Ryutarou Ohbuchi, Yushin Hata, Combining Multiresolution Shape Descriptors for 3D Model Retrieval, *Proc. WSCG 2006*, Plzen, Czech Republic, Jan. 30~Feb. 3, 2006.

Combining Multiresolution Shape Descriptors for 3D Model Retrieval

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Takeda, Kofu-shi
400-8511, Yamanashi-ken, Japan
ohbuchi@yamanashi.ac.jp

Yushin Hata
University of Yamanashi
4-3-11 Takeda, Kofu-shi
400-8511, Yamanashi-ken, Japan
try@ha.bekkoame.ne.jp

ABSTRACT

In this paper, we propose and evaluate a systematic approach for improving performance of 3D model retrieval by combining multiple shape descriptors. We explored two approaches for generating multiple, mutually independent, shape descriptors; (1) application of a (single-resolution) shape descriptor on a set of multiresolution shape models generated from a query 3D shape model, and (2) application of multiple, heterogeneous shape descriptors on the query 3D shape model. The shape descriptors are integrated via the linear combination of the distance values they produce, using either fixed or adaptive weights. Our experiment showed that both multiresolution and heterogeneous sets of shape descriptors are effective in improving retrieval performance. For example, by using the multiresolution approach, the R-precision of the *SPRH* shape descriptor by Wahl, et al, improved by 8%, from 29% to 37%. A combination of three heterogeneous shape descriptors achieved the R-precision of about 42%; this figure is about 5% better than the R-precision of 38% achieved by the *Light Field Descriptor* by Chen, et al., which is arguably the best single shape descriptor reported to date.

Keywords

3D model database, content-based retrieval, geometric modeling, multiresolution analysis, feature combination.

1. INTRODUCTION

3D shape models are increasingly popular in many application domains, ranging from movie special effects, 3D games on cellular phones or on game consoles, to 3D mechanical CAD/CAE systems. The popularity has prompted research into effective management and reuse of 3D shape models by means of shape-based retrieval of 3D models. An example of such database is the 3D Search engine at the Princeton University [Funkhouser03].

Typical steps for shape similarity based retrieval of 3D models starts with query specification (See Figure 1.) As queries, texts, 2D sketches, 2D images, 3D sketches, and 3D shapes have been used in the past. Multiple query specification methods may be combined, as in the work of Funkhouser et al [Funkhouser03]. The next step is to extract feature, or *shape descriptor* (SD), from the query. Also, as a pre-computing step, SDs for 3D shape models in the database have been computed. The querying method and the 3D shape representation used for the database influences the shape descriptor and the similarity (or more often, distance) computation method. During the distance computation, it is often desirable to reflect human judgment. The database retrieves and presents the models most similar to the query based on the distances computed.

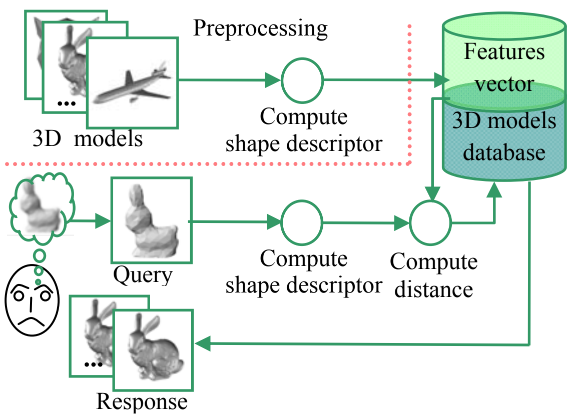

Figure 1. A typical 3D model database system.

Conference proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

In this paper, we explore an approach to boost shape similarity retrieval performance of a 3D model database *by extracting more features* from shape models. Our method uses two method to try to extract more shape feature from a query 3D model; (1) extraction of a multiresolution set of features by applying a SD to the models having multiple resolution levels generated by using the method by Ohbuchi, et al [Ohbuchi03], and (2) extraction of a heterogeneous set of features by applying multiple, heterogeneous SDs to the model.

Multiple SDs are integrated via distance value they produce by using linear combination, which allows integration of features that does not explicitly produce feature vectors, namely, the *Light Field Descriptor* (*LFD*) [Chen03] by Chen, et al. Integration of a set of distances are done through either *fixed weight* linear combination of distances, or knowledge based *adaptive weight* linear combination of distances. The latter is a modification of the "purity" based method of Bustos, et al. [Bustos04a, Bustos04b].

Our experiments showed that the multiresolution approach boosted performances of many, but not all, of the SDs we tested. Combinations of multiresolution, heterogeneous shape descriptors produced the best performance of the combinations we tested. For example, one of the combinations produced R-precision of 42%, which is about 5% higher than the 38% achieved by the LFD by Chen, et al., which is one of the best performing SDs according to Tangelder, et al. [Tangelder04].

This paper is organized as follows. In the following section, we review the previous integrate features for 3D model retrieval. In Section 3, we describe the method to compute multiresolution SDs, and the method to integrate multiple SDs by their distance values. A set of performance evaluation experiments and their results are described in Section 4. We conclude the paper in Section 5.

2. Previous Work

In the field of content based search and retrieval of 2D images, it is typical to extract more than one feature from an image, and combine these features for an overall similarity comparison. Thus it is natural to think of such an approach for the 3D model retrieval. This approach has been taken by several groups; Iyer, et al. [Iyer03], Bustos et al [Bustos04a, Bustos04b], and Atmosukarto, et al. [Atmosukarto05].

Iyer et al. weighted and combined multiple heterogeneous feature vectors, letting the user control the weights explicitly through a user interface or implicitly through a relevance feedback mechanism that employs interactive learning. The method by Atmosukarto et al. also weighted and combined heterogeneous feature vectors. Their experiments showed that combinations of descriptors have better performance than any single shape descriptor they evaluated.

Unlike the former two, the method by Bustos et al. integrated multiple, heterogeneous features via *distance*. Bustos computed the overall distance between a pair of models as a linear combination of the distances using either fixed or adaptive weights. In the fixed weight combination, weights are the same regardless of the model (or the model category.) In the adaptive weights combination, weights are computed by using *purity*, which is an estimate of the performance of the SD determined by using a pre-classified training database. Bustos et al. reported significant performance gain using both fixed weight and adaptive weight linear combination of distances, although the adaptive one performed better.

Integration of multiple shape descriptors can be achieved using either (1) *feature vectors* of the descriptors, if available (the approach by Iyer et al. and Atmosukarto, et al.), or (2) *distance values* computed using the descriptors (the approach by Bustos et al.). Integration using feature vectors potentially allows fine tuning of distance computation, e.g., by weighting each element of the vectors. However, this approach can't be used if a shape descriptor produces distance but not feature vector. For example, one of the most powerful 3D shape descriptors, the LFD by Chen et al. [Chen03] produces distance only, and is useful only for distance based integration. Findings by Atmosukarto et al. and by Bustos et al. are contradictory regarding whether the shape descriptor should be combined as feature vectors or as distances. Atmosukarto reported that a combination of distances does not produce any performance improvement. Bustos et al., on the other hand, reported that combinations of distances are beneficial. Our finding reported in this paper agrees with that of Bustos et al.

3. METHOD

In this section, we describe the method to compute multiresolution shape descriptors and the method to combine multiple shape descriptors by their distance values.

3.1. Computing multiple shape descriptors

To improve retrieval performance through feature integration, features should capture as different shape feature of the model as possible. Our method uses the following two different approaches to extract mutually independent shape descriptors.

1. **Multiresolution approach**: Generate a set of multiresolution (MR) shape models from a model to be compared. A (single-resolution) shape descriptor is applied to the MR shape models to produce multiple SDs.

2. **Heterogeneous shape descriptor approach**: Apply multiple shape descriptors to the model to be compared.

A combination of the two approaches above is also possible. For example, an MR representation having m levels may be combined with n heterogeneous shape descriptors to produce $m \times n$ shape descriptors.

3.1.1 Multiresolution shape descriptors

The method by Ohbuchi, et al. [Ohbuchi03] compares 3D models at multiple scales following the steps below:

1. **Compute a multiresolution representation**: The surface-based input model is converted into a point-based model by *Monte-Carlo* sampling of the surfaces of the model. A set of L-1 scale values α_i, i=1…L-1 is computed based on the size of the model. The set of scale values are used to normalize size among shape models. Then, compute L-1 3D alpha shapes from the point set model by using the L-1 scale values α_i. Of L shape features, those of the coarser (L-1) levels are computed by using the 3D alpha shapes [Edelsbrunner94]. For the finest resolution level L, however, the original, polygon soup model is used. Ohbuchi et al calls this set of multiresolution models *Alpha-Multiresolution Representation* (AMR).

2. **Compute multiresolution shape descriptors**: Apply a shape descriptor x to a model at each resolution level of the AMR, creating a set of multiresolution shape descriptors AMR-x.

3. **Compute distance between a pair of features**: Compute a distance between AMR-x shape descriptors of a pair of models to be compared, using a statically weighted linear combination of L distances.

An advantage of the AMR above is that it can be computed for polygon soup models or even for point set models (without the step 1 above.) Figure 3 shows an example of AMR representation for a surface-based 3D model of rabbit. The AMR is reminiscent of morphological multiresolution representations for 2D images.

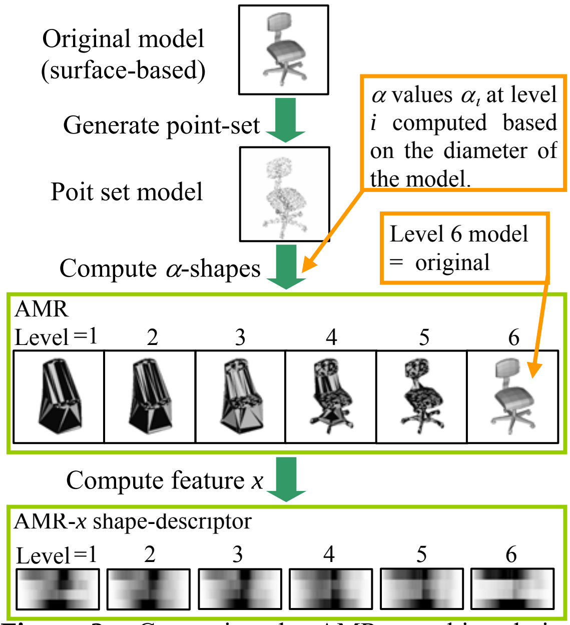

Figure 2. Computing the AMR-x multiresolution shape feature.

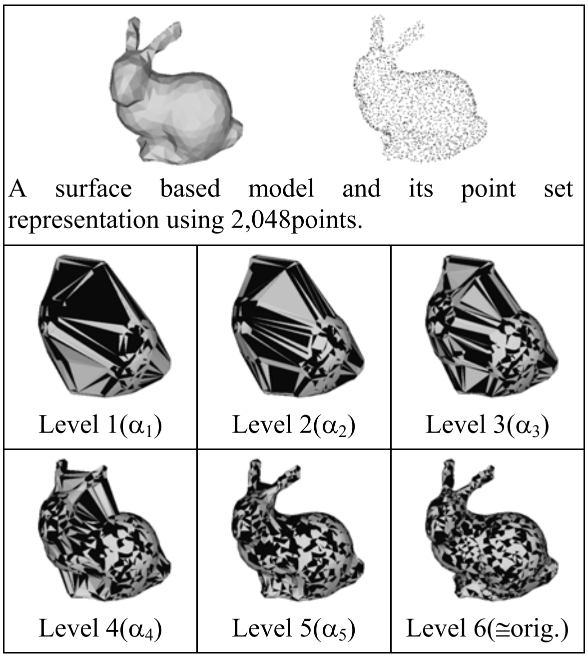

Figure 3. An example of AMR representation for the surface-based model of a rabbit.

3.1.2. Heterogeneous shape descriptors

As the heterogeneous shape descriptors, we used the *D2* by Osada, et al.[Osada02], the *AAD* by Ohbuchi et al. [Ohbuchi05], the *SPRH* by Wahl, et al. [Wahl03], and the *LFD* by Chen, et al. [Chen03]. The D2 is a 1-dimensional (1D) histogram of distances between every pair of points generated on surfaces of a model. The AAD and the SPRH are both extensions to the D2 above. In addition to the distance used in the D2, the AAD and the SPRH extract such features as the mutual orientation of the pair of points, resulting in a 2D histogram for the AAD and a 4D histogram for the SPRH. The LFD by Chen, et al. is different from all of the above. It compares similarities of a set of images generated from multiple viewpoints about the 3D model. According to the survey paper by Tangelder et al. [Tangelder04], the LFD is by far the most powerful shape descriptor.

3.2 Integrating multiple shape descriptors

In combining distances produced by shape descriptors, our method normalizes the distances first, and then computes a weighted linear combination of the distances.

3.2.1 Normalization

Prior to integrating distances, the distances are normalized by their standard deviations. Let U be the set of 3D models in the database, and $o \in U$ be the model from the database, and SD_i, $1 \le i \le N$ be the shape descriptors. For the shape descriptor SD_i, rhe average $\mu(d_i)$ and the standard deviation $\sigma(d_i)$ are computed for the database U. Let $d_i(q,o)$ be the distance prior to normalization computed using the SD_i for the model pair q and o. Then the normalized distance $\hat{d}_i(q,o)$ for the SD_i is computed as below.

$$\hat{d}_i(q,o) = \frac{1}{2}\left(\frac{d_i(q,o) - \mu(d_i)}{3 \cdot \sigma(d_i)} + 1\right) \qquad (1)$$

Figure 4 shows examples of histograms of distances for the four SDs we have used. The distance axis is normalized to its maximum distance value (=100%). It can be seen that the normalization using standard deviation would perform better than the normalization using maximum distance.

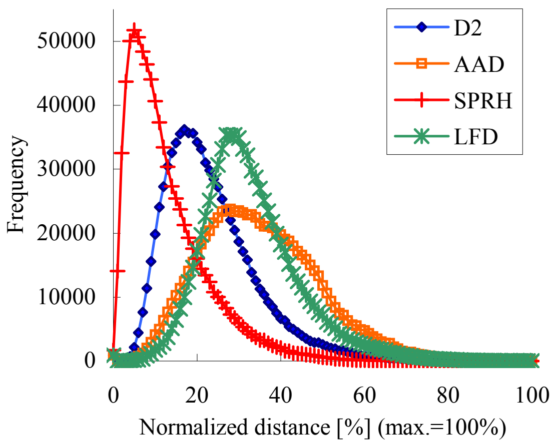

Figure 4. Distribution of distances for the SDs.

Integrating shape descriptors using normalized distance using the formula (1) above should perform better than our previous method described in [Ohbuchi03], which employed no distance normalization at all in integrating SDs generated by using the AMR. Also, we expect our normalization method using standard deviation of distances to be more robust against outliers than the normalization method using maximum distance employed by Bustos, et al. [Bustos04a].

3.2.2. Weighted Linear Combination

Our method computes the overall distance $d(q,o)$ as the linear combination of N normalized distances $\hat{d}_i(q,o)$ using the following formula.

$$d(q,o) = \sum_{i=1}^{N} w_i \hat{d}_i(q,o) \qquad (2)$$

We compared two different methods to determine the weights w_i.

1. Fixed weighting: weights $w_i^{fix.}$ are predetermined and fixed across the query.
2. Adaptive weighting: weights $w_i^{adap.}$ are adaptively varied according to the query and its (estimated) category.

Using fixed weights has two drawbacks. One is that the fixed weights won't produce the best performance across all the classes. Assuming a pre-categorized database, the performance of a SD varies depending on the class the query model (is supposed to) belongs to. That is, for example, one SD is good at querying human figures while another SD is good at querying airplanes. The other is that experimentally finding a best set of weights for a given set of SDs and a database can be computationally expensive for a nontrivial number of SDs and models. An adaptive weighting method that adapts to the query model and/or to the class in which the query model is likely to belong is quite important.

For the adaptive weighting, we adopted Bustos's "*purity*" based weighting scheme [Bustos04a, Bustos04b] with a few minor modifications. The purity is an estimate of "goodness" of a SD in characterizing a category in a given database. The idea behind the purity is the maximum information gain criteria for selecting an attribute in splitting a set in the decision tree learning algorithm. The purity assumes that the database of the 3D models is pre-classified into M classes. Let S_i^k be the number of models retrieved from a class C_k, $1 \le k \le M$, using the shape descriptor SD_i. The purity $purity(SD_i, q, t)$ is computed as below for a shape descriptor SD_i, query q, and a positive integer constant t.

$$purity(SD_i, q, t) = \max_{1 \le k \le M}\left(\left|S_i^k\right|\right) \quad (3)$$

In other words, the purity for the SD_i is higher if the SD returned more model from the category C_k in the top t retrievals, regardless of the class.

Using the $purity(SD_i, q, t)$, the adaptive weight $w_i^{adap.}$ between the query q and the 3D model $o \in U$ is computed as follows.

$$w_i^{adap.} = purity(SD_i, q, t)^n \quad (4)$$

Bustos, et al. used a slightly different $w_i^{adap.}$. Our version could have a larger difference in weights, depending on the selection of the "purity power" parameter n.

$$w_i^{adap.} = purity(SD_i, q, t) - 1 \quad (5)$$

In the following, we call our weighting method purity* and Bustos's original adaptive weighting method purity.

4. EXPERIMENTS AND RESULTS

We implemented our own versions of the D2 [Osada02] and the SPRH [Wahl03] shape descriptors. We used our original implementation of the AAD [Ohbuchi05] shape descriptor. For the LFD [Chan03], we used the executable provided by the original authors of the LFD paper [Chan03] found at their web site. In our previous work [Ohbuchi03], we combined the AMR only with the AAD shape descriptor. In this work, we combine the AMR approach with all the four shape descriptors listed above.

For the performance evaluation experiments, we used the *Princeton Shape Benchmark* (PSB) [Funkhouser03] database. It contains the total of 1914 models, divided into the 907 model training set and the 907 model test set. Each set is classified into 90+ "base" classes. We used the most detailed "base" classifications for the experiment.

As quantitative measures of performance, we used the *R-precision* (1R), the *2R-precision* (2R), the *11 point average precision* (11P) figures, and the precision-recall plot [Baeza-Yates99]. The R-precision is the ratio, in percentile, of the models retrieved from the desired class C_k (i.e., the same class as the query) in the top R retrievals, in which R is the size of the class $|C_k|$. The 2R-precision is similar to the R-precision, except that the figure is computed using the top 2R retrievals. In computing the R- and 2R-precision values, the query q is not counted as the retrieved model. That is, the q is drawn from the database U, the 1R and 2R values are computed by using $|C_k - 1|$. The 11-point average 11P is the average of precision values taken at 11 equally spaced recall values {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. A 11P average precision value can be considered as a summary of the recall-precision plot, which emphasizes overall performance. The 1R and the 2R values favor methods having higher precision for the "near the top" retrievals.

4.1. Multiresolution shape descriptors

In this experiment, we evaluated the effectiveness of integrating multiresolution shape descriptors. We first compared the retrieval performances of the shape descriptors at different resolution levels. Interestingly enough, as shown in Table 1, the most detailed models (the original models) may not achieve the highest retrieval scores. In the cases of the D2 (not shown), the AAD, and the SPRH, retrieval using coarser scales (the level 5, and to a lesser degree, the level 4), produced better 1R score than using level 6 (i.e., most detailed) models. For the LFD, however, retrieval using the most detailed models produced the highest 1R score. A possible explanation for this is that the LFD favors shape details of the models and that the corners and edges of the convex hull models interfere with the performance of the shape descriptor. Note also that retrieval using the coarsest level (i.e., convex hull) models performed surprisingly well.

As shown in Table2 and in Figure 5, for some of the SDs, combinations of multiresolution (MR) shape descriptors using either the fixed or the adaptive weights significantly outperformed their single-resolution (SR) counterparts. In the case of the AAD, 1R figure improved by 10% from 24.4% for the SR version to 34.9% for the MR version using the purity* weighting. The performance of the SPRH increased from 28.6% (SR) to 36.6% (MR), approaching the 38.0% of the LFD. The performance of the LFD, however, did not improve due to the MR combination. Also, adaptive weighting using the purity* showed small but consistent advantage over their fixed weighting counterparts for all of the four SDs tested.

Table 1. R-Precision varies across resolution levels. As the numbers show, the most detailed model may not be the best choice for retrieval.

Resolution levels	R-precision [%]			
	D2	AAD	SPRH	LFD
1	18.53	21.50	24.50	28.32
2	18.12	21.97	25.67	28.71
3	19.96	24.02	28.99	30.62
4	20.27	25.54	30.42	32.59
5	20.92	26.78	31.15	34.83
Orig=6	18.72	24.41	28.57	37.96

Table 2. The multiresolution (MR) versions of D2, AAD, and SPRH outperform their single resolution (SR) counterparts. Such is not the case for the LFD.

SDs	Weights	1R	2R	11P
D2		18.72	27.23	0.314
MRD2	Fixed, 111111	25.14	38.84	0.375
	Fixed, 123456	24.62	33.34	0.369
	Adaptive, purity*	25.45	33.54	0.379
AAD		24.41	34.40	0.364
MRAAD	Fixed, 111111	35.11	44.81	0.446
	Fixed, 123456	33.19	43.06	0.456
	Adaptive, purity*	34.89	45.38	0.471
SPRH		28.57	38.66	0.404
MRSPRH	Fixed, 111111	35.11	44.81	0.468
	Fixed, 123456	35.27	45.69	0.470
	Adaptive, purity*	36.60	45.08	0.482
LFD		37.96	48.74	0.496
MRLFD	Fixed, 111111	35.90	44.22	0.474
	Fixed, 123456	37.10	45.43	0.484
	Adaptive, purity*	37.39	45.61	0.489

4.2 Weighting methods

We compared the performance of the purity* based weighting for different values of the parameter *purity power n* in the equation (2). We also compared the performance of our purity* and the original purity by Bustos, et al. [Bustos04a].

Table 3 shows the performance of the MRLFD-MRAAD-MRSPRH combination using different values of *purity power n*. In terms of 1R scores, $n = 3.0$ produced the best 1R performance, and $n = 9$ produced the best 11P performance. Table 4 compares the Bustos's purity with our purity* for their retrieval performance. In all the combinations tested, our purity* using the selected parameter n performed better than the original purity of Bustos et al. Our purity* performed better probably because the weight can have a larger dynamic range. As a disadvantage, our scheme requires a search for the bester value n, although the search should be relatively easy for it is a 1D search space.

Table 3. Effects of purity power *n* on the retrieval performance for the MRLFD-MRAAD-MRSPRH combination.

	N	1R	2R	11A
Bustos's purity		42.01	52.02	0.538
Purity*	1.0	41.26	51.49	0.527
Purity*	3.0	42.40	52.34	0.550
Purity*	5.0	42.30	51.97	0.562
Purity*	9.0	41.53	50.86	0.567
Purity*	30.0	40.33	49.30	0.563

Table 4. Comparison of retrieval performance between Bustos's *purity* v.s. our *purity**.

Descriptors	Weights		1R	2R	11P
MRAAD	purity		33.19	42.77	0.455
	purity*	*n*=2.0	33.19	43.06	0.456
MRSPRH	purity		35.64	44.75	0.451
	purity*	*n*=3.0	36.60	45.07	0.482
SPRH-AAD	purity		27.69	37.05	0.385
	purity*	*n*=4.0	29.61	39.54	0.417
LFD-AAD	purity		39.15	49.11	0.497
	purity*	*n*=2.0	41.60	51.93	0.531
LFD-SPRH	purity		40.95	50.55	0.517
	purity*	*n*=2.0	42.46	52.68	0.539
LFD-SPRH-AAD	purity		40.57	50.72	0.518
	purity*	*n*=3.0	42.72	52.70	0.542

4.3 Combination of heterogeneous shape descriptors

In this experiment, we compared the performance of integrated SDs using both multiresolution and heterogeneous combinations of various SDs. Table 5 summarizes the experiment. In all the results listed, we used the purity* adaptive weightings. In each combination, a best performing parameter n is chosen out of the 10 candidates values of n={1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0}. The Figure 6 shows the recall-precision plot of five combinations selected from the Table 5.

Table 5 and Figure 6 clearly show that combining multiple, heterogeneous shape descriptors via distances could produce significant performance gain compared to any one of the single resolution SDs. As a reference, the LFD has the 1R precision of 38%. The combination of LFD, AAD, and SPRH produced nearly 5% performance gain over that of the LFD, resulting in the 1R precision of 42.5%.

Table 5. Performance comparison of some of the combinations tested using both heterogeneous and multiresolution shape descriptors. All the combinations used the purity* adaptive weighting.

Shape descriptors	1R	2R	11P
LFD	37.96	48.74	0.496
SPRH	28.57	38.66	0.404
MRSPRH	36.60	45.08	0.482
SPRH+AAD	29.61	39.54	0.417
MRSPRH+MRAAD	37.02	46.96	0.491
LFD+AAD	41.60	51.93	0.531
LFD+MRAAD	39.81	49.94	0.520
LFD+SPRH	42.52	52.68	0.537
LFD+MRSPRH	42.71	51.74	0.541
LFD+AAD+SPRH	42.72	52.70	0.542
LFD+MRAAD+MRSPRH	41.37	51.15	0.534
MRLFD+MRAAD+MRSPRH	42.31	51.97	0.543

On the other hand, combinations of the LFD with the MR shape descriptors did not produce consistent and significant performance gain. The reason for this may be attributed to the AMR model, the purity* adaptive weighting scheme, or both. Further study is needed to determine the exact cause.

5. CONCLUSION AND FUTURE WORK

In this paper, we explored a systematic approach for improving shape-based retrieval performance of 3D shape models. Our approach is to (1) extract as many (mutually independent) shape features as possible, and (2) combine the distances computed using these features by using an adaptive weighting scheme. To extract different shape features, the method employed a combination of multiresolution shape descriptors and heterogeneous shape descriptors. We used the 3D alpha-shape [Edelsbrunner94] based method we have proposed previously [Ohbuchi03] to capture multiresolution shape features. To combine distances computed using these shape descriptors, we adopted Bustos's purity weighting scheme with slight modification.

Experiments showed that the proposed method of integrating multiresolution and heterogeneous shape descriptors is effective in improving retrieval performance. Many combinations of the shape descriptors we tested surpassed the performance of the arguably the best (single) shape descriptor, the Light Field Descriptor by Chen, et al. [Chen03].

We intend to explore better adaptive weighting schemes and better multiresolution shape feature extraction approaches.

6. ACKNOWLEDGMENTS

This research is funded, in part, by the Grants-in-Aid for Scientific Research from the *Japan Society for the Promotion of Science* (No. 17500066).

7. REFERENCES

[Atmosukarto05] I. Atmosukarto, W.K. Leow, Z. Huang, Feature Combination and Relevance Feedback for 3D Model Retrieval, *Proc. 3DPVT* 2005, pp. 334-339, (2005).

[Bustos04a] 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).

[Bustos04b] B. Bustos, D. Keim, D. Saupe, T. Schreck, D. Vranić, Using Entropy Impurity for Improved 3D Object Similarity Search, *Proc. IEEE ICME 2004* (2004).

[Chen03] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On Visual Similarity Based 3D Model Retrieval, *Computer Graphics Forum*, **22**(3), pp. 223-232, (2003).

[Edelsbrunner94] H. Edelsbrunner, E. P. Mücke, Three-dimensional Alpha Shapes, *ACM TOG*, **13**(1), pp. 43-72, (1994).

[Funkhouser03] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3D models, *ACM TOG*, **22**(1), pp. 83-105, (2003).

[Funkhouser04] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, David Dobkin, Modeling by ex-ample, *ACM TOG*, **23**(3), pp.652-663 (2004).

[Iyer03] Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., Ramani, K., A Reconfigurable, Intelligent 3D Engineering Shape Search System Part I: Shape Representation, *Proc. ASME DETC* '03, 23rd CIE Conf. (2003).

[Iyer05] M. Iyer, S. Jayanti, K. Lou, Three Dimensional Shape Searching: State-of-the-art Review and Future Trends, *Computer Aided Design*, **5**(15), pp. 509-530, 2005.

[Leifman03] G. Leifman, S. Katz, A. Tal, R. Meir., Signatures of 3D Models for Retrieval, *The 4th Israel-Korea Bi-National Conf. on Geom. Modeling and Comp. Graph.*, February 2003, 159-163. (2003).

[Ohbuchi05] Ryutarou Ohbuchi, Takahiro Minamitani, Tsuyoshi Takei, Shape-similarity search of 3D models by using enhanced shape functions, *International Journal of Computer Applications in Technology* (*IJCAT*), **23**(3/4/5), pp. 70-85, (2005).

[Ohbuchi03] R. Ohbuchi, T. Takei, Shape-Similarity Comparison of 3D Shapes Using Alpha Shapes, *Proc. PG 2003*, pp. 293-302, IEEE Press, (2003).

[Osada02] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape Distributions, *ACM TOG*, **21**(4), pp. 807-832, (2002).

[Shilane04] 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

[Tangelder04] J. Tangelder, R. C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods, *Proc. SMI '04*, pp. 145-156.

[Wahl03] E. Wahl, U. Hillenbrand, G. Hirzinger, Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, *Proc. 3DIM 2003*, pp. 474-481, IEEE Press, (2003).

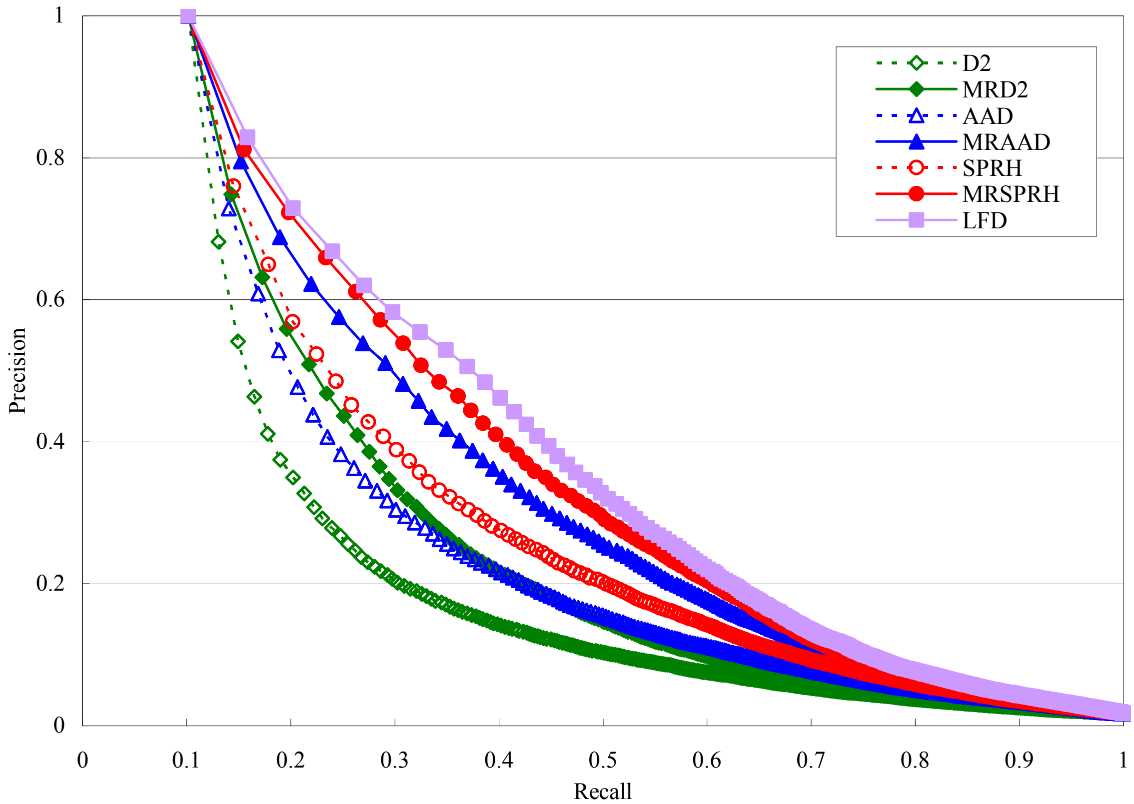

Figure 5. Performance gain due to the multiresolution approach. The multiresolution shape descriptors using AMR increased performance of many but not all of the SDs.

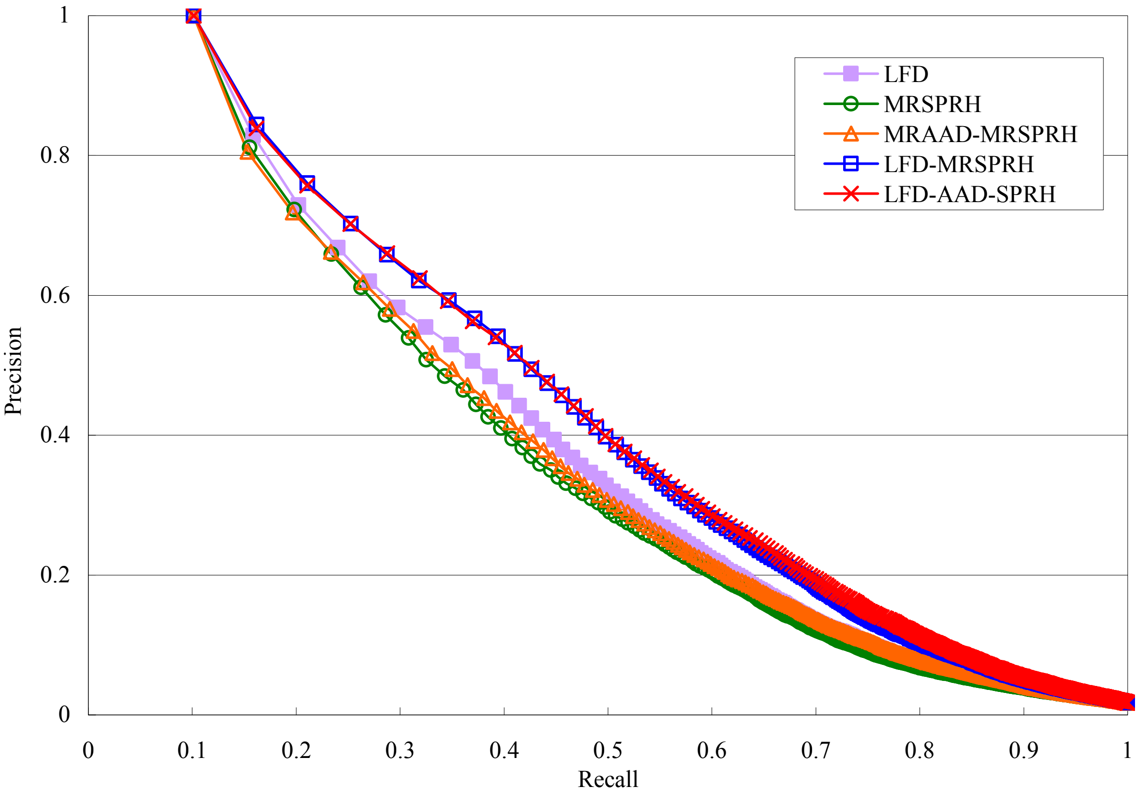

Figure 6. Precision-recall plots of some of the combinations of descriptors. Some of the combinations significantly outperform the LFD [Chen03], arguably the best single shape descriptor to date.