The Princeton Shape Benchmark

Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser
Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540
(pshilane,min,mkazhdan,funk)@cs.princeton.edu

Abstract

In recent years, many shape representations and geometric algorithms have been proposed for matching 3D shapes. Usually, each algorithm is tested on a different (small) database of 3D models, and thus no direct comparison is available for competing methods.

In this paper, we describe the Princeton Shape Benchmark (PSB), a publicly available database of polygonal models collected from the World Wide Web and a suite of tools for comparing shape matching and classification algorithms. One feature of the benchmark is that it provides multiple semantic labels for each 3D model. For instance, it includes one classification of the 3D models based on function, another that considers function and form, and others based on how the object was constructed (e.g., man-made versus natural objects).

We find that experiments with these classifications can expose different properties of shape-based retrieval algorithms. For example, out of 12 shape descriptors tested, Extended Gaussian Images [13] performed best for distinguishing man-made from natural objects, while they performed among the worst for distinguishing specific object types. Based on experiments with several different shape descriptors, we conclude that no single descriptor is best for all classifications, and thus the main contribution of this paper is to provide a framework to determine the conditions under which each descriptor performs best.

Keywords: shape retrieval, geometric matching, shape database, benchmarks.

1 Introduction

Shape-based matching and retrieval from databases of 3D polygonal models is a fundamental problem in computer vision, mechanical CAD, archeology, molecular biology, paleontology, medicine, computer graphics, and several other fields [3, 25].

Despite decades of research on 3D shape representations and matching algorithms [20, 33], there still are no standard ways of comparing the results achieved with different methods. Usually, computed match results are evaluated according to how well they correlate with human-generated classifications. However, it seems that each research group has its own database of 3D models, own classifications, own suites of tests, and own metrics of success. Moreover, few publications contain results of tests comparing several approaches on the same data [4, 10, 19, 36].

In this paper, we describe the Princeton Shape Benchmark (PSB), a publicly-available database of 3D models, software tools, and a standardized set of experiments for comparing 3D shape matching algorithms. The database contains 1,814 polygonal models collected from the World Wide Web and classified by humans according to function and form. It includes a set of hierarchical classifications, separate training and test sets, annotations for each model, and a suite of software tools for generation, analysis, and visualization of shape matching results.

An interesting feature of the benchmark is that it provides mechanisms to define multiple classifications and query sets that can be used to differentiate properties of shape matching algorithms. Our base classification combines both semantics and shape hierarchically. For instance, a model representing a table may be part of the "round table with a single leg" class, as well as the coarser "round table," "table," "furniture," and "man-made" classes. The benchmark also includes several query sets intended to differentiate how matching algorithms work on models with specific properties (e.g., high depth complexity). By evaluating retrieval results with these different classifications and queries, it is possible to expose the differences between different shape matching algorithms. So, rather than simply saying "method X is better than method Y on average," we can now say "method X is better for this type of object, and method Y is better for that type of object, etc."

The main contribution of this paper is the proposed framework for comparison of shape matching algorithms. We demonstrate its use by exploring the differences between twelve shape descriptors, including D2 shape distributions [23], Extended Gaussian Images [13, 15], Shape Histograms [1], Spherical Extent Functions [27, 35], Gaussian Euclidean Distance Transforms [16], Spherical Harmonic Descriptors [16], and Light Field Descriptors [4].

In short, we find that no single shape descriptor performs best for all classifications, and no single classification provides the best evaluation of all shape descriptors. From this

result, we conclude that it is only possible to understand the differences between shape descriptors by looking at the results of several experiments aimed at testing specific properties. The Princeton Shape Benchmark provides a standardized framework for this type of experimentation.

2 Related Work

The benefits of benchmarks have been well-demonstrated in many fields. For example, in computer architecture, the SPEC benchmarks [28] have been used successfully to compare processor performance. In text document retrieval, the Smart Collection [26] and TREC database [31] provide standard benchmarks. In computer vision, benchmarks are available for handwriting recognition (e.g., [18]), face recognition (e.g., [5]), and several other image analysis tasks [6]. There are even benchmark databases for specific types of 3D data – e.g., computer-aided design parts [9] and protein structures [2].

Unfortunately, no standard benchmarks are available for matching of 3D polygonal models representing a wide variety of objects. Instead, several research groups have independently gathered databases of 3D models, generated different classifications, run different sets of tests, employed different metrics to quantify performance, and compared different shape descriptors.

Table 1 shows statistics for several 3D model databases currently in use for shape matching experiments. For each database, the table shows the total number of 3D models in the database, the number of classes, the number of models that have been classified, and the percentage of classified models in the largest class. Also, estimates of what percentage of classified models belong to different object types (vehicle, household, animal, plant, architecture) appear in Table 2. The bottom row of each table shows statistics for the Princeton Shape Benchmark for comparison. From these statistics, we make several observations.

Database	Num Models	Num Classes	Num Classifi ed	Largest Class
Osada [23]	133	25	133	20%
MPEG-7 [38]	1,300	15	227	15%
Hilaga [12]	230	32	230	15%
Technion [19]	1,068	17	258	10%
Zaharia [39]	1,300	23	362	14%
CCCC [35]	1,841	54	416	13%
Utrecht [30]	684	6	512	47%
Taiwan [4]	1,833	47	549	12%
Viewpoint [10]	1,890	85	1,280	12%
PSB [this paper]	6,670	161	1,814	6%

Table 1. Summary of previous 3D model databases.

First, most previous databases contain a small number of classified models. For example, the Osada database [23], which has been used in experiments by several research groups (e.g., [30]), contains only 133 models. Some of them appear in classes with only 2 other models, which makes it difficult to acquire statistically significant results in classification experiments. In other cases, the total number of 3D models in the database is quite large (> 1800), but only a small fraction of them are included in the classification. For instance, the MPEG-7 database [38] contains 1,300 VRML models in all. But, only 227 (18%) of them are included in labeled classes, while the vast majority of models are lumped into a "miscellaneous" class that provides only "background noise" during shape matching experiments. To our knowledge, the only set of more than 1000 classified 3D polygonal models used for shape matching experiments is the Viewpoint database [34], as described in [10]. However, it is not available to the general public, and it is expensive to purchase, which makes its use as a standard benchmark problematic.

Second, most 3D model databases contain a limited range of object types and/or are dominated by a small set of object classes (see Table 2). For example, the Viewpoint database [10] contains only household objects, and the Utrecht database [30] contains mainly vehicles among its classified models. Even databases that have a wide variety of objects often contain a few classes with a disproportionately large number of models. For example, the MPEG-7 database contains 50 (22%) models representing letters of the alphabet among its 227 classified objects, and the Osada database contains 27 (20%) airplanes out of 133 objects. Of course, these large classes significantly bias (micro-)averaged retrieval results.

	Vehicles	Furniture	Animals	Plants	Household	Buildings
Osada [23]	47%	12%	12%	0%	24%	0%
MPEG-7 [38]	12%	0%	14%	13%	0%	7%
Hilaga [12]	12%	0%	23%	2%	12%	0%
Zaharia [39]	35%	0%	7%	7%	11%	0%
CCCC [35]	33%	13%	21%	5%	25%	0%
Utrecht [30]	73%	0%	0%	0%	0%	0%
Taiwan [4]	44%	13%	0%	0%	36%	0%
Viewpoint [10]	0%	42%	1%	0%	50%	0%
PSB [this paper]	26%	11%	16%	8%	22%	6%

Table 2. Types of objects found in previous 3D model databases (shown as percentages of classified models).

Third, current 3D model classifications have significantly different granularities. Some databases have classes with large, diverse sets of objects (e.g., "Kitchenware" [12]), while others have very small and specific classes (e.g., "motorcycles with 3 wheels" [39]). For example, the National Taiwan University database [4] has a single class containing all types of seats (dining room chairs, desk chairs, patio chairs, sofas, recliners, benches, and stools), while the Viewpoint database [10] has a separate class for each specific type. This difference in classification granularity can have an impact on retrieval and classification results, as significant differences between retrieval methods may be masked by classifications that are too coarse or too fine.

Finally, many 3D databases have classifications that mix function and form. For example, the MPEG-7 database contains several classes that group objects with similar semantics (e.g., "buildings"), while others group objects based solely on their shapes (e.g., the "aerodynamic" class contains fish, helicopters, and airplanes). Similarly, the Hilaga database [12] contains some classes corresponding grossly to functions (e.g., "Machine") and others corresponding directly to shapes (e.g., "Stick", "Donut", "Sphere", and "Many Holes"). Results achieved over these disparate class types are averaged together, making it difficult to draw specific conclusions about why and when a shape matching method works well.

3 Overview

The Princeton Shape Benchmark provides a repository of 3D models and software tools for comparing shape matching algorithms. The motivation is to promote the use of standardized data sets and evaluation methods for research in matching, classification, clustering, and recognition of 3D models.

Version 1 of the benchmark contains a database of 1,814 classified 3D models collected from 293 different Web domains. For each 3D model, there is an Object File Format (.off) file with the polygonal surface geometry of the model, a textual information file containing meta-data for the model (e.g., the URL from whence it came), and a JPEG image file containing a thumbnail view of the model. We expect larger databases to be available in future versions.

In addition to the database of 3D models, the benchmark provides guidelines regarding its use. For instance, the 3D models are partitioned equally into training and test sets. The benchmark requires that algorithms be trained only on the training set (without influence of the test set); and then, after all exploration has been completed and all algorithmic parameters have been frozen, results should be reported for experiments with the test set.

In order to enable evaluation of shape matching algorithms for retrieval and classification tasks, the benchmark includes a simple mechanism to specify partitions of the 3D models into classes. In Version 1, we provide a hierarchical classification for 1,814 models (907 from the training set and 907 from the test set). At its finest granularity, this classification provides a tight grouping of objects based on both function and form. For example, there is a class for "birds in a flying pose" in the test database. Yet, it also includes a hierarchy of classes that reflects primarily the function of each object and secondarily its form. Continuing with the example, there are classes for "birds", "flying creatures," and "animals" at coarser levels of the hierarchy. Note that every level of the hierarchy is useful for a different type of evaluation.

Since arbitrarily many semantic groupings are plausible for a given set of 3D models, the benchmark provides a flexible mechanism for specifying multiple classifications. It also includes a method for averaging over queries for models with certain geometric properties (e.g., "roughly spherical"). The differences in matching results achieved with respect to these different classifications and queries yield interesting insights into the properties of the shape retrieval algorithms being tested (e.g., algorithm X works better on round objects, but worse on elongated ones), and the combined results of multiple classifications provide a much more complete view of the differences between competing algorithms.

To standardize analysis of shape matching experiment results, the benchmark includes free source code for evaluation and visualization of 3D model matching scores. For instance, there are programs for: (1) generating precision-recall plots, (2) computing several retrieval statistics (e.g., nearest neighbor, 1st and 2nd tier, discounted cumulative gain, etc.), (3) producing color-coded similarity matrices, and (4) constructing web pages with thumbnails of the best ranked matches for a given query model. These programs provide a standard toolbox with which researchers can compare results of independently run tests in a consistent manner.

In summary, the benchmark provides a flexible framework for comparing shape matching algorithms. The remainder of the paper describes many of the design decisions and issues that were addressed during its construction. Specifically, detailed descriptions of how our database was acquired, classifications were constructed, and models were annotated appear in Sections 4-6. Section 7 describes our software tools for evaluating matching results, and Section 8 presents experimental results obtained during tests with several different shape descriptors, classifications, and databases. Finally, Section 9 summarizes our findings and proposes topics for future research.

4 Acquisition

The 3D models in the PSB were acquired from the World Wide Web by three automated crawls over a two year period. This section describes how they were found, processed to remove duplicates, converted to a common file format, and organized to form a database.

The first crawl was performed in October 2001 and targeted VRML files only. It began with the results of search engine queries for web pages linking to files with extension ".wrl" and ".wrz" and then crawled outward from those pages in a breadth-first fashion. The crawl ran for 48 hours and downloaded 22,243 VRML files from 2,185 different Web sites [21].

The second crawl was executed in August 2002 and targeted VRML, 3D Studio, Autocad, Wavefront, and Lightwave objects, both in plain links as well as in compressed archive files (".tar" and ".zip"). Unlike VRML, the other formats were not designed to be used on the web and often are contained within compressed archives, so they typically cannot be located simply by file name extension. Instead, the second crawler searched for them using a focusing method, where web sites were crawled in priority order according to the number of pages already downloaded from that site containing 3D models. The crawl ran for 2

days and 16 hours and resulted in 13,217 3D model files and 5,539 compressed archive files containing 3D models. After expansion of archive files, there were 20,084 model files retrieved from 455 different sites [21].

The third crawl was executed in August 2003 and targeted models from known 3D model repositories (e.g., 3dcafe.com and avalon.viewpoint.com). The crawl ran for approximately 5 hours and resulted in 1,908 3D models in a variety of formats, downloaded from 16 different web domains.

These three crawls provided 44,235 model files. 3,763 of the models found in the second crawl were ignored because they had URLs in common with ones found in the first crawl. A further 6,863 models were thrown out because they contained no geometry or could not be parsed by our conversion software [22]. 15,035 more models were culled because their shapes were exact-duplicates or near-duplicates of some other model in the database. For example, we found multiple copies of the same model at different URLs (e.g., 483 spheres), multiple levels of detail for the same object, and different colors/textures for models with the same geometry. Finally, 11,904 models were eliminated manually because they came from application domains outside the scope of our benchmark. Specifically, we kept only models of "every-day objects," and threw out molecular structures, CAD parts, data visualizations, and abstract geometric shapes. The remaining 3D models form the database for our shape benchmark. In all, there are 6,670 unique models acquired from 661 distinct Web domains.

All the remaining models were converted to the Object File Format (.off), a simple-to-parse polygonal format designed by the University of Minnesota Geometry Center. During the conversion process, all color, texture, and scene graph information was eliminated, leaving a single indexed face set comprising a list of vertices (x,y,z) and a list of polygons (v1, v2,...). We chose to make only these simple files available in the first version of the benchmark to focus matching experiments on geometric surface information only.

5 Classification

The PSB benchmark splits the 3D model database into training and test sets and partitions both test sets into classes (e.g., telephones, dogs, etc.) that can be used as labels in shape matching, retrieval, and classification experiments. In this section, we first explain how the models are partitioned into classes. Then, we discuss how training and test sets were formed. Finally, we describe the mechanisms provided for creating alternative classifications.

5.1 Base Classification

We manually partitioned the models of the benchmark database into a fine-grained set of classes. During this process, our goal was to create clusters of objects primarily based on semantic and functional concepts (e.g., furniture and table) and secondarily based on shape attributes (e.g., round tables). We use the hierarchical nature of this grouping strategy to form classifications at multiple granularities.

The steps used to produce our *base classification* proceeded as follows. First, we rendered thumbnail images for all 6,670 3D models and stored them in a single directory of a file system. Then, two students used Windows Explorer to create directories representing object classes and to move the thumbnail image files into the directories to indicate membership in the class. This process was executed iteratively until each class represented an atomic concept (e.g., a noun in the dictionary) and could not be partitioned further. Then, where appropriate, a few classes were further partitioned based on geometric attributes (e.g., "humans with arms out" versus "humans with arms down"). No textual information besides an integer model ID was available to the students (e.g., file names were hidden). So, we believe the students were not biased by auxiliary information during the formation of classes. The result of this process was a set of 1,271 classes partitioning the 6,670 models.

Many of the classes contained too few models to be included in meaningful experiments. For example, there were only two drill presses and three fire hydrants. So, we manually selected 161 classes, each containing at least four models, to be included in the first version of the benchmark (the other classes will be included in later versions). We also eliminated models from the largest classes (e.g., fighter jets and humans) so that every class contains at most 100 members ($\sim$6% of the classified models). The net result is our base classification, a set of 161 classes containing a total of 1,814 models.

5.2 Training and Test Sets

We then partitioned the models of the base classification into training and test sets. Our goal was to split the models as evenly as possible, producing two sets with similar types of classes, yet without splitting small classes, and without biasing either set with a large number of models of the same type. To meet these goals, we applied the following steps. First, all classes with 20 or more models were split equally between the training and test sets (models downloaded from the same domain were evenly distributed). Then, smaller classes were alternately placed in the training and test sets in a manner that ensured that both had a balanced number of classes for every object type (plants, animals, vehicles, etc.). Finally, we manually swapped a few small classes so that the training and test sets have an equal number of models. The final result is two sets of classified 3D models, one with 907 models partitioned into 90 classes to be used for training the parameters of shape matching algorithms, and the other with a different 907 models partitioned into 92 classes to be used for comparison with other algorithms. Detailed lists of the classes in both sets appear in Table 3.

Training		Test	
Class Name	# Models	Class Name	# Models
aircraft/airplane/F117	4	aircraft/airplane/biplane	14
aircraft/airplane/biplane	14	aircraft/airplane/commercial	11
aircraft/airplane/commercial	10	aircraft/airplane/fighter_jet	50
aircraft/airplane/fighter_jet	50	aircraft/airplane/glider	19
aircraft/airplane/multi_fuselage	7	aircraft/airplane/stealth_bomber	5
aircraft/balloon_vehicle/dirigible	7	aircraft/balloon_vehicle/hot_air_balloon	9
aircraft/helicopter	17	aircraft/helicopter	18
aircraft/spaceship/enterprise_like	11	aircraft/spaceship/enterprise_like	11
aircraft/spaceship/space_shuttle	6	aircraft/spaceship/flying_saucer	13
aircraft/spaceship/x_wing	5	aircraft/spaceship/satellite	7
animal/arthropod/insect/bee	4	aircraft/spaceship/tie_fighter	5
animal/arthropod/spider	11	animal/arthropod/insect/ant	5
animal/biped/human	50	animal/arthropod/insect/butterfly	7
animal/biped/human/arms_out	21	animal/biped/human	50
animal/biped/trex	6	animal/biped/human/arms_out	20
animal/flying_creature/bird/duck	5	animal/biped/human/walking	8
animal/quadruped/apatosaurus	4	animal/flying_creature/bird/flying	14
animal/quadruped/feline	6	animal/flying_creature/bird/standing	7
animal/quadruped/pig	4	animal/quadruped/dog	7
animal/underwater_creature/dolphin	5	animal/quadruped/horse	6
animal/underwater_creature/shark	7	animal/quadruped/rabbit	4
blade/butcher_knife	4	animal/snake	4
blade/sword	15	animal/underwater_creature/fish	17
body_part/brain	7	animal/underwater_creature/sea_turtle	6
body_part/face	17	blade/axe	4
body_part/head	16	blade/knife	7
body_part/skeleton	5	blade/sword	16
body_part/torso	4	body_part/face	16
bridge	10	body_part/hand	17
building/castle	7	body_part/head	16
building/dome_church	13	body_part/skull	6
building/lighthouse	5	book	4
building/roman_building	12	building/barn	5
building/tent/multiple_peak_tent	5	building/church	4
building/two_story_home	11	building/gazebo	5
chess_piece	17	building/one_story_home	14
chest	7	building/skyscraper	5
city	10	building/tent/one_peak_tent	4
computer/laptop	4	building/two_story_home	10
display_device/tv	12	chess_set	9
door/double_doors	10	city	10
fantasy_animal/dragon	6	computer/desktop	11
furniture/bed	8	display_device/computer_monitor	13
furniture/desk/desk_with_hutch	7	door	18
furniture/seat/chair/dining	11	eyeglasses	7
furniture/seat/chair/stool	7	fireplace	6
furniture/seat/couch	15	furniture/cabinet	9
furniture/shelves	13	furniture/desk/school	4
furniture/table/rectangular	26	furniture/seat/bench	11
furniture/table/round	12	furniture/seat/chair/dining	11
furniture/table_and_chairs	5	furniture/seat/chair/desk	15
gun/handgun	10	furniture/shelves	13
gun/rifle	19	furniture/table/rectangular	25
hat/helmet	10	furniture/table/round/single_leg	6
ice_cream	12	geographic_map	12
lamp/desk	14	gun/handgun	10
liquid_container/bottle	12	hat	6
liquid_container/mug	7	hourglass	6
liquid_container/tank	5	ladder	4
liquid_container/vase	11	lamp/streetlight	8
microchip	7	liquid_container/glass_with_stem	9
microscope	5	liquid_container/pail	4
musical_instrument/guitar/acoustic	4	liquid_container/vase	11
musical_instrument/piano	6	mailbox	7
phone_handle	4	musical_instrument/guitar/electric	13
plant/flower_with_stem	15	newtonian_toy	4
plant/potted_plant	25	plant/bush	9
plant/tree	17	plant/flowers	4
plant/tree/barren	11	plant/potted_plant	26
plant/tree/palm	10	plant/tree/barren	11
sea_vessel/sailboat	5	plant/tree/conical	10
sea_vessel/sailboat/sailboat_with_oars	4	satellite_dish	4
sea_vessel/ship	10	sea_vessel/sailboat/large_sail_boat	6
shoe	8	sea_vessel/ship	11
sign/street_sign	12	sea_vessel/submarine	9
skateboard	5	sign/billboard	4
snowman	6	sink	4
swingset	4	slot_machine	4
tool/screwdriver	5	staircase	7
tool/wrench	4	tool/hammer	4
vehicle/car/antique	5	tool/shovel	6
vehicle/car/sedan	10	umbrella	6
vehicle/car/sports	19	vehicle/car/race	14
vehicle/cycle/bicycle	7	vehicle/car/sedan	10
vehicle/military_tank	16	vehicle/covered_wagon	5
vehicle/pickup_truck	8	vehicle/cycle/motorcycle	6
vehicle/suv	4	vehicle/monster_truck	5
vehicle/train	7	vehicle/semi	7
watch	5	vehicle/suv/jeep	5
wheel/tire	4	vehicle/train/train_car	5
		wheel	4
		wheel/gear	9
Total	907	Total	907
Overall Total = 1,814			

Table 3. The PSB base classification.

5.3 Alternative Classifications

There are many possible classifications for a given set of 3D models. For instance, one person might group models based primarily on function (e.g., like our base classification), while another might group them according to how the objects are constructed (e.g., man-made versus natural), where they are used (e.g., office versus home versus outdoors), or who uses them (e.g., adults versus children). We believe that the results of shape retrieval experiments for multiple classifications are interesting, as they provide information about the circumstances in which each shape matching algorithm performs well/poorly. The cumulative results of experiments with multiple classifications can provide a more complete picture of the differences between competing shape matching algorithms than does any single classification alone.

To support multiple classifications, the benchmark includes a simple language in which researchers can define new classifications. Briefly, an ASCII file is used to specify a hierarchy of class names and to indicate which models belong to each class. We have used this language to create three alternatives to the base classification, each representing a different granularity of grouping. For instance, a coarse classification merges all types of tables into a single class, a coarser classification merges all furniture into one class, and the coarsest partitions objects based only whether they are man-made or appear naturally in the real world. We use these alternative classifications to compare shape matching algorithms in Section 8.

In the future, we expect that other researchers will use the language to define new classifications that we did not anticipate, thereby adding to the suite of experiments that can be used to compare shape matching algorithms.

6 Annotation

The benchmark includes several types of auxiliary information for each model in the database. For instance, the following meta-data is provided to help identify the source and object type for each model:

- **Model URL:** the Web address where the model was found on the Web. The last part of the URL provides the model's file name, which may be useful for semantic labeling. More importantly, the URL can be used to determine the owner of the model for assigning credit and attribution.
- **Referring URL:** the address of the Web page containing a link to the model. The textual anchor and context on this page may be useful for extracting information about the model (if the Web page still exists).
- **Thumbnail image:** an image of the model rendered with colors and textures as seen from a plausible viewpoint. This view of the model with all its surface attributes is useful for seeing what the model looked like in its original form.

In addition, the benchmark lists several geometric attributes for each 3D model (e.g., number of polygons, average dihedral angle, averaged depth complexity over all views, etc.), which are useful for identifying interesting subsets of the database. While these attributes could be derived from the models, and thus are somewhat redundant, they provide a standardized set of values that can be used to avoid the risk that differences in implementations can cause differences in matching results. For instance, the following attributes provide useful data for normalizing 3D models for differences in translation, scale, and orientation:

- **Center of mass:** the average (x, y, z) coordinates for all points on the surfaces of all polygons. These values can be used to normalize the models for translations.
- **Scale:** the average distance from all points on the surfaces of all polygons to the center of mass. This value can be used to normalize the models for isotropic scales.
- **Principal axes:** the eigenvectors (and associated eigenvalues) of the covariance matrix obtained by integrating the quadratic polynomials $x_i \cdot x_j$, with $x_i \in \{x, y, z\}$, over all points on the surfaces of all polygons. These axes can be used to normalize the models for rotations.

7 Evaluation

The benchmark includes several tools for evaluating and comparing how well shape matching algorithms work. These tools assume that every algorithm being evaluated can compute the "distance" between any pair of 3D models, producing positive values that are small if the models are similar and larger for pairs with greater shape differences. So, for a given shape matching algorithm and database of 3D models, we can compute a *distance matrix*, where element (i, j) represents the computed distance between models i and j. Similarly, for any given model Q, we can rank the others from best to worst according to their computed distances from Q. This *ranked list* corresponds to the retrieval result that would be returned if Q were provided as a query to a shape-based search engine.

Given a classification and a distance matrix computed with any shape matching algorithm, a suite of PSB benchmark tools produces statistics and visualizations that facilitate evaluation of the match results (i.e., how many of the top ranked models are from the same class as the query). While none of these statistics are new, we include detailed descriptions so that the reader can get a feel for the tools available in the benchmark and can understand the results presented in Section 8.

- **Best matches:** a web page for each model displaying images of its best matches in rank order. The associated rank and distance value appears below each image, and images of models in the query model's class (hits) are highlighted with a thickened frame. This simple visualization provides a qualitative evaluation tool emulating the output of many 3D model search engines (e.g., [4, 8, 10, 17, 24, 29, 35, 37, 40]).
- **Precision-recall plot:** a plot describing the relationship between precision and recall in a ranked list of matches. For each query model in class C and any number K of top matches, "recall" (the horizontal axis) represents the ratio of models in class C returned within the top K matches, while "precision" (the vertical axis) indicates the ratio of the top K matches that are members of class C. A perfect retrieval result produces a horizontal line across the top of the plot (at precision = 1.0), indicating that all the models within the query object's class are returned as the top ranked matches. Otherwise, curves that appear shifted up represent superior retrieval results (see Figure 2).
- **Distance image:** an image of the distance matrix where the lightness of each pixel (i, j) is proportional to the magnitude of the distance between models i and j [23]. Models are grouped by class along each axis, and lines are added to separate classes, which makes it easy to evaluate patterns in the match results qualitatively – i.e., the optimal result is a set of darkest, class-sized blocks of pixels along the diagonal indicating that every model matches the models within its class better than ones in other classes. Otherwise, the reasons for poor match results can often be seen in the image – e.g., off-diagonal blocks of dark pixels indicate that two classes match each other well.
- **Tier image:** an image visualizing nearest neighbor, first tier, and second tier matches [23]. Specifically, for each row representing a query with model j in a class with $|C|$ members, pixel (i, j) is: (a) black if model i is model j or its nearest neighbor, (b) red if model i is among the $|C| - 1$ top matches (the first tier), and blue if model i is among the $2 * (|C| - 1)$ top matches (the second tier). As with the distance image, models are grouped by class along each axis, and lines are added to separate classes. This image is often more useful than the distance image because the best matches are clearly shown for every model, regardless of the magnitude of their distance values. The optimal result is a set of black/red, class-sized blocks of pixels along the diagonal indicating that every model matches the models within its class better than ones in other classes. Otherwise, more colored pixels in the class-sized blocks along the diagonal represents a better result (see Figure 1).

In addition to these qualitative visualizations, the benchmark includes tools for computing quantitative statistics for evaluation of match results. Usually, the statistics are summarized by averaging over all query models (micro-average), with the query model removed from the matching results. However, our tools also support output of separate statistics for each query model, averages for each class, an average of the averages for each class (macro-average), and

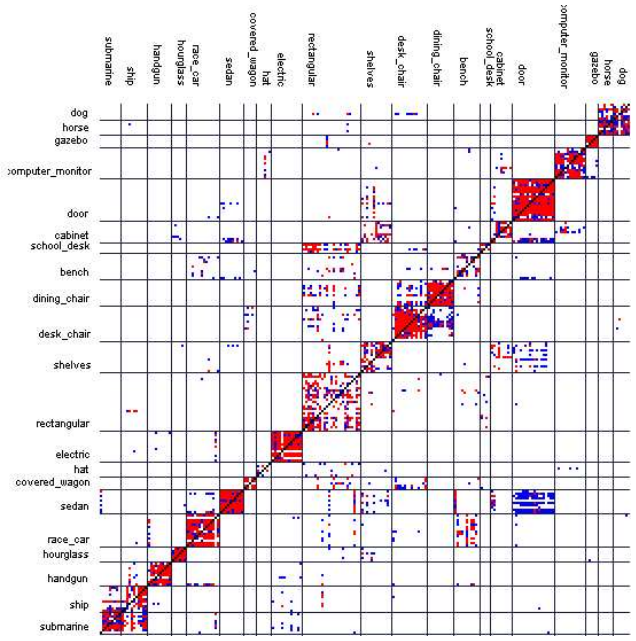

Figure 1. Tier image visualizing nearest neighbor (black), first tier (red), and second tier (blue) computed by matching every model (rows) with every other model (columns) in the base classification of the test set using the LFD algorithm – separating lines and labels indicate classes. Note that the full image is 907x907 "pixels," and only a small portion is shown.

averages over any user-supplied list of query models.[1] As will be shown in Section 8.4, this last feature is particularly useful for studying the quality of matches for models having specific properties. Specifically, our tools compute:

- **Nearest neighbor:** the percentage of the closest matches that belong to the same class as the query. This statistic provides an indication of how well a nearest neighbor classifier would perform. Obviously, an ideal score is 100%, and higher scores represent better results (see column 5 of Table 4).

- **First-tier and Second-tier:** the percentage of models in the query's class that appear within the top K matches, where K depends on the size of the query's class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier, and $K = 2 * (|C| - 1)$ for the second tier. The first tier statistic indicates the recall for the smallest K that could possibly include 100% of the models in the query class, while the second tier is a little less stringent (i.e., K is twice as big). These statistics are similar to the "Bulls Eye Percentage Score" ($K = 2 * |C|$), which has been adopted by the MPEG-7 visual SDs [38]. In all cases, an ideal matching result gives a score of 100%, and higher values indicate better matches (see columns 5 and 6 of Table 4).

- **E-Measure:** a composite measure of the precision and recall for a fixed number of retrieved results [32]. The intuition is that a user of a search engine is more interested in the first page of query results than in later pages. So, this measure considers only the first 32 retrieved models for every query and calculates the precision and recall over those results. The E-Measure is defined as [32, 19]:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

  The E-measure is equivalent to subtracting van Rijsbergen's definition of the E-measure from 1. The maximum score is 1.0, and higher values indicate better results (see column 7 of Table 4).

- **Discounted Cumulative Gain (DCG):** a statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. Specifically, the ranked list R is converted to a list G, where element G_i has value 1 if element R_i is in the correct class and value 0 otherwise. Discounted cumulative gain is then defined as follows [14]:

$$DCG_i = \left\{ \begin{array}{ll} G_1, & i = 1 \\ DCG_{i-1} + \frac{G_i}{lg_2(i)}, & otherwise \end{array} \right\}$$

  This result is then divided by the maximum possible DCG (i.e., that would be achieved if the first C elements were in the correct class, where C is the size of the class) to give the final score:

$$DCG = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{lg_2(j)}}$$

  where k is the number of models in the database (see column 10 of Table 4).

The entire query result list is incorporated in an intuitive manner by the discounted cumulative gain [19], so we typically use it to summarize results when comparing algorithms. More specifically, we usually look at the "normalized DCG," which scales the DCG values down by the average over all algorithms tested and shifts the average to zero:

$$NormalizedDCG_A = DCG_A / AvgDCG - 1$$

where DCG_A is the DCG value computed for algorithm A, and AvgDCG is the average DCG value for all algorithms being compared in the same experiment. Positive/negative normalized DCG scores represent above/below average performance, and higher numbers are better (see the rightmost column of Table 4).

[1] For precision-recall plots, the precision for each model (micro) or class (macro) is averaged using linear interpolation over the recall range $[1/|C|, 1]$, if there are C models in a class.

8 Results

In order to investigate the utility of the proposed benchmark, we used it to compare 12 shape matching algorithms recently described in the literature. While the results of these experiments are interesting in their own right, the focus of our investigation is whether the database, classifications, annotations, and evaluation tools provided by the benchmark are useful for understanding the differences between the algorithms. Our hypothesis is that we might learn something about the algorithms that would have been difficult to discover without the benchmark tools.

8.1 Shape Descriptors

The 12 shape matching algorithms are all similar in that they proceed in three steps: the first step normalizes the models for differences in scale and possibly translation and rotation; the second step generates a *shape descriptor* for each model; and the third step computes the distance between every pair of shape descriptors, using their L_2 difference unless otherwise noted. The differences between the algorithms lie mainly in the details of their shape descriptors:

- **D2 Shape Distribution (D2)**: a histogram of distances between pairs of points on the surface [23].
- **Extended Gaussian Image (EGI)**: a spherical function giving the distribution of surface normals [13].
- **Complex Extended Gaussian Image (CEGI)**: a complex-valued spherical function giving the distribution of normals and associated normal distances of points on the surface [15].
- **Shape Histogram (SHELLS)**: a histogram of distances from the center of mass to points on the surface [1].
- **Shape Histogram (SECTORS)**: a spherical function giving the distribution of model area as a function of spherical angle [1].
- **Shape Histogram (SECSHEL)**: a collection of spherical functions that give the distribution of model area as a function of radius and spherical angle [1].
- **Voxel**: a binary rasterization of the model boundary into a voxel grid.
- **Spherical Extent Function (EXT)**: a spherical function giving the maximal distance from center of mass as a function of spherical angle [27].
- **Radialized Spherical Extent Function (REXT)**: a collection of spherical functions giving the maximal distance from center of mass as a function of spherical angle and radius [35].
- **Gaussian Euclidean Distance Transform (GEDT)**: a 3D function whose value at each point is given by composition of a Gaussian with the Euclidean Distance Transform of the surface [16].
- **Spherical Harmonic Descriptor (SHD)**: a rotation invariant representation of the GEDT obtained by computing the restriction of the function to concentric spheres and storing the norm of each (harmonic) frequency [16].
- **Light Field Descriptor (LFD)**: a representation of a model as a collection of images rendered from uniformly sampled positions on a view sphere. The distance between two descriptors is defined as the minimum L_1-difference, taken over all rotations and all pairings of vertices on two dodecahedra. [4].

All computations were performed on a Windows PC with a Pentium 4 CPU running at 2.00 GHz and 512 MB of memory, except the LFD computations, which were executed on a Windows PC with Pentium 4 CPU running at 2.4GHz with 256MB RAM and an NVIDIA GeForce 2 MX graphics card.[2]

8.2 Base Classification Results

In our first experiment, we used each of the 12 shape matching algorithms to compute the distances between all pairs of models in the test set and analyzed them with the benchmark evaluation tools to quantify the matching performance with respect to the base classification (the training set was not used for training any of the algorithms). Figure 2 shows a precision-recall plot showing the micro-averaged retrieval results achieved for this experiment, and Table 4 shows micro-averaged storage requirements, processing times, and retrieval statistics for each algorithm. We found that the micro and macro-average gave consistent results, and we decided to present micro-averaged statistics.

Surprisingly, we find that the shape descriptor based on 2D views (LFD) provides the best retrieval precision in this experiment. Although we might expect shape descriptors that capture 3D geometric relationships would be more discriminating than ones based solely on 2D projections, the

[2]Every model was normalized for size by isotropically rescaling it so that the average distance from points on its surface to the center of mass is 0.5. Then, for all descriptors except D2 and EGI, the model was normalized for translation by moving its center of mass to the origin. Next, for all descriptors except D2, SHELLS, SHD, and LFD, the model was normalized for rotation by aligning its principal axes to the x-, y-, and z-axes. The ambiguity between positive and negative axes was resolved by choosing the direction of the axes so that the area of the model on the positive side of the x-, y-, and z-axes was greater than the area on the negative side [7].

Every spherical descriptor (EGI, CEGI, Sectors, etc.), was computed on a 64×64 spherical grid and then represented by its harmonic coeffi cients up to order 16. Similarly, every 3D descriptor (e.g., Voxel and GEDT) was computed on a $64 \times 64 \times 64$ axial grid, translated so that the origin is at the point $(32, 32, 32)$, scaled by a factor of 32, and then represented by thirty-two spherical descriptors representing the intersection of the voxel grid with concentric spherical shells. Values within each shell were scaled by the square-root of the corresponding area and represented by their spherical harmonic coeffi cients up to order 16. Histograms of distances (D2 and Shells) were stored with 64 bins representing distances in the range $[0, 2]$. All descriptors, except LFD, were scaled to have L_2-norm equal to 1.

The LFD comprises 100 images encoded with 35, 8-bit, coeffi cients to describe Zernike moments and 10, 8-bit, coeffi cients to represent Fourier descriptors.

Shape Descriptor	Storage Size (bytes)	Timing		Discrimination					
		Generate Time (s)	Compare Time (s)	Nearest Neighbor	First Tier	Second Tier	E-Measure	DCG	Normalized DCG
LFD	4,700	3.25	0.001300	65.7%	38.0%	48.7%	28.0%	64.3%	21.3%
REXT	17,416	2.22	0.000229	60.2%	32.7%	43.2%	25.4%	60.1%	13.3%
SHD	2,184	1.69	0.000027	55.6%	30.9%	41.1%	24.1%	58.4%	10.2%
GEDT	32,776	1.69	0.000450	60.3%	31.3%	40.7%	23.7%	58.4%	10.1%
EXT	552	1.17	0.000008	54.9%	28.6%	37.9%	21.9%	56.2%	6.0%
SECSHEL	32,776	1.38	0.000451	54.6%	26.7%	35.0%	20.9%	54.5%	2.8%
VOXEL	32,776	1.34	0.000450	54.0%	26.7%	35.3%	20.7%	54.3%	2.4%
SECTORS	552	0.90	0.000014	50.4%	24.9%	33.4%	19.8%	52.9%	-0.3%
CEGI	2,056	0.37	0.000027	42.0%	21.1%	28.7%	17.0%	47.9%	-9.6%
EGI	1,032	0.41	0.000014	37.7%	19.7%	27.7%	16.5%	47.2%	-10.9%
D2	136	1.12	0.000002	31.1%	15.8%	23.5%	13.9%	43.4%	-18.2%
SHELLS	136	0.66	0.000002	22.7%	11.1%	17.3%	10.2%	38.6%	-27.3%

Table 4. Comparing 12 shape descriptors using the PSB base classification.

opposite is true. However, this view-based descriptor takes more time to compare than the other descriptors, since it requires searching over multiple possible image correspondences. Among the other descriptors, REXT, SHD, GEDT, and EXT provide the best matching performance. While REXT provides slightly better discrimination than the others, SHD and EXT are smaller and quicker to compare, suggesting they provide more "bang for the buck." The least discriminating descriptors are D2 and SHELLS. However, they are also the smallest and fastest to compare, which may be useful in certain applications.

Overall, we conclude that there is a quality-cost trade-off in the choice between shape descriptors, and no one descriptor beats the others in all respects.

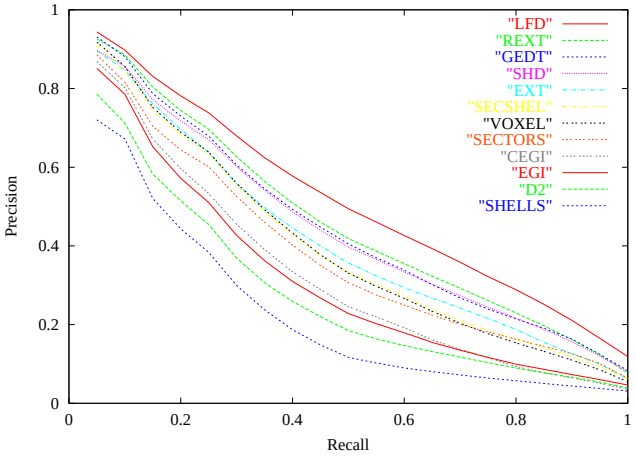

Figure 2. Precision-recall curves computed for 12 shape descriptors for tests with the PSB base classification.

8.3 Multi-Classification Results

In our second experiment, we investigated the impact of alternative classifications on the analysis of retrieval results. Specifically, we created three new classifications representing increasingly coarser groupings for the 907 models in the benchmark test set, and then we tested how these different classifications affect the evaluation of the 12 shape matching algorithms.

The base classification provides the grouping with finest granularity in this experiment. It contains the 92 classes listed in Table 3. Most classes contain all the objects with a particular function (e.g., microscopes). Yet, there are also cases where objects with the same function are partitioned into different classes based on their forms (e.g., round tables versus rectangular tables). In the alternative classifications, we recursively merge classes to form coarser granularity groups. Specifically, the "Coarse" classification merges objects with similar overall function to form 44 classes, the "Coarser" classification merges groups further to form the 6 classes listed in Table 1, plus a miscellaneous class not included in averaged retrieval results. Finally, the "Coarsest" classification merges those classes until just two classes remain: one with man-made objects and the other with naturally occurring objects.

Table 6 lists the normalized DCG scores achieved by the 12 shape matching algorithms (rows) when evaluated with respect to the four different classifications (columns). From this table, we make two observations. First, as you might expect, the differences between shape matching algorithms are diminished when evaluated with coarser granularity classifications - i.e., the normalized DCG scores, which measure differences from the average, become less in columns further to the right. Second, we observe that the relative rankings of algorithms can vary significantly for different classifications. In particular, the EGI algorithm performs tenth best with respect to the base classification (10.9% below the average). However, it performs best of all for the coarsest classification (3.0% above the average).

Shape Descriptor	All Models	Semantic						Procedural	Geometric	
		Animal	Building	Furni-ture	House-hold	Tree & Plant	Vehicle	Rotation Aligned	Stick Shape	Complex Shape
LFD	21.3%	15.8%	41.6%	35.4%	25.3%	24.4%	17.9%	18.8%	14.8%	28.2%
REXT	13.3%	8.6%	3.9%	11.2%	13.9%	8.8%	16.2%	12.3%	10.0%	15.0%
SHD	10.2%	10.3%	-8.3%	10.1%	12.8%	19.7%	7.2%	7.6%	4.9%	8.9%
GEDT	10.1%	9.6%	3.6%	8.1%	10.8%	6.6%	10.6%	13.0%	8.1%	13.5%
EXT	6.0%	6.4%	10.9%	5.3%	8.5%	-6.8%	7.7%	5.0%	6.6%	6.1%
SECSHEL	2.8%	0.6%	1.1%	-5.6%	7.2%	-11.4%	7.1%	5.2%	3.6%	2.2%
VOXEL	2.4%	4.0%	1.1%	-1.7%	3.3%	-2.5%	5.4%	4.7%	5.3%	0.2%
SECTORS	-0.3%	-2.2%	1.8%	-6.1%	4.0%	-21.4%	3.6%	2.0%	4.7%	-1.6%
CEGI	-9.6%	-4.1%	-22.6%	1.1%	-19.6%	11.2%	-13.4%	-8.7%	-7.9%	-12.7%
EGI	-10.9%	-9.7%	-12.2%	0.0%	-21.1%	11.8%	-12.7%	-11.2%	-9.8%	-9.1%
D2	-18.2%	-16.1%	-0.3%	-26.4%	-15.2%	-21.5%	-21.1%	-19.7%	-11.5%	-19.9%
SHELLS	-27.3%	-23.1%	-20.8%	-31.4%	-29.9%	-19.0%	-28.5%	-29.1%	-28.8%	-30.9%

Table 5. Evaluating retrieval performance for 12 shape descriptors on query lists with specific object types and geometric properties using the PSB base classification. Numbers represent normalized DCG value averaged over models with the property listed in the column heading.

Apparently, it is very good at determining the difference between man-made and natural objects, but not that good at telling apart the differences between specific classes. We conjecture that man-made objects have a narrower distributions of normals, making detection easy with EGIs.

These results provide a simple example of the value of using multiple classifications when evaluating shape matching algorithms. The information available in multiple classifications is more than in any one classification alone. We expect that many alternative semantic classifications will be made for these models in the future, exposing further differences between algorithms.

Shape Descriptor	Base (92)	Coarse (44)	Coarser (6)	Coarsest (2)
LFD	21.3%	11.7%	3.2%	0.3%
REXT	13.3%	6.8%	2.0%	0.2%
SHD	10.2%	5.7%	0.9%	-0.6%
GEDT	10.1%	4.8%	1.2%	-0.3%
EXT	6.0%	2.0%	0.7%	-0.6%
SECSHEL	2.8%	-0.3%	0.1%	-0.4%
VOXEL	2.4%	0.0%	-0.1%	-0.4%
SECTORS	-0.3%	-1.4%	-0.7%	-0.7%
CEGI	-9.6%	-1.2%	0.6%	2.6%
EGI	-10.9%	-2.1%	0.2%	3.0%
D2	-18.2%	-10.3%	-3.4%	-1.6%
SHELLS	-27.3%	-15.7%	-4.5%	-1.5%

Table 6. Evaluating 12 shape descriptors using classifications of different granularity. The columns represent different classifications (with the number of classes listed in parenthesis), and the rows represent different shape descriptors. The numbers show normalized DCG scores averaged over all models.

8.4 Query List Results

In our third experiment, we studied the properties of the 12 shape matching algorithms further by looking at retrieval results with respect to the base classification averaged over sets of models with specific properties. Some of the properties were semantic (e.g., is a piece of furniture), others were procedural (e.g., aligned well with other members of its class), and the rest were geometric (e.g., roughly linear in shape). Our hope is that we can infer the conditions under which each shape matching algorithm performs best by comparing the retrieval results of this experiment.

Table 5 lists normalized DCG scores achieved by the 12 shape matching algorithms (rows) with respect to the base classification when averaged over all models with specific properties (columns). The first column of numbers ("All Models") shows the average for all models, as a reference for comparison. The next six columns ("Animal"-"Vehicle") correspond to averages over the sets of models of the same object type. The next column ("Rotation Aligned") shows the average over all models for which our normalization steps were successfully able to align the model consistently with other members of is class. The following column ("Stick Shape") lists averages over the 200 models whose shape is most stick-like (as determined by the ratio of the largest and second largest eigenvalues of the covariance matrix of second order moments). Finally, the right-most column ("Complex Shape") shows averages over the 200 models with the most "complex shapes" (as estimated by the average pixel depth complexity when the model is rendered with parallel projection from viewpoints at the vertices of an icosahedron). These latter properties are derived directly from the annotations provided with the benchmark.

With these results, we confirm that shape matching algorithms do not perform equally well on all object types. Al-

though the ranking of algorithms is fairly consistent, there is sometimes a big difference in the relative performance of algorithms when focusing on models with specific properties. For instance, we note that SECTORS does better than EGI on household objects (4.0% above average versus 21.1% below average), while the opposite is true for trees and plants (21.4% below average versus 11.8% above average). Also, we see that the top ranked algorithms (LFD, REXT, and SHD) do worse on stick-shaped objects relative to other algorithms (the normalized DCG scores averaged for stick shaped objects are worse than the average over all models by 6.5%, 3.3%, 5.3%, respectively), probably because the principal axes of sticks align well and/or the descriptors eliminate high-frequency information. Finally, we note that queries with "Rotation Aligned" models produce significantly different retrieval results, indicating that misalignment of models during normalization significantly affects the results achieved with some algorithms (GEDT, SECSHEL, VOXEL, and SECTORS).

8.5 Comparison with Other Databases

In our final experiment, we compared results of the Princeton Shape Benchmark database versus those achieved with other databases previously described in the literature [10, 23, 30, 35, 38]. Our goal in this experiment was to validate whether our benchmark produces results consistent with those previously reported.

Table 7 shows the normalized DCG scores computed for the 12 shape matching algorithms on six different databases. We see that the results computed with the Osada [23] and MPEG-7 [38] databases are less consistent with the others. We conjecture that the reason is that they are relatively small (133 and 227 models, respectively) and have less variation of object types. The classified models of the Utrecht [30] database are mostly airplanes, which probably explains why the retrieval results showed little variation. Meanwhile, the relative performance of the algorithms on the other three databases appear fairly consistent. We expect that the minor differences between the databases can be explained by the differences in their object types.

Shape Descriptor	Osada [23]	MPEG-7 [38]	Utrecht [30]	CCCC [35]	VP [10]	PSB [ours]
LFD	14.9%	5.8%	5.4%	20.3%	17.7%	21.3%
REXT	8.6%	3.6%	2.4%	11.3%	8.5%	13.3%
SHD	12.1%	5.5%	2.3%	12.5%	10.6%	10.2%
GEDT	5.2%	2.5%	4.3%	5.5%	6.3%	10.1%
EXT	2.9%	0.4%	2.4%	5.5%	5.6%	6.0%
SECSHEL	-0.7%	-0.2%	2.2%	-0.8%	0.7%	2.8%
VOXEL	2.2%	1.3%	2.5%	-0.5%	0.4%	2.4%
SECTORS	-0.8%	-2.3%	2.3%	-1.9%	-1.6%	-0.3%
CEGI	-13.9%	-1.8%	-6.9%	-4.7%	-7.6%	-9.6%
EGI	-10.7%	-1.0%	-7.0%	-7.3%	-9.5%	-10.9%
D2	-1.1%	-4.3%	-3.1%	-16.6%	-12.8%	-18.2%
SHELLS	-18.7%	-9.6%	-6.8%	-23.2%	-18.2%	-27.3%

Table 7. Evaluating shape descriptors using different databases. Numbers represent normalized DCG averaged over all models in each database.

9 Conclusion

In summary, this paper describes the Princeton Shape Benchmark, a publicly available framework for comparing shape matching algorithms. The benchmark includes a database of annotated 3D polygonal models, multiple classifications, and software tools for evaluating the results of shape matching experiments. All data and source code is freely available on the Web (http://shape.cs.princeton.edu/benchmark).

The main research contribution of this work is the methodology proposed for comparing shape matching algorithms. In particular, we advocate experimenting with several different classifications and query lists targeted at exposing specific differences between shape matching algorithms. Using this methodology, for example, we were able to discover that EGIs are good at discriminating between man-made and natural objects, but not that good at making detailed class distinctions. We also find that the Light Field Descriptor [4], which is computed from multiple 2D images of a 3D model, is the most discriminating among the shape descriptors tested, but at higher storage and computational cost than many other 3D descriptors. We hope that results of this type encourage shape matching researchers to use the benchmark in future experiments, possibly creating new classifications and query lists of their own. In time, we expect that a common set of tests will emerge to form a de facto standard for shape matching experiments.

This paper suggests several avenues for future research. First, the benchmark should be expanded to support other shape analysis tasks, such as recognition, registration, and retrieval. As a first step, we intend to provide annotations for human-generated alignment transformations to facilitate evaluation of automatic registration algorithms, and we will include measures of indexing performance as a metric in future versions of the benchmark (i.e., for retrieval applications rather than matching applications). Second, we plan to investigate multi-classifiers. The results of Section 8.4 suggest that it is possible to build an adaptive multi-classifier that first checks the geometric properties of a given query model and then dynamically weights the distances computed by several shape matching algorithms to produce more discriminating results (e.g., [11]). Finally, as more and more data gets added to the benchmark, it will become possible to consider multi-classifiers that take into account both geometric and non-geometric attributes of 3D models (e.g., color, texture, scene graph structure, textual annotation, etc.). We believe that the benchmark described in this paper provides a solid infrastructure to begin research in these directions.

Acknowledgements

We would like to thank David Bengali, who partitioned thousands of 3D models into classes. Ming Ouhyoung and his students provided an implementation of the Light Field Descriptor. Dejan Vranic provided the CCCC and MPEG-7 databases; Viewpoint Data Labs donated the Viewpoint

database; and Remco Veltkamp and Hans Tangelder published the Utrecht database. Finally, the National Science Foundation provided funding under grants CCR-0093343 and 11S-0121446.

References

[1] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. Nearest neighbor classification in 3D protein databases. In *Proc. ISMB*, 1999.

[2] H. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, (28):235–242, 2000.

[3] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, March 1985.

[4] D.-Y. Chen, M. Ouhyoung, X.-P. Tian, and Y.-T. Shen. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, pages 223–232, 2003.

[5] CMU. Pose, illumination, and expression (PIE) database, 2003. http://www.ri.cmu.edu/projects/project_418.html.

[6] P. Courtney and N. Thacker. Peformance characterization in computer vision, 2003. http://peipa.essex.ac.uk/benchmark.

[7] M. Elad, A. Tal, and S. Ar. Content based retrieval of VRML objects - an iterative and interactive approach. In *6th Eurographics Workshop on Multimedia 2001*, 2001.

[8] T. T. Elvins and R. Jain. Web-based volumetric data retrieval. In *VRML '95*, pages 7–12, 1995.

[9] C. Foster, E. Hayes, C. Y. Ip, D. McWherter, M. Peabody, Y. Shapirsteyn, V. Zaychik, and W. C. Regli. National design repository project: A status report. In *Int'l Joint Confs. on Artificial Intelligence (IJCAI) AAAI/SIGMAN Workshop on AI in Manufacturing Systems*, August 2001.

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

[11] G. Giacinto and F. Roli. Dynamic classifier selection. *Lecture Notes in Computer Science*, 1857, 2000.

[12] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 203–212, August 2001.

[13] B. Horn. Extended Gaussian images. *Proc. of the IEEE*, 72(12):1671–1686, December 1984.

[14] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

[15] S. Kang and K. Ikeuchi. Determining 3-D object pose using the complex extended Gaussian image. In *CVPR*, pages 580–585, June 1991.

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

[17] I. Kolonias, D. Tzovaras, S. Malasiotis, and M.G.Strintzis. Fast content-based search of VRML models based on shape descriptors. *IEEE Transactions on Multimedia*, 2003. accepted for publication.

[18] Y. Lecun. The MNIST database of handwritten digits, 2003. http://yann.lecun.com/exdb/mnist/.

[19] G. Leifman, S. Katz, A. Tal, and R. Meir. Signatures of 3D models for retrieval. pages 159–163, February 2003.

[20] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[21] P. Min, J. Halderman, M. Kazhdan, and T. Funkhouser. Early experiences with a 3D model search engine. In *Proceeding of the eighth international conference on 3D web technology*, pages 7–18, 2003.

[22] Okino. Polytrans, 2003. www.okino.com/conv/conv.htm.

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

[24] E. Paquet and M. Rioux. Nefertiti: a query by content software for three-dimensional models databases management. *Image and Vision Computing*, 17(2):157–166, 1999.

[25] A. R. Pope. Model-based object recognition: A survey of recent research. Technical Report TR-94-04, University of British Columbia, January 1994.

[26] G. Salton. The smart document retrieval project. In *ACM SIGIR Conference on Research and development in Information Retrieval*, pages 356–358, 1991.

[27] D. Saupe and D. V. Vranic. 3D model retrieval with spherical harmonics and moments. In B. Radig and S. Florczyk, editors, *DAGM 2001*, pages 392–397, September 2001.

[28] SPEC. Standard performance evaluation corporation, 2003. www.specbench.org/benchmarks.html.

[29] M. T. Suzuki. A web-based retrieval system for 3D polygonal models. *Joint 9th IFSA World Congress and 20th NAFIPS International Conference (IFSA/NAFIPS2001)*, pages 2271–2276, July 2001.

[30] J. Tangelder and R. Veltkamp. Polyhedral model retrieval using weighted point sets. In *Shape Modeling International*, May 2003.

[31] TREC. Text REtrieval Conference data, 2003. http://trec.nist.gov/data.html.

[32] C. K. van Rijsbergen. *Information Retrieval*. Butterworths, 1975.

[33] R. C. Veltkamp. Shape matching: Similarity measures and algorithms. In *Shape Modelling International*, pages 188–197, May 2001.

[34] Viewpoint Corporation. http://www.viewpoint.com, 2001.

[35] D. V. Vranic. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *IEEE International Conference on Image Processing (ICIP 2003)*, volume 3, pages 757–760, September 2003.

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

[37] Y. Yang, J. Yang, H. Yang, and O. Gwun. Indexing VRML objects with triples. In *SPIE Proceedings*, volume 4311, pages 236–243, 2001.

[38] T. Zaharia and F. Preteux. 3D shape-based retrieval within the MPEG-7 framework. In *SPIE Conf. on Nonlinear Image Processing and Pattern Analysis XII*, volume 4304, pages 133–145, January 2001.

[39] T. Zaharia and F. Preteux. Shape-based retrieval of 3D mesh models. In *IEEE International Conference on Multimedia and Expo (ICME '2002)*, August 2002.

[40] C. Zhang and T. Chen. Indexing and retrieval of 3D models aided by active learning. In *ACM Multimedia*, 2001.