt [11] with 3 matrices.

For details on the algorithms and the different runs proposed by the participants, the reader is referred to their papers, included in these proceedings.

4 Performance measures

As observed in section 2, each query has its own set of 20 relevant items. We evaluated all the methods using the standard measures briefly described below.

1. **Precision** and **recall** are two fundamental measures often used in evaluating search strategies. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database, while precision is the ratio of the number of relevant records retrieved to the size of the return vector [4]. In our contest, for each query the total number of relevant records in the database is always 20, that is the size of each class. Starting from here, we evaluate the precision-recall measures for each query, and then average it over each class and over the entire database. An averaged recall value can be calculated through the so-called **average dynamic recall**, defined in our context as $ADR = \frac{1}{20}\sum_{i=1}^{20}\frac{RI(i)}{i}$, where $RI(i)$ indicates the number of retrieved relevant items within the first i retrieved items. $ADR \in [0,1]$ and its best value is $ADR = \frac{1}{20}\sum_{i=1}^{20}\frac{i}{i} = 1$.

2. We compute the **percentage of success** for the **first (PF)** and the **second (PS)** retrieved items, i.e. the probability of the first and second elements in the return vector to be relevant to the given query, and average them over the whole set of queries. For an ideal method $PF = PS = 100\%$.

3. With respect to a query, the **average ranking** is computed averaging the retrieval ranking (i.e. the positions in the return vector of ordered items) of all relevant items. The lower this value, the better the method. The optimal value in our experimental setting is $\frac{1+2+...+20}{20} = 10.5$.

4. The **last place ranking** is defined as $L = 1 - \frac{Rank-n}{N-n}$, where $Rank$ indicates the rank at which the last relevant object is found, n is the number of relevant items ($n = 20$), and N is the size of the whole database ($N = 400$) [1]. In this case, the performance of a method is as good as L is high; in fact $L \in [0,1]$, and the best value, occurring when the last relevant object is in the 20^{th} position, is $L = 1 - \frac{20-20}{400-20} = 1$.

5. There is a series of vectors to describe the performance of retrieval methods that descend from the so called *gain vector* G, which is, in our context, a 400-sized vector such that $G(i) = 1$ if the *i-th* retrieved item is in the same class of the query, 0 otherwise. The ideal gain vector is $IG(i) = 1, \forall i = 1, \ldots, 400$. The **cumulated gain vector** is defined as

$$CG(i) = \begin{cases} G(1) & i = 1 \\ CG(i-1) + G(i) & otherwise \end{cases}$$

in our case the ideal vector would be $ICG(i) = i$, $i = 1, \ldots, 400$.

The **discounted cumulated gain vector** is

$$DCG(i) = \begin{cases} CG(1) & i = 1 \\ DCG(i-1) + \frac{G(i)}{\log(i)} & otherwise \end{cases}$$

The ideal vector $IDCG$ is obtained using ICG and IG instead of, respectively, CG and G.

We implemented these measures using the software Matlab, version 7.1 (R14), installed on a Intel(R) Pentium(R) 4 CPU 3.00Ghz, 1.00 Gb of RAM.

5 Results and discussion

Most of the participants sent more than one matrix. In what follows, we compare the performance of the participants using their single best run, selected using the previously described measures. Our choice is motivated by reasons of readability of tables and figures; anyway, the results of each run are detailed in *SHape REtrieval Contest 2007: Watertight Models Track*, Technical Report IMATI-CNR-GE 9/07[6].

For all participants, the selected run coincides with the best one according to all measures used, except for Agkul et al., that proposed two runs with very similar performances, so that the choice of the best one was not unique; in this case, we choose the best performing one in terms of the area of the precision-recall graph. The comparison of the performance of different runs for the same authors reported in the technical report mentioned above allows to evaluate the dependence of each method on different choices of parameters. A general observation is that almost all the methods of the same authors perform more or less the same.

Fig.2 shows the standard **precision-recall graph**, plotting for each participant precision values vs. recall values. We remind that curves shifted upwards and to the right indicate a superior performance.

Numerical values for both the average (w.r.t. all the queries) **precision** and **recall** for return vectors of 20, 40, 60 and 80 items (i.e. 1, 2, 3, and 4 times the number of relevant items to each query) are reported in Table 1 (a) and (b), respectively.

The **average dynamic recall (ADR)** values for each participants are listed in the following table. As before, these values refer to the average value with respect to all the queries.

	ADR
Akgul et al.	0.7931
Chaouch et al.	0.7206
Napoleon et al.	0.7795
Daras et al.	0.7546
Tung et al.	0.8577

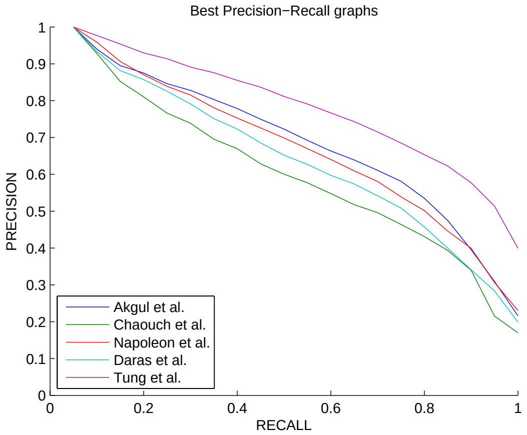


Figure 2. Comparing the best final Precision-recall graphs of each participant

Deepening in this kind of analysis, we can also deal with **class precision**, so that it is possible to make consideration not only on the average performance on the given database, but also on the specific class results. In this sense we invite the reader to have a look at the graphs in Fig. 3. It can be seen that the methods can perform in a very different manner when dealing with different classes of objects; for example, while in (a) Chaouch et al. gives the worst result (armadillo class), in (b) it yields the best one (tables class). Moreover, some classes are uniformly easy to deal with, such as the class of pliers (c), while others are uniformly difficult, as the class of vases (d).

Returning to the whole database, we observed that all the methods guarantee the identity property (i.e. $d(Q_i, Q_i) = 0$ $\forall i = 1, \ldots, 400$, Q_i i^{th} model in the database), so that the **percentage of success PF** of the first retrieved item is always 100%:

	PF
Akgul et al.	100%
Chaouch et al.	100%
Napoleon et al.	100%
Daras et al.	100%
Tung et al.	100%

,

The **percentages of success PS** w.r.t. the second retrieved item are listed below:

	Ps
Akgul et al.	93.97%
Chaouch et al.	92.81%
Napoleon et al.	95.87%
Daras et al.	93.33%
Tung et al.	97.68%

.

[6] available at http://watertight.ge.imati.cnr.it/watertight-global.pdf

Precision after	20	40	60	80
Akgul et al.	0.626	0.366	0.262	0.205
Chaouch et al.	0.546	0.329	0.241	0.190
Napoleon et al.	0.604	0.366	0.262	0.205
Daras et al.	0.564	0.346	0.252	0.199
Tung et al.	0.714	0.414	0.290	0.225
Ideal	1	0.5	0.333	0.25

(a)

Recall after	20	40	60	80
Akgul et al.	0.626	0.732	0.786	0.821
Chaouch et al.	0.546	0.658	0.724	0.763
Napoleon et al.	0.604	0.732	0.788	0.822
Daras et al.	0.564	0.692	0.756	0.798
Tung et al.	0.714	0.828	0.872	0.902
Ideal	1	1	1	1

(b)

Table 1. Precision and Recall after 20, 40, 60 and 80 retrieved items

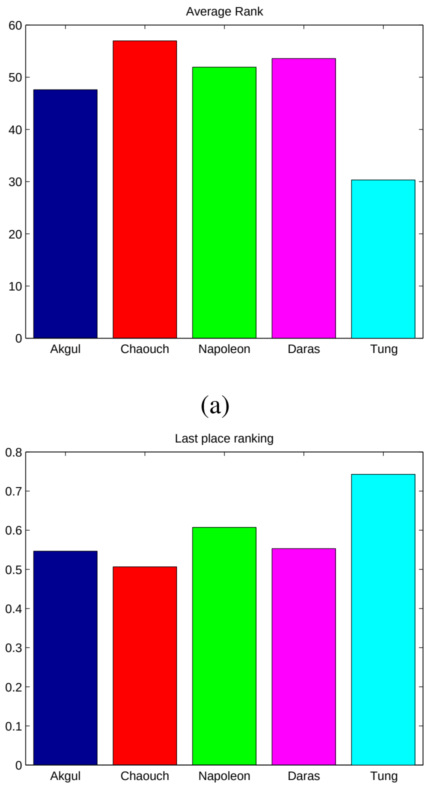
Concerning the **average ranking** we refer the reader to the histogram in Figure 4(a). Let us remark that in our case the ideal value for the average rank is 10.5, and a lower height of the bars indicate a superior performance.

The **last place ranking** is evaluated in Figure 4(b). The value yield by an ideal method is equal to 1. This measure gives an estimate of the number of the retrieved items a user has to search in order to have a reasonable expectation of finding all relevant items. The higher the bars in the histogram, the lower the number of items to check, meaning better results.

Finally, let us deal with the **mean cumulated** and **mean discounted cumulated gain vectors**, illustrated in Fig. 5. We show the 400-elements vectors (top), as defined in Section 4, and also a detailed view of the behavior of the first 20 components of the vectors (bottom), that is the behavior for the very first part of the return vector.

6 Conclusions

This paper proposed a comparative analysis of the retrieval performances of 5 different techniques, using a benchmark database of 400 models represented by watertight triangular meshes. Anyway, we warn the reader that, despite the care used in designing a benchmark and evaluating the results, it may happen that a single test collection delivers only a partial view of the whole picture.



(a)

(b)

Figure 4. Average ranking (a) and last place ranking (b) of all the methods.

References

[1] J. Eakins, J. Boardman, and M. Graham. Similarity retrieval of trademark images. *Multimedia*, 2(5):53–63, 1998.

[2] D. P. Huijsmans and N. Sebe. How to complete performance graphs in Content-Based Image Retrieval: Add generality and normalize scope. *IEEE Trans. PAMI*, (27):245–251, 2005.

[3] M. Mäntylä. *An introduction to solid modeling*. Computer Science Press, 1988.

[4] G. Salton and M. McGill. *Introduction to modern information retrieval*. McGraw Hill, 1983.

[5] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark. In *Shape Modeling International*, 2004.

[6] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the end of the early years. *IEEE Trans. PAMI*, 12(22):1349–1380, 2000.

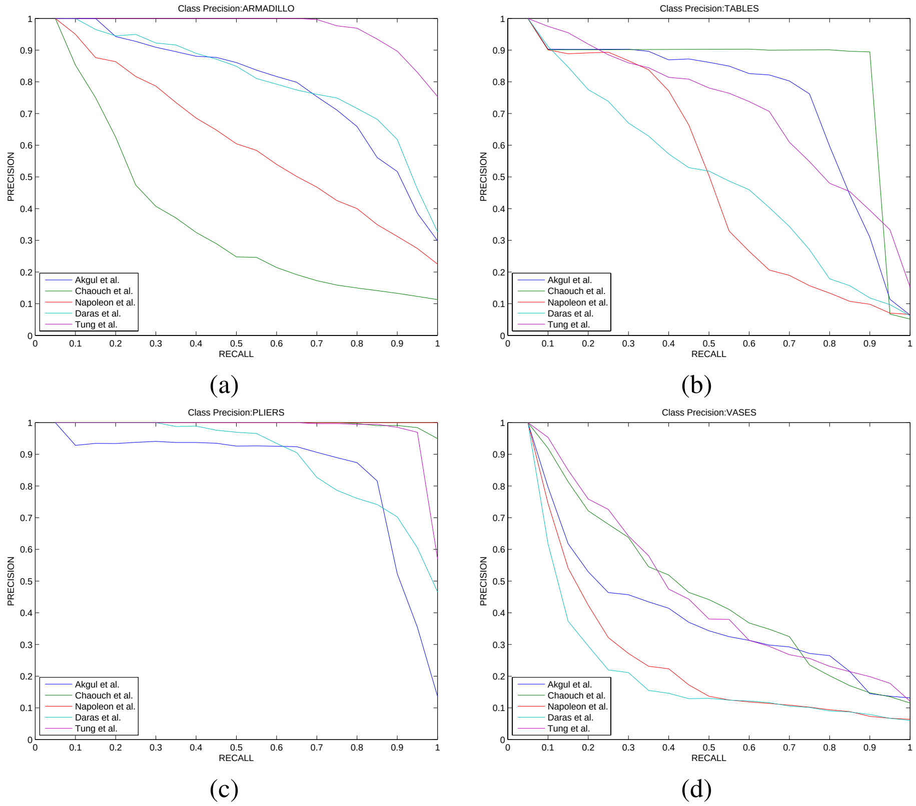
[7] Ceyhun Burak Akgül, Francis Schmitt, Bülent Sankur, Yücel Yemez. Multivariate Density-Based 3D Shape Descriptors. These proceedings.

[8] Mohamed Chaouch, Anne Verroust-Blondet. 2D/3D Descriptor based on Depth Line Encoding. These proceedings.

[9] Thibault Napoléon, Tomasz Adamek, Francis Schmitt, Noel E. OConnor. Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation. These proceedings.

[10] P. Daras, D. Tzovaras, A. Axenopoulos, D. Zarpalas, A. Mademlis and M. G. Strintzis. The Spherical Trace Transform. These proceedings.

[11] Tony Tung and Francis Schmitt. Shape retrieval of watertight models and CAD models using aMRG These proceedings.



(a) (b)

(c) (d)

Figure 3. Precision-recall graphs on 4 classes of the database.

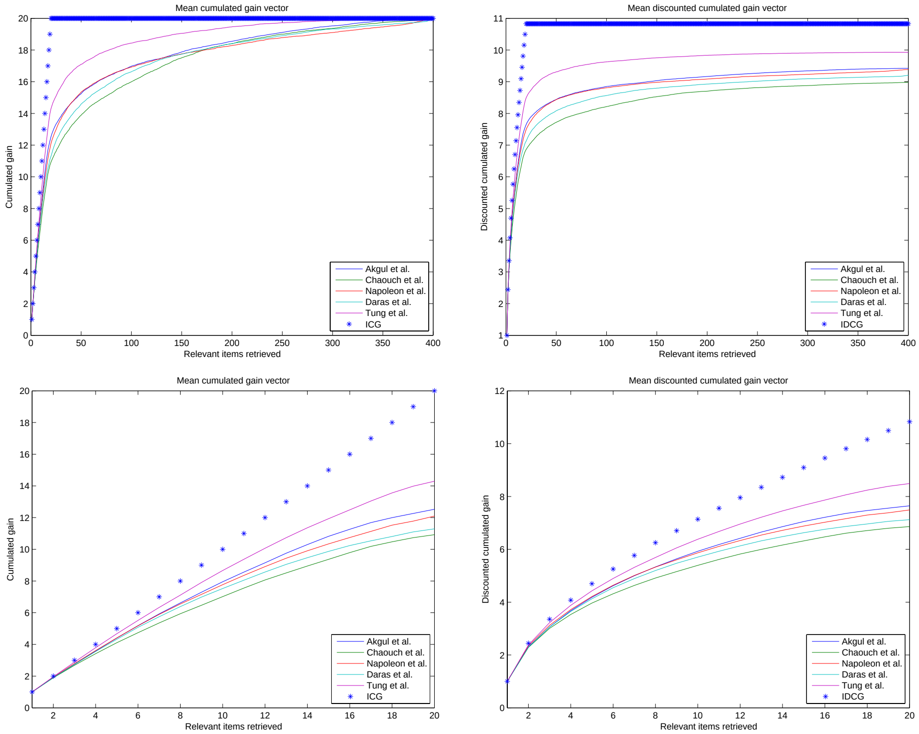


Figure 5. Mean cumulated gain vector (left) and mean discounted cumulated gain vector (right).

Shape Retrieval Contest 2007: CAD Models Track

Yagnanarayanan Kalyanaraman and Karthik Ramani
Purdue Research and Education Center for Information Sciences in Engineering
Purdue University, West Lafayette, Indiana, USA 47906.
{yagna, ramani} @purdue.edu

1. Introduction

Engineering parts typically have high genus, rounding features (fillets, chamfers), presence of internal structure. They are closed watertight volumes. Engineering models can be parts or assemblies. A part is an atomic unit and many parts are assembled to make an assembly. For example, a wheel can be a part where as a bike is an assembly. Moreover the engineering context is unique where in part families and parametric models, i.e. models differ by relative dimensions of various local geometries, are common. So this track focuses on engineering parts and the search tasks in an engineering context.

2. Dataset

The engineering track uses the Purdue Engineering Shape Benchmark (ESB) [1]. This established database consists of closed triangulated meshes of CAD parts in vendor neutral formats. This dataset is classified into a ground truth classification which has two levels of hierarchy. Overall there are three super-classes with 45 sub-classes under them. This classification can be browsed at http://purdue.edu/shapelab.

3. Query set

The query set was developed to articulate four main search conditions in the engineering context. All the models in the query set except six of them were constructed from the current version of Engineering Shape Benchmark. The search conditions that were considered are:

- **Subdivided/Decimated**: In engineering, the models are mostly stored in proprietary formats containing exact geometry information. Triangulated meshes are generated for neutral format data transfer between vendors or for rapid prototyping applications. Hence the different triangulation parameters for the same model will generate meshes with different sets of triangles. This is equivalent to having a subdivided or decimated mesh with respect to a reference mesh.
- **Parametric Variation**: Typically vendors manufacture classes of parts. A class or family of parts has the same overall shape. The different instances of models in a class are obtained by choosing different values for parameters such as dimensions and constraints. So some portions of the model may be in different sizes and proportions.
- **Slightly modified**: During many instances in typical engineering scenarios the small holes and features in a model are overlooked. Consider a single part being manufactured in three stages and each stage as being handled by different vendors. Then the level of detail of the CAD model they use will vary. The initial stage vendor may have a very coarse model with no holes and other machining features at all. The next stage vendor may machine the major features and the last stage vendor will create the small holes and other finishing effects. So we created models that reflect these situations with minor variation in the shape details.
- **Partial Shape**: Sometimes the user might be interested in a specific portion of the shape of a part. So we altered models by chopping off some portion of the model to create partial shape queries. These may be advanced level tasks for the shape search engines at this time.

Details regarding individual queries are available at https://engineering.purdue.edu/PRECISE/shrec/results/queryset.

Search Type	Queries
Subdivided/Decimated	9
Parametric Variation	10
Slightly modified	10
Partial Shape	4

4. Submissions

There were 4 participants in the CAD models track in SHREC 2007. The participants were Thibault Napoleon et al. [2] from ENST-TSI (2 runs), Petros Daras et al. [3] from ITI (5 runs), Tony Tung et al. [4] from Telecom Paris (4 runs), and Jun Kobayashi et al. [5] from University of Yamanashi (2 runs). The 13 runs of all the participants were all evaluated as described below.

5. Evaluation Measures

We have evaluated all the performance measures that were used in the SHREC 2006 [6]. The detailed results of all the runs' preformances are available at https://engineering.purdue.edu/PRECISE/shrec/results.

Based on the query model condition, the performance can be aggregated and studied for the different search types. From the evaluation for parametric models, it will be interesting to observe the results for the family of parts. We believe that the partial shape queries are advanced level tasks at this point of time. Nevertheless we incorporated few partial shape queries to get an idea of the current level of performance. There are six models which are totally new and not present in ESB. These models are expected to test the performance of any training that was used. All other queries are very similar to the ones currently existing in ESB.

6. Overall Evaluation

Figures 1 and 2 plot the overall mean normalized cumulative gains for each run. From the figures, it can be seen that Kobayashi et al.'s methods perform the overall best among all the submissions. More evaluations for each query is available on the website.

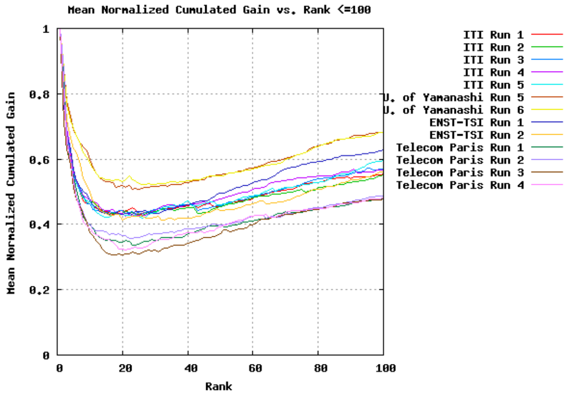


Figure 1: Mean Normalized Cumulative Gain vs. Rank

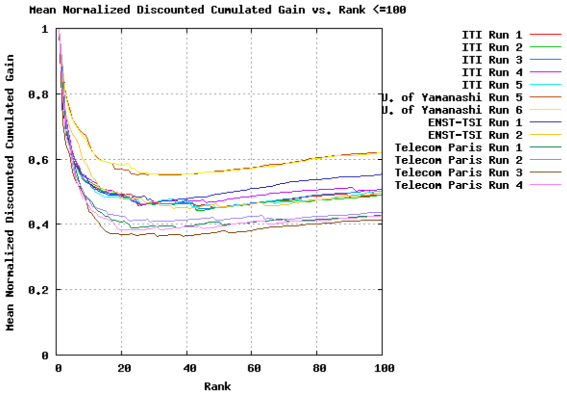


Figure 2: Mean Normalized Discounted Cumulative Gain vs. Rank

Acknowledgements

We would like to thank AIM@SHAPE for giving us the opportunity to conduct a dedicated track for CAD models and for the base evaluation code.

References

[1] S. Jayanti, Y. Kalyanaraman, N. Iyer and K. Ramani, Developing An Engineering Shape Benchmark For Cad Models, Computer-Aided Design, Volume 38, Issue 9, Shape Similarity Detection and Search for CAD/CAE Applications, September 2006, Pages 939-953.

[2] Thibault Napoléon, Tomasz Adamek, Francis Schmitt, Noel E. O'Connor. Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation. These proceedings.

[3] P. Daras, D. Tzovaras, A. Axenopoulos, D. Zarpalas, A. Mademlis and M. G. Strintzis. The Spherical Trace Transform. These proceedings.

[4] Tony Tung and Francis Schmitt. Shape retrieval of watertight models and CAD models using aMRG These proceedings.

[5] Jun Kobayashi, Akihiro Yamamoto, Toshiya Shimizu, and Ryutarou Ohbuchi. A database-adaptive distance measure for 3D model retrieval. These proceedings

[6] SHREC06, http://give-lab.cs.uu.nl/shrec/shrec2006.

SHape REtrieval Contest 2007: Partial Matching Track

Simone Marini, Laura Paraboschi, Silvia Biasotti

CNR-IMATI, Genova, Italy,
{simone.marini,laura.paraboschi, silvia.biasotti}@ge.imati.cnr.it

1. Introduction

The general objective of SHREC is to evaluate the effectiveness of 3D-shape retrieval algorithms. The aim of this track is to evaluate the performance of algorithms for the partial matching of 3D-shapes.

The dataset used within this track consists of 400 watertight models (i.e., object models represented by seamless surfaces) and 30 query models, where each query model shares common subparts with (possibly) more than one model belonging to the dataset. Given a query model, the goal of this track is the retrieval of the largest number of models from the dataset that share similar subparts with the given query.

2. Description of the Data Set and the Query Set

The Dataset used for the track consists of 400 3D objects grouped into 20 classes consisting of 20 objects each one. Every model is represented as a watertight manifold mesh encoded in the ’‘object file format’‘ (.off). The dataset is obtained by selecting its elements from a set of more than one thousand models harvested from the following well known 3D web repositories: the AIM@SHAPE repository[1], the National Design Repository at Drexel University[2], the CAESAR Data Samples[3] and the McGill 3D Shape Benchmark[4].

The models of the Dataset associated with the name and the classes they belong to can be found in Figure 1.

The query set consists of 30 3D models each one obtained by combining subparts of models belonging to the dataset. Also in this case each query model is a watertight manifold mesh encoded in the .off file format and it is obtained by cutting, scaling and rotating the original models. The whole set of queries is shown in Figure 2.

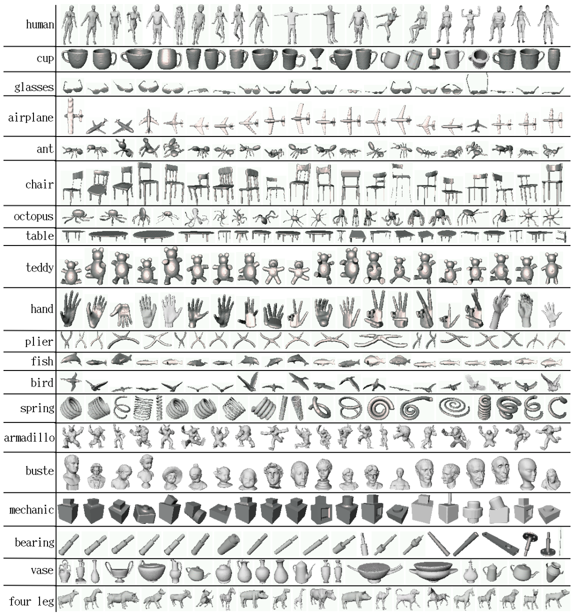


Figure 1. The dataset used to evaluate the algorithms proposed by the participants of the track.

3. Performance Measures

In order to evaluate the performance of the algorithms for the partial matching, a set of **highly relevant**, **marginally relevant** and **non-relevant** models belonging to the dataset has been associated to each query model.

Highly relevant models correspond to the members of the classes whose models compose the query.

The set of the marginally relevant models consists of the models that are ‘‘reasonably‘‘ similar to the given query. Since it is not easy to define such a set we have chosen an arbitrary set of models whose shape resembles the overall shape, or a sub-part, of the query model.

Table 1 shows the ground-truth associated to the queries

[1] http://shapes.aim-at-shape.net

[2] http://www.designrepository.org

[3] http://www.hec.afrl.af.mil/HECP/Card1b.shtml#caesarsamples

[4] http://www.cim.mcgill.ca/∼shape/benchMark/

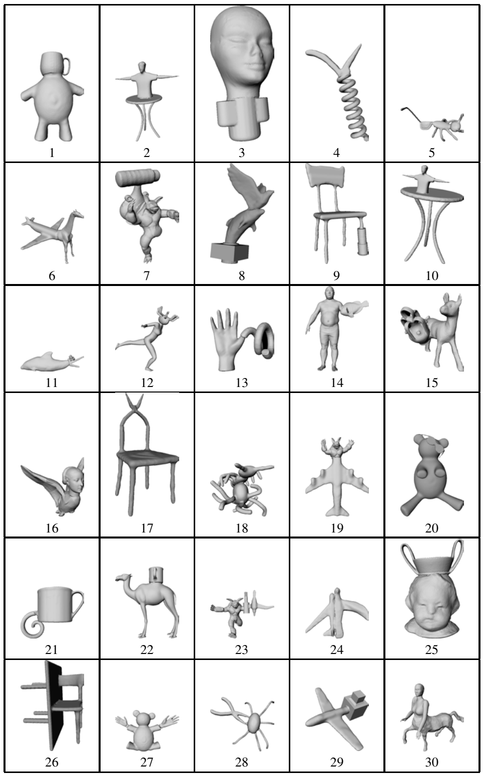


Figure 2. The set of queries. Each query is obtained by composing sub-parts of objects belonging to the dataset.

used in this track. Classes not mentioned neither in the Relevant nor in the Marginally Relevant classes have to be considered non-relevant.

The performance indicator used to compare the algorithms is the **Normalized Discounted Cumulated Gain vector** (NDCG), [2].

In order to compute the NDCG vector, the gain vector G has to be defined accordingly to the ground-truth. The gain vector is obtained by the ranked list where the model's identifiers are substituted with their ''relevance scores'' and where the relevance scores depend on the definition of the ground-truth. In particular, highly relevant, marginally relevant and non-relevant models have relevance scores 2, 1 and 0 respectively. The Discounted Cumulated Gain vector is recursively defined as:

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

where $G[i]$ represents the value of the gain vector at the position i. The normalized discounted cumulated gain vector NDCG is obtained by dividing DCG by the ideal cumulated gain vector, see [2] for details.

Query Number	Relevant Classes	Marginally Relevant Classes
1	cup, teddy	vase, four legs
2	human, table	armadillo, chair
3	buste, mechanic	
4	plier, spring	airplane, bird
5	ant, glasses	octopus
6	four legs, airplane	bird, plier, teddy
7	armadillo, vase, bearing	human, cup
8	fish, bird, mechanic	airplane, plier
9	chairs, bearings	tables
10	human, table	armadillo, chair
11	fisher, hand	
12	human, octopus	armadillo, ant
13	hand, spring	
14	human, fish	armadillo
15	four legs, vase	cup, teddy
16	bird, buste	airplane, plier
17	chair, plier	airplane, bird, table
18	ant, octopus	
19	airplane, armadillo	human, bird, plier
20	teddy, spectacle	four leg
21	cup, springs	vase
22	four legs, cup	vase, teddy
23	armadillo, bearing, bird	human, airplane, plier
24	airplane, bird	plier
25	head, vase	cup
26	chair, table	
27	teddy, hand	four legs
28	octopus, plier	bird, airplane, ant
29	airplane, mechanical	bird, plier
30	four legs, human	armadillo, teddy

Table 1. The ground-truth.

4. Results

Two algorithms have been submitted to this track:

- **Cornea:** Nicu. D. Cornea (Rutgers University) and M. Fatih Demirci (Utrecht University)
- **ERG:** S. Marini, S. Biasotti, M.Spagnuolo and B. Falcidieno (National Research Council of Italy, IMATI, Genova)

Each algorithm has been executed with two different setting of parameters: *Cornea 1* and *Cornea 2* for the algorithm described in [1] and *ERG1* and *ERG 2* for the algorithm described in [3]. The results of the two algorithm can be found at the SHREC07 homepage.[5]

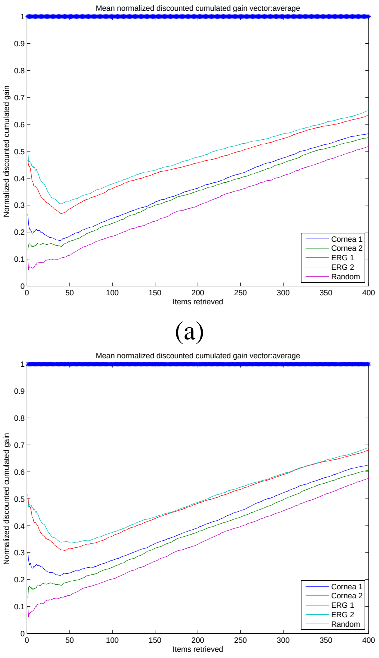
Figure 3(a) shows the overall performance of the two algorithms by considering only the highly relevant models, while in Figure 3(b) both highly relevant and marginally relevant models are considered. The diagrams are obtained by averaging the NDCG values obtained for each query.

Figures 4, 5 and 6 show the NDCG diagrams associated to each query model. Each diagram is obtained by considering both the highly relevant and the marginally relevant models.

5. Conclusion

The limited number of participants illustrates that while for shape retrieval a lot of methodologies have been defined, only few of that methods are focused on partial matching.

[5] http://www.aimatshape.net/event/SHREC/



(a)

(b)

Figure 3. Overall Normalized Discount Cumulated Gain considering only highly relevant models (a) and both highly relevant and marginally relevant models (b).

Due to this lack of methods it is difficult to draw the conclusion of this track. Even if the overall results are slightly improved by considering the marginally relevant models, the performance of the two methods are not radically changed. On the contrary, the single-query diagrams show that the two methods provide relevant differences depending on the heterogeneity of the query set.

6. Acknowledgments

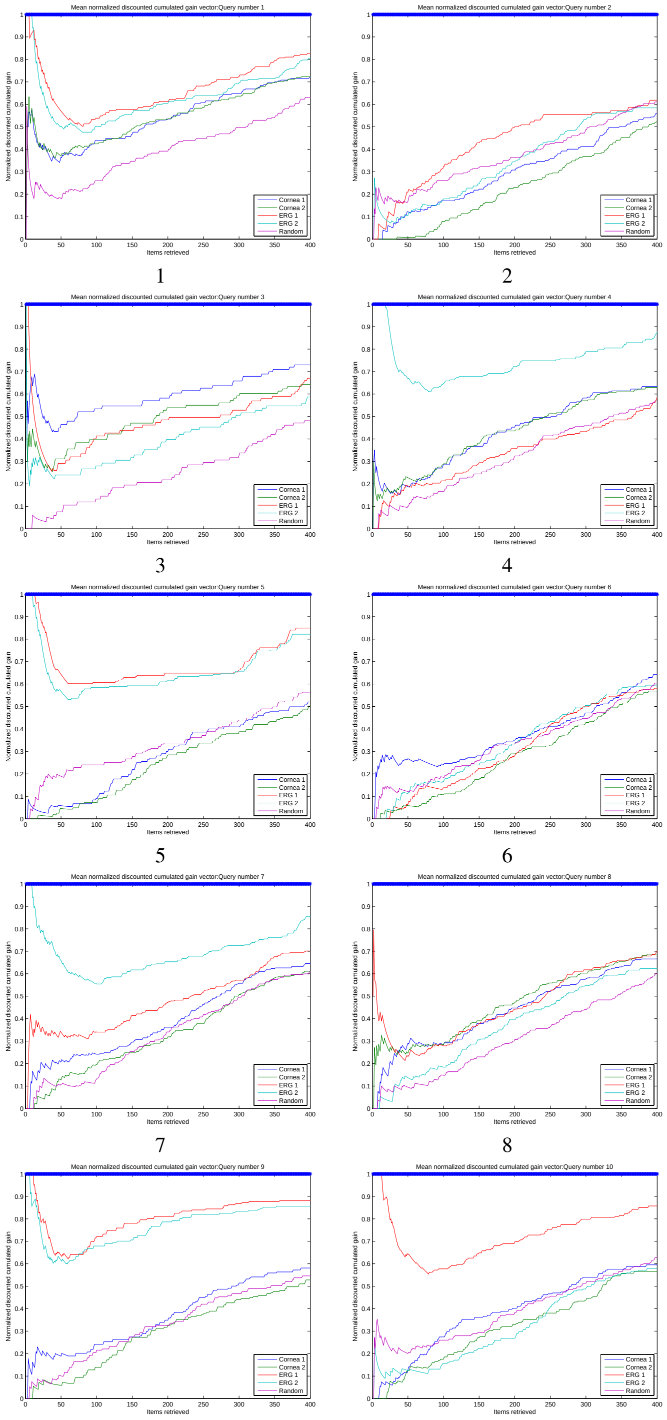
This research was supported by the FP6 IST Network of Excellence 506766 AIM@SHAPE.

References

[1] N. D. Cornea and M. F. Demirci. 3D Object Retrieval using Many-to-many Matching of Reconstruction-based Curve Skeletons. In this Proceedings.

[2] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[3] S. Marini, L. Paraboschi, and S. Biasotti. Sub-part Correspondence by Structural Descriptors of 3D Shapes. In this Proceedings.



1

2

3

4

5

6

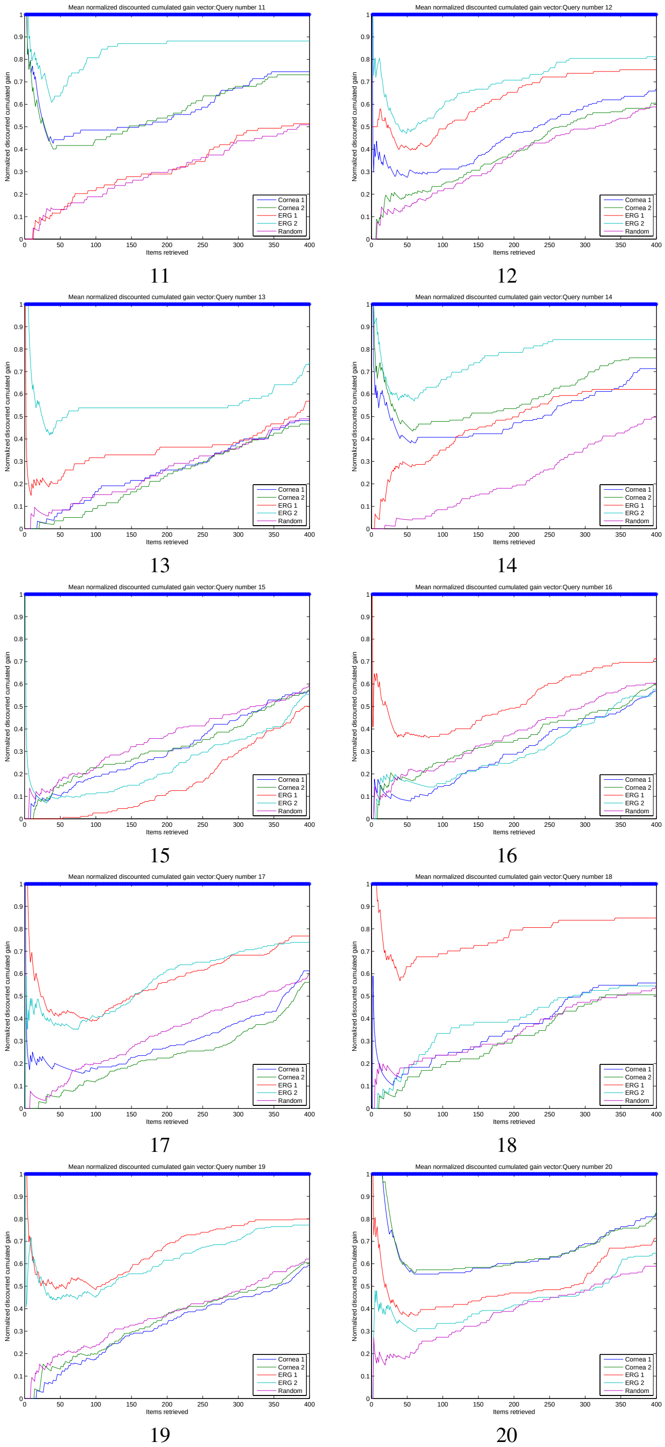
7

8

9

10

Figure 4. The NDCG vector diagrams for queries from 1 to 10.



11

12

13

14

15

16

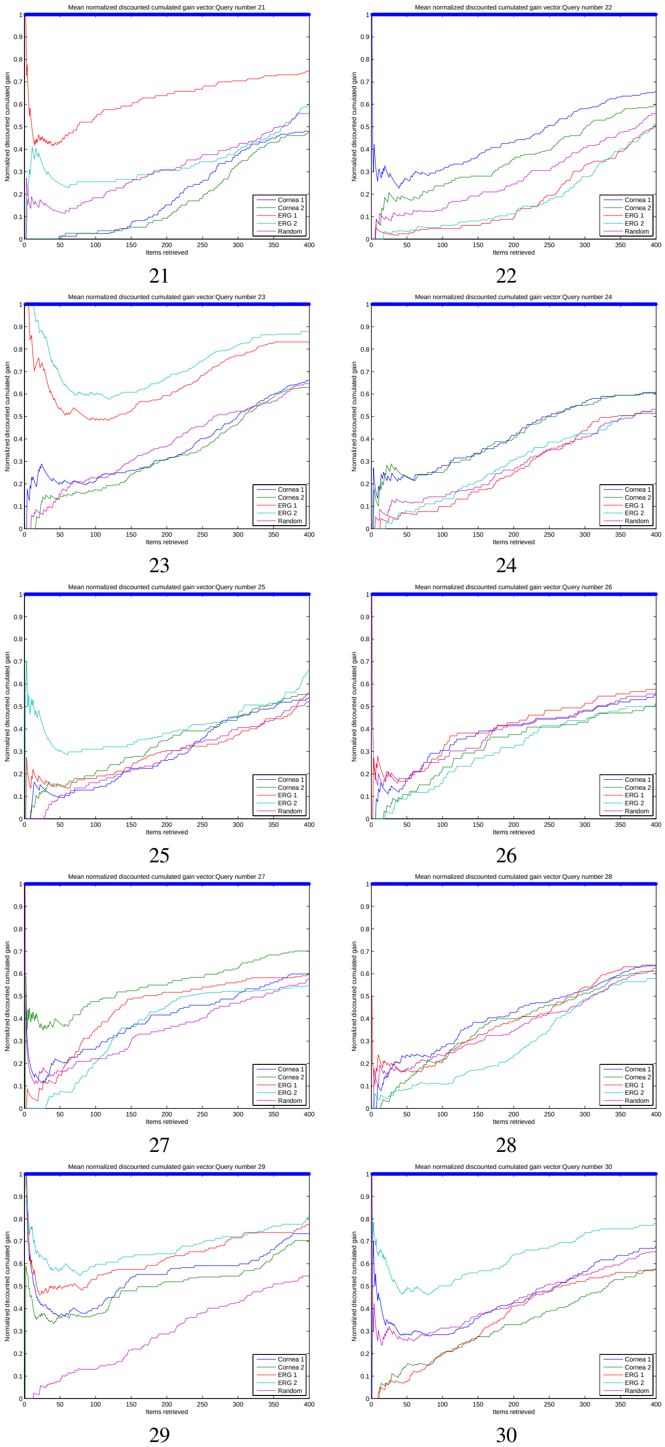
17

18

19

20

Figure 5. The NDCG vector diagrams for queries from 11 to 20.



21

22

23

24

25

26

27

28

29

30

Figure 6. The NDCG vector diagrams for queries from 21 to 30.

SHape REtrieval Contest 2007: Protein Retrieval Track

Maja Temerinac, Marco Reisert and Hans Burkhardt
Albert-Ludwig University Freiburg
Computer Science Laboratory
79110 Freiburg, Germany
temerina@informatik.uni-freiburg.de

Abstract

The SHREC - 3D Shape Retrieval Contest aims at evaluating the effectiveness of 3D-shape retrieval algorithms on various types of data. In this particular track the structure of three dimensional proteins is taken under consideration. The Protein Database (PDB) offers over 30000 protein structures. To cope with such a huge amount of data automatic classification and search tools get more and more important in biomolecular research. Feature based approaches are known to be the tool to provide a fast content based retrieval. In this contest we want to evaluate such methods. Each protein is attached with a fingerprint which is relying solely on coordinates of the atom sequence of the protein, no further information is used. Four different methods are evaluated.

1 Motivation

Proteins are linear sequences of amino acids which fold into three dimensional structures. Throughout evolution the amino acid composition can change, but the three dimensional structure of the protein stays conserved. The three dimensional structure of a protein is closely linked to its function. So, by finding similar three dimensional protein structures, their function and evolutionary linkage can be determined.

Molecular biologists are often interested in getting a survey of the objects in a biomolecular database making classification one of their basic tasks: To which of the recognized classes in the database does a new molecule belong? Several classification schemata such as SCOP [1], CATH and DALI/FSSP are available in the Internet. When a new object is inserted into the database the supervision by experts that are very experienced and have a deep knowledge in the domain of molecular biology is necessary in most cases. An efficient classification algorithm is desired that can speed up the classification process by acting as a fast filter for further investigation.

While SCOP and CATH require classification by human experts, a fully automatic classification is available from the FSSP database (Families of Structurally Similar Proteins), generated by the DALI (Distance matrix ALIgnment) system. The evaluation of a protein query by the DALI method is very expensive; comparing a single molecule against the entire FSSP database currently takes an overnight run.

2 The task

The task of this competition is to classify protein domains to one of the SCOP folds. The participants were able to train their feature extraction algorithms on the provided data set. One day before the end of the competition, the participants were provided with a set of 30 unknown protein domains. The query files had contained all atoms of the protein domain and their 3D coordinates. The task was then to assign the query protein domains to SCOP folds. Since the entire SCOP database is divided into more than 970 folds, we limited the task to assigning the unknown protein domains to one of the 27 folds provided in the data set.

We provided a dataset, which consists of 685 protein domains divided into 27 folds according to their SCOP classification. We have chosen this dataset because it is rather difficult, it does not contain any close-by related structures. Thus, the performance differences between the competitors become more apparent. For each protein only the atom positions are allowed to be used for retrieval. No additional information like chemical properties or others, e.g. temperature were allowed to be used. The 3D coordinates were provided in the common *pdb* file format.

2.1 SCOP

In the SCOP[1] (Structural Classification of Proteins) database published in 1995 all proteins of known structure are ordered according to their evolutionary and structural relationship. The protein domains are hierarchically grouped into families, superfamilies, folds and classes The basic unit in SCOP is a protein domain. The domain is either a monomer or a part of a protein and it should reflect a structure that did not change throughout evolution. Since this definition is very hard to measure by an algorithm, SCOP solely relies on visual inspection by experts.

Each domain can be addressed either by a unique integer (sunid) or by a concise classification string (sccs). For example, the protein with the PDB identity 1dlr has the sunid 34906 and the sccs 'c.71.1.1', where 'c' stands for the class, '71' the fold, '1' the superfamily and the last '1' for the family. In the 'dir.des.scop.txt' file the domains sunid, sccs and English names for proteins, families, superfamilies, folds and classes are listed. Also the sequence number where the domain in the chain starts and ends is contained in this file.

A family consists of proteins which either have residue identities over 30% or have similar structure or functions. Globins and Triosephosphate isomerase (TIM) are examples of protein families. A superfamily consists of proteins with lower than 30% sequential identity and a probable common evolutionary origin. Examples for superfamilies are Actin-crosslinking proteins. A fold contains proteins having same major secondary structures in same arrangement with the same topological connections. The most interesting members of a fold are those with low sequential similarity where there exists an evolutionary link to the other proteins of the fold. A class contains folds with similar secondary structure and is the most general way of defining a protein structure.

3 Participants

In this track we had two groups participating:

- B. Li, Y. Fang, K. Ramani, D. Kihara [6] (Purdue University, USA)
- P. Daras, V. Tsatsaias [7] (ITI, Greece)

The group from ITI participated with two different methods:

- a three dimensional shape-structure comparison method (Trace) [5]
- a graph based method (Graph) (not yet published)

Each group submitted a ranked list of the unknown 30 protein structures (See Figure 2) together with the distance of each query to each protein from the 633 training set computed by their method. The submitted ranked lists are available at *http://lmb.informatik.uni-freiburg.de/events/shrec07/results.html.* The SCOP classifiaction [1] was considered as the ground truth. Only the ATOM section of the PDB [2] files was provided.

We also compared the results to the classification achieved by our method (LMB, Germany) [3]. Since we organized the track, our results are out of competition. But we want to emphasize here that our features were not tuned on the 30 test proteins, we only used the training set for parameter tuning.

4 Methods

Li et al. focus on the topology of each protein: they use STRIDE [4] to detect the secondary structure, including the hydrogen bond. Then, they compute the beta sheets (beta strands connected with hydrogen bond) and the order. For main class a, b, c, d, g, and folds of a and g, they used the length and percentage of alpha helix and beta strand to classify. For each fold in each class b, c, d, they used the orders to classify.

P. Daras and V. Tsatsaias submitted two ranked lists computed with two different methods. The first method (Trace) is described in the paper [5]. The second method (Graph) is called '3D Protein Classification Using Toplogical and Geometrical Information' [7]. The 3D objects are firstly segmented to their molecular structure. Then, descriptors are extracted for each segment using spherical harmonics algorithms, and graphs are constructed for the molecules. Next, a sub-graph matching procedure is utilized in order to provide final similarity distances between the graphs.

Our method (LMB) was proposed in [3]. The basic idea of our approach is to obtain invariant fingerprint of the 3D structure. Therefore, we use a group integration approach. Practically the features can be seen as joint histograms over spatial distances, sequential distances and 2 angle-like quantities.

5 Evaluation

The ranked lists were evaluated by the following simple method: The next neighbor in the ranked list, meaning the protein domain with the least distance to the query protein is considered and the query protein

[1] http://scop.mrc-lmb.cam.ac.uk/scop/

is assigned to its class. One point is scored for the correct SCOP class only, two points for the correct SCOP fold and zero points if neither of them is correct. The maximal amount of points is 60, when the fold for each query protein is correctly classified.

As can be seen in Figure 1, from the three submitted methods , the team from Purdue performed (total score 45) best even though using simple features. The two methods submitted by team ITI misclassified half of the query proteins and their best method Graph scored 29 points. However, even better classification could be achieved by the LMB team, total score 52.

The query set (Figure 2) was chosen randomly from the 27 scop folds. Some proteins consisted of only one domain, others (e.g. Protein2, Protein8, Protein23) of several domains which were however all belonging to the same fold. Also, the size of the protein domains ranged from 31 amino acids (Protein11) to 364 amino acids (Protein16).

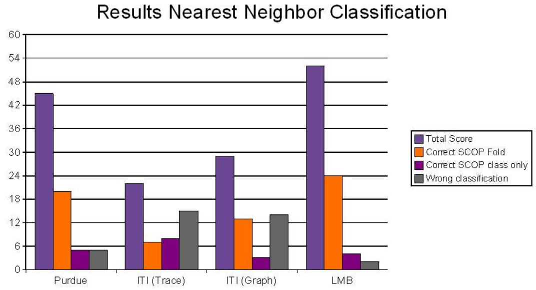


Figure 1. Comparison of the methods in terms of nearest neighbor classification.

Group	Wrong	Correct Class	Correct Fold	score
Purdue	5	5	20	**45**
ITI(Trace)	15	8	7	**22**
ITI(Graph)	14	3	13	**29**
LMB	2	4	24	**52**

Table 1. The results of the nearest neighbor classification according to the ranked lists submitted by each group.

6 Discussion and Conclusion

It is astonishing that the very simple method from Purdue is working so well in comparison to the methods proposed by ITI. It seems that properties and frequencies of secondary structure elements are a very important information for the presented task. The method (Trace) by ITI covers more the overall 3D structure, the tertiary structure, and is less sensitive for secondary and primary structure elements. Another issue is that (Trace) uses a normalization approach (by the center of mass) to obtain invariance against translations of the 3D structure. This approach can be very unstable when only partial structures are matching. Although the (Graph) method works a little bit better, it still achieves less than 50 % classification correctness. Statistical features seem to be a more discriminative feature than trying to establish one-to-one correspondences by a matching approach. The good performance of our method (LMB) could be explained by the fact that it describes all structural levels uniformly. Primary and secondary structure elements are described by cooccurences of small sequential and spatial distances, the tertiary structure is contained in occurrences of larger distances.

In conclusion, this competition has shown that statistics that rely upon low-level features as primary and secondary structure are much more important for the protein retrieval task, than features of the tertiary structure, that is the overall shape of the protein.

References

[1] A.G. Murzin., S.E Brenner, T. Hubbard and C. Chothia, *SCOP: a structural classification of proteins database for the investigation of sequences and structures*, J. Mol. Biol. 247, pp. 536-540, 1995.

[2] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhata, H. Weissig, I.N. Shindyalov and P.E. Bourne, *The Protein Data Bank*, Nucleic Acids Research, Vol. 28, pp. 235-242, 2000.

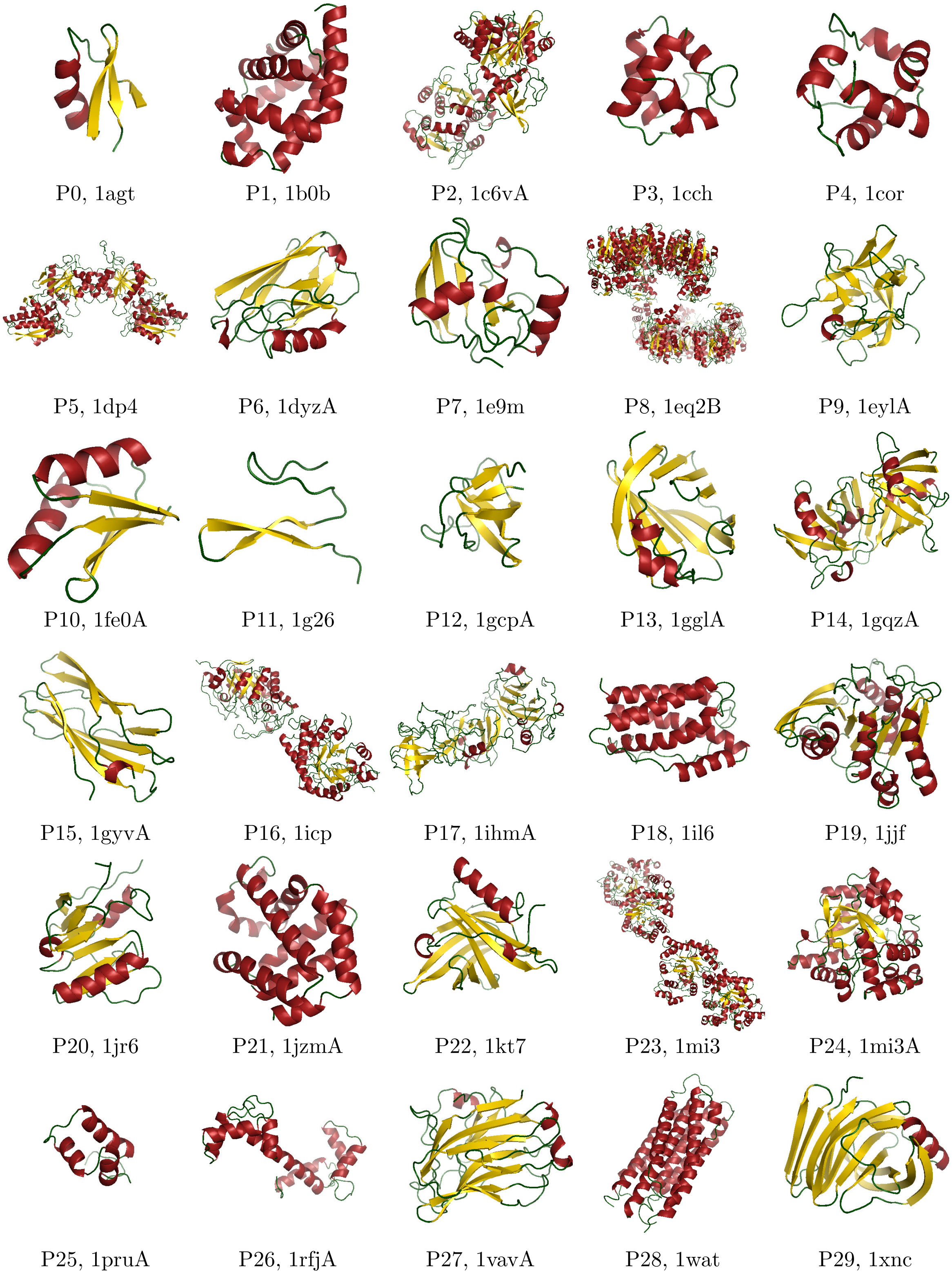
[3] M. Temerinac, M. Reisert and H. Burkhardt, *Invariant Features for Searching in Protein Fold Databases*, International Journal on Computer Mathematics , 'Special Issue on Bioinformatics', to appear 2007.

[4] D. Frishman, P. Argos *Knowledge-Based Protein Secondary Structure Assignment*, Proteins: Structure, Function, and Genetics 23:566-579, 1995

[5] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras and M.G. Strintzis, *Three-Dimensional Shape-Structure Comparison Method for Protein Classification*, IEEE/ACM transactions on Computational Biology and Bioinformatics, Vol. 3, No. 3, pp. 193-207, July 2006.

[6] Bin Li, Yi Fang, Karthik Ramani, Daisuke Kihara. Auto classification of SHREC 2007-Protein Challenge. These proceedings.

[7] P. Daras, V.Tsatsaias, M. G. Strintzis. 3D Protein Classification Using Topological and Geometrical Information. These proceedings.



P0, 1agt
P1, 1b0b
P2, 1c6vA
P3, 1cch
P4, 1cor

P5, 1dp4
P6, 1dyzA
P7, 1e9m
P8, 1eq2B
P9, 1eylA

P10, 1fe0A
P11, 1g26
P12, 1gcpA
P13, 1gglA
P14, 1gqzA

P15, 1gyvA
P16, 1icp
P17, 1ihmA
P18, 1il6
P19, 1jjf

P20, 1jr6
P21, 1jzmA
P22, 1kt7
P23, 1mi3
P24, 1mi3A

P25, 1pruA
P26, 1rfjA
P27, 1vavA
P28, 1wat
P29, 1xnc

Figure 2. The evaluation dataset.

SHape REtrieval Contest 2007: 3D Face Models Track

Frank B. ter Haar and Remco C. Veltkamp
Department of Information and Computing Sciences
Utrecht University, the Netherlands

1. Introduction

The general objective of SHREC is to evaluate the effectiveness of 3D-shape retrieval algorithms. This shape retrieval contest focusses on the retrieval of 3D face models using geometric shape information only. Insight in the performance of face retrieval algorithms may lead to better face recognition methods.

Seven participants registered for “SHREC 2007 - Shape Retrieval Contest of 3D Face Models”, four of them withdrew during the contest. Each participant was asked to 1) submit a list of queries for which relevant faces were automatically generated, 2) apply their method(s) to retrieve the relevant faces for each of the queries, that is, generate for each query a ranking of all faces in the database (run file), 3) submit up to five run files from different runs on the database, and 4) write a short paper about their method(s). All of the information contained in this track-document is available online [5].

2. Queries

Before the retrieval contest, each participant had access to the database by means of snapshots of the 1000 face models sorted on increasing face area. Each participant who registered to the contest had to provide ten potential queries using three model names per query, describing the query model and two other models to morph the query to (as described in Section 3). To remove duplicate queries and to increase the diversity of differently sized queries, eight queries were selected from each participant and eight additional queries were added, with 64 queries as a result (see Figure 1). Each participant was requested to submit a sorted ranked list of all database faces for each of the 64 queries, which we refer to as a run file.

3. Database

For this contest we generated a initial database of 1000 emotionless 3D face models with the use of the morphable face model [2]. Each instance of the morphable model was created by assigning m (m=100) random weights α_i within the range [-1.5,1.5] to the shape vectors of the morphable model: $\mathbf{S} = \sum_{i=1}^{m} \alpha_i \mathbf{S_i}$. A morphable model consists of 75,972 vertices and 150,958 faces. Snapshots of these random faces were used to select the queries in the previous section. Important to note is that each instance of the morphable model could have been a post-processed scan obtained with a Cyberware Head and Face 3D laser range scanner.

For each of the 64 query faces we generated eight relevant face models by morphing the query face towards and away from mean face (α_i=0) and towards two other faces in the database, using a 60-40 and a 40-60 weighting scheme (e.g. $\gamma_i = 0.6\alpha_i + 0.4\beta_i$). These relevant faces are added to the database which builds up a database of 1512 different face models. Four copies of database models were added to the database to be able to check the consistency of methods, none of these models were selected as part of a query.

The following items explain the creation of relevant faces in more detail and Figure 2 shows two examples of automatically generated clusters of relevant faces. Note that a generated cluster of 3D face models contains look-a-like faces, we can not assume that such a cluster resembles multiple range scans of truly the same person.

1. Given the mean face M and the selected query of randomly generated faces “face A - face Q - face B”.
2. Face Q is used as query.
3. Face Q is selected from the database, which makes it a highly relevant (i.e. identical) instance that needs to be retrieved.
4. Highly relevant instance R1 is created by scaling the weights of Q with 0.6, which morphs Q towards the mean face (less detail/noise).
5. Highly relevant instance R2 is created by scaling the weights of Q with 1.4, which morphs Q away from the mean face (more detail/noise).
6. Highly relevant instance R3 is created by averaging the weights of Q and A according a 60-40 ratio.
7. Highly relevant instance R4 is created by averaging the weights of Q and B according a 60-40 ratio.

Participant	Affiliation	Reference	Number of runs
Berretti et al.	University of Florence, Italy	[1]	5
ter Haar et al.	Utrecht University, the Netherlands	[4]	3
Tung-Ying Lee et al.	National Tsing Hua University, Taiwan	[3]	2

Table 1. Participants of SHREC 2007 - Shape Retrieval Contest of 3D Face Models

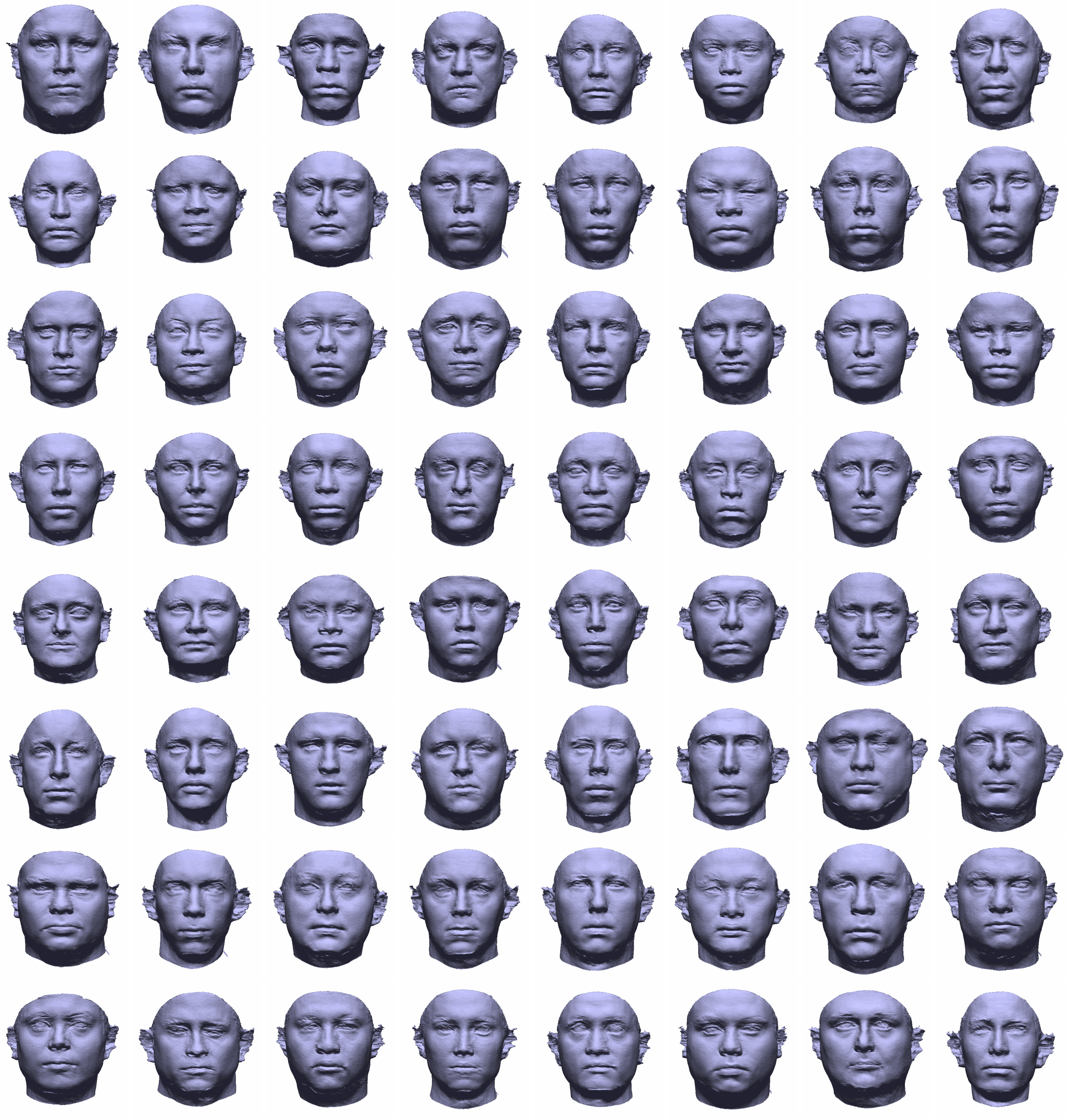


Figure 1. The 64 query faces of the contest.

8. Marginally relevant instance R5 is created by scaling the weights of Q with 0.4, which morphs Q further towards the mean face (less detail/noise).

9. Marginally relevant instance R6 is created by scaling the weights of Q with 1.6, which morphs Q further away from the mean face (more detail/noise).

10. Marginally relevant instance R7 is created by averaging the weights of Q and A according a 40-60 ratio.

11. Marginally relevant instance R8 is created by averaging the weights of Q and B according a 40-60 ratio.

12. Averaging two faces as in R3, R4, R7, and R8 causes the outcome to move towards the mean face, because the weights ai are pairwise averaged. We compensate for this by scaling the weights uniformly such that the total absolute sum of these weights equals 75, which is a measure for the level of detail.

A query face can have additional relevant models in case the query is used for another query to morph to. Consider a query face Q' and the selected faces "face A - face Q - face B". If face Q' equals face A, then face Q' has face R7 as additional highly relevant face. If face Q' equals face B, then face Q' has face R3 as additional marginally relevant face. In our contest we have eleven query faces that were used for one other query to morph to. Therefore, the final classification of face models has 53 queries with one identical, four highly relevant, and four marginally relevant faces and eleven queries with one identical, five highly relevant, and five marginally relevant faces. For each query the goal is to retrieve its nine or eleven relevant faces.

The generated instances of morphable models have normalized poses and full correspondences between vertex and face indices. Therefore, all faces in the database and query set 1) are centered with the center of mass in the origin, 2) are randomly rotated (at most 30 degrees around the x-, y-, and z-axis), 3) have their vertex and face indices randomly reassigned, 4) are renamed using a unique random index number. Note that a query's identical database face has a different orientation, which makes its retrieval non-trivial.

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 (cited from [6]).

Because the ranked lists are rankings of all the database items, the evaluation is performed on a certain scope of the ranked list. In our case the scope is the subset of the first **S** highest ranked items. The number of relevant items in the scope (true positives) is often used as part of an evaluation measure, which makes the scope size an important variable. In our contest we have a scope size equal to the size of the database, therefore we focus on evaluation measures that take into account the position of retrieved relevant items (e.g. Average Precision), or that use a selected subset of the ranked lists (e.g. the k-th Tier). For the actual performance measures and their definitions we refer to our website [5].

5. Results

For each ranked list, the following performance measures are calculated: Average Precision, First Tier and Second Tier (for both highly relevant and all relevant items), Average Dynamic Precision, Normalized Cumulated Gain and Normalized Discounted Cumulated Gain (for the first 5, 10, 25, 50 and 100 ranked items), and whether or not the best ranked item is identical to the query. For each query, these evaluation scores are listed per run file per participant (See our website [5]). In addition to these evaluation scores, there are two performance graphs showing the performance of individual run files per query, namely the Precision versus Recall graph and the Normalized Discounted Cumulative Gain versus Recall graph both based on all relevant items (Figure 3).

To evaluate a run file, we have averaged the evaluation scores over all 64 queries. These evaluation scores are listed per evaluation measure. Two performance graphs are included on the website to visualize some results, namely the the Mean Normalized Cumulated Gain and the Mean Normalized Discounted Cumulated Gain for the top 100 ranked items (See Figure 4 for the latter).

The retrieval results according four evaluation measures are shown in Table 2. The evaluation scores shown are total number of identical items found on top of the ranked lists, the Mean Average Precision of highly relevant items (MAPH), the Mean Average Precision of relevant items (MAPR) and the Mean Average Dynamic Precision.

6. Concluding remarks

For this contest a new database was generated using the Morphable model. The Morphable model was selected for the following reasons: 1) a large number of different faces

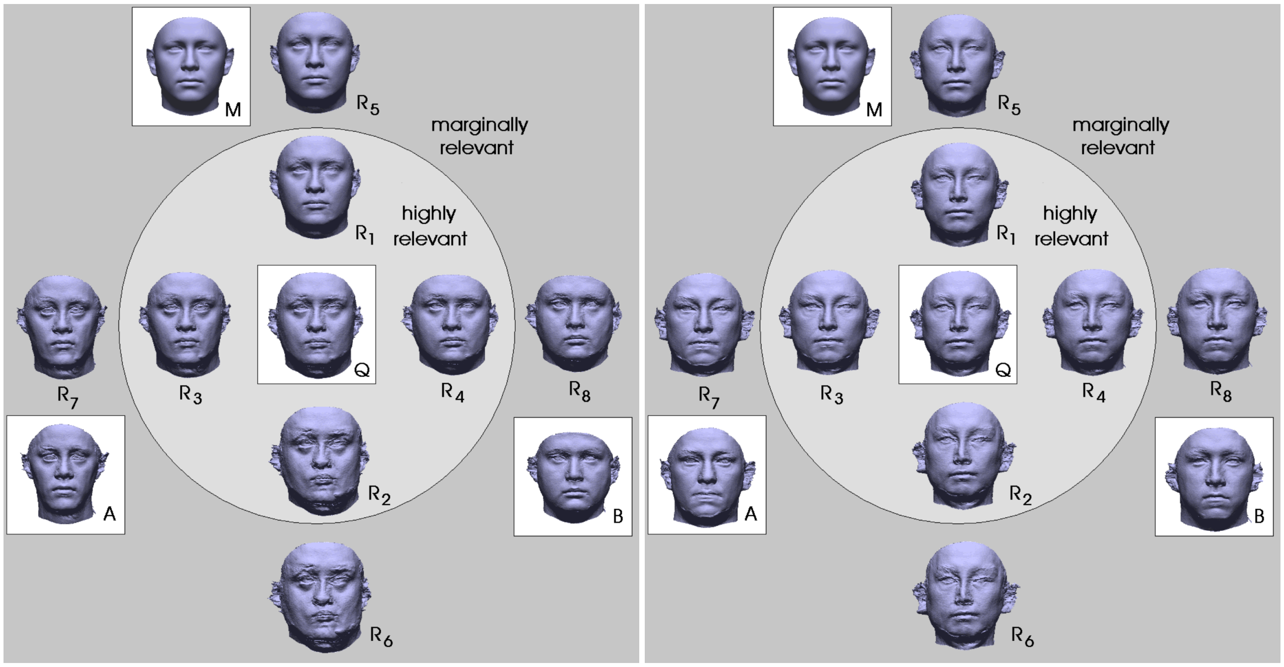


Figure 2. Two examples of automatically generated clusters of relevant models.

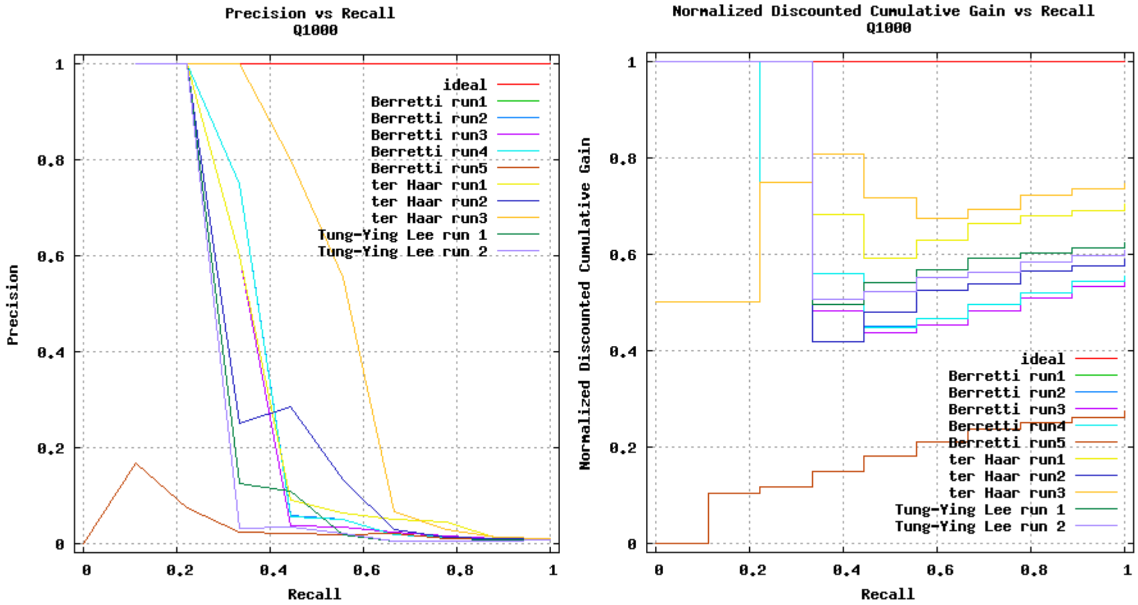


Figure 3. For each run file, the scores for the first query. Left: Precision versus Recall(relevant). Right: Normalized Discounted Cumulative Gain versus Recall(relevant).

can be automatically generated, 2) the morphing between faces allows the automatic creation of relevant faces, 3) the morphable model generates faces with proper topology and without holes. An existing database such as the Notre Dame database could have been used, but many face matching methods are developed for and tested on this database increasing the risk for overfitting.

The retrieval of relevant faces for each query poses to problems. Firstly, a face matching method has to cope with the random reorientation of the face models. Secondly, a face matching method has to select a set of features that distinguish relevant faces from non-relevant faces. The scores

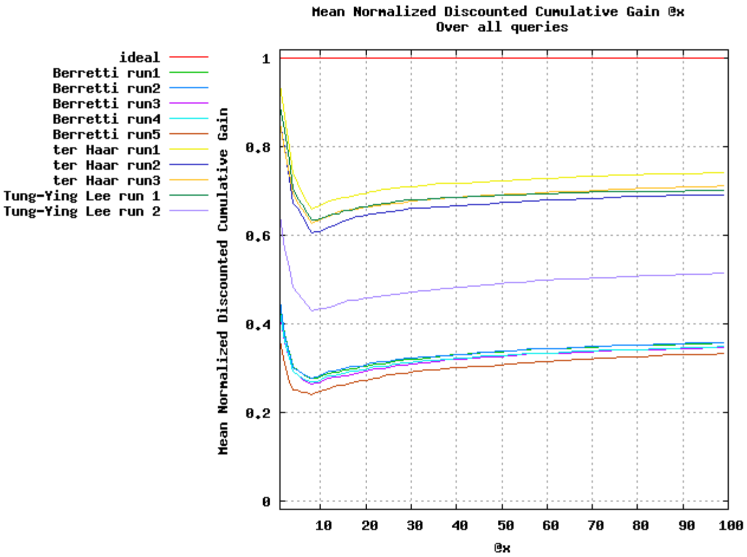


Figure 4. For each run file, the Mean Normalized Discounted Cumulated Gain @x[0,100] over all queries is shown.

Method	Identical 1st	MAPH	MAPR	MADP
ideal	64/64	1.0	1.0	1.0
ter Haar run1	63/64	0.6613	0.4775	0.6809
Tung-Ying Lee run 1	50/64	0.6159	0.4785	0.6519
ter Haar run3	48/64	0.5945	0.4782	0.6404
ter Haar run2	43/64	0.5618	0.4536	0.6186
Berretti run3	35/64	0.2253	0.1741	0.2903
Berretti run1	35/64	0.2320	0.1778	0.3018
Berretti run4	34/64	0.2213	0.1735	0.2877
Berretti run2	34/64	0.2288	0.1777	0.2986
Tung-Ying Lee run 2	32/64	0.3694	0.2976	0.4402
Berretti run5	20/64	0.2073	0.1598	0.2486

Table 2. Retrieval results all methods

from Figure 2 show that the retrieval of the relevant classified faces from a very large database is a difficult problem. The fact that the face identical to the query model is not always found on top of the ranked list, shows how difficult it is for face recognition methods to handle differently orientated faces. Because of the limited number of participants, care should be taken not to draw too far reaching conclusions from this retrieval contest.

Nowadays, numerous amount of techniques exist to obtain 3D face data using a laser range scanner, such as taking a single scan from one view, merging multiple scans from different views, and taking cylindrical scans around the face. Therefore, the assumption about a certain pose of the face model often doesn't hold. This indicates the need for further research in the direction of pose normalization methods invariant to different scan data and even pose normalization methods and face matching methods that are entirely rotation invariant.

Acknowledgements

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

References

[1] S. Berretti. 3D Face Recognition by Matching Facial Regions. In *These proceedings*.

[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[3] T.-Y. Lee and S.-H. Lai. An ICP-Based Approach to Retrieving Similar 3D Face Models. In *These proceedings*.

[4] F. B. ter Haar and R. C. Veltkamp. 3D Face Retrieval using Advanced Profile Matching. In *These proceedings*.

[5] Utrecht University. SHREC 2007 - Shape Retrieval Contest of 3D Face Models at `http://give-lab.cs.uu.nl/shrec/shrec2007`, Apr 2007.

[6] R. C. Veltkamp, R. Ruijsenaars, M. Spagnuolo, R. van Zwol, and F. ter Haar. SHREC2006: 3D Shape Retrieval Contest. Technical report, Utrech University, Technical Report UU-CS-2006-030, 2006.

Multivariate Density-Based 3D Shape Descriptors

Ceyhun Burak Akgül[1,2], Francis Schmitt[2], Bülent Sankur[1], Yücel Yemez[3]
[1] *Boğaziçi University, Electrical and Electronics Engineering Dept., Istanbul, Turkey*
[2] *GET - Télécom Paris - CNRS UMR 5141, Paris, France*
[3] *Koç University, Computer Engineering Dept., Istanbul, Turkey*
{akgul,schmitt}@enst.fr, bulent.sankur@boun.edu.tr, yyemez@ku.edu.tr

Abstract

In this brief communication, we provide an overview of the density-based shape description framework and the details of the runs we have submitted to the SHREC'07 Watertight Models Track. We also present the results of an improved matching scheme leading to more effective 3D retrieval.

1 Introduction

Density-based shape description is a framework to extract 3D shape descriptors from local surface features characterizing the object geometry [1, 2]. The feature information is processed with the kernel methodology for density estimation (KDE) [3] and the probability density function (pdf) of the local feature is estimated at chosen target points. The shape descriptor vector is then simply a discretized version of this probability density. This density-based approach provides a mechanism to convert local shape evidences, using KDE, into a global shape description. Our recent work on density-based shape descriptors [1, 2] for 3D object retrieval has proven that this scheme is both computationally rapid and effective compared to other state-of-the-art descriptors. In this brief communication, we provide an overview of the density-based shape description framework in Section 2 and the details of the runs we have submitted to the SHREC'07 Watertight Models Track in Section 3. More details about the methods and its performance can be found in [1, 2].

2 Overview of the Description Framework

A density-based descriptor of a 3D shape is defined as the discretized pdf of some surface feature S taking values within a subspace $\mathcal{R}_S$ of $\mathbb{R}^m$. The feature S is local to the surface patch and treated as a random variable. For an object O_l represented as a triangular mesh, evaluating the feature S at each triangle and/or vertex, we can obtain a set of observations about S, that we call *the source set*, denoted as $\{s_k \in \mathcal{R}_S\}_{k=1}^K$. Let $f(\cdot|O_l)$ be the pdf of S. Using the source set $\{s_k \in \mathcal{R}_S\}_{k=1}^K$, the value of this pdf at an arbitrary m-dimensional point t (which is in the range space $\mathcal{R}_S$ of the feature S) can be estimated by the following KDE equation [3]:

$$f(t|O_l) = \sum_{k=1}^{K} w_k \left|H\right|^{-1} \mathcal{K}(H^{-1}(t - s_k)), \qquad (1)$$

where $\mathcal{K} : \mathbb{R}^m \to \mathbb{R}$ is a kernel function, H is a $m \times m$ matrix composed of a set of design parameters called bandwidth parameters, and w_k is the importance weight associated with the kth observation s_k. Suppose that we have specified a finite set of points $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$, called *the target set*, within $\mathcal{R}_S$. The density-based descriptor $\mathbf{f}_{S|O_l}$ for the object O_l (with respect to the feature S) is then simply an N-dimensional vector whose entries consist of the pdf values computed at the target set $\overline{\mathcal{R}}_S = \{t_n \in \mathcal{R}_S\}_{n=1}^N$, that is, $\mathbf{f}_{S|O_l} = [f(t_1|O_l), \ldots, f(t_N|O_l)]$. To convert this general framework into a practical description scheme, we have to address the following issues:

Feature Design: Which surface features should be used to capture local characteristics of the surface? (See Section 2.1.)

Target Selection: How to determine the targets $t_n \in \overline{\mathcal{R}}_S$ at which we will evaluate the KDE equation in (1)? (See Section 2.2.)

Density Estimation: How to choose the kernel function $\mathcal{K}$, how to set the bandwidth parameter matrix H in (1) and how to evaluate this equation in a computationally efficient manner? (See Section 2.3.)

2.1 Feature Design

In the following, we assume that the 3D object is embedded in a canonical reference frame of $\mathbb{R}^3$ whose origin coincides with the center of mass of the object. The reference frame can be computed using the continuous PCA approach [4].

1. *Radial distance* R measures the distance of a surface point Q to the origin (centroid). Though not an effective shape feature all by itself; when coupled to other local surface features, it helps to manifest their distribution at varying radii.

2. *Radial direction* $\hat{\mathbf{R}}$ is a unit-norm vector $(\hat{R}_x, \hat{R}_y, \hat{R}_z)$ collinear with the ray traced from the origin to a surface point Q. This unit-norm vector is obviously scale-invariant.

3. *Normal direction* $\hat{\mathbf{N}}$ is simply the unit normal vector at a surface point and represented as a 3-tuple $(\hat{N}_x, \hat{N}_y, \hat{N}_z)$. Similar to the radial direction $\hat{\mathbf{R}}$, the normal $\hat{\mathbf{N}}$ is scale invariant.

4. *Radial-normal alignment* A is the absolute cosine of the angle between the radial and normal directions and is computed as $A = \left|\left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle\right| \in [0,1]$. This feature measures crudely how the surface deviates locally from sphericity. For example, if the local surface approximates a spherical cap, then the radial and normal directions align, and the alignment A approaches unity.

5. *Tangent-plane distance* D stands for the absolute value of the distance between the tangent plane at a surface point Q and the origin. This scalar feature D is related to the radial distance R by $D = R.A$.

6. *Shape index* SI provides a local categorization of the shape into primitive forms such as spherical cap and cup, rut, ridge, trough, or saddle. In the present work, we consider the following parameterization

$$SI = \frac{1}{2} - \left(\frac{2}{\pi}\right) arctan \left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2}\right),$$

   where κ_1 and κ_2 are the principal curvatures at the surface point. SI is confined within the range $[0,1]$ and not defined when $\kappa_1 = \kappa_1 = 0$ (planar patch).

Our previous work [1, 2] has shown that the density-based framework is more effective when we consider the joining of the local features described above, to obtain even higher dimensional features. In the present work, we consider three such instances:

Radial feature S_1 is obtained by augmenting the scale-invariant radial direction vector $\hat{\mathbf{R}}$ with the rotation-invariant radial distance R. The resulting 4-tuple $S_1 \triangleq (R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ can be viewed as an alternative to Cartesian coordinate representation of the surface point. In this parameterization, however, the distance and direction information are decoupled. With this decoupling, the range of individual features can be determined independently. In fact, $\hat{\mathbf{R}}$ lies on the unit 2-sphere, and the scalar R lies on the interval $(0, r_{max})$, where r_{max} depends on the size of the surface.

Tangent plane feature S_2 is obtained by joining the tangent plane distance D with the normal direction $\hat{\mathbf{N}}$, providing a 4-dimensional vector $S_2 \triangleq (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ that corresponds to the representation of the local tangent plane. As in the radial case, this representation also separates the distance and direction information concerning the tangent plane.

Finally, we define *a third feature* $S_3 \triangleq (R, A, SI)$, which aims at encoding the interaction between the radial and normal directions through the alignment feature A and at adding further local surface information through the shape index SI. Again the radial distance R augments the two-tuple (A, SI) to capture the characteristics of the shape at different distances from the center of the object.

In our runs, we will use the discretized pdfs of all these three features as descriptors and combine their individual discrimination capabilities (see Section 3).

2.2 Target Selection

The targets sets for our local features occur as the Cartesian products of their individual constituents. For instance, the S_1-feature is composed of a scalar feature $R \in (0, r_{max})$ and a unit-norm 3-vector $\hat{\mathbf{R}} \in \mathcal{S}^2$. Accordingly to determine the target set $\overline{\mathcal{R}}_{S_1}$, we first uniformly sample the interval $(0, r_{max})$ to obtain a distance set $\overline{\mathcal{R}}_R$, then partition the unit 2-sphere using the octahedron subdivision scheme described in [1, 6] to obtain a direction set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}}$, and finally take their Cartesian products to obtain $\overline{\mathcal{R}}_{S_1} = \overline{\mathcal{R}}_R \times \overline{\mathcal{R}}_{\hat{\mathbf{R}}}$. Note that r_{max} depends on the type of scale normalization applied to the object. The target set for $S_2 = (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$ can be obtained likewise. Finally the target set for S_3 is given by $\overline{\mathcal{R}}_{S_3} = \overline{\mathcal{R}}_R \times \overline{\mathcal{R}}_A \times \overline{\mathcal{R}}_{SI}$, where both $\overline{\mathcal{R}}_A$ and $\overline{\mathcal{R}}_{SI}$ are uniformly sampled versions of the interval $[0,1]$ since both A and SI share the same unit-interval as range.

2.3 Density Estimation

It is known that the particular functional form of the kernel does not significantly affect the accuracy of KDE [3]. In our scheme, we choose the Gaussian kernel since there exists a fast algorithm, the fast Gauss transform (FGT) [5], to rapidly evaluate large KDE sums in $O(K + N)$ instead of $O(KN)$-complexity of direct evaluation, where K is the number of sources and N is the number of targets.

The settings of the bandwidth matrix H has been shown to be critical for accurate density estimation [3], which in turn affects shape discrimination and retrieval performance [1]. The optimal bandwidth for KDE depends on the unknown density itself [3], making the appropriate choice of the bandwidth parameter a challenging problem. Guided by the results of our previous work in [1], we set the bandwidth parameter by averaging all object-level bandwidths given by Scott's rule [3] or by averaging the covariance matrix of the observations over the objects.

3 Experiments

Prior to descriptor computation, all models have been normalized so that descriptors are translation, rotation, flipping and scale invariant. For translation invariance, the object's center of mass is considered as the origin of the coordinate frame. For rotation and flipping invariance, we applied the continuous PCA algorithm [4]. For isotropic scale invariance, we calculate a scale factor so that the average point-to-origin distance is unity.

In each of the three runs that we have submitted to the SHREC'07 Watertight Models Track, we have used three different density-based descriptors: $S_1 = (R, \hat{R}_x, \hat{R}_y, \hat{R}_z)$, $S_2 = (D, \hat{N}_x, \hat{N}_y, \hat{N}_z)$, and $S_3 = (R, A, SI)$. Again in each of the runs, we have calculated the bandwidth matrix, required by the KDE, by averaging the covariance matrices of the observations over the meshes. We have used the l^1-metric to obtain three distance measures (each of which corresponds to one of S_1, S_2, and S_3) that we sum up to obtain a final dissimilarity value between each pair of objects. The runs differ in the following aspects:

Run 1. The mesh is reoriented so that the angle between radial and normal directions at a surface point is always acute.

Run 2. The orientation of the mesh is kept as given by the list of triangles and the *radial-normal alignment* feature is defined *with* the absolute value, i.e., $A = \left|\left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle\right| \in [0, 1]$.

Run 3. The orientation of the mesh is kept as given by the list of triangles and the *radial-normal alignment* feature is defined *without* the absolute value, i.e., $A = \left\langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \right\rangle \in [-1, 1]$.

The DCG performance of these runs on the SHREC'07 Watertight Database are 85.0%, 84.0%, and 84.9% respectively. As we have pointed out earlier in this section, these results have been obtained using the l^1-metric. However, as we explain in the sequel, the discrimination ability of S_1- and S_2-descriptors can be further increased by exploiting a certain property of the directional features $\hat{\mathbf{R}}$ and $\hat{\mathbf{N}}$ involved in these descriptors respectively. Since these features are explicity parameterized by Cartesian coordinates, i.e., $\hat{\mathbf{R}} = (\hat{R}_x, \hat{R}_y, \hat{R}_z)$ and $\hat{\mathbf{N}} = (\hat{N}_x, \hat{N}_y, \hat{N}_z)$, an error in the labeling or the sign of x-,y- or z-axes alters the resulting pdf completely, leading to an unreliable descriptor. PCA-based normalization methods assign the axes based on the eigenvalue of the covariance matrix, and if two eigenvalues are close to each other there remains an ambiguity about which is the correct one. Assigning a sign to an axis is also problematic.

A matching scheme to circumvent discrimination errors to such undesirable situations can be implemented by considering all possible configurations of the Cartesian coordinate system. The first axis can be labeled as one of the 6 elements of the set $\{x+, x-, y+, y-, z+, z-\}$, let this be $x+$; the second axis can be labeled as one of the 4 elements of the set $\{y+, y-, z+, z-\}$, let this be $y+$; finally the remaining axis can be labeled as either $z+$ or $z-$. Clearly, there are $6 \times 4 \times 2 = 48$ such configurations. A naive way to achieve invariance against mislabelings is to compute, for a given mesh, 48 different descriptors each of which corresponding to a fixed coordinate labeling. The dissimilarity between two objects can then be calculated as the minimum of 48 distance values obtained by holding one descriptor fixed and alternating the other over the 48 configurations. Such a naive approach is computationally and memory-wise impractical.

Hopefully, since the target sets of the $\hat{\mathbf{R}}$- and $\hat{\mathbf{N}}$-features arise from a uniform sampling of the unit 2-sphere by octahedron subdivision [6], a descriptor corresponding to a certain labeling of the coordinate system can be derived from one another by just permuting the entries appropriately. To give an example, let $t_n = (t_{n,x}, t_{n,y}, t_{n,z})$ be a fixed target in the target set $\overline{\mathcal{R}}_{\hat{\mathbf{R}}} \in \mathcal{S}^2$ (or $\overline{\mathcal{R}}_{\hat{\mathbf{N}}} \in \mathcal{S}^2$ identically). Octahedron subdivision ensures that, for instance, $t_{n'} = (t_{n,y}, t_{n,x}, t_{n,z})$ is also in the target set (actually, all the 48 possibilities are in the target set). Thus, evaluating 48 distance values between two descriptors as described above, is just a matter of holding one descriptor fixed and permuting the entries of the other appropriately. The final dissimilarity value is simply the minimum among these 48 distance values. This metric adds a factor of 48 to the overall matching complexity, way smarter then calculating and storing 48 descriptors each of which corresponding to a different labeling of the coordinate system.

The improvements gained by the matching scheme are tabulated in Table 1 for each of the runs. In this table, we provide DCG figures corresponding to single descriptors S_1 and S_2 as well. Observe that, in all runs/cases, the improved matching scheme does indeed improve the discrimination as measured by the DCG gain, defined as percent increase in DCG. It appears that S_1 benefits more from this improvement than S_2, DCG gains for Run 1 being 4.8% and 2.9% respectively. Enriched by the S_3-descriptor (which is invariant to all affine transformations but scale through the

Table 1. DCG Performance of the Improved Matching Scheme

Run	Desc.	l^1	Imp. l^1	Gain $(\%)$
1	S_1	74.4	78.0	4.8
	S_2	80.3	82.6	2.9
	$S_1 + S_2$	80.9	84.2	4.1
	$S_1 + S_2 + S_3$	85.0	86.7	2
2	S_1	73.2	76.8	4.9
	S_2	78.7	81.0	2.8
	$S_1 + S_2$	80.1	83.1	3.7
	$S_1 + S_2 + S_3$	84.0	85.4	1.7
3	S_1	73.3	76.8	4.8
	S_2	78.9	81.1	2.8
	$S_1 + S_2$	80.1	83.1	3.7
	$S_1 + S_2 + S_3$	84.9	86.3	1.6

Table 2. Classwise DCG Performance of the Density-Based Framework

Class	Avg. DCG $(\%)$	St. Dev. DCG $(\%)$
Ant	99.3	1.1
Armadillo	93.8	10.5
Bearing	74.8	14.9
Bird	73.0	11.5
Bust	82.7	9.9
Chair	98.3	1.7
Cup	85.8	16.3
Fish	94.7	6.7
Four-legged	93.2	10.3
Glasses	88.9	11.1
Hand	79.3	13.0
Human	85.2	11.6
Mechanic	95.9	10.8
Octopus	70.0	14.0
Plane	93.1	14.5
Plier	99.2	1.2
Spring	80.0	17.1
Table	84.4	21.6
Teddy	98.7	1.8
Vase	61.4	12.5

R-component), the overall descriptor in Run 1 achieves a DCG of 86.7% on the Watertight Models database.

It's also instructive to take a classwise look to the DCG performance in order to see what kind of discrimination can be achieved by the density-based framework on a database where classification semantics are topologically-guided (see Figure 1 in the main paper in this proceedings). In Table 2, we display classwise averaged DCGs and their standard deviations. From this table, we observe that the density-based descriptor performs nearly perfectly for the "Ant" and "Plier" classes with DCGs of 99.3% and 99.2% respectively. Worst DCG is obtained for the "Vase" class (DCG=61.4%) with a rather high standard deviation of 12.5%, indicating an inhomogeneity of the shapes within the class. The latter fact corroborates with the visual investigation of Figure 1 in the the main paper in this proceedings: the "Vase" instances differ significantly in geometry and appearence.

4 Conclusion

Discrimination performance of density-based descriptors have already been established in [1, 2] through experiments on two different databases: Princeton Shape Benchmark and Sculpteur (see references therein for these databases). In the present paper, we have experimentally demonstrated that the density-based framework provides adequate discrimination also on the SHREC'07 Watertight database. The density-based description framework being geometrical by foundation, it's comforting to see that it enables effective retrieval on a database where shape equivalences are topologically-induced. We also remark that our framework is computationally very efficient [1, 2] thanks to the fast Gauss transform [5], it's not demanding in terms of mesh quality or degeneracies, and it's able to support multiple 3D modalities be they 3D point clouds, meshes, or parametric surfaces. Our research on effective 3D retrieval continues with score fusion and similarity learning, guided by statistical techniques, which are able to incorporate domain-specific classification semantics to the matching scheme.

References

[1] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Density-based 3D shape descriptors. *EURASIP Journal on Advances in Signal Processing*, 2007:Article ID 32503, 16 pages, 2007. doi:10.1155/2007/32503.

[2] C. B. Akgül, B. Sankur, Y. Yemez, and F. Schmitt. Multivariate density-based 3D shape descriptors. In *Shape Modeling International (SMI '07)*, to appear 2007.

[3] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer, 2004.

[4] D. V. Vranić. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[5] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. *ICCV*, 1:464, 2003.

[6] T. Zaharia and F. Prêteux. Shape-based retrieval of 3D mesh models. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME'2002)*, Lausanne, Switzerland, August 2002.

2D/3D Descriptor based on Depth Line Encoding

Mohamed Chaouch, Anne Verroust-Blondet
INRIA Rocquencourt
Domaine de Voluceau, B.P. 105
78153 Le Chesnay Cedex, FRANCE
Email: mohamed.chaouch, anne.verroust@inria.fr

1. Introduction

For the Watertight Models Track of SHREC'07 (SHape REtrieval Contest 2007 organized by the Network of Excellence AIM@SHAPE), we tested our new 2D/3D approach based on depth lines (DLA) with two different similarity measures.
Our method is detailed in the article "3D Model Retrieval based on Depth Line Descriptor" [1] with the following abstract: *"In this paper, we propose a novel 2D/3D approach for 3D model matching and retrieving. Each model is represented by a set of depth lines which will be afterward transformed into sequences. The depth sequence information provides a more accurate description of 3D shape boundaries than using other 2D shape descriptors. Retrieval is performed when dynamic programming distance (DPD) is used to compare the depth line descriptors. The DPD leads to an accurate matching of sequences even in the presence of local shifting on the shape. Experimentally, we show absolute improvement in retrieval performance on the Princeton 3D Shape Benchmark database."*

2 Depth Line Descriptor Extraction

Our 3D shape retrieval system compares 3D models based on their visual similarity using depth lines extracted from depth images: The process first normalizes and scales 3D model into a bounding box. Then, it computes the set of $N \times N$ depth-buffer images associated to the six faces of the bounding box. The system then generates $2 \times N$ depth lines per image, considering each depth image as a collection of N horizontal and N vertical depth lines. Finally, each depth line is encoded in a set of N states called sequence of observations. The shape descriptor consists in the set of 6 $\times$ 2 $\times$ N sequences, with $N = 32$.
Please see our paper [1] for further details.

3 Experimental results

We submitted two runs:

1. In Run1 (DLA_DPD), to perform retrieval results, we tested the dynamic programming distance (DPD) because it tolerates some local shifting on the shape. We used here the Needleman-Wunsch algorithm [2].

2. In Run2 (DLA_HD), to compare the depth line descriptors, we used the Hamming distance (HD) which returns the number of corresponding state positions that differ.

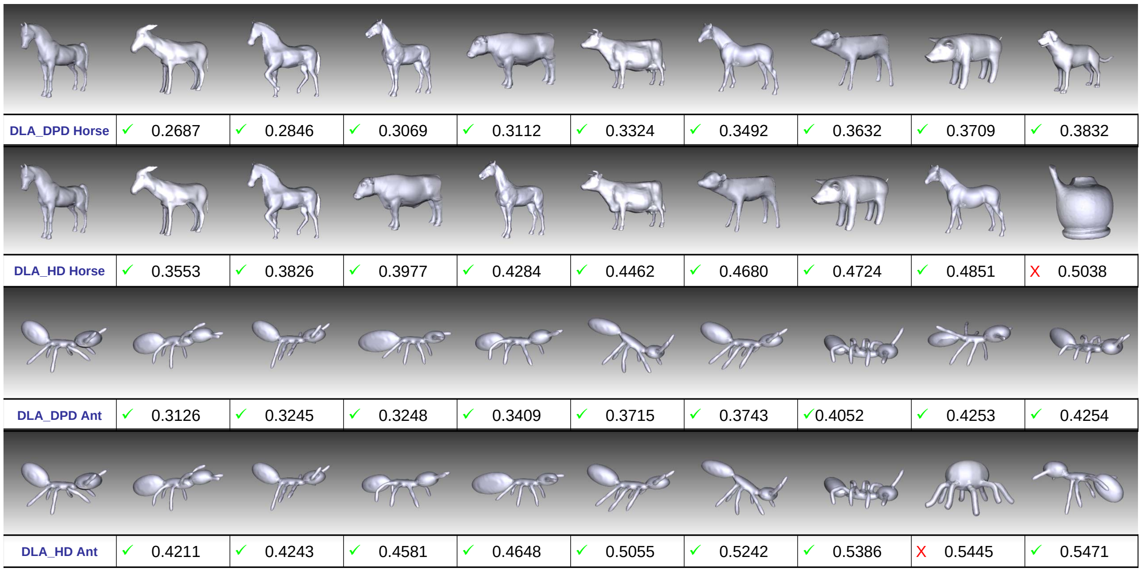
Our method does not need to have watertight models as input, as we work on the depth buffer images of the 3D objects. Thus it can be used on more general databases, such as the Princeton Shape Benchmark database [3]. We choose not to mix our descriptors with others, which may have increased the overall performance, but to test here the performance of our depth line approach to detect the classes of shapes where they are efficient and the classes of objects where other descriptors may be more suitable.
The experimental results show that:

- Run1 always performs better than Run2.

- Our descriptors are very well adapted to classes of objects that are homogeneous in shapes, such as tables, chairs and bearing. In fact, Run1 obtained the best precision-recall curve on table class in the Watertight Models Track of SHREC 2007.

- Articulated objects, such as human, armadillo, teddy, ant, glasses, hand and octopus classes present a great variation of shape inside their classes. As this variation is related to the structure of the objects, they are adapted to skeleton-based descriptors.

The presence of many classes of articulated objects in the watertight models track has influenced the overall performance of our approach.



DLA_DPD Horse	✓ 0.2687	✓ 0.2846	✓ 0.3069	✓ 0.3112	✓ 0.3324	✓ 0.3492	✓ 0.3632	✓ 0.3709	✓ 0.3832
DLA_HD Horse	✓ 0.3553	✓ 0.3826	✓ 0.3977	✓ 0.4284	✓ 0.4462	✓ 0.4680	✓ 0.4724	✓ 0.4851	X 0.5038
DLA_DPD Ant	✓ 0.3126	✓ 0.3245	✓ 0.3248	✓ 0.3409	✓ 0.3715	✓ 0.3743	✓0.4052	✓ 0.4253	✓ 0.4254
DLA_HD Ant	✓ 0.4211	✓ 0.4243	✓ 0.4581	✓ 0.4648	✓ 0.5055	✓ 0.5242	✓ 0.5386	X 0.5445	✓ 0.5471

Figure 1. Examples of similarity search. For each query (from "Four-legged animals class" and "Ants class"), we show the top 9 objects matched with DLA approach (using DPD and HD). The similarities between the query models and the retrieved models are given below corresponding images. ✓ and x indicate that the retrieved models belong or don't belong to the query's class, respectively.

Acknowledgments

This work has been supported in part by the DELOS NoE on Digital Libraries (EU IST NoE G038-507618).

References

[1] M. Chaouch and A. Verroust-Blondet. 3D model retrieval based on depth line descriptor. In *IEEE International Conference on Multimedia & Expo (ICME'07)*, Beijing, China, July 2007.

[2] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[3] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *SMI'04*, pages 167–178, Genova, Italy, June 2004.

Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation

Thibault Napoléon
Telecom Paris
CNRS UMR 5141
75013 Paris, France
napoleon@enst.fr

Tomasz Adamek
CDVP
Dublin City University
Dublin 9, Ireland
adamekt@eeng.dcu.ie

Francis Schmitt
Telecom Paris
CNRS UMR 5141
75013 Paris, France
schmitt@enst.fr

Noel E. O'Connor
CDVP
Dublin City University
Dublin 9, Ireland
oconnorn@eeng.dcu.ie

Abstract

We describe in this paper two methods for 3D shape indexing and retrieval that we apply on two data collections of the SHREC - SHape Retrieval Contest 2007: Watertight models and 3D CAD models. Both methods are based on a set of 2D multi-views after a pose and scale normalization of the models using PCA and the enclosing sphere. In all views we extract the models silhouettes and compare them pairwise. In the first method the similitude measure is obtained by integrating on the pairs of views the difference between the areas of the silhouettes union and the silhouettes intersection. In the second method we consider the external contour of the silhouettes, extract their convexities and concavities at different scale levels and build a multiscale representation. The pairs of contours are then compared by elastic matching achieved by using dynamic programming. Comparisons of the two methods are shown with their respective strengths and weaknesses.

1 Introduction

We proposed two methods for the 3D Shape Retrieval Contest 2007. Each one is based on a multi-view approach which keeps 3D model coherence by considering simultaneously a set of 2D images in specific view directions. The various silhouettes of a model being strongly correlated, using a set of them help to better discriminate one model among others.

First of all, we have to get a robust normalization of the model pose and model scale in order to remain invariant to various geometrical transformations (translation, rotation, scaling). We used a Principal Continuous Component Analysis [4][5] and the smallest enclosing sphere [3] to solve these problems.

The first method is based on silhouettes intersection. We capture a set of views of a model and we extract its silhouette in each view. The distance between two silhouettes is chosen as equal to the number of pixels that are not common to the two silhouettes intersection. The distance between two models is defined as the sum of the distances between their two sets of silhouettes.

The second approach is based on a multiscale representation of the external closed contour of non rigid 2D shapes presented in [1]. We capture a set of views of a model and for each view we extract and normalize the external border of the silhouette, and we buid its multi-scale shape representation where for each contour point we store information on the convexities and concavities at different scale levels. We then search the optimal elastic match between each pair of silhouettes by minimizing the distance between matched contour points and we integrate the distance over the silhouettes pairs.

Section 2 presents the normalization method for the 3D models. Section 3 describes the intersection methods and section 4 presents the contour convexities and concavities approach. Experimental results are shown in section 5.

2 Model normalization

Before comparing 3D models we need to proceed to a robust normalization of their pose and scale in order to remain invariant to various geometrical transformations (translation, rotation, scaling). For the center and the scale, we use the smallest enclosing sphere S [3] (see Figure 1). The normalization then becomes:

$$x = \frac{x - c_x(S)}{d(S)},\ y = \frac{y - c_y(S)}{d(S)}\ and\ z = \frac{z - c_z(S)}{d(S)}$$

where $d(S)$ is the diameter of S and $c_i(S),\ i = x,\ y,\ z$ are the i-th coordinates of its centre. The use of the smallest enclosing sphere has several advantages: it is fast to calculate, it allows maximizing the model size inside the unit sphere

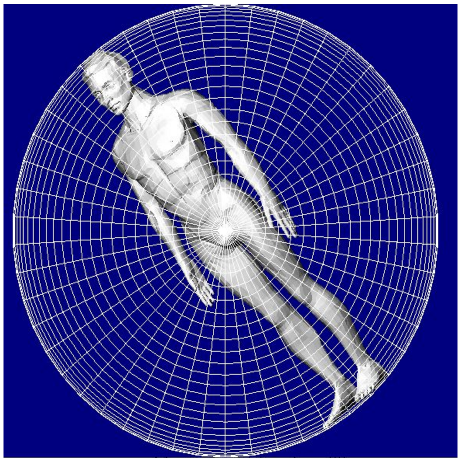


Figure 1. *Smallest enclosing sphere.*

and then its silhouette size in any view direction, with the guaranty that the silhouette remains inside the unit disc inscribed in the image domain associated to this view (no risk of accidental cropping of the silhouette).

For the normalization of the model pose we use the Continuous Principal Component Analysis [4][5] which defines and orientates the three principal axis of a model in a robust way and at a very reasonable computation cost.

3 Intersection descriptor

3.1 Signature extraction

The various silhouettes of a model being strongly correlated, using a set of them help to better discriminate one model among others [2]. For this, we can use any set of view directions regularly distributed in space. We consider here simply the three orthogonal views along the oriented principal axis with parallel projections. A higher number of views could be used but we limit it to 3 to keep a reduced size for the shape descriptors.

We choose an image size of 256x256 for each silhouette. This resolution gives a good tradeoff between precision and computation time. To keep the maximum information from a silhouette, the simplest way is just to keep its image (see Figure 2). We will then compare two silhouettes by superposing them and comparing their intersection with their union. A silhouette being a binary image we can stored it in a lossless compression format fast to read and to decode when comparing silhouettes. The signature of a model is then simply constituted by the three compressed silhouettes corresponding to the three oriented principal directions.

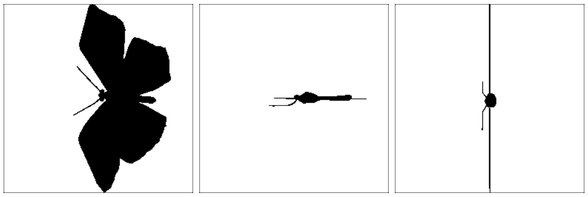


Figure 2. *Three silhouettes of a model.*

3.2 Signature matching

The distance between two models is defined as the distance between their two sets of silhouettes. The three silhouettes of each set being sorted according the three principal axis, this distance is then just defined as the sum of the distances of the three pairs of silhouettes, one pair per axis. The distance between two silhouettes is chosen as equal to the number of pixels that are non common to the two silhouettes, i.e. the difference between the areas of the silhouettes union and the silhouettes intersection (see Figure 3). This measure can be computed very efficiently directly on the files compressed with a simple run length encoding. The distance computation between two models is then straightforward and fast. To answer a query we just measure its distance to every database models and sort the list accordingly (see Figure 4).

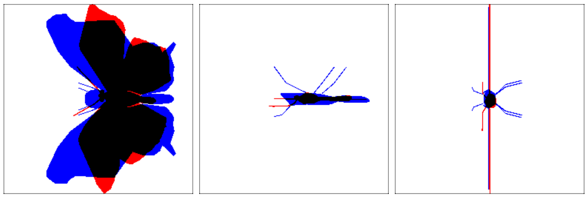
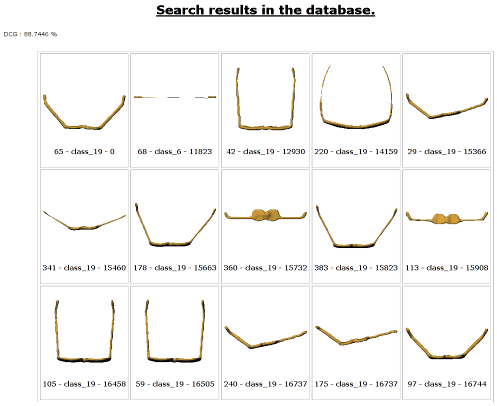


Figure 3. *Intersections of two models. In black their common parts, in blue the parts of the first model and in red the parts of the second one*

4 Contour convexities and concavities descriptor

4.1 Signature extraction

We use 256x256 silhouettes for both the CAD and Watertight models tracks. We also test 64x64 silhouettes on the Watertight models: this smaller image resolution allows a reduction of the descriptor size and of the computation time at the price of a stronger sampling noise leading to a lower retrieval precision. The descriptor of a silhouette contour C is obtained by normalizing the contour length



Search results in the database.

DCG : 88.7446 %

65 - class_19 - 0
68 - class_6 - 11823
42 - class_19 - 12930
220 - class_19 - 14159
29 - class_19 - 15366
341 - class_19 - 15460
178 - class_19 - 15663
360 - class_19 - 15732
383 - class_19 - 15823
113 - class_19 - 15908
105 - class_19 - 16458
59 - class_19 - 16505
240 - class_19 - 16737
175 - class_19 - 16737
97 - class_19 - 16744

Figure 4. *15 first results for shape retrieval using the intersection method on the Watertight model database. The query 65.off is on the top left.*

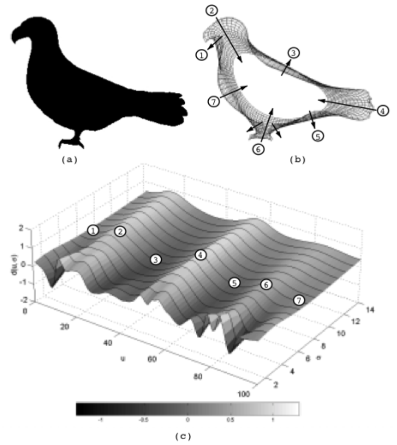


Figure 5. *Example of extracting the MCC shape representation: (a)-original shape image, (b)-filtered versions of the original contour at different scale levels, (c)-final MCC representation for 100 contour points at 14 scale levels.*

with 100 sampled contour points and by extracting convexity/concavity information at each sampled contour point and at 10 scale levels [1]. The representation can be stored in the form of a 2D matrix where the columns correspond to contour points (contour parameter u) and the rows correspond to the different scale levels σ. The position (u, σ) in this matrix contains information about the degree of convexity or concavity for the contour point u at scale level σ. The simplified boundary contours at different scale levels are obtained via a curve evolution process. It should be noted that we use the same number of sample contour points at each scale. Let the contour C be parameterized by arc-length $u:\ C(u) = (x(u),\ y(u))$, where $u \in [0, N]$. The coordinate functions of C are convolved with a Gaussian kernel ϕ_σ of bandwidth $\sigma \in \{1, 2 ... \sigma_{max}\}$. The resulting contour C_σ becomes smoother with increasing σ value, until finally the contour becomes convex (see Figure 5).

We propose a very simple measure for the convexity/concavity of the curve. It is defined as the displacement of the contour between two consecutive scale levels. If we denote the contour point u at scale level σ as $p(u, \sigma)$, the displacement $d(u, \sigma)$ of the contour between two consecutive scale levels at point $p(u, \sigma)$ can be defined as the Euclidian distance between positions of $p(u, \sigma)$ and $p(u, \sigma - 1)$.

4.2 Signature matching

When comparing two contours A and B, it is necessary to examine the distance between each sampled contour point of both contours. If two contour points u_A and u_B are represented by their multi-scale features $d_A(u_A, \sigma)$ and $d_B(u_B, \sigma)$ respectively, then the distance between the two contour points can be defined as:

$$d(u_A, u_B) = \frac{1}{K} \sum_{\sigma=1}^{K} |d_A(u_A, \sigma) - d_B(u_B, \sigma)|$$

where K is the number of scale (here 10).

As part of the matching process, the best correspondence between contour points must be determined. We use a dynamic programming method with an $N * N$ distance table to conveniently examine the distances between corresponding contour points on both shapes. The columns represent contour points of one shape representation and the rows represent the contour points of the other. Each row/column entry in the table is the distance between two corresponding contour points calculated according to the previous equation.

Finding the optimal match between the columns corresponds to finding the lowest cost diagonal path through the distance table (see Figure 6 where the contours feature vectors are illustrated as grey levels along each axis).

The three silhouettes of each set being sorted according to the three principal axis, the distance between two models is just defined as the sum of the distances of the three pairs of silhouettes, one pair per axis (see Figure 7).

5 Experimental results

5.1 Experimental results for Watertight track

We propose three runs for the SHREC'07 Watertight models track:

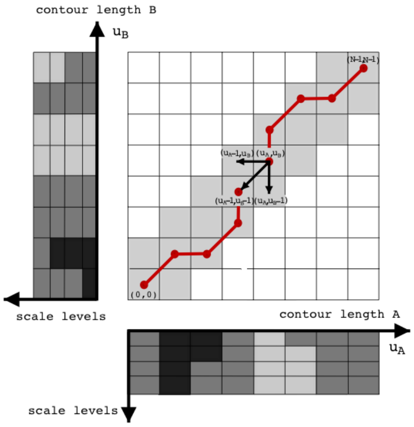
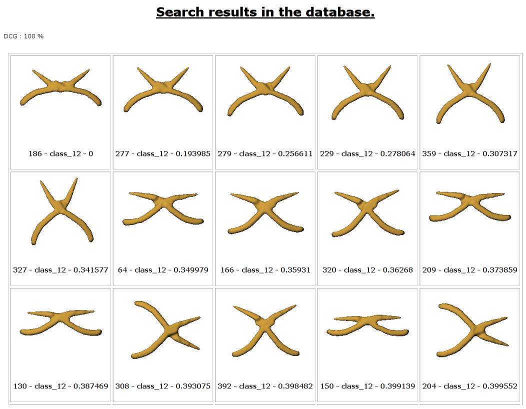


Figure 6. *Matching of two MCC representations by using dynamic programming.*

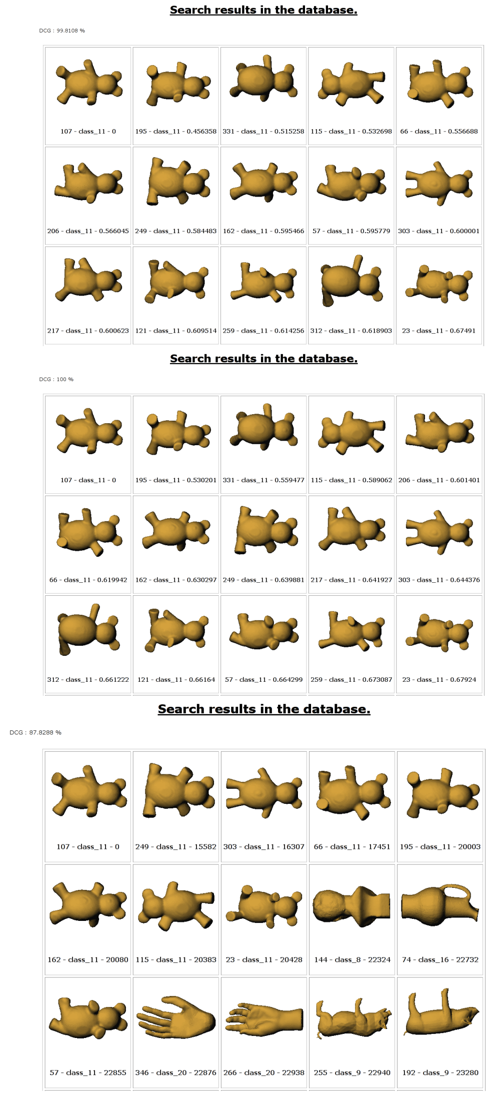


Search results in the database.

DCG : 100 %

186 - class_12 - 0	277 - class_12 - 0.193985	279 - class_12 - 0.256611	229 - class_12 - 0.278064	359 - class_12 - 0.307317
327 - class_12 - 0.341577	64 - class_12 - 0.349979	166 - class_12 - 0.35931	320 - class_12 - 0.36268	209 - class_12 - 0.373859
130 - class_12 - 0.387469	308 - class_12 - 0.393075	392 - class_12 - 0.398482	150 - class_12 - 0.399139	204 - class_12 - 0.399552

Figure 7. *15 first results for shape retrieval using the contour convexities and concavities method on the Watertight model database. The query 186.off is on the top left.*



Search results in the database.

DCG : 99.8108 %

107 - class_11 - 0	195 - class_11 - 0.456358	331 - class_11 - 0.515258	115 - class_11 - 0.532698	66 - class_11 - 0.556688
206 - class_11 - 0.566045	249 - class_11 - 0.584483	162 - class_11 - 0.595466	57 - class_11 - 0.595779	303 - class_11 - 0.600001
217 - class_11 - 0.600623	121 - class_11 - 0.609514	259 - class_11 - 0.614256	312 - class_11 - 0.618903	23 - class_11 - 0.67491

Search results in the database.

DCG : 100 %

107 - class_11 - 0	195 - class_11 - 0.530201	331 - class_11 - 0.559477	115 - class_11 - 0.589062	206 - class_11 - 0.601401
66 - class_11 - 0.619942	162 - class_11 - 0.630297	249 - class_11 - 0.639881	217 - class_11 - 0.641927	303 - class_11 - 0.644376
312 - class_11 - 0.661222	121 - class_11 - 0.66164	57 - class_11 - 0.664299	259 - class_11 - 0.673087	23 - class_11 - 0.67924

Search results in the database.

DCG : 87.8288 %

107 - class_11 - 0	249 - class_11 - 15582	303 - class_11 - 16307	66 - class_11 - 17451	195 - class_11 - 20003
162 - class_11 - 20080	115 - class_11 - 20383	23 - class_11 - 20428	144 - class_8 - 22324	74 - class_16 - 22732
57 - class_11 - 22855	346 - class_20 - 22876	266 - class_20 - 22938	255 - class_9 - 22940	192 - class_9 - 23280

Figure 8. *15 first results for the Watertight query 107.off. Top: run 1, Middle: run 2, Bottom: run 3.*

- Run 1: The contour convexities and concavities descriptor, with 3 silhouettes aligned with the principal axis and a resolution of 256x256 pixels for each silhouettes.
- Run 2: The contour convexities and concavities descriptor, with 3 silhouettes aligned with the principal axis and a resolution of 64x64 pixels for each silhouettes.
- Run 3: The multi-view intersection descriptor, with 3 silhouettes aligned with the principal axis and a resolution of 256x256 pixels for each silhouettes.

The contour convexities and concavities descriptor provides better results than the multi-view intersection descriptor, see Figure 8 (top and bottom). The two different image resolutions produce practically the same results, see Figure 8 (middle). We present in Table 1 the classwise DCGs for each run. We observe that run 1 and run 2 perform perfectly results for the *plier* class. For the contour convexities and concavities descriptor the worst DCGs are obtained for *spring* and *vase* classes. These two classes contains models with heterogeneous shapes. For the multi-view intersection descriptor the worst DCG is obtained for *armadillo* and *octopus*.

5.2 Experimental results for CAD track

We propose two runs for the SHREC'07 CAD models track:

- Run 1: The contour convexities and concavities de-

Class	DCG Run 1	DCG Run 2	DCG Run 3
airplane	92.9	93.6	66.5
ant	91	89.7	46.7
armadillo	82.4	80.3	46.3
bearing	86.1	80.1	80.8
bird	87.1	89.7	61.6
buste	87	85.9	66.3
chair	92	93.4	87.7
cup	77.2	77.1	73.7
fish	94.3	93.2	76.9
four leg	90.4	85.5	68.1
glasses	90.8	91	92.5
hand	81.8	77.2	56.6
human	82.8	77.4	68
mechanic	89.1	91	85.6
octopus	79.2	74.2	41.2
plier	100	100	97
spring	55.3	54.5	50.6
table	77.7	81.6	81.9
teddy	95.6	96	80
vase	52.2	53.7	56.3

Table 1. *Classwise DCGs performance for the three runs on the Watertight models.*

scriptor, with 3 silhouettes aligned with the principal axis and a resolution of 256x256 pixels for each silhouettes.

• Run 2: The multi-view intersection descriptor, with 3 silhouettes aligned with the principal axis and a resolution of 256x256 pixels for each silhouettes.

The contour convexities and concavities descriptor provides again better results than the multi-view intersection descriptor (see Figure 9). The first method is more robust against small variations of the shape and against the mirrored silhouettes problem. But the computation time per query model is very different between the two methods: with the contour convexities and concavities descriptor the CPU time per query is $\sim$ 20 s and with the multi-view intersection descriptor $\sim$ 0,06 s.

6 Conclusion

We have tested two different methods on the SHREC'07 contest. We observe that we obtain the best results with the convexities/concavities descriptor. We notice that the intersection method is not very robust with small deformations of the models. The contour convexities and concavities method needs much more computation time, but this weakness could be strongly reduced by optimizing the source code.

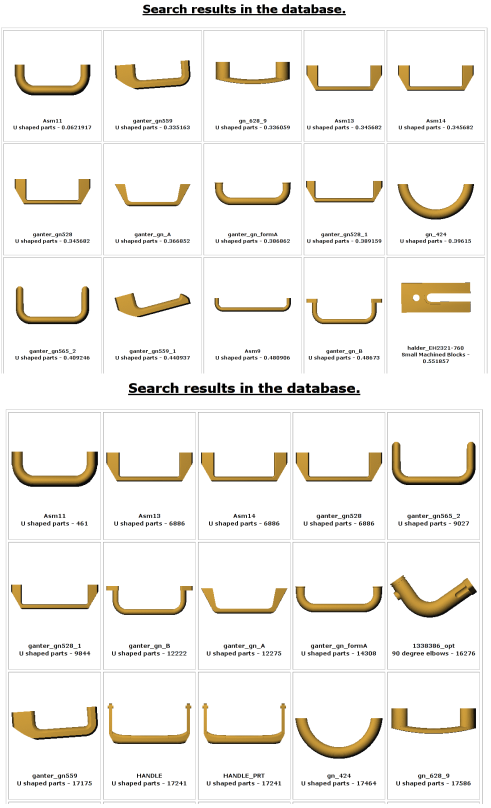


Figure 9. *15 first results for the CAD query 40.stl. Top: run 1, Bottom: run 2.*

7 Acknowledgments

This research was partly supported by the European Commission under contract FP6-027026-K-SPACE.

References

[1] T. Adamek and N. E. O'Connor. A multiscale representation method for nonrigid shapes with a single closed contour. *IEEE Trans. Circuits Syst. Video Techn*, 14(5):742–753, 2004.

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

[3] K. Fischer and B. Gartner. The smallest enclosing ball of balls: Combinatorial structure and algorithms. *International Journal of Computational Geometry and Applications (IJCGA)*, 14:341–387, 2004.

[4] D. V. Vranic. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[5] D. V. Vranic, D. Saupe, and J. Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In J.-L. Dugelay and K. Rose, editors, *Proceedings of the IEEE 2001 Workshop Multimedia Signal Processing*, pages 271–274, Budapest, Hungary, Sept. 2001.

The Spherical Trace Transform

P. Daras, D. Tzovaras, A. Axenopoulos, D. Zarpalas, A. Mademlis and M. G. Strintzis
Informatics & Telematics Institute
1st km Thermi-Panorama Rd
57001 (PO Box 60361)
daras@iti.gr

Abstract

The method used for the CAD track, the watertight models track and the one method of proteins track, is based on the Spherical Trace Transform. After the proper positioning of the 3D object using translation and scaling normalization techniques, the object is fully decomposed into a set of planes tangential to a set of concentric spheres. In this new domain, a new set of functionals is applied, resulting in a descriptor vector which is completely rotation invariant, and thus suitable for 3D model matching. The matching is based on Minkowski L1 Distance.

1. Introduction

The large amount of the available 3D models and their increasingly important role for 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 that 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. The method is briefly presented in the sequel. A detailed description of the method can be found in [1] for 3D models, and in [2] for protein classification.

2. Watertight and CAD track

2.1. Descriptor Extraction Method

Every 3D object is expressed in terms of a binary volumetric function. In order to achieve translation invariance, the center of mass of the 3D object is calculated and the model is translated so that its center of mass coincides with the coordinates system origin. Scaling invariance is also accomplished, by scaling the object in order to fit inside the unit sphere. Then, a set of concentric spheres is defined. For every sphere, a set of planes which are tangential to the sphere is also defined. Further, the intersection of each plane with the object's volume provides a spline of the object, which can be treated as a 2D image. Next, 2D rotation invariant functionals F are applied to this 2D image, producing a single value. Thus, the result of these functionals when applied to all splines, is a set functions defined on every sphere whose range is the results of the functional. Finally, a rotation invariant transform T is applied on these functions, in order to produce rotation invariant descriptors. For the needs of the SHREC, the implemented functionals F are the 2D Krawtchouk moments, the 2D Zernike Moments and the Polar Fourier Transform, while the T function is the Spherical Fourier Transform.

2.2. Matching

Firstly, the descriptors are normalized so that their absolute sum is equal to 1. Then, the matching is based on the Minkowski L_1 distance.

Acknowledgement

This work is supported by the VICTORY EC project (contract no: 044985) and by PENED project, co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development - GSRT (25%).

References

[1] D.Zarpalas, P.Daras, A.Axenopoulos, D.Tzovaras, and M.G.Strintzis: "3D Model Search and Retrieval Using the Spherical Trace Transform", EURASIP Journal on Advances in Signal Processing Volume 2007

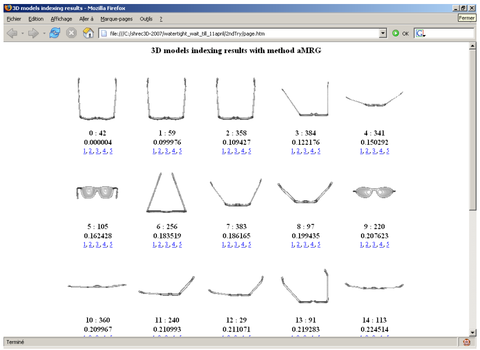
[2] Petros Daras, Dimitrios Zarpalas, Apostolos Axenopoulos, Dimitrios Tzovaras and Michael Gerassimos Strintzis. "Three-Dimensional Shape-Structure Comparison Method for Protein Classification". IEEE/ACM Transactions on Computational Biology and Bioinformatics July 2006 Vol 3 issue 3 pp 193-207.

Shape retrieval of watertight models and CAD models using aMRG

Tony Tung   Francis Schmitt
Telecom Paris, CNRS UMR 5141, France
{tony.tung, francis.schmitt}@enst.fr

Abstract

This paper presents an application of the augmented Multiresolution Reeb graph (aMRG) [3] for shape retrieval of watertight 3D models and CAD models. The method is based on a Reeb graph construction which is a well-known topology based shape descriptor. Using multiresolution property and additive geometrical and topological informations, aMRG has shown its efficiency to retrieve high quality 3D models [4]. The SHREC - SHape REtrieval Contest 2007 Watertight Models track data collection is composed of 20 classes of 20 models (sunglasses, humans, ants, etc.). Most classes contain deformed models with same topology. The Engineering track proposed by Purdue is composed of 3D CAD models. The models have high genus, rounding features (fillets, chamfers), and presence of internal structure. We propose to evaluate the performance of the aMRG with new embedded topological features [5] to retrieve the different classes of both data collections. Our experiments show interesting results. We give comments on the retrieval performed by aMRG compared to the ground truth classification. Our approach obtained the best results at the contest on the Watertight Models track.


3D models indexing results - Mozilla Firefox
Fichier
Edition
Affichage
Aller à
Marque-pages
Outils
?
Fermer
file:///C:/shrec3D-2007/watertight_wait_till_11april/2ndTry/page.htm
OK
Terminé

3D models indexing results with method aMRG

0 : 42 0.000004 1, 2, 3, 4, 5	1 : 59 0.099976 1, 2, 3, 4, 5	2 : 358 0.109427 1, 2, 3, 4, 5	3 : 384 0.122176 1, 2, 3, 4, 5	4 : 341 0.150292 1, 2, 3, 4, 5
5 : 105 0.162428 1, 2, 3, 4, 5	6 : 256 0.183519 1, 2, 3, 4, 5	7 : 383 0.186165 1, 2, 3, 4, 5	8 : 97 0.199435 1, 2, 3, 4, 5	9 : 220 0.207623 1, 2, 3, 4, 5
10 : 360 0.209967	11 : 240 0.210993	12 : 29 0.211071	13 : 91 0.219283	14 : 113 0.224514

Figure 1. aMRG is powerful to retrieve similar 3D models with various deformations and same topology. Here, the table presents the most similar models to a query. The query is the model on top-left. Distance to query is shown under each compared models. Similar sunglasses can be retrieved, even strongly deformed.

1 Introduction

Our paper proposes a scheme to retrieve similar shapes in collections of watertight 3D models and CAD models using a Reeb graph based approach. In the framework of the SHREC - SHape REtrieval Contest 2007 Watertight Models track, the data collection consists on 400 watertight models (20 classes of 20 similar models). As most classes are composed of deformed objects with same topology, the Reeb graph suits well to describe their shape. In the framework of the Engineering track, the data collection is composed of 866 CAD models. The models are watertight and have particular topology (high genus). Therefore it is interesting to study the Reeb graph performance for describing their shape as well. The graph is built using a Morse function μ based on the mesh connectivity. The surface of the object is divided in regions according to the values of μ, and a node is associated to each region. The graph structure is then obtained by linking the nodes of the connected regions. Then a multiresolutional Reeb graph can be constructed hierarchically, based on a coarse-to-fine approach node merging [1]. Keeping advantage of the multiresolutional representation, the augmented multiresolution Reeb graph (aMRG) [3] is an enhanced Reeb graph which embeds topological, geometrical and visual (color or texture) information in each graph node. Afterwards similarity between two aMRGs can be computed to retrieve the most similar nodes. Its efficiency has been tested on high quality 3D model databases acquired from art museums [4]. Recently, the aMRG was adapted for full topology matching of human models in 3D video sequence [5]. We propose to evaluate the performance of the aMRG with the new topological features to retrieve the data collection classes.

As shown in Figure 1, the method is able to retrieve similar 3D models. Three different runs are proposed fr the Watertight Models track, and four different runs are proposed for the Engineering track of the SHREC 2007.

The next section presents an overview of the aMRG. Section 3 presents our experimental results on the watertight models track. Section 4 gives comments on the Watertight Models track contest results. Section 5 presents our experimental results on the Engineering track. Section 6 gives comments on the Engineering track contest results.

2 Overview of the aMRG

According to the Morse theory, a continuous function defined on a closed surface characterizes the topology of the surface on its critical points. Therefore, a Reeb graph can be obtained assuming a continuous function μ calculated over the 3D object surface.

In both tracks, models are defined by their surface and represented as 3D triangular meshes with vertices located in a Cartesian frame. We chose the function μ proposed in [1], which is defined as the integral of the geodesic distance $g(\mathbf{v}, \mathbf{p})$ from $\mathbf{v}$ to the other points $\mathbf{p}$ of the surface:

$$\mu(\mathbf{v}) = \int_{\mathbf{p} \in S} g(\mathbf{v}, \mathbf{p}) dS. \tag{1}$$

This function μ has the property to be invariant to rotations. Its integral formulation provides a good stability to local noise on surface and gives a measure of the eccentricity of the object surface points. A point with a great value of μ is far from the center of the object. A point with a minimal value of μ is close to the center of the object. Therefore μ is normalized to $\mu_N = \frac{\mu_{max} - \mu_{min}}{\mu_{max}}$, so that values of μ keep an information on the distance to the object center. The corresponding Reeb graph is then obtained by iteratively partitioning the object surface into regular intervals of μ_N values and by linking connected regions. For each interval, a node is associated to each different set of connected triangles.

To construct a Reeb graph of R levels of resolution, μ_N is subdivided into 2^R intervals from which the object surface is partitioned at the highest level of resolution. Afterwards, using a hierarchical procedure, Reeb graphs of lower resolution levels are obtained by merging intervals by pairs [1]. The multiresolutional aspect results from the dichotomic discretization of the function values and from the hierarchical collection of Reeb graphs defined at each resolution. Topology is more precisely recovered at higher resolutions (cf. Figure 2).

The original approach, mainly based on the 3D object topology, is not accurate enough to obtain satisfying matching. Therefore in [3, 4], the multiresolution Reeb graph has been augmented by merging global and local geometric properties and visual properties extracted from the object surface region $S(m)$ associated to each node m. Topological aspects of the graph matching procedure was extended, and similarity calculation with the new features was adapted. The different features are:

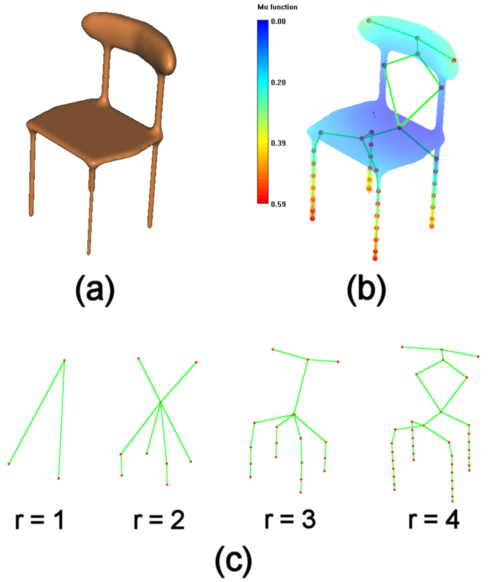


Figure 2. a) 3D model. b) Values of μ function on the model surface and Reeb graph at resolution $r = 4$. c) Reeb graphs at resolution $r = 1, 2, 3$ and 4. Topology is more precisely recovered at higher resolutions.

- the relative volume of $S(m)$,
- a statistic measure of the extent of $S(m)$,
- a statistic of the Koenderink shape index for local curvature estimation on $S(m)$,
- a statistic of the orientation of the triangle normals associated to $S(m)$,
- a statistic of the texture/color mapped on $S(m)$ (not used here).

The choice of these attributes has been guided by the litterature on 3D content-based retrieval [2]. The result is a flexible multiresolution and multicriteria 3D shape descriptor including merged topological, geometrical and colorimetric properties.

In order to obtain a better control of the node matching, graph topology was exploited in [5]. Topological features can be deduced by the edge orientations given by μ values. In addition, multiresolution gives valuable information to characterize the global shape of models. The shape description is intuitive and completely topological. Its efficiency has been tested as described in the next section.

3 Experiments on Watertight Models track

aMRG calculations were performed on a laptop with Pentium(R) M processor 1.60 GHz and RAM 512 Mo. aMRG were computed with resolution up to $r = 5$. The computation time depends on the number of vertices, and the complexity of the model. For example, aMRG of watertight model 1.off which contains 15000 vertices was computed in ~ 20 s. And aMRG of model 3.off which contains 6833 vertices was computed in ~ 12 s.

The calculation of the function μ remains the most time consuming, even with the Dijkstra coding scheme of $O(NlogN)$ complexity on N vertices. Our experiments have pointed out the importance of the choice of the function μ. The invariance to rotation properties was necessary for the watertight model database.

For the SHREC 2007 Watertight Models track, three runs were proposed with different aMRG resolution:

- Run1: similarity are computed till resolution $r = 3$
- Run2: similarity are computed till resolution $r = 4$
- Run3: similarity are computed till resolution $r = 5$

Results are slightly similar. The main difference is the computation time which is a little bit longer for higher resolutions. The matching procedure based only on topological information is well adapted for non oriented object. Using geometrical information can sharpen the retrieval in some case. Computation time for retrieval of a query on the watertight model database of 400 models at $r = 5$ is ~ 6 s. Computation time for a query at $r = 3$ is ~ 3 s.

4 Comments on Watertight Models track contest results

Our approach obtained the best results amongst the contest participants at the SHREC 2007 Watertight Models track (cf. Figure 3). The data collection is composed of classes with quite heterogeneous retrieval difficulty. Some classes contain some different models that can lead to bad retrieval performances. Looking at the contest results, we can observe that aMRG is very robust to shape deformation. In general, aMRG returns unsatisfactory retrieval results only on some very rare queries on the watertight models data collection.

Our method performed far better retrievals than other methods on the classes: amardillo (same model strongly deformed), octopus, spectacles and springs (models with remarkable topology). We got a perfect retrieval on the teddybear class. Results were very good on the classes: ants (remarkable topology), fishes, blends, cups and pliers as well. Results were good on the classes: aircrafts, chairs, heads, and vases. Results were average compared to other methods on the classes: birds, bearings, four limbs, hands, humans, and tables.

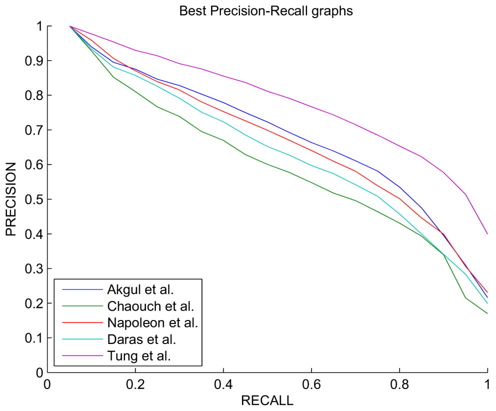


Figure 3. Watertight Models Track of SHape REtrieval Contest 2007 Precision-Recall results. aMRG performed better than other methods.

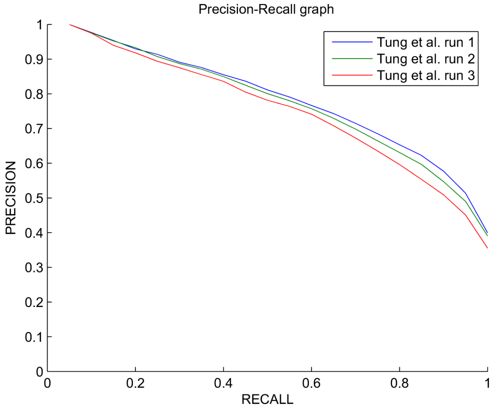


Figure 4. Run 1 performed slightly better than Run 2 and Run 3.

In our matching procedure we use several weighted parameters that are experimentally determined. If some parameters are not well set, then we can obtained some mismatch like a human with a four limb (which have the same topology). Therefore we believe parameters can be optimized to refine the similarity calculation. Moreover, as a matter of fact, Run 1 (aMRG at resolution $r = 3$) performed better than Run 2 and Run 3 (corresponding to $r = 4$ and $r = 5$ respectively) as can be seen on Figure 4. Using higher resolutions give more precision in the retrieval process and similarity calculation but can introduce noises: fine details on model surface can be matched as well and can increase the similarity score whereas it should not. This can be enhanced by filtering the graphs before the matching procedure.

5 Experiments on Engineering track

aMRG calculations were performed on a laptop with Pentium(R) M processor 1.60 GHz and RAM 512 Mo. aMRG were computed with resolution up to $r = 5$. The computation time depends on the number of vertices, and the complexity of the model. For example, aMRG of CAD model bat_gear2.stl which size is 124 K was computed in $\sim$ 2 s. And aMRG of CAD model 72t_12d_05w.stl which size is 731 K was computed in $\sim$ 5 s.

The calculation of the function μ remains the most time consuming, even with the Dijkstra coding scheme of $O(NlogN)$ complexity on N vertices. Our experiments have pointed out the importance of the choice of the function μ. The invariance to rotation properties was necessary for the Purdue CAD model database.

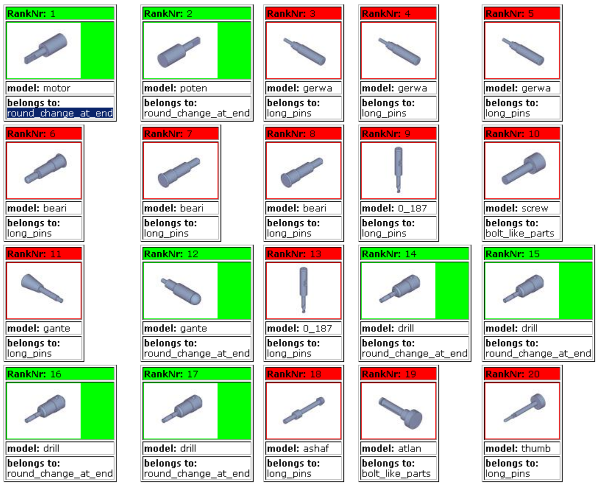
For the SHREC3D CAD models 2007, four runs were proposed with different parameters:

- Run1: similarity are computed till resolution $r = 4$, and aMRG matching contains only topological information.
- Run2: similarity are computed till resolution $r = 3$, and aMRG matching contains only topological information.
- Run3: similarity are computed till resolution $r = 4$, and aMRG matching contains geometrical and topological information.
- Run4: similarity are computed till resolution $r = 3$, and aMRG matching contains geometrical and topological information.

Results are slightly similar. The main difference is the computation time which is a little bit longer for higher resolution. The matching procedure based only on topological information is well adapted for non oriented object. Using geometrical information can sharpen the retrieval in some case. Computation time for retrieval of query 40 on the CAD database of 866 models at $r = 4$ is $\sim$ 5 s. Computation time for the same query at $r = 3$ is $\sim$ 3 s.

6 Comments on Engineering track contest results

aMRG is based on a coarse-to-fine approach. Hence we care about global shape as well as local details. In particular, model topology has a strong impact on the shape description. Therefore for example, a CAD model with 3 holes will not be described as similar as a model with 5 holes. This could obviously be interpreted as a limitation of our approach, as well as a strength. In deed the ground truth classification reveals that the similarity concept of the SHREC for CAD models 2007 data collection is not defined exactly as for our aMRG approach. For example some models from classes *round_change_at_end* and *long_pins* are retrieved as similar by aMRG, whereas these classes were defined as different by the ground truth classification (cf. Figures 5 and 6). And similarly, some models from classes *simple_pipes* and *90_degree_elbows* are retrieved together.



RankNr: 1
model: motor
belongs to: round_change_at_end

RankNr: 2
model: poten
belongs to: round_change_at_end

RankNr: 3
model: gerwa
belongs to: long_pins

RankNr: 4
model: gerwa
belongs to: long_pins

RankNr: 5
model: gerwa
belongs to: long_pins

RankNr: 6
model: beari
belongs to: long_pins

RankNr: 7
model: beari
belongs to: long_pins

RankNr: 8
model: beari
belongs to: long_pins

RankNr: 9
model: 0_187
belongs to: long_pins

RankNr: 10
model: screw
belongs to: bolt_like_parts

RankNr: 11
model: gante
belongs to: long_pins

RankNr: 12
model: gante
belongs to: round_change_at_end

RankNr: 13
model: 0_187
belongs to: long_pins

RankNr: 14
model: drill
belongs to: round_change_at_end

RankNr: 15
model: drill
belongs to: round_change_at_end

RankNr: 16
model: drill
belongs to: round_change_at_end

RankNr: 17
model: drill
belongs to: round_change_at_end

RankNr: 18
model: ashaf
belongs to: long_pins

RankNr: 19
model: atlan
belongs to: bolt_like_parts

RankNr: 20
model: thumb
belongs to: long_pins

Figure 5. Results of Query 7. Models from classes *round_change_at_end* and *long_pins* are retrieved as similar by aMRG.

Besides aMRG obtains good retrieval performance on the *simple_pipes* class (cf. Figure 12 in Appendix). In the other hand, shape models of the class *90_degree_elbows* seems to be too heterogeneous to be well retrieved by our approach (cf. Figure 7). More generally, we have observed that aMRG performances on the CAD data collection depend on the query type: for example, moderate for *90_degree_elbows* class, and very good for *u_shaped_parts* class (cf. Figures 12 and 13 in Appendix). For further investigation, it should be worthwhile to optimize the weight of the different parameters embedded in the aMRG nodes. In addition, the matching process could be extended to Reeb

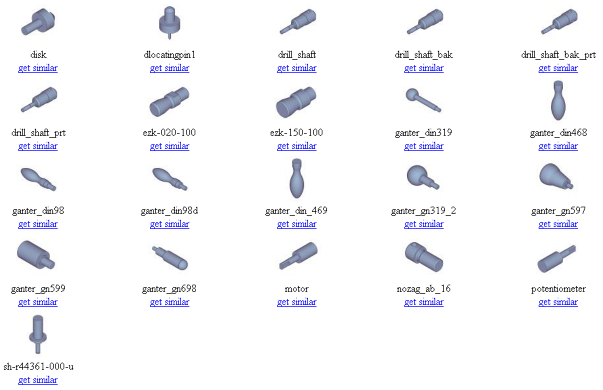


Figure 6. Ground truth classification of the class *round_change_at_end***.**

graphs with different topology. Thus, the retrieval of CAD models with different number of holes would be improved.

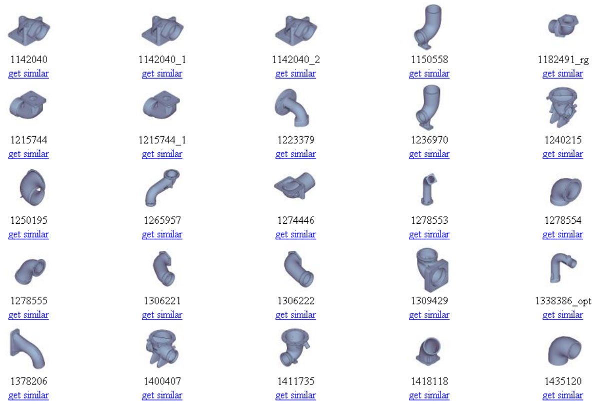


Figure 7. The class *90_degree_elbows* **contains models with heterogeneous shape.**

7 Conclusion

We have presented an application of the augmented Multiresolution Reeb graph (aMRG) for the SHape REtrieval Contest 2007 (SHREC) for Watertight Models and for CAD models (Engineering track). aMRG is a topology based shape descriptor. The method uses a multiresolution scheme for the graph matching process. Moreover additional topological and geometrical features are used to improve the matching. The Watertight Models collection models have remarkable topology. We obtained interesting results with our approach. A full topological approach was used for the graph matching procedure and three runs at different aMRG resolution were proposed. We obtained the best results amongst the contest participants. In general, the approach is efficient to retrieve classes of deformed models, especially with same topology. It is more difficult to retrieve a class of objects with different shapes (e.g. glasses and mugs are not considered as similar by our approach). As lots of CAD models of the data collection have complex shape, as well it was interesting to evaluate the aMRG performance. In particular, we compare the aMRG retrieval results with the ground truth classification. Four runs with different parameters were proposed. We have observed that some classes were well retrieved by aMRG (e.g. *simple_pipes*, *u_shaped_parts*), whereas some classes were not adapted to our approach (*90_degree_elbows*). Model topology plays a major role in our matching process, as well as global shape. Hence CAD models from a same class with different numbers of holes, or with significative shape difference are not considered as similar by the aMRG method. For further evaluations, the weights of the different embedded attributes could be optimized to improve the similarity scores.

References

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

[2] J. Tangelder and R.C.Veltkamp. A survey of content based 3d shape retrieval methods. *IEEE International Conference on Shape Modeling and Applications (SMI)*, pages 145–156, 2004.

[3] T. Tung and F. Schmitt. 3d shape matching using reeb graph. *IEEE International Conference on Shape Modeling and Applications (SMI)*, pages 157–166, 2004.

[4] T. Tung and F. Schmitt. The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes. *International Journal of Shape Modeling (IJSM)*, pages 91–120, 2005.

[5] T. Tung, F. Schmitt, and T.Matsuyama. Topology matching for 3d video compression. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

Appendix 1

Figure 8 presents a query results on the amardillo model. aMRG resolution is 3. Similar models are retrieved, even with strong deformations. Figure 9 presents a query results on an ant model. aMRG resolution is 3. The retrieval is very efficient thanks to the model topology. Figure 10 presents a query results on a chair model. aMRG resolution is 4. Chairs are well retrieved. Figure 11 presents a query results on a teddybear model. aMRG resolution is 4. Teddybears are very well retrieved, even models with missing arms or legs.

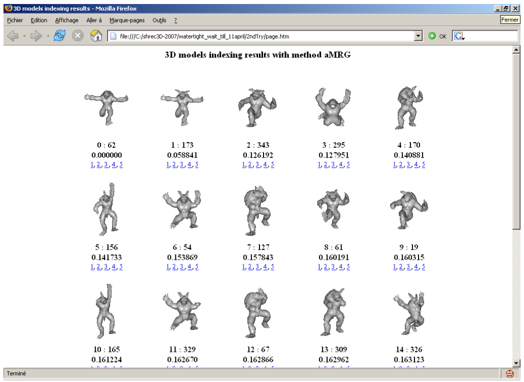


Figure 8. Query 62 with run 1. aMRG resolution is 3. Similar models are retrieved, even with strong deformations.

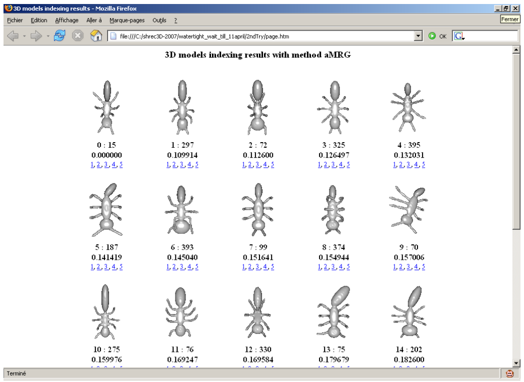


Figure 9. Query 15 with run 1. aMRG resolution is 3. Ants models from the database are very well retrieved due to their remarkable topology.

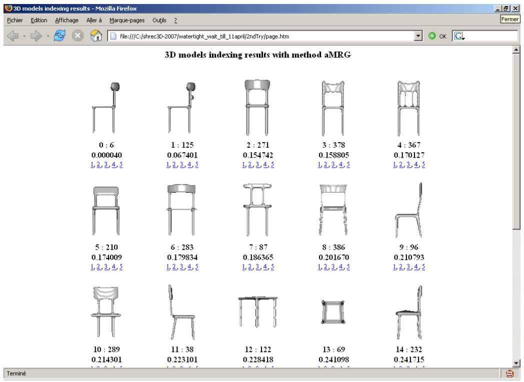


Figure 10. Query 6 with run 2. aMRG resolution is 4. Chairs are well retrieved.

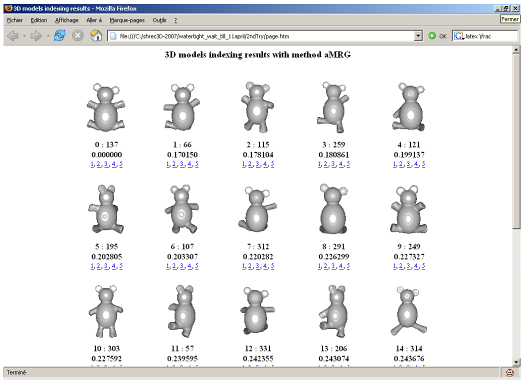


Figure 11. Query 137 with run 2. aMRG resolution is 4. Teddybears very are well retrieved, even models with missing arms or legs.

Appendix 2

Figure 12 presents a query results on a *simple_pipes* model with run 4. aMRG resolution is 3, and the graph matching approach contains geometrical and topological features. The retrieval contains model from the correct class and some models from class *90_degree_elbows* having similar shape.

Figure 13 presents a query results on a *u_shaped_parts* model with run 1. aMRG resolution is 4, and the graph matching approach is fully topologic. The class is very well retrieved.

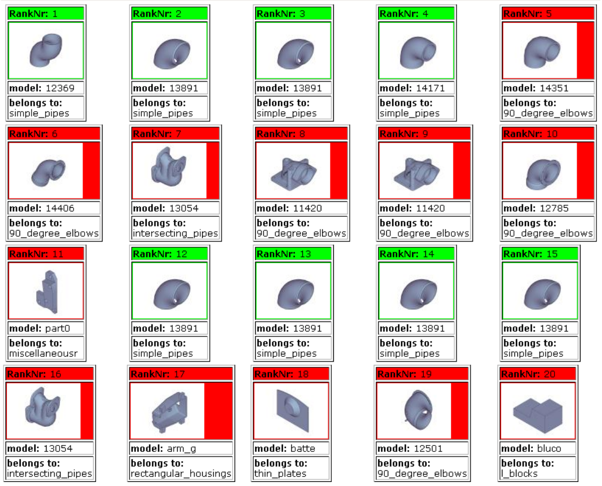

RankNr: 1 model: 12369 belongs to: simple_pipes
RankNr: 2 model: 13891 belongs to: simple_pipes
RankNr: 3 model: 13891 belongs to: simple_pipes
RankNr: 4 model: 14171 belongs to: simple_pipes
RankNr: 5 model: 14351 belongs to: 90_degree_elbows
RankNr: 6 model: 14406 belongs to: 90_degree_elbows
RankNr: 7 model: 13054 belongs to: intersecting_pipes
RankNr: 8 model: 11420 belongs to: 90_degree_elbows
RankNr: 9 model: 11420 belongs to: 90_degree_elbows
RankNr: 10 model: 12785 belongs to: 90_degree_elbows
RankNr: 11 model: part0 belongs to: miscellaneousr
RankNr: 12 model: 13891 belongs to: simple_pipes
RankNr: 13 model: 13891 belongs to: simple_pipes
RankNr: 14 model: 13891 belongs to: simple_pipes
RankNr: 15 model: 13891 belongs to: simple_pipes
RankNr: 16 model: 13054 belongs to: intersecting_pipes
RankNr: 17 model: arm_g belongs to: rectangular_housings
RankNr: 18 model: batte belongs to: thin_plates
RankNr: 19 model: 12501 belongs to: 90_degree_elbows
RankNr: 20 model: bluco belongs to: l_blocks

Figure 12. Query 1 from *simple_pipes* class with run 4. aMRG resolution is 3, and the graph matching approach contains geometrical and topological features. The retrieval contains model from the correct class and some models from class *90_degree_elbows* having similar shape.

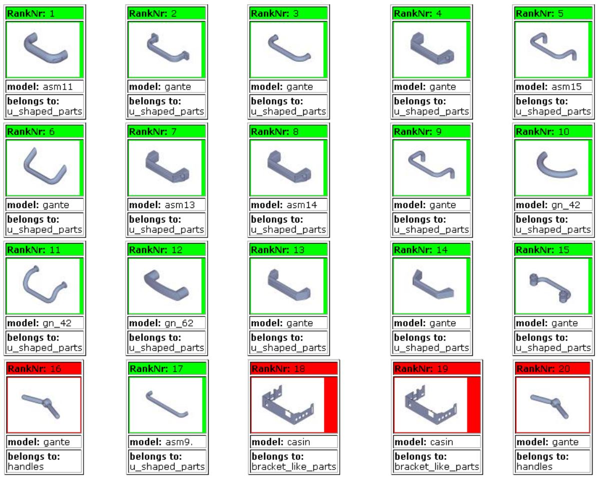

RankNr: 1 model: asm11 belongs to: u_shaped_parts
RankNr: 2 model: gante belongs to: u_shaped_parts
RankNr: 3 model: gante belongs to: u_shaped_parts
RankNr: 4 model: gante belongs to: u_shaped_parts
RankNr: 5 model: asm15 belongs to: u_shaped_parts
RankNr: 6 model: gante belongs to: u_shaped_parts
RankNr: 7 model: asm13 belongs to: u_shaped_parts
RankNr: 8 model: asm14 belongs to: u_shaped_parts
RankNr: 9 model: gante belongs to: u_shaped_parts
RankNr: 10 model: gn_42 belongs to: u_shaped_parts
RankNr: 11 model: gn_42 belongs to: u_shaped_parts
RankNr: 12 model: gn_62 belongs to: u_shaped_parts
RankNr: 13 model: gante belongs to: u_shaped_parts
RankNr: 14 model: gante belongs to: u_shaped_parts
RankNr: 15 model: gante belongs to: u_shaped_parts
RankNr: 16 model: gante belongs to: handles
RankNr: 17 model: asm9. belongs to: u_shaped_parts
RankNr: 18 model: casin belongs to: bracket_like_parts
RankNr: 19 model: casin belongs to: bracket_like_parts
RankNr: 20 model: gante belongs to: handles

Figure 13. Query 40 from *u_shaped_parts* class with run 1. aMRG resolution is 4, and the graph matching approach is fully topologic. The class is very well retrieved.

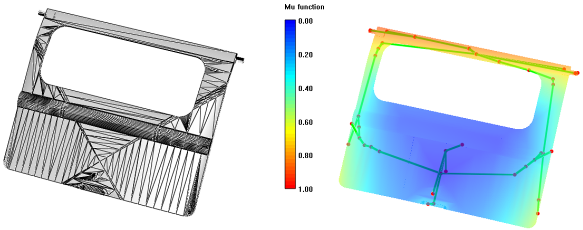


Figure 14. Left: 3D mesh model of *backdoor*. Right: values of function μ on the surface, with Reeb graphs at resolution $r = 4$. The graph structure contains topological and geometrical information.

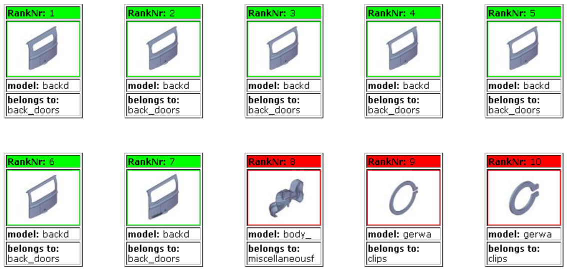

RankNr: 1 model: backd belongs to: back_doors
RankNr: 2 model: backd belongs to: back_doors
RankNr: 3 model: backd belongs to: back_doors
RankNr: 4 model: backd belongs to: back_doors
RankNr: 5 model: backd belongs to: back_doors
RankNr: 6 model: backd belongs to: back_doors
RankNr: 7 model: backd belongs to: back_doors
RankNr: 8 model: body_ belongs to: miscellaneousf
RankNr: 9 model: gerwa belongs to: clips
RankNr: 10 model: gerwa belongs to: clips

Figure 15. Good retrieval of *backdoor* using aMRG.

A database-adaptive distance measure for 3D model retrieval

Jun Kobayashi, Akihiro Yamamoto, Toshiya Shimizu, Ryutarou Ohbuchi
Graduate School, University of Yamanashi.
jun066 AT quartz.ocn.ne.jp, g05mk039 AT yamanashi.ac.jp,
g06mk009 AT Yamanashi.ac.jp, ohbuchi AT yamanashi.ac.jp

Abstract

The distance measure, in addition to the shape feature is a key factor in shape based 3D model retrieval. We employed a database adaptive distance measure for 3D model retrieval in the SHREC 2007 CAD track. The method described in this paper uses a feature dimension reduction based on an unsupervised learning of features to produce salient, lower dimensional feature from the original feature. Our method also combines a multiresolution shape comparison approach with the database adaptive distance measure. Our experiments the SHREC 2007 CAD track showed that both adaptive distance measure and multiresolution shape comparison approach added a few percent each to the original shape feature. Our method came in first in the SHREC 2007 CAD track despite the fact that the distance measure was trained by using a set of "generic" 3D models that are not CAD specific.

1. Introduction

Two of the most important components in a hape-based 3D model retrieval system [13] are the shape feature and the distance measure used. Intuitively, shape similarity decision would depend on shapes of models in the database to be compared, or a shape of the specific model to be queried. For example, a feature might excel at retrieving a class of models but not for the others classes. The shape similarity decision would also have inter-user variations as well intra-user variation depending on intention of the specific query. Despite these possible variations, most of the previous method uses a fixed feature and a fixed distance measure. Several researchers experimented with relevance feedback for on-line, interactive adaptation of distance measure [6] while the other tried query-adaptive combination of distances generated from multiple features [2].

Xhaofei He et al proposed, in the context of 2D image retrieval, a distance measure that *adapts to the database* of 2D images by using unsupervised learning. The method learns the non-linear subspace, or manifold, of features from large number of (unlabeled) 2D images. The dimension of an input feature is reduced by projecting the feature onto the manifold. A distance among a pair of dimension reduced and "salient" features corresponds to a geodesic distance on the manifold. He et al used the *Laplacian Eigenmaps* (LE) proposed by Belkin, et al [1] to learn the non-linear manifold. As the manifold generated by LE is defined only at the training samples, the manifold is smoothly approximated to handle queries not in the training set.

We adopted He's approach for 3D model retrieval in [9]. We applied the dimension reduction using the LE algorithm on two shape features. For both of the features, the dimension reduced feature overtook the original, untrained feature in retrieval performance when the number of training samples is more than about 1,500. For the unsupervised learning of feature manifold, we (quasi-) randomly sampled a set of models from the union of 1,814 model *Princeton Shape Benchmark* database [12] and 10,911 model *National Taiwan University* database [7].

Later, we further explored the approach by experimenting with six learning-based dimension reduction algorithms, both linear and non-linear [10]. These six dimension reduction algorithms are paired with eight shape features in a set of experiments. Of the dimension reduction methods we compared, the locally constrained, non-linear methods such as LE and *Locally Linear Embedding* (LLE) [11] performed the best. The best performing pair among those tested performed on a par with the top finisher in the SHREC 2006 contest, that is, Makadia's method.

To enter the SHREC 2007 CAD track, we used the same approach as described in [9, 10]. We used the set of "generic" 3D models as in [9, 10], not CAD specific 3D models, for learning the manifold. Our method came in first in every performance index for the contest even though the training samples are not CAD model specific.

This paper is organized as follows. We will describe in Section 2, the algorithm we employed. It is followed in Section 3 by the experimental results we have done using the SHREC 2007 CAD track benchmark. Section 3 will present the summary and future work.

2. The retrieval method

Our retrieval algorithm uses data-driven unsupervised learning to find a non-linear subspace of shape features for dimension reduction and then distance computation (See Figure 1);

Learning phase:

1. **Feature extraction:** Extract n-dimensional feature vectors from the K models in the training database (i.e., corpus).
2. **Sample selection:** If necessary, to reduce computational costs, sub-sample the training set down to L ($L \le K$) features.
3. **Manifold learning:** Perform unsupervised learning of the m-manifold ($m \le n$) from the n-dimensional training samples by using a manifold learning algorithm. Certain learning algorithm produces a manifold defined only at the set of training samples. In such a case, to handle queries outside of the training set, continuously approximate the manifold.

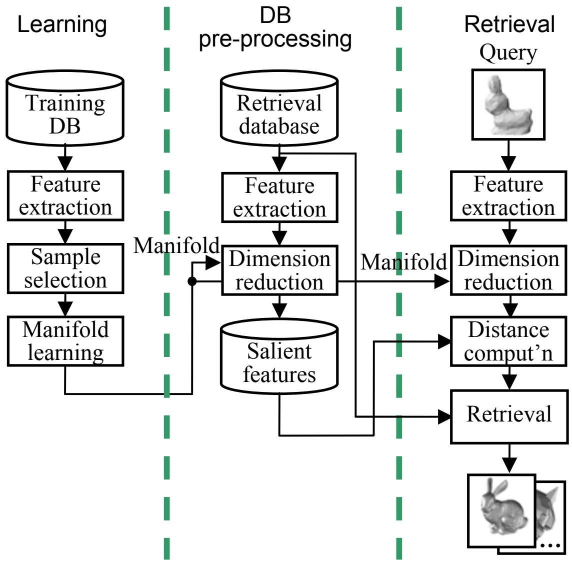


Figure 1. Our 3D model retrieval algorithm using database-adaptive distance measure.

Database pre-processing phase:

1. **Dimension reduction:** Project the feature of each 3D model in the database onto the (approximated) m-manifold to obtain an m-dimensional "*salient*" feature. Store each salient feature together with a corresponding 3D model.

Retrieval phase:

1. **Feature extraction:** Extract an n-dimensional feature from the query model.
2. **Dimension reduction:** Project the n-dimensional feature of the query onto the (approximated) m-manifold to obtain m-dimensional salient feature.
3. **Distance computation:** Compute distances from the query model to all the models in the database using their m-dimensional salient features using *cosine* distance.
4. **Retrieval:** Retrieve the models in the database having the p-smallest distances from the query.

To choose the combination of feature and dimension reduction method for the SHREC 2007 CAD track, we chose six dimension reduction algorithms including PCA, Kernel PCA, LE, and LLE [10]. We coupled these with the eight shape features including the SPRH and Spherical Harmonics (SH) [5]. Note that all the features we compared are purely geometric and applicable to polygon soup models as well as to watertight surface based models. Topological feature such as position and/or number of through holes is not used. The distance between a pair of salient features is computed by using Cosine distance. We chose the Cosine distance after comparing *L1*-norm, *L2*-norm, and Cosine distance for their retrieval performance.

For the training, we would have two major alternatives; (1) use CAD specific 3D models, e.g., that of the *Purdue University Engineering Shape Benchmark (PUESB)*, (2) use "generic" 3D models found, for example, the *National Taiwan University 3D Model Database* (NTUD) [7].

What we used for the SHREC 2007 CAD track is a set of "generic" 3D models, that is, the union of the training set of the *Princeton Shape Benchmark* (PSB) database containing 907 models and the *National Taiwan University 3D Model Database* (NTUD) ver. 1.0 containing 10,911 models. The NTU database does not have any labels. The labels in the PSB training set are simply ignored. By using a quasi-random sequence (*Niederreiter* sequence), we sub-sampled the union (12,775 models) down to 4,000, 5,000, or 10,000 models for the training.

Some of the learning algorithms have parameters, e.g., number of output dimensions, neighborhoods size k for manifold reconstruction, and spreads of RBF

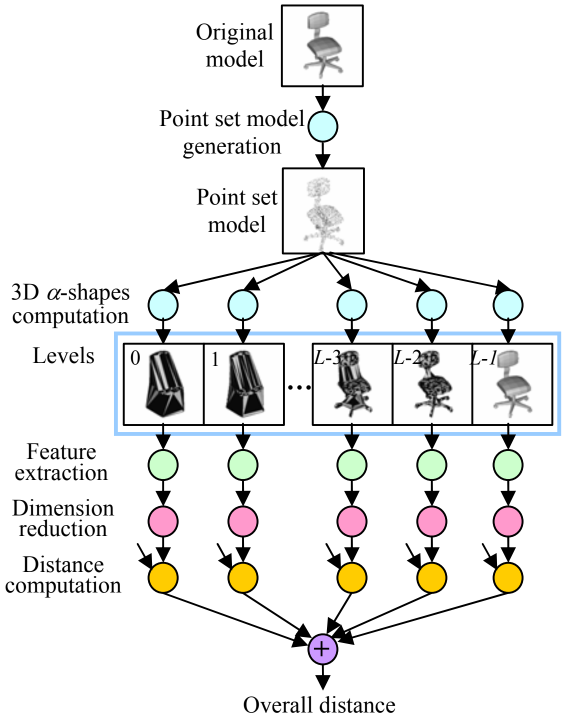


Figure 2. The multiresolution set of features is dimension reduced and distance computed separately at each resolution levels before integrated into an overall distance.

kernels for manifold approximation. We chose, through preliminary experiments using SHREC 2006 benchmark [14], a set of best-performing parameters.

We integrated the multiresolution 3D shape comparison approach we have previously proposed [8] with the learning based approach described above (See Figure 2.) The multiresolution shape comparison approach uses a mathematical morphology-like multiresolution (MR-) representation to obtain multiresolution set of shape features. We applied the manifold learning and dimension reduction separately at each resolution level. That is, if the MR representation has L resolution levels, a total of L manifolds are learned using features computed for each resolution level. Then, dimension reduction is performed independently at each level. To compare a pair of 3D models, a distance is calculated at each of the L levels of the MR representation. These L distance values are then combined into an overall distance between the pair of models by using a fixed-weight linear combination of the distances. We used the weight 1.0 for all the levels.

We conducted a set of preliminary experiments to compare retrieval performances of various combinations of the methods. To enter the SHREC 2007 CAD track, we chose the best performing combination, the multiresolution (MR) SPRH feature dimension reduced by using the LLE algorithm and 5,000 training models.

3. The experimental results

Table 1a shows the retrieval performance figures by using the SHREC 2006 benchmark. For example, the salient (6-level) multiresolution feature LLE-MR-SPRH-C trained using 5,000 models gained nearly 18% in First Tire Highly-Relevant (FT-HR) compared to the original SPRH (=SR-SPRH-K), that is, the single resolution SPRH without the learning-based adaptation of distance measure. The SR-SPRH-K used *Kullback-Leibler Divergence* (KLD) as the distance measure [15]. The table also lists the performance figures for the top finisher in the SHREC 2006 contest, the Makadia's method.

Table 1b shows the performance indices using the SHREC 2007 CAD track. The best performing is the LLE-MR-SPRH-C (Run6), which used a dimension reduced, 6-level multiresolution set of SPRH features. It has the FT-HR of 41.23%, which is 6.5% better than the original SPRH (=SR-SPRH-K) in the same benchmark. The LLE-MR-SPRH-C (Run5) used 5-level MR feature, omitting the level 0 (=convex hull). Overall performance of the 5-level version (Run 5) is slightly lower than that of the 6-level version (Run 6).

The performance gain due to learning appears to be more significant in the SHREC 2006 than in the SHREC 2007 CAD track. This may be due to the fact that we used the manifold trained by using the SHREC 2006-like 3D models for the SHREC 2007 CAD track.

It is interesting to note that the feature that uses only the convex-hulls of the models for feature computation, the MR-L0-SPRH-K, performed rather well. It would have placed at about 4th place in the SHREC 2007 CAD track. This might be an indication that the similarity decision of the SHREC 2007 CAD track depended heavily on the overall shape of the models.

4. Summary and future work

In this paper, we described the method we used for our entries in the SHREC 2007 CAD track. The method employs unsupervised learning to estimate a non-linear subspace, or manifold of features. Dimension reduction of the original feature using the manifold produces a salient feature that results in better retrieval performance. To enter the contest, we used the LLE algorithm [11] for the manifold learning, and used a set of 5,000 "generic" 3D models for the training samples. By combining the learning based adaptive distance measure with a multiresolution shape comparison approach we have previously proposed, we gained significant performance advantage to win the SHREC 2007 CAD track.

First in the list of items to do is to see if a training set consisting of CAD specific models would improve learning and thus retrieval performance. We are also interested in applying the database-adaptive approach to the other tracks, e.g., the face model retrieval track.

Acknowledgements

This research has been funded, in parts, by the *Ministry of Education, Culture, Sports, Sciences, and Technology of Japan* (No. 17500066 and No. 18300068).

References

[1] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* **15**, pp. 1373–1396, (2003).

[2] B. Bustos, D. Keim, D. Saupe, T. Schreck, D. Vranić, Automatic Selection and Combination of Descriptors for Effective 3D Similarity Search, *Proc. IEEE MCBAR'04*, pp. 514-521, (2004).

[3] S. Haykin, *Neural network a comprehensive foundation*, Second Edition, Prentice Hall, 842 pages, (1999).

[4] Xiaofei He, et al., Learning an Image Manifold for Retrieval, *Proc. ACM Multimedia 2004*, pp. 17-23 (2004).

[5] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation Invariant Spherical Harmonics Representation of 3D Shape Descriptors, *Proc. Symposium of Geometry Processing* 2003, pp. 167-175 (2003).

[6] G. Leifman, R. Meir, A. Tal, Semantic-oriented 3d shape retrieval using relevance feedback, *The Visual Computer (Pacific Graphics),* **21**(8-10)*,* pp. 865-875*, October 2005.*

[7] NTU 3D Model Database ver.1. http://3d.csie.ntu.edu.tw/

[8] R. Ohbuchi, T. Takei, Shape-Similarity Comparison of 3D Shapes Using Alpha Shapes, *Proc. Pacific Graphics 2003*, pp. 293-302, (2003).

[9] R. Ohbuchi, Jun Kobayashi, Unsupervised Learning from a Corpus for Shape-Based 3D Model Retrieval, *poster paper*, Proc. *ACM MIR 2006* (2006).

[10] R. Ohbuchi, J. Kobayashi, A. Yamamoto, T. Shimizu, Comparison of dimension reduction methods for database-adaptive 3D model retrieval, *accepted*, *Proc. 5th International Workshop on Adaptive Multimedia Retrieval* (AMR) 2007, Paris, France, July, 2007.

[11] S. T. Roweis, L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, **290**(5500), pp. 2323-2326, (2000).

[12] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton Shape Benchmark, *Proc. SMI '04*, pp. 167-178, (2004). http://shape.cs.princeton.edu/search.html

[13] J. Tangelder, R. C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods, *Proc. SMI '04*, pp. 145-156.

[14] R. C. Veltkamp, et al., SHREC2006 3D Shape Retrieval Contest, Utrecht University Dept. Information and Computing Sciences *Technical Report UU-CS-2006-030* (ISSN: 0924-3275)

[15] E. Wahl, U. Hillenbrand, G. Hirzinger, Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, *Proc. 3DIM 2003*, pp. 474-481, (2003).

Table 1a. Retrieval performance of our method measured using the SHREC 2006 [14].

Features	# training samples	# of MR levels	Distance	AP-HR	FT-HR [%]	DAR	NCG @25	NDCG @25
SR-SPRH-K	-	SR	KLD	0.2886	26.68	0.3990	0.3920	0.4384
MR-SPRH-K	-	6	KLD	0.3761	34.93	0.4631	0.4519	0.5101
LLE-MR-SPRH-C	5,000	6	Cosine	0.4614	44.46	0.5341	0.5604	0.5966
LLE-MR-SPRH-C	10,000	6	Cosine	0.4747	44.44	0.5382	0.5584	0.6013
Makadia (Run 2)	-	-	-	0.4364	44.77	0.5499	0.5498	0.5906

Table 1b. Retrieval performance of our method measured using the SHREC 2007 CAD track.

Features	# training samples	# of MR levels	Distance	AP-HR	FT-HR [%]	DAR	NCG @25	NDCG @25
SR-SPRH-K	-	SR	KLD	0.3721	34.76	0.4642	0.3916	0.4478
MR-SPRH-K	-	6	KLD	0.4055	36.38	0.5002	0.4369	0.4912
LLE-MR-SPRH-C (Run 6)	5,000	6	Cosine	0.4337	41.23	0.5357	0.5023	0.5341
LLE-MR-SPRH-C (Run 5)	5,000	5	Cosine	0.4319	40.25	0.5345	0.4850	0.5270
MR-L0-SPRH-K	-	1	Cosine	0.3437	30.97	0.4400	0.3896	0.4327

AP-HR: Mean Average Precision (highly relevant)
FT_HR: Mean First Tier (Highly Relevant)
ST_R: Second Tier (Relevant)
NCG @25: Mean Normalized Cumulated Gain @25

AP-R: Mean Average Precision (relevant)
FT_R: Mean First Tier (Relevant)
DAR: Mean Dynamic Average Recall
NDCG @25: Mean Normlized Discounted Cumulated Gain @25

- "MR" denotes a multiresolution feature, while "SR" denotes a single resolution feature.
- Suffix "-K" denotes distance computed by using Kullback-Leibler Divergence, while "-C" denotes Cosine distance.

3D Object Retrieval using Many-to-many Matching of Reconstruction-based Curve Skeletons

Nicu D. Cornea
Rutgers University
cornea@ece.rutgers.edu

M. Fatih Demirci
Utrecht University
mdemirci@cs.uu.nl

Abstract

The results presented in this report were obtained using a 3D matching framework based on a many-to-many matching algorithm that works with skeletal representations of 3D volumetric objects. Skeletal matching has an intuitive quality that helps in defining the search and visualizing the results. In particular, the matching algorithm produces a direct correspondence between two skeletons and their parts, which can be used for image registration and juxtaposition.

1. Approach

The general approach used for this report is described in [1]. Below, we outline the framework and highlight the differences from the above mentioned publication.

Our framework begins with computing a curve-skeleton from each object in the database and the query set. Our curve-skeleton extraction algorithm works on a volumetric representation of the 3D object. It is based on the method presented by Chuang et. al. [2], which uses a generalized potential field, generated by charges placed on the surface of the object. As the original implementation (described in [5] and used in [1]) was computationally too expensive to be efficiently used for large datasets, in [3] we developed a faster methodology, based on propagating surface normals toward the interior of the object. In summary, surface normals computed for the original object are subject to a number of smoothing steps (a default value of 10 was used here) and then, using a front propagation scheme, they are propagated to all the interior voxels by averaging, generating a vector field. The new methodology has the advantage of partially maintaining the averaging effect of the generalized potential field, while having a lower computational cost (linear).

Given a 3D vector field, we use concepts from vector field visualization to extract a "core" skeleton, connecting the critical points of the underlying vector field. The critical points are detected on a cell by cell basis, by looking for voxel cells where each of the three vector components changes sign, over the eight nodes of the voxel cell. Candidate cells are recursively subdivided and the above candidacy test is repeated until either the cell fails the test, or the cell is small enough, in which case a critical point is assumed to exist at the center of the cell.

Skeleton segments are discovered using a force following algorithm on the underlying vector field, starting at each of the critical points. The force following process evaluates the vector (force) value at the current point and moves in the direction of the vector with a small pre-defined step. Since at critical points the force vector is zero, the initial step is taken in the direction of the outgoing flow, given by the eigenvectors corresponding to the positive eigenvalues of the Jacobian at the critical point. The skeleton obtained by connecting the critical points, is called the "core skeleton".

The skeleton obtained using the above algorithm consists of a set of points sampled by the force following algorithm. Each skeleton point is then equipped with a distance transform value: a real number specifying the distance to the closest point on the surface of the object. This additional information is used by the many-to-many matching process.

Different from the approach taken in [1] and [5], we construct the rest of the curve-skeleton guided by how much of the original object is reconstructed by the current skeleton. The idea behind this approach can be expressed as follows: a curve-skeleton represents a particular object better if it can reconstruct more of the original object. We start with the core skeleton and reconstruct the object from this representation using the distance transform information, associated with each skeleton point. We compute the difference between the original object and the object reconstructed from the current curve-skeleton. If the difference is more than a given percentage of the original object, we choose a point from the un-reconstructed part of the object and derive a new curve-skeleton branch using the selected point as a seed point for the force-following algorithm described above. The process is then repeated started from the new curve-skeleton until the target reconstruction percentage is reached. The representative point from the un-reconstructed part is selected using the divergence criterion presented in [5].

In the experiments presented in this report, we used two target reconstruction percentages for the two runs: 30% and 80% of the original object. We name the two runs MMM-RCS-30 (for 30%) and MMM-RCS-80 (for 80%), corresponding to Many-to-Many Matching of Reconstruction-based Curve-Skeletons. Note that some objects, especially those containing surface-like regions, cannot be well reconstructed using a curve-skeleton. Therefore, the skeleton construction algorithm stops when there is no significant reconstruction progress from one step to the next.

For more details on the force-following algorithm and other details about curve-skeleton computation, please see [3] and [5].

Given a pair of 3D skeletons, our next objective is to find the correspondences between their vertices and produce a similarity score. We use a distribution-based similarity measure, known as the Earth Mover's Distance (EMD) [4] to achieve this goal. The Earth Mover's Distance framework is designed to evaluate dissimilarity between two multi-dimensional distributions. Each distribution consists of a set of points along with their attributes. Coordinates in multi-dimensional space, weights, and scales are only a few examples of point attributes.

The EMD approach assumes that a distance measure between single features, called the ground distance, is given. The EMD then “lifts” this distance from individual features to full distributions. Intuitively, given a pair of distributions, one may see the first distribution as a set of piles and the other as a set of holes in the same space. Let us define a unit work as transporting one unit of earth by a unit of ground distance. The EMD then computes the minimum amount of work that is necessary to fill the holes with earth. This problem is formalized as linear programming problem.

The EMD approach extends the concept of distance between two points to that of a distance between two sets. The distance between two sets can be used as their dissimilarity score. It was shown in [4] that when the ground distance is perceptually meaningful, the EMD obtains perceptual similarity better than other measures. In addition, one main advantage of using EMD lies in the fact that it subsumes many histogram distances and permits matches in a natural way. If the size of each set in the problem is not equal, the EMD approach results in a partial matching. This important property allows the similarity measure to deal with uneven clusters and noisy datasets. This is especially significant for some applications such as image retrieval and registration. We used this property to compute a partial matching between two skeletons.

Computing the EMD is based on a solution to the well-known transportation problem, whose optimal value determines the minimum amount of “work” required to transform one distribution into the other. While computing this optimal value, the approach also finds the correspondences between input skeletons. The reader is referred to [1] for details.

2. Results

The overall results (Figure 3) show that, in general, our methodology has discriminative power in retrieving relevant classes of objects, being consistently better than random choices. The graphs also seem to indicate that a target reconstruction of 80% performs worse on the retrieval task than a 30% target, on the given query set. This can also be seen from the individual graphs (Figures 4, 5 and 6), where MMM-RCS-30 performs better than MMM-RCS-80 in most cases.

The MMM-RCS method performs very well for objects which can be represented well by generalized cylinders (see queries 3, 20, 22, 24). For these objects, the curve-skeleton reconstructs most of the object, while at the same time remaining very simple and thus easier to match to other similar curve-skeletons. The method performs particularly badly (worse than or, similar to, random selection) on shapes with surface-like sections (see queries 2, 9, 10, 17, 21, 26). These sections cannot be well reconstructed by a simple curve-skeleton. As a result, our reconstruction-based skeleton building process generates very complicated skeletons, with many branches, and clearly no discriminative power.

It is interesting to note that this experiment does not disprove the initial hypothesis that reconstruction is a good measure of quality for shape descriptors used for matching. Although the results show that the 30% target reconstruction skeleton performs better than the 80% skeleton, this can be explained by the increased complexity of the 80% skeleton for objects that cannot be represented well by curve-skeletons. For mostly tubular objects, 80% is actually better than 30%. The solution for a better matching result seems to be a shape descriptor which combines curve and surface descriptors, so that both of these types of sections are easy to reconstruct from a simple descriptor.

3. References

[1] N.D. Cornea, M.F. Demirci, D. Silver, A. Shokoufandeh, S. Dickinson, P.B. Kantor. 3D Object Retrieval using Many-to-many Matching of Curve Skeletons. IEEE International Conference on Shape Modeling and Applications (SMI 2005), pp. 368-373, MIT, Boston, MA. June 15-17, 2005.

[2] J. Chuang, C. Tsai, and M.-C. Ko. Skeletonization of three-dimensional object using generalized potential field. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(11):1241-1251, 2000.

[3] N.D. Cornea. Curve-skeletons: Properties, Computation and Applications. Ph.D. Thesis, Department of Electrical and Computer Engineering, Graduate School New Brunswick, Rutgers, The State University of New Jersey.

[4] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision, 40(2):99-121, 2000.

[5] N.D. Cornea, D. Silver, X. Yuan, R. Balasubramanian. Computing Hierarchical Curve-Skeletons of 3D Objects. The Visual Computer 21(11):945-955, Springer-Verlag, October, 2005.

Sub-part Correspondence by Structural Descriptors of 3D Shapes

S. Marini, S. Biasotti, M. Spagnuolo and B. Falcidieno
Istituto di Matematica Applicata e Tecnologie Informatiche
Consiglio Nazionale delle Ricerche - Genova, Italy
{simone.marini, silvia.biasotti, michela.spagnuolo, bianca.falcidieno}@ge.imati.cnr.it

Abstract

This extended abstract describes a method for partial shape-matching able to recognize similar sub-parts of objects represented as 3D polygonal meshes. The geometry and the structure of the shapes are coupled in a descriptor that provides a flexible coding, grounded on solid mathematical theories, and that can be adapted to the user's needs and to the context of applications.

The matching framework for sub-part correspondence is achieved through a graph-matching technique, which builds the common sub-graphs between the two shapes and highlights the maximal sub-parts having similar structure and similar space distribution.

1. Approach

Aim of this extended abstract is to describe a method for recognizing the sub-parts of two objects the most similar both in geometry and structure. The main innovation of the method is the coupling of a structural descriptor [1] with a geometric descriptor and the inexact graph-matching techniques [5, 7, 2].

The structural information of the object is captured by the Extended Reeb Graph (ERG) computed with respect to a position invariant function [1, 2]. The function used for the experiments performed on this track is the integral geodesic distance proposed in [3]. The shape characterization and the Reeb graph construction naturally induce a decomposition of the shape into topologically significant regions used to produce a directed graph in which each node corresponds to an object sub-part and each edge connects two nodes. Node attributes represent the geometric descriptors associated to corresponding sub-part [2]. The mathematical properties of the extended Reeb graph assure that the structural descriptor can be represented as a directed and acyclic graph even if the shape model, the graph is extracted from, has genus grater than zero.

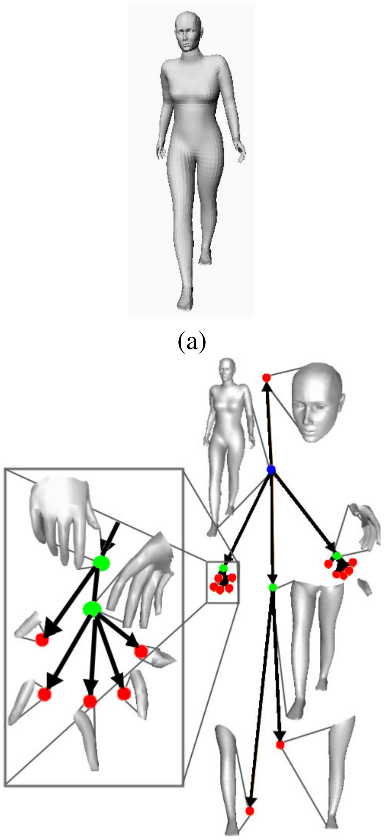
Figure 1 shows an example of a 3D model 1(a) and the corresponding structural descriptor 1(b). Since the graph is directed and acyclic, each node of the graph can be considered as a root of its induced subgraph. This subgraph identifies the sub-part of the model that can be described by a geometrical descriptor (see Figure 1(b)). Within this track, the shape descriptor used to produce the geometrical attributes associated to each node of the graph is the rotation invariant spherical harmonic representation proposed in [4].

The sub-part shape correspondence between two objects is obtained by matching the directed attributed graphs. The output of the matching process should be the largest maximal common subgraph that minimizes the geometrical and the structural differences between the two models. The proposed matching algorithm can be synthetically described by the two following steps:

1. Select a mapping M among the nodes of the two graphs G_1 and G_2. The mapping M is a set of node pairs (v_1, v_2), where v_1 is a node of G_1 and v_2 is a node of G_2;

2. compute the common subgraph between G_1 and G_2 by expanding the mapping M.

Figure 2 provides a graphical explanation of the algorithm described by the steps 1 and 2. The initial mapping described by step 1 is shown in figure 2(a). This step allows the use of several heuristics techniques to identify the initial mapping depending on the application context the algorithm is used for. Step 2, shown in figure 2(b), expands the initial mapping M as much as possible while respecting the definition of common subgraph. Also in this case, as for the previous step, the expansion process can be driven by heuristic techniques depending on the semantics of the models and the application context. For the track on partial matching have been used general-purpose techniques based on the relevance of the graph nodes and the combination of geometry and structure as described in [2]. In particular, it has to be observed that the initial mapping among relevant nodes makes the algorithm robust with respect to structural



(a)

(b)

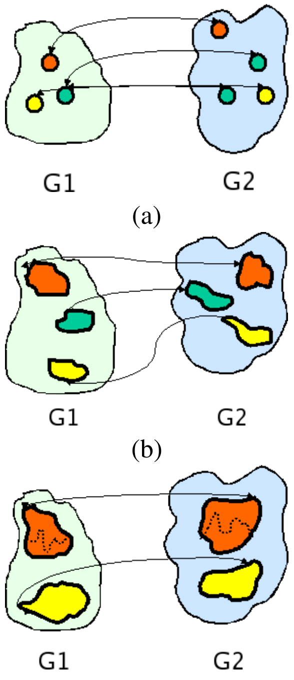
Figure 1. A 3D model (a) and its corresponding structural descriptor (b). The direction of the edges and the sub-parts of the model identified by each node are shown in (b).

noise, allowing the construction of a not necessarily connected common subgraph, as shown in figure 2(c), that enables the recognition of similar sub-parts even if the overall objects shape/structure is dissimilar.

2 Results and Discussion

The results produced within this track of the SHREC07 contest, see [6], have been obtained by applying the matching algorithm previously described to the ERG structural descriptor obtained by analyzing the models with two different resolutions. In particular the function based on the integral geodesic distance has been sampled into 15 and 63 level sets, corresponding to the runs ERG 1 and ERG 2 respectively.

As shown in Figure 3 of the report on the partial matching track [6], the performance improvement of the two runs ERG1 and ERG2 are marginal by considering highly relevant and marginally relevant models instead of only highly relevant models. This means that for both ERG1 and ERG2 the performance are mainly driven by the highly relevant models.



(a)

(b)

(c)

Figure 2. Synthetic description of the matching algorithm: the initial mapping (a), the expansion process (b) and the common subgraph (c).

Even though the overall performance of both the runs ERG1 and ERG2 are quite good, different behavior can be observed in the Figures 4, 5, 6 shown in [6]. For example query 3 is not characterized by relevant structural features. In this case both ERG1 and ERG2 do not provide good performance. Moreover, it should be noticed that while for the queries 5, 9, 17, 19 and 23 both ERG1 and ERG2 have similar performance, while for the queries 4 and 10 their performance are quite different. This means that due to the different resolutions corresponding to the two runs, the relevant features of the query models are not correctly recognized by the matching process. A similar remark must be made for the queries 2 and 10, where the involved sub-parts are the same but their relative size is different.

Finally, it has to observed that for the results corresponding to the queries 15 and 22 the performance of both ERG1 and ERG2 are quite bad. Also in this case the problem may be caused by a bad recognition of the relevant features.

3. Acknowledgments

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

References

[1] S. Biasotti. *Computational topology Topology Methods for Shape Modelling Applications*. PhD thesis, University of Genova, May 2004.

[2] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Subpart correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002–1019, September 2006.

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

[4] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In L. Kobbelt, P. Schröder, and H. Hoppe, editors, *Proceedings of Symposium in Geometry Processing*, pages 156–165, Aachen, Germany, June 2003.

[5] S. Marini. *3D Shape Similarity Through Structural Descriptors*. PhD thesis, University of Genova, April 2005.

[6] S. Marini, L. Paraboschi, and S. Biasotti. SHape REtrieval Contest 2007: Partial Matching Track. In these Proceedings.

[7] S. Marini, M. Spagnuolo, and B. Falcidieno. From exact to apporximate maximum common subgraph. In *Proceedings of the 5rd IAPR -TC-15 Workshop on Graph-based Representations in Pattern Recognition*, Lecture Notes in Computer Science, pages 263–272, Poitiers (France), April 11-13 2005.

Auto classification of SHREC 2007-Protein Challenge

Bin Li, Yi Fang, Karthik Ramani, Daisuke Kihara
Purdue University
lib@purdue.edu

Abstract

We attend the SHREC 2007-Protein model, we focused on the topology of each protein: they use STRID to detect the secondary structure, including the hydrogen bond. Then, they compute the beta sheets (beta strands connected with hydrogen bond) and the orders. For main class a, b, c, d, g, and folds of a and g, we used the length and percentage of alpha helix and beta strand to classify. For each fold in each class b, c, d, they used the keys of the orders to classify. Our method performed best in the contest.

1. Introduction

We attended the SHREC 2007-Protein Challenge competition, SHREC 2007 was organized by the AIM@SHAPE network to evaluate the effectiveness of 3D features on protein 3D models. The website address for the competition is available at http://lmb.informatik.uni-freiburg.de/events/shrec07/index.en.html. Our method performed best using some simple features based on proteins' secondary structures and their topologies.

The SCOP database is the well-known ground truth for protein classification recognized by biological community. In SCOP database, protein classes are defined by protein's secondary structures and protein folds are mainly defined by the topologies of secondary structures.

For the 653 provided dataset, protein classes and folds were first manually checked against their classes' and folds' definitions in SCOP database; The class clustering are mostly related to the size of alpha helix, beta sheets or the number of residues, but most of folds are not accurate exactly match. For example: in class a, most of proteins have alpha helixes only, but in 1opc for example, beta sheets also exist. However in protein folds, most of the beta strands are not exactly but closely ordered as defined.

1.1. Class classification

For class a, b, c, d, and g classification, we used the size of alpha helix, beta sheets, and counted the number of residues to classify.

1.2. Fold classification

For fold classification in class a, we mainly used alpha helix information, and all class g were classified as fold g.3 since most of the folds in class g belong to g.3.

For fold classification in class b, c, and d, first, we used STRIDE to compute the alpha helixes, beta sheets, and hydrogen bonds, then we computed the order of each core strands; Here, to avoid the cases of reversed order such as 4123 and 3214, the order of 1 would always close to the first half of the string; when 1 is in the middle position, the smaller number would be first such as we would use 213 instead of 312.

Since most of the datasets are not exactly ordered as defined in the SCOP database, the problem became, technically, finding the approximate definition of each classes and folds, which should not be too specific for similar structure, and also it should not be too general in order to distinguish each classes and folds.

One approach could be, for a given protein, with its orders of 4132 and 213 for example, the order with the largest number of beta sheets 4132 and the number of beta strand 2 are used as the key for grouping, and is recorded like 2_4123. With this approach, we found that same keys will most likely fall into same fold, and same fold usually have multiple keys. For example: order 2_4123 were only observed in fold b.1, and 2_213 also mostly happened in class b.1, where the formal definition of fold b.1 is "sandwich; 7 strands in 2 sheets; greek-key".

For a given query, the process of fold classification in class a and g was only based on residue length on alpha helix, beta sheets, or others. For folds in class b, c, and d, the process is: first group all the keys of orders with their folds in dataset, ordered by their occurrences in dataset, then look for the exact string matching of giving query's ordering key. The fold with the maximum number of key occurrence was the hit. If not record was found, a default value would be used.

1.3. Scoring

As requested in contest, we need to provide a distance matrix for the given query. The distance between two proteins was defined as major class distance + medium fold distance + minor size distance, the fold distance would be 0 if two proteins' class were same. For class distance, class b should be more close to class c than class a, because class a do not have beta sheets, so we simply map each class into a 2D space, which would satisfy the distance constrain. For example: class a is mapped to (-1, 1), class b is mapped to (1,-1) and class c is mapped to (0,-1); and the class distance is computed by the Euclidean distance. Similar to class distance, fold distances were also mapped by the guidance of missing classification of our method. For example, fold a.24 and a.26 were miss classified mostly then we mapped a.24 closely to a.26. Size distance was defined as 1-(shorter number of residue)/(longer number of residue), then the distance value would be between 0 and 1. And then, different weights 0.6, 0.4 and 0.1 were applied to class, fold and residue distance.

2. Database

Totally 633 proteins are provided, which falls into 27 folds in SCOP database. Details of the dataset are available from http://lmb.informatik.uni-freiburg.de/events/shrec07/task.html.

3. Method

The first step was to clean the protein by removing duplicated residues and multiple frames, and then STRIDE was used to detect the secondary structure, including the hydrogen bond. Next step was to compute the orders of beta sheets, the beta strands connected by hydrogen bond, and then we computed the keys, such as 2_3124, which means two beta sheets and the largest one is ordered as 3124.

Then classes were classified as following: Computed the total number of residues in alpha helixes, NA, and total number of residues in beta sheets, NB, let PA=NA/(NA+NB).

```
If NA+NB are less than 18, then it is class g.
If PA is larger than 0.94, then it is class a.
If PA is less than 0.35, then it is class b.
If the total number of atoms is less than 104, then it is class d, else it is class c.
```

To classify the folds in class a:

```
If number of alpha strand >= 6
    If the average length of alpha strand >= 12.5 then it is fold a.1.
    If the average length of alpha strand < 12.5 then it is fold a.39.
Else
    Compute each ith alpha helix's vector by subtracting the coordinates of the first CA and the last CA, then average the angles between ith alpha helix and (i+1)th alpha helix, named as BUNDLE.
    Compute the average residue length between adjacent alpha helixes, named as AVGLEN.
        If (BUNDLE >0.57) then
            If AVGLEN < 9.5 then a.24
            If AVGLEN>=9.5 then a.26
        Else
            If AVGLEN > 8 then a.3
            If AVGLEN <= 8 then a.4
        End if
    End if
```

To classify the folds in class b, c, d:

```
Compute the key of the order K.
If K is found in the dataset
It is the corresponding fold with the maximum number of occurrence.
Else
    It is b.29 for class b.
    It is c.93 for class c.
End if.
```

3. Scoring

To compute the distance between two proteins:

Map each class of a, b, c, d, and g to (-1, 1), (1, -1), (0,-1), (0, 1), and (0, 0);

Map each fold of 1, 26, 24, 29, 45, 39, 3, 4, 85, and 86 in class a to:

(0, 0), (1, 0), (2, 0), (2, 0.5), (2.5, -0.5), (0, 1), (2, 2), (2.5, 2.5), (0, 1.5), and (-0.5, 1).

Map each fold of 1, 6, 121, and 40 in class b to:
(0, 0), (0, 1), (1, -1), and (-1,-1).

Map each fold of 2, 23, 55, 47, 3, 69, 1, and 93 in class c to:
(0, 1), (-1, 1), (1, 1), (0, 1.5), (1, 1.5), (0, -1), (-1, -1), and (1, -1)

Scoring:
The final distances were computed as: (class distance)*0.6 + (fold distance) *0.4 + (size distance) *0.1 (The fold distance is set to 0 if two proteins belong to same class)

5. Results

The method can correctly identify 77% of 632 proteins in fold level. Where the details of the results are available from http://lmb.informatik.uni-freiburg.de/events/shrec07/results.html

5. Conclusion

After we computed the secondary structures using STRIDE, some beta sheets or alpha helixes were shown in PYMOL but not identified in STRIDE. Since the secondary structure would affect our results significantly, a more robust method is needed.

Unlike other models in SHREC, the problems in protein model were not just to find the optimal methods, it is also needed to identify the correct questions, since this is still not very straightforward in SCOP database even it’s the golden standard. It seemed that most proteins do not follow a strict rule to be classified into a certain category, which looked like to be protein’s nature character, so there would be lots of "errors" and the errors were hard to be avoided.

Our method is simple, so it might tolerant more "errors". Also, using keys of orders are likely memorizing the answers. So, more studies on clustering were needed.

3D Protein Classification Using Topological and Geometrical Information

P. Daras, V.Tsatsaias, M. G. Strintzis
Informatics & Telematics Institute
1st km Thermi-Panorama Rd
57001 (PO Box 60361)
daras@iti.gr

1. Introduction

Computational approaches for protein classification have been proposed over the last years in order to speed up the analysis of the biological mechanics in living organisms. Most of the approaches tend to focus in geometrical comparison of the 3D molecules to reach their goals. This method is suitable for partial (sub) graph matching of 3D proteinic models, in order to achieve fast and accurate classification, is proposed. The method is analytically described in [1].

2. Descriptor Extraction Method

Firstly, the pdb files are parsed so as to create the 3D protein representations. Then, each proteinic chain is being segmented based on the type of secondary structures of the molecule. Further, the Spherical Harmonic expansion is applied to every protein segment, in order to extract geometrical descriptors invariant to geometric transformations. The segmentation process leads in the construction of a graph which is further enriched with the extracted geometrical descriptors and topological information such as the angles between two neighboring edges. Thus, every protein is fully described by an attributed graph, where the attributes are the topological and geometrical features.

3. Matching

For the matching purposes, an attributed graph matching process has been utilized. Parts with topological and geometrical similarities are identified and finally a single dissimilarity metric is computed.

Acknowledgement

This work is supported by the VICTORY EC project (contract no: 044985)

References

[1] V. Tsatsaias, P. Daras and M.G. Strintzis: "3D Protein Classification Using Topological, Geometrical and Biological Information", IEEE International Conference on Image Processing, (ICIP 2007), San Antonio, Texas, USA, September 2007.

3D Face Recognition by Matching Facial Regions

Stefano Berretti
University of Firenze
Dipartimento Sistemi e Informatica
via S.Marta 3, 50139 Firenze, Italy
berretti@dsi.unifi.it

Abstract

In this work, we describe a 3D face recognition approach that we tested in the SHREC - 3D Shape Retrieval Contest of 3D Face Models organized by the Network of Excellence AIM@SHAPE.

1. Introduction

Human face identification has been addressed mainly focussing on detection and recognition of faces in 2D still images and videos [6]. Recently, three-dimensional (3D) facial data has been proposed to improve the effectiveness of face recognition systems [2]. This is motivated by the fact that solutions based on 3D face models are less sensible, if not invariant, to lighting conditions and pose variations. This opens the way to systems supporting face recognition by combining information extracted from 2D images with the information of facial characteristics derived from 3D models. Motivated by these considerations, an increasing number of face recognition approaches using 3D data has been proposed.

2. Face representation

In our approach, in order to capture structural information of 3D faces, the complete face surface is used and a smooth real valued function is defined on the 3D surface in order to capture its local characteristics. In particular, we computed the normalized geodesic distance between the surface vertices and a reference (fiducial point) identified at the nose tip. Then, distances are quantized, and surface points which belong to the same interval of distance from the nose tip are grouped together to form different *bands* on the face. Every band is then modeled by considering its spatial relationships with respect to any other band of the face. In so doing, a modeling technique is used to represent spatial relationships between 3D extended sets of points. This representation is based on integral measures which evaluate the relative arrangement of individual pairs of points by considering their mutual position in a 3D absolute reference system. Since this latter representation is not rotation invariant, models require a common alignment in the 3D absolute Cartesian reference system. To this end, the plane formed by the nose tip and by the two fiducial points located at the base of the eyes is used. As a consequence, the approach relies on a pre-processing step in which fiducial points of the face are identified, and a common alignment of the models is obtained.

2.1. Fiducial points identification

Fiducial points identification requires the computation of *Gaussian* and *Mean* curvatures of models surfaces. Since the mesh surface is discrete and the computation of Gaussian and Mean curvature requires C^2 functions, we interpolated the surface using the bi-quadratic function of a paraboloid (with unknown coefficients a, b, c, d and f):

$$\zeta = a\xi^2 + b\xi\psi + c\psi^2 + d\xi + e\psi + f \quad (\in C^2)$$

Once the 6 coefficients are determined, values of Gaussian (K), and Mean (H) curvatures can be directly obtained:

$$K = \frac{(4ac - b^2)}{(1 + d^2 + e^2)^2}, \quad H = \frac{(a + c + ae^2 + cd^2 - bde)}{(1 + d^2 + e^2)^{3/2}}$$

In particular, the discrete surface is locally approximated, vertex by vertex, using the following steps:

- A local reference system centered on the vertex is defined, with the z-axis oriented along the normal $\mathbf{n}$ to the surface at the vertex. Coordinates of the neighborhood vertices are transformed into the vertex local reference system according to the rotation $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)^T$, where $\mathbf{r}_1 = (\mathbf{I} - \mathbf{n}\mathbf{n}^T)\mathbf{i}/||(\mathbf{I} - \mathbf{n}\mathbf{n}^T)\mathbf{i}||$, $\mathbf{r}_2 = \mathbf{r}_3 \times \mathbf{r}_1$ and $\mathbf{r}_3 = \mathbf{n}$;

- Neighborhood vertices $\mathbf{q}(x,y,z)$ to the current vertex $\mathbf{p}(x,y,z))$, are transformed according to $\mathbf{R}$: $\mathbf{q}(\xi,\psi,\zeta) = \mathbf{R}\cdot(\mathbf{q}(x,y,z) - \mathbf{p}(x,y,z))$ and a redundant system of equations is solved:

$$\begin{bmatrix} \xi_{l,1}^2 & \xi_{l,1}\psi_{l,1} & \psi_{l,1}^2 & \xi_{l,1} & \psi_{l,1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{l,n}^2 & \xi_{l,n}\psi_{l,n} & \psi_{l,n}^2 & \xi_{l,n} & \psi_{l,n} & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \zeta_{l,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \zeta_{l,n} \end{bmatrix}$$

For almost all the vertices, a system of six equations has been sufficient. Once curvature values are computed for every vertex of the surface, a median filtering is used to remove impulsive noise.

Our face representation relies on the identification of three fiducial points on the model surface. These are the *pronasale* (i.e., the nose tip) and the left and right *endocanthion* (i.e., the points at the inner commissure of the left and right eye fissure, respectively), which have been verified to be stable with respect to face variations [3]. These can be identified using the curvature information observing that points near to the nose tip identify a convex region (characterized by negative values of the Mean curvature and positive values of the Gaussian curvature), while points at the two endocanthion originate two concave regions (characterized by positive values of the Mean and Gaussian curvatures).

In particular, if $H(v)$ and $K(v)$ are the Mean and Gaussian curvatures for a vertex v, the vertex is classified according to the following rules:

$$convex\ vertex\ if: \quad H(v) < -t_H \ \ and \ \ K(v) > t_K$$
$$concave\ vertex\ if: \quad H(v) > t_H \ \ and \ \ K(v) > t_K$$

with thresholds $t_H = 0.04$ and $t_K = 0.005$. These values for the thresholds were previously determined on databases of 3D faces of real subjects scanned at one-to-one scale (about $70mm$ as average distance between mesh vertices and the centroid of the mesh) and resolution of about 15000 vertices. On the left of Fig.1, a face model is shown with the convex and concave regions identified according to the Gaussian and Mean curvature.

However, this curvature estimation is not effective on meshes of high resolution like those used in SHREC which have a complexity of 75972 vertices. In this case, interpolating polynomials of higher degree are needed or different approximating surfaces (like those defined by Bèzier [5]). Due to the limited time to provide results for the SHREC competition, we faced this difficulty by reducing the mesh resolution to the 15% of the original one using the Heckbert/Garland sampling approach. This simplification tries to reduce the number of vertices in planar regions of the mesh, while preserving the surface complexity of curved regions.

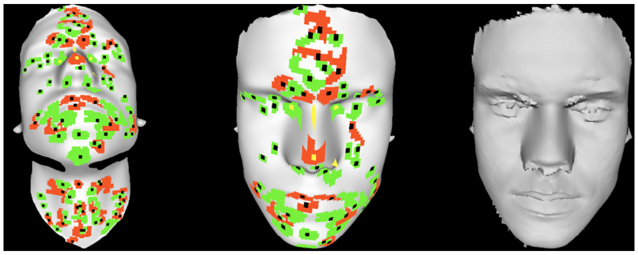


Figure 1. Face processing to identify fiducial points and to provide a common alignment of the models.

Once convex and concave regions have been identified, a representative point for every region must be determined in order to be used as candidate fiducial point. To this end, the vertex with the normal more close to the mean normal of the region (averaged on all the vertices of the region) is considered. The squared black markers inside regions of Fig.1 are used to highlight these points.

To determine the convex and concave regions corresponding to the fiducial points, the most plausible *triangle* representing the arrangement of the nose tip and of the two endocanthion is determined. For this purpose, constraints on the shape, position and size of the nose and the eyes are used to prune the number of possible valid triangles. For the remaining triangles, a fitting operation is performed which measures the error between the surface points and those of a "template nose" adjusted on the three vertex of the triangle (see Fig.2). Vertices of the triangle providing the smaller error are finally selected as fiducial points of the model.

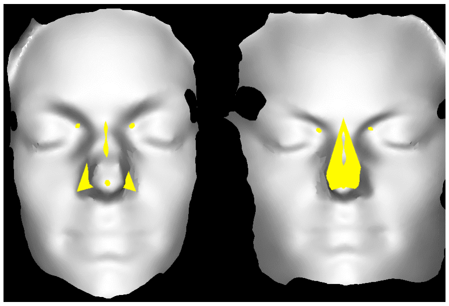


Figure 2. The "template nose" superimposed to a face model is shown: frontal view on the left; behind view on the right.

The complete preprocessing for a sample face is shown in Fig.1. The face reported in the center of Fig.1 evidences the "template nose" and the corresponding fiducial points. On the right, the model rotated in its final aligned position is shown.

2.2. Face partitioning

Instead of performing a-priori rigid partitioning of the face surface, the complete face surface information is used to define a smooth real valued function on the surface of the 3D model in order to capture its local characteristics. In particular, the *normalized geodesic distance* between surface points and the nose tip, taken as the reference point are computed. The face surface S is approximated through a discrete triangular mesh $\mathcal{M}$ with n vertices $v_1, \ldots, v_n$, and v_f is the fiducial vertex located at the nose tip. Geodesic distances are calculated on the piecewise planar mesh according to the *Dijkstra*'s algorithm. This approximates the actual geodesic distance between vertices v_i and v_f with the length of the shortest piecewise linear path on mesh vertices $\mu(v_i, v_f) = \min_P L(P(v_i, v_f))$, where: $P(v_i, v_f)$ is the path from v_i to v_f defined as an ordered sequence of adjacent vertices $(v_{p(1)}, \ldots, v_{p(k)})$; $L(P(v_i, v_f))$ is the length of the path measured as the sum of the Euclidean distances between adjacent vertex pairs. More sophisticated approximations (like for example the *fast marching* algorithm [4]) can be used which are not constrained to the edge length, but use the entire surface of the mesh. In the presence of non-triangular meshes, or non-regular meshes, approximation of the actual geodesic distance may be inaccurate. To avoid this, meshes undergo a pre-processing step accounting for triangularization and regularization.

Normalized values $\bar{\mu}(v_i, v_f)$ of the geodesic distance are obtained dividing $\mu(v_i, v_f)$ by the Euclidean eyes-to-nose distance. This normalization guarantees invariance of function values with respect to scaling of the face model. Furthermore, it does not bias the values of the function under expression changes. Once values of $\bar{\mu}(v_i, v_f)$ are computed for every surface vertex, the face surface can be partitioned into *iso-geodesic bands*. For this purpose, the range of $\bar{\mu}$ values is quantized into N intervals $c_1, \ldots, c_N$. Accordingly, N bands are identified on the model surface, the i-th band corresponding to the set of surface vertices for which the value of the distance falls within the limits of interval c_i. In so doing, the effectiveness of the representation depends on two main factors: the maximum allowed value of geodesic distances from the nose tip (this defines the *face coverage*, that is the portion of the face surface used for recognition), and the number of levels quantizing this interval (this determines the *number of bands* as well as their *width*).

For the purpose of 3D face recognition, the face coverage includes the main face traits, like eyes, nose, mouth, cheeks, eyebrows and the main part of the forehead, and doesn't include parts which can be affected by noise, like the borders of the model, or the hair. This is particularly the case of models acquired by real scanners where the sole frontal capture of the face is available (and not the whole model of the head like in SHREC). Centering this region on the nose tip, its radial geodesic range spans $8 \div 9 cm$. The optimal width of facial bands, instead, is mainly related to the modeling technique used for their description. A compromise has been found using bands which span $1cm$ in the geodesic radial distance. This results in $8 \div 9$ bands covering the part of the face relevant for recognition, with approximately a maximum of $15 \div 20\%$ of the points comprised by a band, changing their positions due to expression variations.

Different runs of our approach submitted to SHREC, have been obtained using different face coverages and different widths of facial bands. In general, we observed that better results have been obtained using 9 bands of normalized width of 0.08 (run 1).

2.3. Modeling spatial relationships between facial regions

Information about the arrangement of facial bands is represented through integral measures capturing the mutual spatial relationships (in all the possible elementary directions of the space, left, right, up, down, in front, behind, etc.) between any pair of points of the same band (*intra-band* spatial relationship), and between any pair of points of different bands (*inter-band* spatial relationships). For any pair of bands, S_i and S_j, variations in the relative spatial arrangement of pairs of points in S_i, S_j, impact on the overall value of the measure according to the proportion between the number of varying points with respect to the overall number of points in S_i and S_j. More details on this modeling approach can be found in [1].

This permits, at the same time, to perform efficient computation of the structural information of a face model and smooth the effect of local variations induced by expression changes. Iso-geodesic bands and their spatial relationship measures are cast into a graph-like representation, where graph nodes represent facial bands and graph edges their relative 3D spatial arrangement. Graph matching is used to perform effective and efficient comparison of face models.

3. Results

For our approach, results show a lower average *Normalized Cumulative Gain* and a lower average *Normalized Discounted Cumulative Gain*, with respect to the others two approaches that submitted the final runs to SHREC. Results are not completely satisfactory, but are encouraging. This is motivated by the fact that the overall performances of the approach have been negatively affected by some factors that we have not encountered in our previous experimentations.

First, the methods has been affected by the fact that all the models of SHREC are artificially rotated up to 30 degrees in every direction. In fact, our approach requires a common alignment of the models with respect to a global reference system, and can cope with small rotations. In real contexts of use, usually only small differences in the alignment of face models, due to different acquisitions, are present. In the context of SHREC, our solution for face alignment resulted not completely adequate to align many of the models. This is evidenced by the fact that also for our best submitted result series (run 1), in the 45% of the cases (29 queries out of the 64), the query model itself has not been retrieved in the first position. This evidences a difficulty for these queries in determining a common alignment between models. If the results of the *Normalized Cumulative Gain* and of the *Normalized Discounted Cumulative Gain*, are examined by considering this fact, it can be observed that the curves of our approach have the same increasing slope of the others two competitors (at increasing gain), thus evidencing a similar retrieval trend. The difference is mainly due to the difference in the starting point which depends from the missed retrieval of the database model equal to the query.

In addition, as reported in Sect.2.1, the high resolution of the models required us to reduce the number of vertices by sub-sampling the model surface. This has been necessary to have good performances for the process which extract fiducial points of the face. Of course, working at a considerably lower resolution has determined the loss of face details, thus contributing to further reduce the performance with respect to the others competitors.

Moreover, our approach was devised to perform 3D face recognition on real face models acquired by common acquisition devices (like 3D laser or structured light scanners) even in presence of different facial expression. Since these devices are typically used to capture only the frontal part of the face, our method does not consider the whole head in comparing different models. This reduces the overall information used in the match, that can instead be used by different solutions specifically targeted to use the whole model of the head. Actually our model can also use the whole head information, but we did not experimented this before, and so we decided to only use the face of the head models in order to compare SHREC models.

The combination of these factors has impacted the average performance measures of our approach. However, its capability to capture salient structural information of the face and to provide good similarity retrieval is evidenced by those queries for which the above difficulties have had a lower influence. As an example, Fig.3 and Fig.4 report the *Precision vs. Recall* and the *Normalized Discounted Cumulative Gain*, respectively, for the query model $Q1001$.

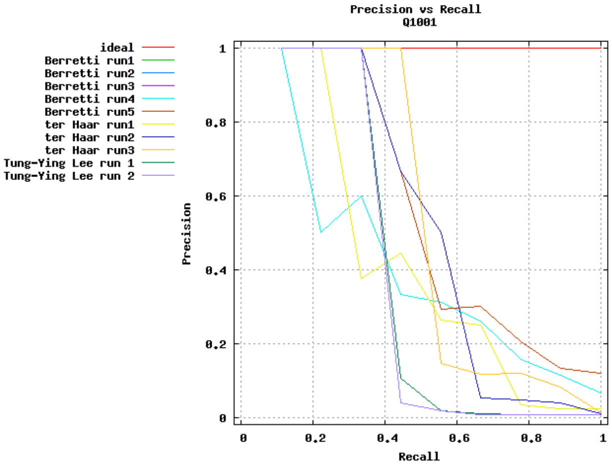


Figure 3. Precision vs. Recall curves for the query model Q1001.

It can be observed that, in the cases for which face alignment has been successfully solved, the approach provides results which are comparable with that of the others two competitors.

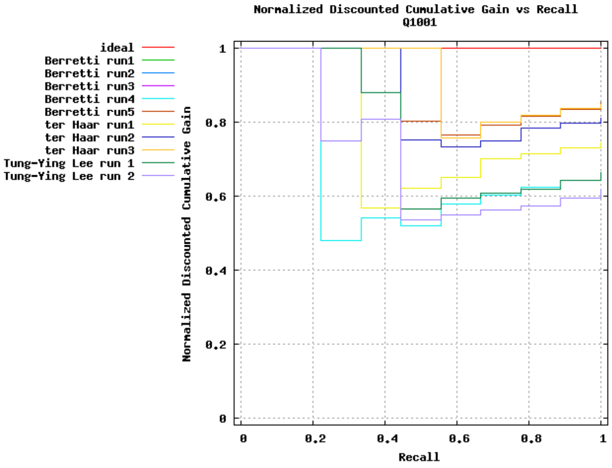


Figure 4. Normalized Discounted Cumulative Gain for the query model Q1001.

Finally, it is relevant to note that the proposed approach is capable to match a query model against the overall SHREC database comprising 1516 models in approximately $1sec$ (using non optimized Java code on a Pentium machine with 1Gb memory). This makes the proposed solution scalable to very large face databases, and applicable in real contexts of use, where recognition must be performed in real time. However, this performance measure was not considered among the evaluation criteria comparing different solutions in SHREC.

4. Discussion

In this work, we experimented our 3D face recognition approach in the context of 3D face similarity retrieval. The approach has proved to be highly effective and efficient in 3D face recognition. Results for 3D similarity retrieval on the SHREC database indicated that for this kind of applications improvement to the pre-processing steps of the approach are needed. In particular, a main factor that limited the average performance of the approach in 3D similarity retrieval, resulted the difficulty to obtain a common alignment of the models with respect to a global 3D reference system. Since the approach assumes a global reference system for all the models, wrong alignments resulted in an initial error that biases the representation and reduces the accuracy of retrieval in many cases. The difficulty in model alignment has been further complicated by the need to down-sample the vertices of the models in order to perform the detection of the fiducial points needed for models alignment, and for the computation of geodesic distances on the model surface. Based on these observations, future work will be mainly devoted to improve the pre-processing of the models in order to accurately identify fiducial points, and to correctly align models in the reference system. In particular, the problem emerged in locating fiducial points in models of high resolution will be addressed by using interpolating polynomials of higher degree or the surfaces defined by Bèzier. As a further step, we aim to reformulate the representation in a local reference system.

A final observation regards the way models of heads in the SHREC database are obtained. A first consideration is that the common practice in real applications for face recognition/authentication is to have scans only of the frontal facial surface that can be acquired with devices like laser or structured light scanners. Differently, in SHREC whole head models are provided. This, and the fact that a morphing operation is used to derive head models from deformable templates, make the test set somewhat biased towards methods that use morphing to evaluate models similarity. In general, these methods are capable to use the entire head information, have a small sensitivity to models rotations and have a good accuracy, but at the cost of a very high computational complexity. Other solutions, which are more targeted for recognition or authentication in real applicative contexts, and so use only the facial surface for similarity evaluation, can be impaired by the SHREC models.

In addition, since the evaluation is targeted to similarity retrieval, the retrieval time is a factor of performance highly relevant. Of course, registration or morphing methods can provide very good similarity measures but at the cost of prohibitive computational time to be applied in large databases for real time recognition applications. As so, these methods are more adequate for an authentication scenario, where a person claims its identity and comparison between only two models suffices.

Finally, in real 3D face databases only small differences in the alignment of models are present. In the SHREC database, instead, high differences in the models alignment are artificially created. Of course, this is useful for the investigation of the behavior of different approaches with respect to this feature, but mixing alignment variations with face differences makes it difficult to evaluate the real potentiality of the approaches with respect to the sensitivity to the face features alone. Since not all methods are invariant with respect to alignment, maybe the evaluation of robustness with respect to rotation of same faces and with respect to face variations should be evaluated separately. Differently, similarity results are difficult to be interpreted.

In the perspective to make competitions like SHREC of ever increasing usefulness and popularity, maybe some of the previous considerations could be addressed in constructing the test database and in performing the final evaluation.

5. Acknowledgments

This work is partially supported by the Information Society Technologies (IST) Program of the European Commission as part of the DELOS Network of Excellence on Digital Libraries (Contract G038-507618).

We would like to thank M.Migliorini and S.Paladini to implement the code for the identification of fiducial points, and the models alignment.

References

[1] S. Berretti and et. al. Modeling spatial relationships between 3d objects. In *Proc. IEEE International Conference on Pattern Recognition*, volume 1, pages 119–122, Hong-Kong, China, August 2006.

[2] K. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition. *Computer Vision and Image Understanding*, 101(1):1–15, January 2006.

[3] L. Farkas and I. Munro. *Anthropometric Facial Proportions in Medicine*. Thomas Books, Springfield, Illinois, 1987.

[4] R. Kimmel and J. Sethian. Computing geodesic paths on manifolds. *Proc. National Academy of Science, USA*, 95(15):8431–8435, July 1998.

[5] A. Razdan and M. Bae. Curvature estimation scheme for triangle meshes using biquadratic bézier patches. *Computer-Aided Design*, 37:1481–1491, 2005.

[6] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Survey*, 35(4):399–458, December 2003.

An ICP-Based Approach to Retrieving Similar 3D Face Models

Tung-Ying Lee
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
d9562818@oz.nthu.edu.tw

Shang-Hong Lai
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
lai@cs.nthu.edu.tw

Abstract

In this paper, we describe two methods used in Shape Retrieval Contest of 3D Face Models. The first applies Iterative Closest Point Algorithm (ICP) to register two reduced point clouds. Each reduced point cloud is downsampled based on the distances for all points to their mean. The second method takes local linear fitting error into account for retrieval. Finally, the evaluation scores for these two methods on 3D face model retrieval are shown.

1. Introduction

Recently, the number of available 3D shape models is rapidly increasing because of the advances in 3D modeling and acquisition. Researchers have focused not only on traditional content-based image/video retrieval, but also on 3D shape retrieval. Many 3D search engines and shape matching methods are developed. A good survey was presented by Tangelder et al. in [1]. In the Shape Retrieval Contest of 3D Face Models, the 64 queries are issued on a huge database with 1516 three-dimension face models. Although these face models are generated by a morphable model [2], the correspondence between two face models are not kept. Efficient and effective matching methods are needed due to the size of database.

For achieving computational efficiency, we apply the iterative closest point (ICP) algorithm [3] on the reduced point clouds. For effectiveness, the local property of the surface is considered. The evaluation scores show that the first implementation has better performance in precision than the second one.

In section 2, some preliminaries and the two methods are described in details. Some experimental results are shown in section 3. Finally, the evaluation scores and conclusion are given in the last section.

2. Preliminaries and Method Description

2.1. Preliminaries

The methods for comparing two 3D models can be roughly categorized into two types. One type is to first register one model to the other model and computing a distance between the two registered models. The other type uses invariants of surfaces under rotation and translation.

The registration between two models involves aligning two models in the same coordinate after performing certain transformation on one of them. Given a few correspondence point pairs between two models, we can easily compute the registration between these two models. Without correspondence pairs, the model registration problem is much more difficult.

The ICP method can be used to align two models whenever the correspondence pairs are given or not. For each point at one model, it iteratively finds the nearest point at the other model.

The invariants can be used in registration or matching if they are Euclidean invariants that are invariant to 3D motion. Curvature, moment, and spherical harmonic invariants are usually used. Sharp et al. [4] integrate these invariants to the ICP algorithm.

2.2. The First Method (RUNNR 1)

Our retrieval system performs the following algorithm and lists query results in runfile1 (RUNNR 1). The flow chart of the algorithm is shown in Figure 1. First, a face model is served as the reference model R and is downsampled by a factor of 10 (this downsampled version is denoted by R_d). All face models (query and data models) are also downsampled by a factor of 10. In order to downsample models efficiently, we adopt a simple downsampling method. Given a model, all points in a model are sorted by their distance to the mean of this model and their indices are reassigned according this list (starting from 1). All points whose new indices are divided by 10 are collected into the downsampled version of this model. The system performs ICP for aligning a downsampled model (a query model or a data model) to reference R_d. Hence, all models have the same pose. When the system is given a query and a data model in DB, a point of the data model corresponds to the nearest point in the query (many-to-one mapping). The dissimilarity between two models is estimated by summing distances over all corresponding pairs.

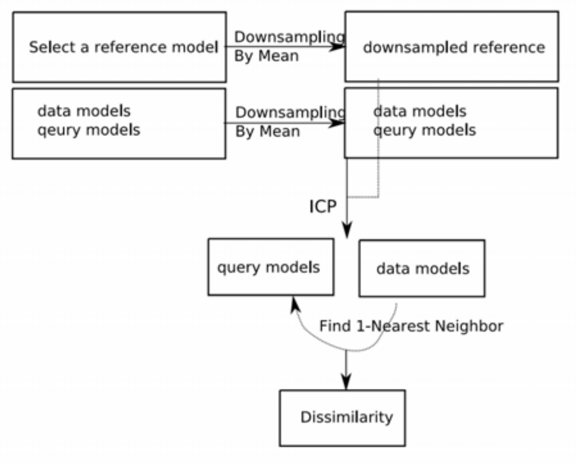


Figure 1. The flow chart of the first method.

2.3. The Second Method (RUNNR 2)

This method is based on the model registration method in RUNNR 1. Moreover, we introduce a feature in this run. We not only care about the distances of pseudo correspondence pairs but also the local property of face models. (We call these correspondence pairs, calculated by ICP, with pseudo ones because the faces are not the same and these correspondence pairs are usually exactly correct.) Our feature value at a vertex p is a minimum error $\mathbf{E}(p)$ which represents the least square error of fitting the neighbors of p with a plane. The neighbors of a point are obtained by the face structure. The face structure in a model consists of many polygons which are represented by several vertices. For a vertex p, we select a set F_p of all faces containing the vertex p. The neighbors of p are all vertices that are contained in a face of F_p. A weighted distance of a correspondence pair (p,q) is **abs**($\mathbf{E}(p)$-$\mathbf{E}(q)$)***dis**(p,q) where **abs** means absolute value and **dis**(p,q) is the distance between p and q. The dissimilarity between two models is estimated by summing weighted distances over all corresponding pairs. The flow chart of the second method is shown in Figure 2.

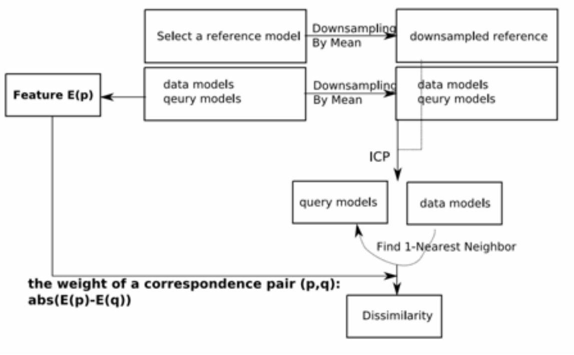


Figure 2. The flow chart of the second method.

3. Experimental Results

3.1. Downsampling by Mean

In the first step, we downsample all models by a factor of 10. An example (including original and downsampled face models) is depicted in Figure 3. The rebuilded surface is constructed by the downsampled version with ball size 3 (in Meshlab [5] Setting). Some model are properly downsampled, such as D0004 depicted in Figure 4. Note that there are a few selected points in the nose.

3.2. Aligning Data Models to a Reference Model

In the second step, all models are aligned to a reference model. This step has great influence on performance. Three queries out of sixty-four queries are not registered correctly. Our method failed in these three queries, Q1046, Q1048, Q1057. They are shown in Figure 5. Their average dynamic precisions are about 4%; the other queries like Q1003~Q1007 have high ADPs from 97% to 75%. The Q1003 has 100% precision and 97.3765% ADP. Note that all highly relevant items are registered correctly and shown in Figure 6.

4. Evaluation Scores and Conclusion

4.1. Evaluation Scores

Ranking results based on mean ADP is summarized in Table 1.

Table 1. Mean Average Dynamic Precision.

Rank	RunFile	Value
0	ideal	1
1	ter Haar run1	0.680894
2	Tung-Ying Lee run 1	0.651938
3	ter Haar run3	0.640431
4	ter Haar run2	0.618641
5	Tung-Ying Lee run 2	0.440214
6	Berretti run1	0.301797
7	Berretti run2	0.298608
8	Berretti run3	0.290289
9	Berretti run4	0.287672
10	Berretti run5	0.248614

4.2. Conclusion and Discussion

The first method basically performs well. However, the wrong pose is a tough problem. The good feature is needed to correct the wrong pose. The linear fitting error in the second method may not be suitable in downsampled point clouds. In the future, we may use this kind of simple invariants in a part of dense point clouds.

References

[1] J.W. Tangelder and R.C. Veltkamp, “A survey of content based 3d shape retrieval”, Int’l Conf. on *Shape Modeling and Applications (SMI)*, Genova, Italy, June 2004, pp. 145-156.

[2] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces”, Int’l Conf. on *Computer Graphics and Interactive Techiques (SIGGRAPH)*, 1999, pp. 187-194.

[3] P.J. Besl and N.D. McKay, “A Method for Registration of 3-D Shapes”, In IEEE Trans. on *Pattern Analysis and Machine Intelligence (PAMI)*, 1992, 14(2), pp. 239-256.

[4] G.C. Sharp, S.W. Lee, and D.K. Wehe, “ICP Registration Using Invariant Features”, IEEE Trans. on *Pattern Analysis and Machine Intelligence (PAMI)*, 2002, 24(1), pp. 90-102.

[5] MeshLab Visual Computing Lab – ISTI – CNR http://meshlab.sourceforge.net/

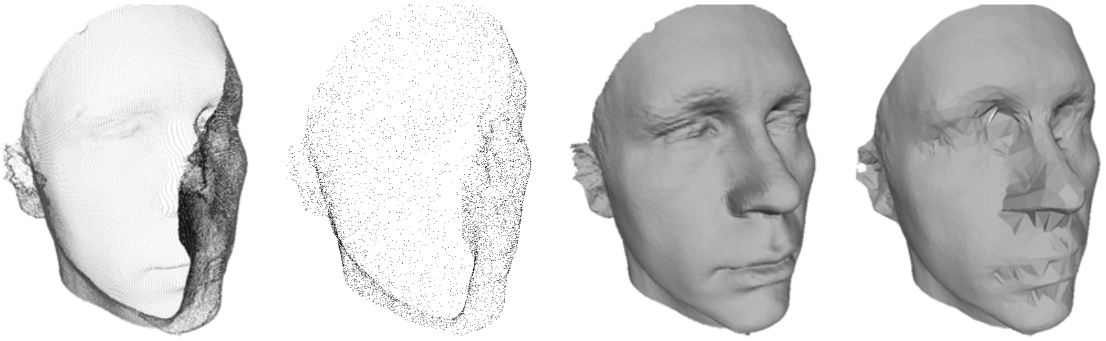


Figure 3. An original point cloud, a downsampled version, an original surface, a rebuilt surface from a downsampled version (D0001)

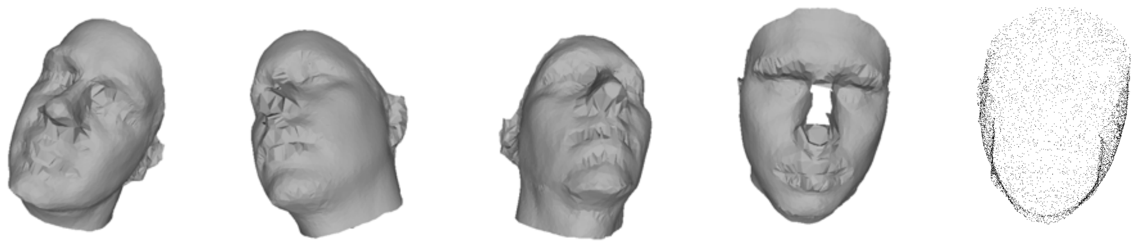


Figure 4. Rebuilt surfaces from downsampled versions (D0000, D0002, D0003, D0004) and the downsampled point cloud (D0004)

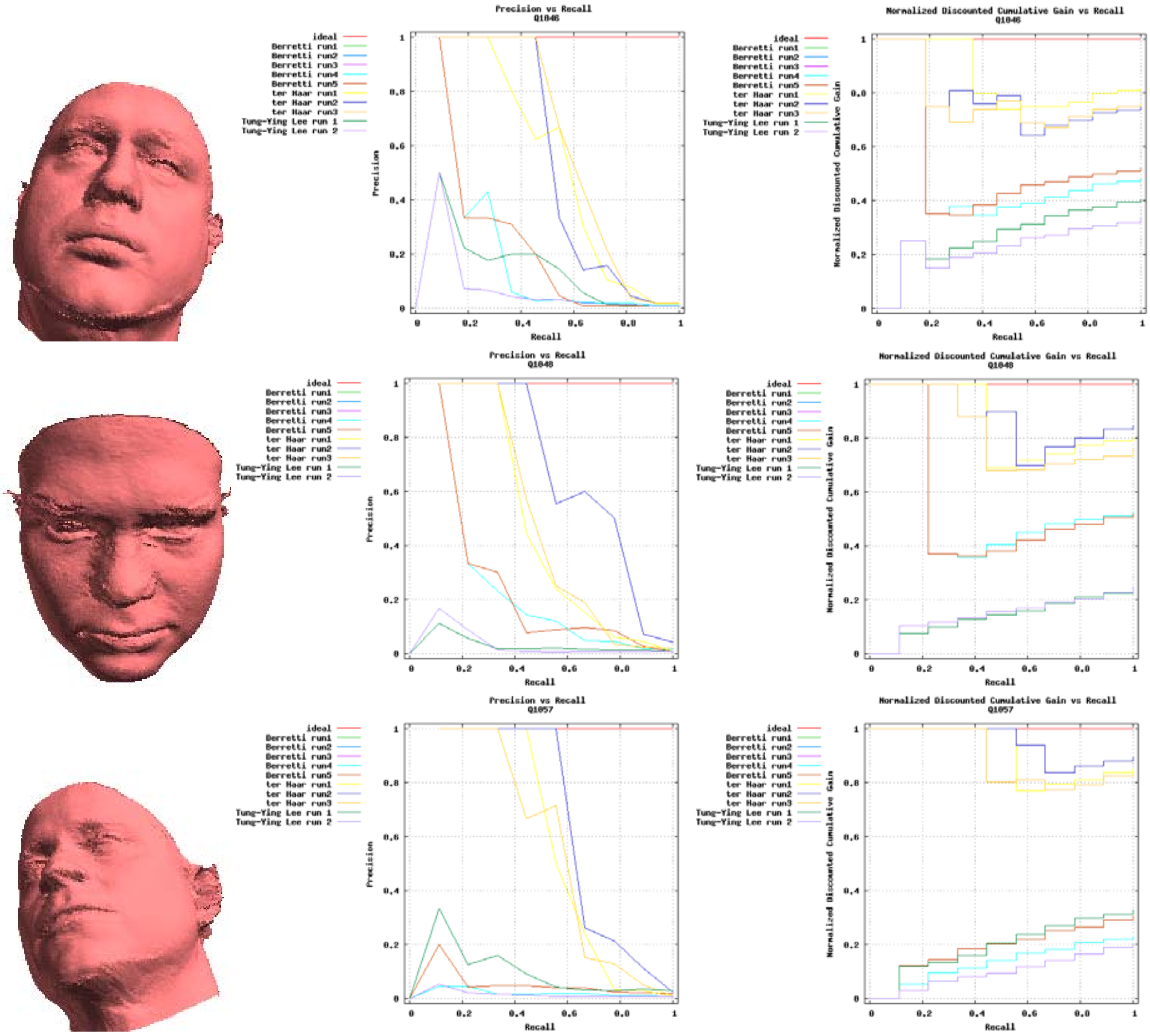


Figure 5. After registering to the reference model (Q1046, Q1048, Q1057)

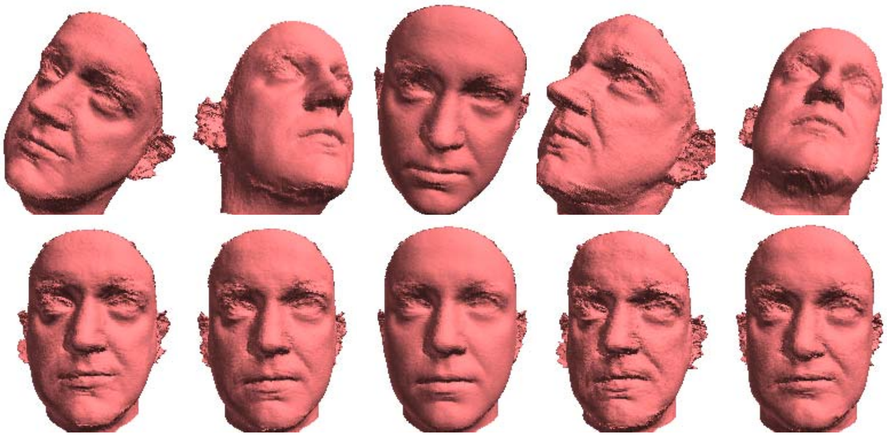


Figure 6. Before/After registering to the reference model (D1112, D0456, D0947, D0624, D1123)

3D Face Retrieval using Advanced Profile Matching

Frank B. ter Haar and Remco C. Veltkamp
Department of Information and Computing Sciences
Utrecht University, the Netherlands

Abstract

Many 3D face matching techniques have been developed to perform face recognition. Among these techniques are several variants of 3D profile matching, which are techniques that reduce the amount of face data to one or a few 3D curves. 3D profile curves proved their use in 3D face recognition, but the selection of the optimal profile or set of profiles and the best way to match them is still under exposed. In our work we decribe a new pose normalization method that finds the best fit for a nose template and the tip of the nose as well. A set of profile paths over the 3D surface starting at the tip of the nose is extracted. These profiles are sampled according three different sampling strategies and then matched assuming correspondence between the samples. Retrieval results on a large database of 1516 face models with difficult sets of relevant classified models shows Mean Average Dynamic Precisions upto 0.68.

1. Introduction

Many techniques have been developed to compare human face data using its 3D shape information obtained with a 3D scanner. Because the 3D shape information of two scans from the same person may vary in resolution, accuracy, pose, coverage and emotion it is difficult to develop a technique that is generally applicable. However, the use of 3D shape information is believed to have a useful contribution to the field of face recognition, because traditional face recognition based on 2D images suffer from the same variances and has to deal with illumination and scale diversity as well. In this work we focus on 3D face matching with the use of 3D profiles, a technique with high potential which has been applied in a very limited manner.

Contribution Our contribution includes a new pose normalization technique and an advanced 3D profile matching technique. Pose normalization is performed by fitting a nose template to the scan data and using the inverse transformation of the best fit to normalize the pose.

The tip of the nose is extracted from the scan data automatically and is used as the origin from which a set of profile curves is extracted in different directions. Such sets of profiles are used to determine the similarity between faces. The profiles are sampled according three different sampling strategies. These samples are assumed to correspond when two profiles from different faces are matched, this way the amount of potential correspondences is limited.

2. Material

In this work we compare 3D faces using surface curves sampled according different strategies. To test the effectiveness of these different strategies for face matching, we applied them as different runs on the database from “SHREC 2007 - Shape Retrieval Contest of 3D Face Models” (see [4] for more details). For each run relevant scores are computed based on the retrieval of relevant classified faces for 64 different queries.

3. Method

3.1. Overview

To normalize the pose of a 3D face model and to detect its tip of the nose, we apply template matching with a nose template in high curvature areas. Starting from the tip of the nose we then extract profile curves in different directions, which are sampled according three different sampling schemes. To determine the similarity of two different faces, we match the profiles of the two faces using the samples as predefined correspondences.

3.2. Pose normalization

In this paper we describe a pose variant method for 3D face recognition. Therefore we need to normalize the pose of the 3D face before we can do face matching. This is done by matching a nose template in high curvature areas of the face. Each vertex of a face model can be considered as a potential location for the initial placement of our nose template. Generally, the tip of the nose is a location with high (positive) curvature, which makes it possible to exclude a large number of potential placements (i.e. vertices) based on a simple curvature threshold heuristic.

At first, we need to determine the areas of high curvature which presumably contain the tip of the nose, but also the areas around the ears, eyebrows, chin, etc. To extract these areas we apply Rusinkiewicz's curvature estimation algorithm [2] to obtain an estimation of the mesh's curvature at each vertex. Secondly, we select the t_c (t_c=8) percent of vertices with the highest curvature as potential nosetip locations to place our nose template on.

For each of the selected vertices, we have its position (p_v) and normal direction (n_v). For our *nose template* we determine the position of the nosetip (p_t) and its normal direction (n_t). To determine the optimal location for the nose template in the scan data, we place the template on each selected vertex and apply the Iterative Closest Point (ICP) algorithm [1] to obtain the best local fit based on the Minimum Least Squares (MLS) distance of the templates vertices to their closest points in the scan data. Because the ICP algorithm is known to get stuck in local minima, the initial placement of the template is important. Therefore we place the nose template with p_t on p_v and with n_t aligned to n_v. This limits the degrees of freedom to an unknown rotation around the normal n_t. Because the nose template has a characteristic shape, the unknown rotation around the normal n_t causes a problem. To overcome this problem we fit a *nosetip template* to the selected vertices first. This template covers only a small part around p_t of our nose template, which resembles a symmetric bump. Because of the symmetry we can fit this template to the scan data independent of the rotation around its normal n_t. Finally, the entire *nose template* is fit on the locations where the *nosetip template* obtained a good fit (i.e. low MLS error). The nose template is placed using eight different rotations (every 45°) around its normal n_t to obtain the optimal fit with the ICP algorithm. The inverse transformation matrix for the optimal fit is used to normalize the face's pose. The point in the scan data closest to p_t is defined as the tip of the nose used during profile extraction.

3.3. Profile extraction

The information we use to match faces is a set of N_p (N_p=180) profiles. We define a profile as a curve that starts from the tip of the nose and follows a path over the surface mesh with a fixt angle in the XY-plane. Such a path is defined by the intersection points of the mesh's triangles encountered along the way. Basically we extract a path (or profile curve) for every 360/N_p degrees in the XY-plane with the tip of the nose as origin. The path ends whenever the Euclidean distance between the current path location and the tip of the nose becomes larger than 90 mm.

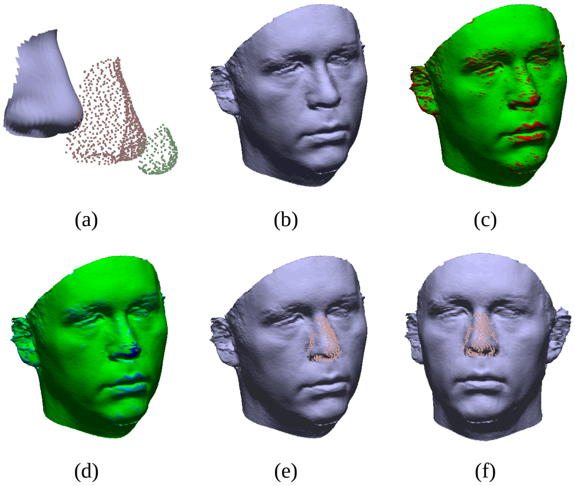


Figure 1. The visualization of our pose normalization steps. The applied nose (pink) and nosetip (green) templates. The original face (b) with high curvature areas (c), optimal (dark blue) locations for the nosetip template (d) and optimal fit for the nose template (e). The final normalized face (f).

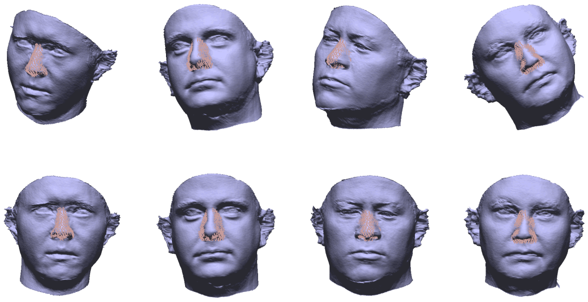
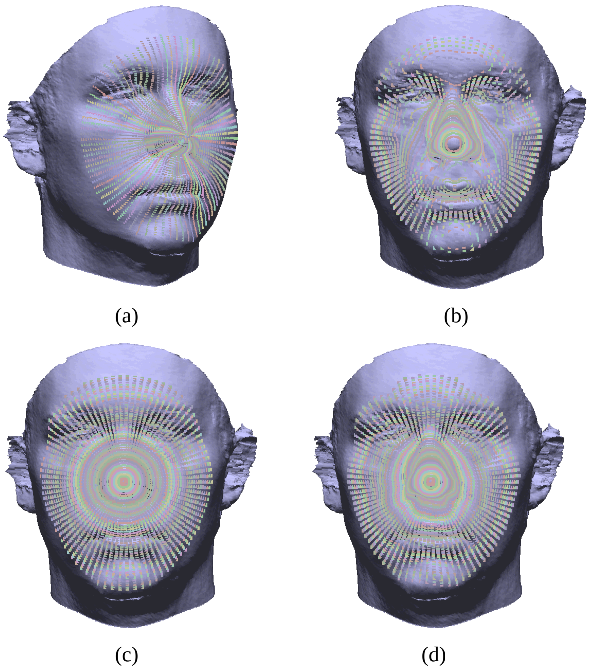


Figure 2. Pose normalization of different faces. Top: initial faces. Bottom: pose normalized faces with fitted nose templates.

Beyond this distance the chance of missing data or hair covering parts of the face increases. Before profile extraction the pose normalized face is centered with its nosetip in the origin, so that the extracted sets of profiles of two different faces are aligned.

3.4. Profile matching

A single profile defines a 3D curve over a surface starting from the origin. To compare two corresponding curves, that is, two curves extracted with the same XY-orientation


(a)

(b)

(c)

(d)

Figure 3. A set of 90 profiles (shown from a different view) are sampled using increasing (b) Z-, (c) XY-, and (d) C-distances resulting in different contour curves.

as shown in Figure 4, we select and match samples from both curves according three different sampling strategies: samples with similar (1) Z-distance, (2) XY-distance and (3) curve(C)-distance to the tip of the nose. Combining samples along all profiles with a similar sample-distance generates approximations of contours curves (see Figure 3), such as the Z-contour used for face recognition by Samir et al. [3].

The distance between a point p on curve P and a point q on curve Q, using the tip of the nose (p_{nt}=origin) and the Euclidean distance $e(p,q)$ is defined as:

$$d(p,q) = (e(p,p_{nt}) - e(q,p_{nt}))^2 = (|p| - |q|)^2$$

A sample $\mathbf{p}$ with a fixt Z-distance can consist of multiple points p and a single or no point at all for all three sampling strategies. In the distance measure for two curves we neglect corresponding samples without a profile intersection point, and only the best matching point in case of multiple sample points. The distance between two profiles sampled with N_s samples ($\mathbf{p_i}$ and $\mathbf{q_i}$) defined by either the Z-, XY-, or C-contours is then defined as the sum of best matching points for corresponding samples:

$$d(P,Q) = \sum_{i=1}^{N_s} min_{\forall p \in \mathbf{p_i}, \forall q \in \mathbf{q_i}} d(p,q)$$

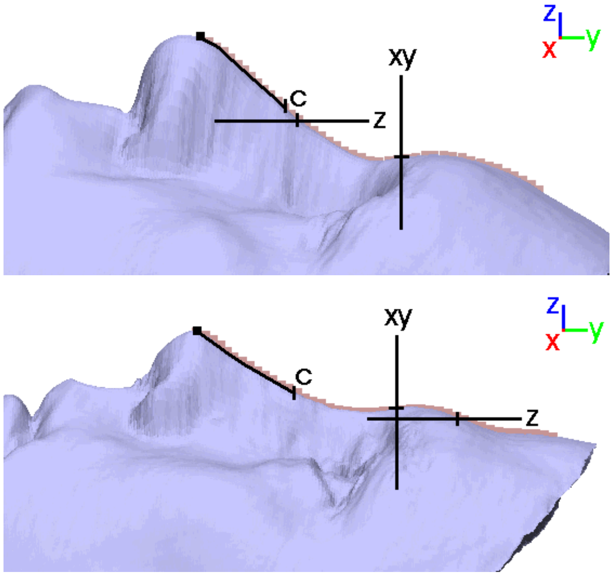


Figure 4. Example of an Z-, XY-, and C-sample on the profile curve from nose to forehead.

The advantage of this distance measure for two profiles is that it matches samples invariant to translations and rotations, which is a useful property when matching two non-corresponding profile curves as in Section 3.5. Furthermore, missing data has no influence and the appearance of multiple contours for Z-sampling is dealt with.

3.5. Face matching

To determine the similarity between two faces we pose normalize the faces and extract for each face a set of N_p profiles and for each profile a set of N_s samples. When two faces are similar we expect them to be pose normalized similarly, however, small misalignments can occur. Therefore, we match each profile of face A to a small subset of t_s (t_s=$\frac{30}{360}$) percent of the N_p profiles around the 'corresponding' profile of face B and retain the distance of the best matching profile. The distance measures of profiles from face A to face B and vice versa are summed and normalized to the number of profiles. Theoretically, the comparison can be made more robust to noisy profiles and/or emotion variance by selecting a percentage of best matching profile curves to describe the difference between two faces.

4. Results and evaluation

To prove the usefulness of our pose normalization and profile matching scheme using different sampling strategies, we applied our face matching system to the SHREC

Method	Identical 1st	MAPH	MAPR	MADP
C sampling (run1)	**63**/64	**0.6613**	0.4775	**0.6809**
XY sampling (run2)	43/64	0.5618	0.4536	0.6186
Z sampling (run3)	48/64	0.5945	**0.4782**	0.6404

Table 1. Retrieval results of our methods

3D Face database to retrieve the relevant faces for each query. For each normalized face in the database we extracted profiles in N_p=180 directions starting from the located tip of the nose. Each profile is then sampled with at most N_s=50 samples following the three different sampling strategies: C-distance (run 1), XY-distance (run 2), and Z-distance (run 3). The sample step size is fixt for each strategy, in such a way that all sampling strategies have approximately the same number of samples along a profile curve. We matched two faces using all N_p best matching profiles.

Some retrieval results of our experiment are shown in Table 1, namely the Mean Average Precision of both highly and all relevant faces (MAPH and MAPR), the Mean Average Dynamic Precision (MADP) and the number of identical query faces on top of the ranked list (Identical 1st). These results lead to the following observations. Firstly, the pose normalization is not perfect, otherwise all faces in the database identical to their query face would have a dissimilarity value equal to zero and thus retrieved as first element in the ranked list (Identical 1st). Secondly, the C-sampling strategy is more robust to small misalignments, because the number of identical faces retrieved first is significantly larger than for the other two strategies. Thirdly, the highly relevant classified faces are easier to retrieve than the marginally relevant faces, because the MAPR is lower than the MAPH for all strategies.

4.1. Discussion

When we compare the results of our methods to those of other methods of the 3D face retrieval contest of SHREC'07 we see that our methods perform relatively well based on the performance measures described earlier (see Table 2). The fact that all methods except one have trouble retrieving the identical models first, shows that pose normalization is a highly difficult task. Our XY-sampling method (run2) returns 43 identical models first, while our C-sampling method (run1) returns almost all (63) identical models on top of the list. This raises the question whether 3D face recognition methods should focus on perfect pose normalization or on methods that are more robust or even completely invariant under rotations.

A disadvantage of our method is that extracting a profile path requires a manifold mesh with proper topology and preferably no holes. Our method can be extended to overcome these problems by defining each profile as the intersection curve of the face with a plane, which is less efficient.

Method	Identical 1st	MAPH	MAPR	MADP
ideal	64/64	1.0	1.0	1.0
ter Haar run1	63/64	0.6613	0.4775	0.6809
Tung-Ying Lee run 1	50/64	0.6159	0.4785	0.6519
ter Haar run3	48/64	0.5945	0.4782	0.6404
ter Haar run2	43/64	0.5618	0.4536	0.6186
Berretti run3	35/64	0.2253	0.1741	0.2903
Berretti run1	35/64	0.2320	0.1778	0.3018
Berretti run4	34/64	0.2213	0.1735	0.2877
Berretti run2	34/64	0.2288	0.1777	0.2986
Tung-Ying Lee run 2	32/64	0.3694	0.2976	0.4402
Berretti run5	20/64	0.2073	0.1598	0.2486

Table 2. Retrieval results of SHREC 2007

The retrieval results shown in Table 2 are in general not very high. Besides the difficulty of pose normalizing the 3D face models, we expect the large embedding of the database (most face recognition databases are small) and the dissimilarity of the relevantly classified faces are the main contributions to this outcome of the retrieval scores.

5. Concluding remarks

In this work we proposed a new pose normalization technique and an advanced 3D profile matching scheme. We performed pose normalization by fitting a nose template to the scan data and using the inverse transformation of the best fit to normalize the pose. The fitted template is used to extract the tip of the nose. Starting from the tip of the nose we extracted a set of profile curves, which were sampled according three different sampling strategies, namely: C-sampling, XY-sampling, and Z-sampling.

Results from the 3D face retrieval contest show that C-sampling has the highest performance based on the number of identical query faces retrieved first, the mean average performance of highly relevant faces, and the mean average dynamic precision.

Acknowledgements

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

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Trans. PAMI*, 14(2):239–256, 1992.

[2] S. Rusinkiewicz. Estimating Curvatures and Their Derivatives on Triangle Meshes. In *3DPVT*, pages 486–493, 2004.

[3] C. Samir, A. Srivastava, and M. Daoudi. Three-Dimensional Face Recognition Using Shapes of Facial Curves. *IEEE trans. PAMI*, 28(11):1858–1863, 2006.

[4] Utrecht University. SHREC 2007 - Shape Retrieval Contest of 3D Face Models at `http://give-lab.cs.uu.nl/shrec/shrec2007`, Apr 2007.