Automatic Selection and Combination of Descriptors for Effective 3D Similarity Search

Benjamin Bustos Daniel Keim Dietmar Saupe Tobias Schreck Dejan Vranić
Department of Computer and Information Science, University of Konstanz
{bustos,keim,saupe,schreck,vranic}@informatik.uni-konstanz.de

Abstract

We focus on improving the effectiveness of similarity search in 3D object repositories from a system-oriented perspective. Motivated by an effectiveness evaluation of several individual 3D retrieval methods, we research a selection heuristic, called purity, *for choosing retrieval methods based on query-dependent characteristics. We show that the purity selection method significantly improves the search effectiveness compared to the best single methods. We then show that retrieval effectiveness can be further boosted by considering* combinations *of multiple retrieval methods to perform the search. We propose to use a dynamically weighted combination of feature vectors based on the purity concept, and we experimentally show that the search effectiveness of our combined methods by far exceeds the effectiveness of our best implemented single method.*

Keywords: 3D objects, information retrieval, query by content, effectiveness, feature selection.

1. Introduction

The development of effective content-based multimedia search systems is an important research issue due to the growing amount of digital audio-visual information. In the case of images and video, the growth of digital data has been observed since the introduction of 2D capture devices. A similar development is expected for 3D data, as acquisition and dissemination technology is constantly improving. In digital libraries, it is possible to search using annotation information, which describes the content of an object in textual form, or using the multimedia data itself, the so-called *content-based search*. The latter is the more promising approach, because in general textual descriptions are manually created, which is prohibitively expensive, and they are subject to the opinion of the person who creates them. In contrast, content-based search algorithms allow an implementation of fully automatic retrieval systems.

There are many practical applications of similarity search in 3D libraries. In medicine, the detection of similar organ deformations can be used for diagnostic purposes. In the manufacturing industry, the search for similar standard parts can help to reduce costs. There are also applications in the entertainment industry, e.g., film production and video games. Figure 1 illustrates the concept of content-based 3D similarity search.

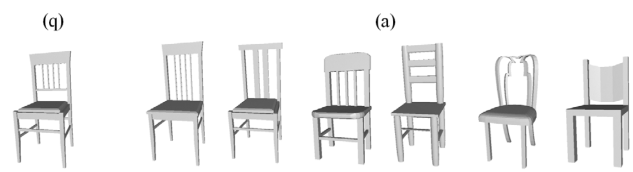

Figure 1. Example of a similarity query in a 3D object database, showing a query object (q) and a set of possibly relevant objects (a).

In this paper, we experimentally compare a range of different 3D feature vectors (FVs), and we propose methods for improving the effectiveness of the 3D similarity search process. Our new method, called *purity selection*, determines which one of the available FVs should be engaged depending on the query object. This leads to a significant improvement in retrieval effectiveness compared with the best single FV. Also, we propose to use combinations of FVs, which leads to further significant improvements of the effectiveness of the search system. Our experimental results show that the relative ordering of FVs by retrieval effectiveness depends on the query point, which means that no single FV outperforms all other FVs on all queries, and that linear combinations of multiple FVs provide a significant improvement of the retrieval effectiveness compared to the best single FV.

Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04)

2. Similarity search of 3D objects

2.1. Feature-based approach to 3D retrieval

Usually, a *feature vector* approach is adopted for performing similarity search in multimedia databases. The basic idea is to derive a vector of numerical (real) values for each of the objects in the repository, extracting those properties of the 3D objects that best support an application-dependent notion of similarity. For a given feature extraction technique, it is usually possible to generate FVs of different dimensionality by setting the resolution with which the FV extraction proceeds. FVs describing 3D objects may be derived from object geometry and/or other attributes and should be *invariant* to changes in the orientation (translation, rotation and reflection) and scale of 3D models. Good FVs should also be *robust* with respect to small changes in the level-of-detail, geometry and topology of the models.

Given the FVs for all of the objects in a database and for a query point, the retrieval of similar objects is performed by returning the k nearest neighbors (k-NN) of the query point. To this end, a metric in the vector space $\mathbb{R}^d$ (for dimensionality d depending on parameterization of the FV extraction method at hand), is used, e.g., the unweighted *Minkowski* (l_p) distance, given by $l_p(\vec{x}, \vec{y}) = \left(\sum_{1 \leq i \leq d} |x_i - y_i|^p\right)^{1/p}$, $p \geq 1$. More sophisticated metrics for vector spaces, e.g., quadratic forms [2], exist, but their applicability depends on the FV definition and computational efficiency considerations.

2.2. Related work and studied feature vectors

The last few years have seen a strongly increasing interest in content-based retrieval of 3D models, and its popularity may be expected to eventually approach the popularity of similarity search in image databases. Algorithms have been proposed to extract FVs based on many different 3D object characteristics. Statistical FVs include geometric moments [21, 16, 14, 19], and histograms of measures like the distribution of distances between points on an object's surface [15]. Some of the extention-based methods treat 3D objects as functions defined on spheres, and describe the objects in terms of samples taken from these functions [25, 20, 13]. Many algorithms derive object descriptions from certain space partitioning schemes [1, 10, 22, 5, 18, 12]. Furthermore, the curvature of an object's surface may be considered as in [26]. FVs may also be obtained from 2D renderings of the objects as in [7, 4]. There also exist non-FV approaches to 3D retrieval, which rely on topological [8] or skeletal descriptions of the models [17]. Considering the specific problem of securing rotation invariance of the description, there exists an ongoing discussion whether this should be achieved by the application of an rotation normalization step prior to feature calculation [20], or by the definition of FVs that are implicitly rotational invariant [11]. In our work, we consider FVs that rely on rotation normalization by using a variant of the Principal Component Analysis (PCA) [25], as we believe it is stable in many cases, and able to contribute valuable information to the object description.

While we have implemented many different FVs from our own as well as other researchers work in our 3D similarity search system, for clarity reasons we focus the remainder of this paper to a set of six algorithms which belong to the FVs providing the best retrieval precision in our experiments. Specifically, we consider two FVs based on the Fourier transformation of rendered silhouettes and Z-buffers of the 3D models, resulting in the *silhouette* and *depth buffer* FVs respectively, as presented in [7]. Also, we consider the spherical harmonics transform of samples taken from model extention, as well as samples taken from a combination of model extention and surface orientation properties, resulting in the *ray-based* and the *complex* FVs respectively, as introduced in [24]. Furthermore, we include in this study a FV based on the discretization of model surface into a voxel grid (the *voxel* FV, as introduced in [23]), and our implementation of the implicitly rotation invariant algorithm introduced in [5] which is based on the spherical harmonics transform of concentric functions defined on the voxelization of models, and that we would like to call the *harmonics 3D* FV in this paper.

3. Measuring retrieval effectiveness

3.1. Description of the experiments and the effectiveness measures

The database used for retrieval experiments contains 1,837 3D objects collected from the Internet [1]. From this set, 472 objects were classified by shape into 55 different model classes, yielding the ground truth, and the rest of them were left as "unclassified". Each classified object was used as a query object, and the objects belonging to the same model class, excluding the query, were considered relevant to it.

Table 1 gives a partial description (first 20 classes) of the classified objects of the database. The first column indicates the class identification number. The second column describes the 3D class models. The last column lists the number of objects per model class.

For comparing the effectiveness of the search algorithms, we use *precision vs. recall figures*, a standard evaluation technique for retrieval systems [3]. *Precision* is the frac-

1 The database is available for downloading at `http://-merkur01.inf.uni-konstanz.de/CCCC/`

Class id #	Description	# of models
1	ants	6
2	rabbits	4
3	cows	7
4	dogs	4
5	sea animals	13
6	bees	5
7	CPU's	4
8	keyboards	8
9	cans	4
10	bottles	14
11	bowls	4
12	pots	4
13	cups	8
14	wine glasses	9
15	teapots	4
16	biplanes	5
17	helicopters	9
18	missiles	16
19	jet planes	18
20	fighter jet planes	26

Table 1. Partial description of the classified set of our 3D object database.

tion of the retrieved objects which are relevant to a given query, and *recall* is the fraction of the relevant objects which have been retrieved from the database. In addition, we also consider the widely used *R-precision* measure [3]. It is defined as the precision when retrieving as many objects as there are relevant answers in the database, w.r.t. the query. We average these measures over all queries that belong to one of the predefined query classes. As the metric of choice we employ the l_1 (Manhattan) distance, as we experimentally found this gives us the best retrieval results compared to other Minkowski distances (l_1 was slightly but consistently better than l_2 in our experiments).

3.2. Results using single feature vectors

In our first experiments, we compared the retrieval performance of six FVs using our ground truth. To first assess the influence of FV resolution, we evaluated a range of FV dimensionality settings. Figure 2 shows the effect of the FV dimensionality on the overall effectiveness of the FVs, measured in terms of R-precision. The figure shows that the effectiveness improvement rate diminishes quickly for roughly more than 64 dimensions for most FVs. It is interesting to note that the saturation effect is reached at roughly the same dimensionality level. This is not an expected result, considering that the different FVs describe different characteristics of the 3D objects.

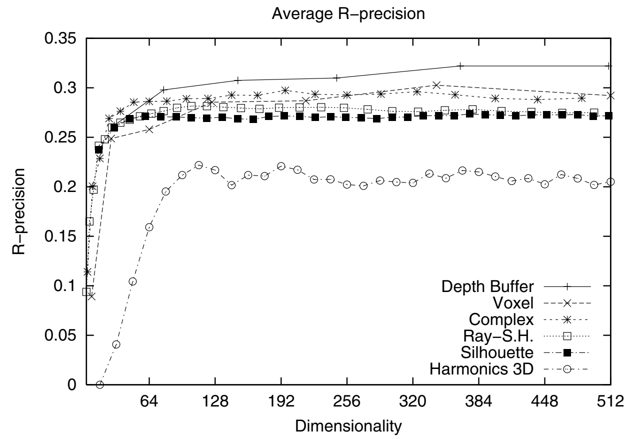

Figure 2. R-precision as a function of the dimensionality of the feature vectors.

Figure 3 shows the database-average effectiveness performance (precision vs. recall curves and R-precision values) of the six FVs when using their best dimensionality parameterization, respectively. The best performing FV on average is the image-based depth buffer FV. Between the 1^{st} and the 3^{rd} best FV, the performance differences are small, implying that in practice all these FVs have similar retrieval capabilities.

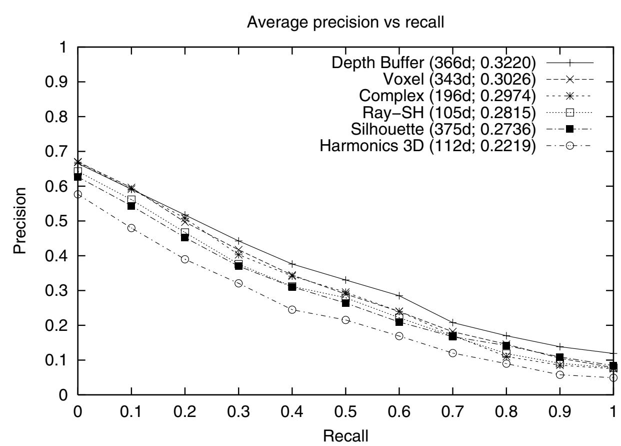

Figure 3. Average precision vs. recall figures for all feature vectors (the legend includes the optimal dimensionality and the R-precision values).

Next, we present the results obtained for two specific

0-7695-2217-3/04 $20.00 © 2004 **IEEE**

model classes. Figure 4 shows the average precision vs. recall figures for the Formula 1 cars model class. In this case, the best effectiveness is obtained with the depth buffer and the harmonics 3D FVs. Note that the best FV for this model class is also the best FV on average. The R-precision value is also given for each FV.

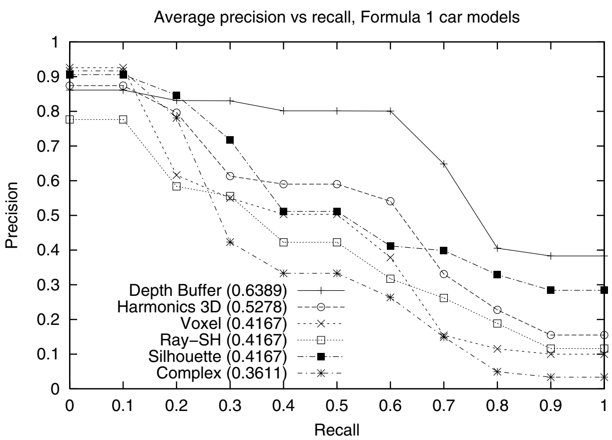

Figure 4. Average precision vs. recall figures and R-precision values for the F-1 cars model class.

Figure 5 shows the average precision vs. recall figure for the sea animals model class. For this class, the best FVs are the silhouette and the ray-based spherical harmonics FVs. This result shows that for some model classes the best average FV (depth buffer) does not perform well. Moreover, the best three FVs for this class are different from the best three FVs of the F-1 cars model class.

In general, we observed that for many query classes, the respective ranking of FVs by retrieval precision differs from the average ranking. It follows that an *appropriate selection* of the FV used for the similarity search, depending on the query object, will improve the overall retrieval effectiveness as compared with the standard policy of always choosing a certain default FV.

4. Purity-based feature selection

The previous results indicate the need for appropriate feature selection, based on the query to be evaluated. At our disposal is a set of FVs, but how can we automatically estimate the quality of the result when choosing one of them? Note that we want to support similarity queries. In classification, feature selection refers to choosing the features that optimize an objective function (usually, classification accuracy). For similarity search, we first need to find such an objective function estimating the quality of a similarity ranking.

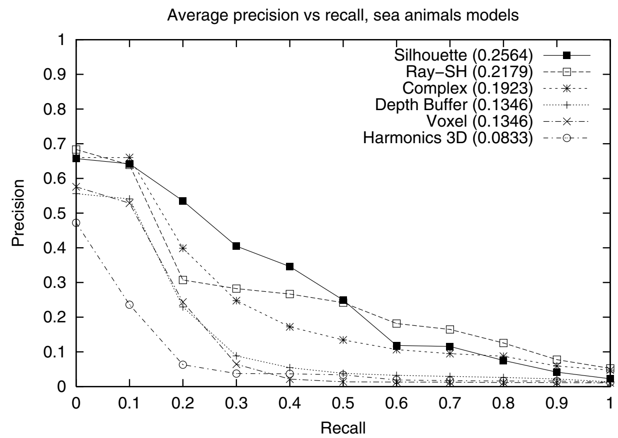

Figure 5. Average precision vs. recall figures and R-precision values for the sea animals model class.

4.1. Query dependent selection of features

Let X be the universe of valid 3D objects, $U \subseteq X$ the 3D database, and $Q \subseteq U$ a *set of classified objects*, that is, $Q = \biguplus_{i=1}^{m} Q_i$, where Q_i is a *model class* (i.e., a set of similar objects), and Q is the disjoint union of m model classes.

Definition 1 *Given a set of ℓ FVs $\{f_1, \ldots, f_\ell\}$, a 3D query object $q \in X$, and a constant value $k \in \mathbb{N}^+$, we generate ℓ object rankings, one for each FV, consisting of the distances between q and every object of Q sorted in ascending order. Let R_{qk}^j be the first k positions of the ranking using f_j, and let $S_i^j = R_{qk}^j \cap Q_i$. The* purity *of f_j for the query q is defined as:*

$$purity(f_j, q, k) = \max_{1 \leq i \leq m} (|S_i^j|)$$

The purity value indicates the maximum number of objects that belong to a same model class in the first k positions of each ranking. The FV that has the maximum purity is selected for performing the search. In case of ties, we select the FV that has the best average R-precision, using the values of Figure 3 as reference. This *purity selection* method tries to measure the "coherence" of the retrieved objects in the first positions of the ranking. Our hypothesis is that a good FV will rank objects from the same model class at the first positions of the ranking. On the other hand, if a FV ranks objects from different model classes in the first positions, then one can assume that the answer is not coherent and hence the FV is not suitable for this query.

Figure 6 shows a comparison between the purity selection technique and the best FVs, using the set of classified objects as the set Q for the purity computation. We tested values for k from 3 up to 10, noticing only small variations in the results (in the figure, we show the results using $k = 7$). With the purity selection method, we obtained an improvement of 21% in R-precision compared to the best average single FV, which is a significant gain in retrieval effectiveness.

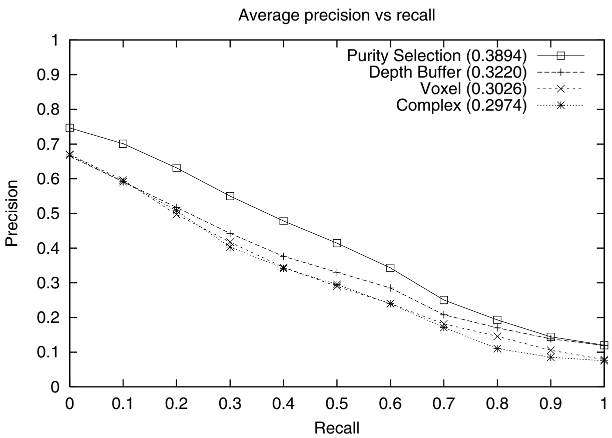

Figure 6. Comparison between the purity selection method and the best feature vectors.

To discard that the selection of the model classes could have some influence on the obtained results, we qualitatively validated the obtained results running a cross-validation test [6]. We divided a subset of the classified objects into two equally sized groups: A *query set* and a *test set*. The query set was used as in the described experimental framework. For computing the purity values we only used the test set, which was considered to be "out" of the database. That is, the test set was not considered for computing the effectiveness of the search system. Partitioning of the classified set was randomly performed, and we average over 100 random partitions. The results of the cross-validation test confirm the significant improvement of our purity selection method.

5. Combinations of feature vectors

The retrieval performance analysis in Section 3.2 suggests that there exist a number of FVs that achieve good average retrieval performance on the majority of query classes, but that there is no clear winner among them. Instead, the individual FVs have different strengths and weaknesses, and they represent complementary information regarding the description of 3D objects.

Because FVs capture different aspects and characteristics of the models, we propose to use *combinations of FVs* for further improving the retrieval effectiveness of the similarity search, thus avoiding the disadvantages of using a single feature, which captures only a single characteristic of an object.

So, how can different FVs be combined in a search system? A simple concatenation of all available FVs is not advisable due to effectiveness and efficiency reasons [9]: Effectiveness would degrade with the inclusion of FVs irrelevant to the queries, and efficiency would also degrade because of the large dimensionality of the resulting FV, a problem known as the *curse of dimensionality*. Therefore, it is an interesting problem to find whether there are combinations of FVs that are better suited for performing similarity search on certain object classes, or even if there are combinations that dominate others for all types of queries.

We propose two methods for combining FVs: An unweighted combination method, and a weighted combination method based on the purity concept.

5.1. Unweighted combinations of feature vectors

We ran retrieval experiments on all possible combinations of all FVs, using their best dimensionality given by Figure 3. This gives a total of $\sum_{k=2}^{6}\binom{6}{k} = 57$ different combinations of FVs. To construct the combinations, we use the sum of the unweighted normalized distances.

Definition 2 *The unweighted normalized combined distance d^c is defined as:*

$$d^c(q,o) = \sum_{i=1}^{N} b_{c_i} \frac{d_i(q,o)}{dmax_i(q)}$$

where N is the total number of FVs, b_{c_i} is a binary variable that indicates whether FV f_i is included in combination c, $d_i(q,o)$ is the distance of a query object q from another object o under f_i, and $dmax_i(q)$ is the maximum distance of object q to any other object in the database as measured by f_i.

As in the single FVs experiments, the combined distance d^c gives the ranking of objects w.r.t. a query q. The unweighted combination approach treats all FVs of the combination as equally important in determining the ranking.

Table 2 shows the average effectiveness of the best combinations of FVs in terms of R-precision and combination cardinality. The results confirm our assumption that there exist FV combinations that significantly improve the retrieval performance over the best single FV (depth buffer) in the average case. The maximum R-precision value reached on average over all query classes by a combination amounts to 42.89%, which is equal to an improvement of more than 33% compared to the performance of the depth buffer. This

Comb. #	R-precision	Feature vectors
1	0.3220	Depth Buffer
2	0.3803	Voxel, Complex
3	0.4108	Depth Buffer, Voxel, Complex
4	0.4200	Depth Buffer, Voxel, Complex, Silhouette
5	0.4287	Depth Buffer, Voxel, Complex, Silhouette, Harmonics 3D
6	0.4289	All feature vectors

Table 2. Average R-precision for the best unweighted combinations of feature vectors.

best combination is composed of all six FVs. The largest improvement occurs when changing from the single to the 2-combination case (voxel and complex). The improvement increases further with combination cardinality, but the increment becomes smaller as we add more FVs to the combination. For the last one, the improvement in effectiveness is negligible. Figure 7 shows the precision vs. recall curves for the best unweighted combinations.

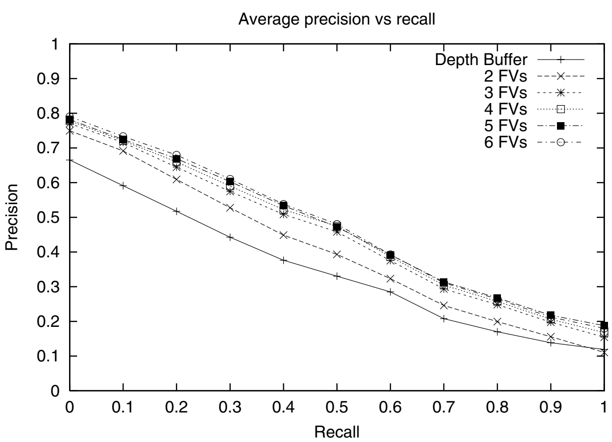

Figure 7. Average precision vs. recall for the best feature vector and the best unweighted combinations for an increasing numbers of feature vectors.

Note that we also performed a much larger series of experiments considering combinations of up to nine FVs. In these experiments we found that the retrieval effectiveness even starts to decrease when adding more FVs after a certain saturation point has been reached.

5.2. Weighted combination of feature vectors

A further improvement over the unweighted combination of FVs can be achieved by assigning *weights* to each FV in the combined distance, because it is expected that not all FVs are equally relevant to all queries, and using a non-suitable FV can even lower the effectiveness of the search. We tested all possible weightings for the combination of the six FVs using three different weight values (0, 1, 2), resulting in $3^6 - 1 = 728$ different combinations. We call this approach *fix-weighted combination*, because each combination uses the same set of weight values $w = \{w_1, \dots, w_6\}$ for all queries. The weights are assigned to each FV in the order given by Figure 3 (e.g., w_1 corresponds to depth buffer, w_2 corresponds to voxel, and so on).

Definition 3 *The fix-weighted combined distance is defined as:*

$$d_{fix\text{-}weighted}(q, o) = \sum_{i=1}^{N} w_i \frac{d_i(q, o)}{dmax_i(q)}$$

The experimental results show that the set of weights $w^* = \{2, 1, 2, 0, 1, 1\}$ provides the best performance. The precision vs. recall plot is shown in Figure 8. While this weight vector provides excellent retrieval performance, it is expected to be highly correlated to our database. Thus, it will probably not be useful for another 3D objects database, because the optimal average weighting may be different. Moreover, in a dynamic database, it is not possible to determine the best weighting factors by experimentally analyzing all combinations of weighting factors for all possible queries. All these negative attributes make this approach unpractical for real-world applications.

To overcome these problems, we propose another combination technique. The *purity-weighted combination* uses a dynamically determined weighting scheme based on the purity concept. The combined distance is defined as follows.

Definition 4 *The purity-weighted combined distance is defined as:*

$$d_{p\text{-}weighted}(q, o) = \sum_{i=1}^{N} (purity(f_i, q, k) - 1) \frac{d_i(q, o)}{dmax_i(q)}$$

Figure 8 shows the average precision versus recall figures for both weighted FV combination methods, and for each single FV. For the purity value computation, we show the results using $k = 4$ (results using values between 3 and 10 are all similar). For the fix-weighted combination, we show the result of using w^* as the fixed weighting scheme. The improvement obtained with the weighted combinations (38% improvement in R-precision compared with the best single FV) is far superior to the improvement obtained when switching from one single FV to the next best single FV. Both weighting combination methods have almost the same effectiveness on average, but notice that in the case of the fix-weighted combination we had to perform a brute force search to find the best weighting values (which, probably, are not optimal w.r.t. a different database). In contrast, the purity-weighting method automatically determines the weights for each FV depending on the query object.

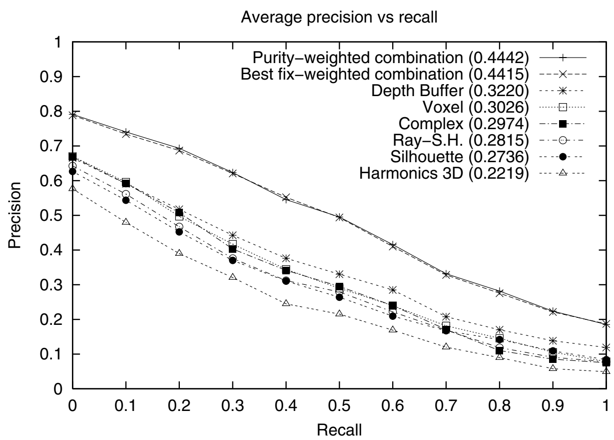

Figure 8. Average precision vs. recall figures for weighted combinations and its components.

Figure 9 shows the average precision versus recall figures for the Formula 1 cars model class, where significant improvements in retrieval effectiveness occur with the purity-weighted combination of FVs over the best single FV (39% in terms of R-precision). The method sustains a near-perfect precision level for almost all recall levels, with just a small degradation for the very high recall levels.

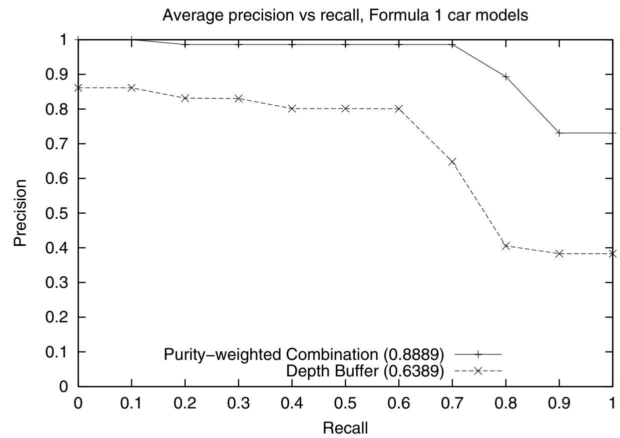

Figure 9. Average precision vs. recall figures, purity-weighted combination of feature vectors, F-1 cars model class.

Method	R-precision	Improvement
Best single feature vector	0.3220	0%
Purity selection	0.3894	21%
Best unweighted combination	0.4289	33%
Best fix-weighted combination	0.4415	37%
Purity-weighted combination	0.4442	38%

Table 3. Improvements in effectiveness obtained with the proposed techniques.

6. Conclusions and future work

In this paper, we described the challenges involved in the implementation of a content-based 3D similarity search system. Our first contribution is the proposal of a new selection method based on our purity concept, which determines the FV to be used for retrieval. The results show a significant effectiveness improvement over the best single FV. Secondly, we established that combinations of FVs may be highly beneficial for improving retrieval effectiveness in a 3D search system. We proposed to use a dynamically weighted combination of FVs based on the purity measure, thus avoiding the disadvantages of using just a single FV for the search. The experimental results show that the weighted combination of FVs further improves the retrieval effectiveness of the search system, and this improvement is far superior to the improvement obtained when switching from one type of FV to another. Table 3 summarizes the improvements obtained with the proposed techniques. Note that an improvement of 38% in effectiveness is very significant compared to the improvements of recently proposed FVs, which in most cases is in the order of 5% over previous methods.

It is worth noting that the proposed techniques are general and not restricted to 3D objects, and that they can be used with any multimedia data type (images, audio, etc.) on which a distance metric is defined. Future work involves further researching query dependent feature selection and combination methods for other types of multimedia data. The final goal is to define a query processor that does not need a classified set of objects but is still capable of determining a good combination of feature vectors given a query object. It is also an open issue, how the efficiency of the search system can be improved. The need for appropriate indexing techniques, considering the very high dimensionality of the combined feature vectors (hundreds of dimensions) is obvious.

Acknowledgments

This work was partially funded by the German Research Foundation (DFG), Projects No. KE 740/6-1 and No. SA 449/10-1, within the strategic research initiative "Distributed Processing and Delivery of Digital Documents" (V3D2), SPP 1041. It was also partially funded by the Information Society Technologies program of the European Commission, Future and Emerging Technologies under the IST-2001-33058 PANDA project (2001–2004). The first author is on leave from the Department of Computer Science, University of Chile.

References

[1] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *6th Intl. Symp. on Advances in Spatial Databases*, LNCS 1651, pages 207–226. Springer, 1999.

[2] M. Ankerst, H.-P. Kriegel, and T. Seidl. A multistep approach for shape similarity search in image databases. *IEEE Trans. on Knowledge and Data Engineering*, 10(6):996–1004, 1998.

[3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

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

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

[6] J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kauffman, 2001.

[7] M. Heczko, D. Keim, D. Saupe, and D. Vranic. Methods for similarity search on 3D databases. *Datenbank-Spektrum*, 2(2):54–63, 2002. In German.

[8] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. ACM Intl. Conf. on Computer Graphics and Interactive Techniques*, pages 203–212. ACM Press, 2001.

[9] A. Hinneburg, C. Aggarwal, and D. Keim. What is the nearest neighbor in high dimensional spaces? In *Proc. 26th Intl. Conf. on Very Large Databases*, pages 506–515. Morgan Kaufmann, 2000.

[10] T. Kato, M. Suzuki, and N. Otsu. A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. In *Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics*, pages 2946–2952, 2000.

[11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*, pages 156–164. Eurographics Association, 2003.

[12] H.-P. Kriegel, S. Brecheisen, P. Kröger, M. Pfeifle, and M. Schubert. Using sets of feature vectors for similarity search on voxelized CAD objects. In *Proc. ACM Intl. Conf. on Management of Data (SIGMOD'03)*, pages 587–598. ACM Press, 2003.

[13] G. Leifman, S. Katz, A. Tal, and R. Meir. Signatures of 3D models for retrieval. In *Proc. 4th Israel-Korea Bi-National Conf. on Geometric Modeling and Computer Graphics*, pages 159–163, 2003.

[14] R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Proc. 10th Pacific Conf. on Computer Graphics and Applications*, pages 265–274, 2002.

[15] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. on Graphics*, 21(4):807–832, 2002.

[16] E. Paquet, M. Murching, T. Naveen, A. Tabatabai, and M. Rioux. Description of shape information for 2-D and 3-D objects. *Signal Processing: Image Communication*, 16:103–122, 2000.

[17] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proc. Intl. Conf. on Shape Modeling and Applications*, pages 130–142, 2003.

[18] J. Tangelder and R. Veltkamp. Polyhedral model retrieval using weighted point sets. *Intl. Journal of Image and Graphics*, 3(1):209–229, 2003.

[19] G. Taubin and D. Cooper. Recognition and positioning of rigid objects using algebraic moment invariants. In *Proc. SPIE Conf. on Geometric Methods in Computer Vision*, 1991.

[20] D. Vranic. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *Proceedings of the IEEE Intl. Conf. on Image Processing (ICIP 2003)*, volume 3, 2003.

[21] D. Vranić and D. Saupe. 3D model retrieval with spherical harmonics and moments. In *Proc. DAGM-Symp.*, LNCS 2191, pages 392–397. Springer, 2001.

[22] D. Vranić and D. Saupe. 3D shape descriptor based on 3D Fourier transform. In *Proc. EURASIP Conf. on Digital Signal Processing for Multimedia Communications and Services (ECMCS'01)*, pages 271–274. Comenius University, 2001.

[23] D. Vranic and D. Saupe. A feature vector approach for retrieval of 3D objects in the context of MPEG-7. In *Proc. Intl. Conf. on Augmented, Virtual Environments and Three-Dimensional Imaging (ICAV3D'01)*, pages 37–40, 2001.

[24] D. Vranić and D. Saupe. Description of 3D-shape using a complex function on the sphere. In *Proc. IEEE Intl. Conf. on Multimedia and Expo*, pages 177–180, 2002.

[25] D. Vranić, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In *Proc. IEEE 4th Workshop on Multimedia Signal Processing*, pages 293–298, 2001.

[26] T. Zaharia and F. Prêteux. Three-dimensional shape-based retrieval within the MPEG-7 framework. In *Proc. SPIE Conf. on Nonlinear Image Processing and Pattern Analysis XII*, volume 4304, pages 133–145, 2001.