Using Entropy Impurity for Improved 3D Object Similarity Search

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

***Abstract*— Similarity search in 3D object databases is becoming an important problem in multimedia retrieval, with many practical applications. We investigate methods for improving the effectiveness in a retrieval system that implements multiple feature extraction algorithms to choose from. Our techniques are based on the *entropy impurity* measure, widely used in the context of decision trees. We propose a method for the a priori estimation of individual feature vector performance given a query. We then define two approaches that use this estimator to improve the retrieval effectiveness. Our experimental results show that significant improvements are achievable using these methods.**

I. Introduction

Improvements in 3D scanner technology and the availability of 3D models distributed over the Internet are both contributing to create large databases of this type of multimedia data. Searching 3D databases by content has many promising applications in domains like CAD, medicine, molecular biology, and entertainment, just to name a few.

The 3D similarity search problem can be stated as follows: "Given a 3D object database and a 3D query object, return the objects in the database that are most similar, according to some similarity notion, to the query". Broadly, two different notions of 3D similarity can be distinguished: *Shape* similarity (e.g., two round tables that have similar shapes), and *semantic* similarity (e.g., two tables, regardless their shapes). Both notions can refer to global or partial similarity between 3D objects. Also, the effectiveness of the search is more important than the efficiency, as usually it is not possible to specify an exact matching criterion. We propose two techniques based on the concept of entropy impurity, both aimed at improving the effectiveness of global shape-based similarity search.

II. Feature vector approach for similarity search

In many methods for 3D similarity search proposed until now, the *feature vector* (FV) approach [1] is used. Usually, before feature extraction is performed, the 3D objects are *normalized* by one of the variants of the *Principal Component Analysis* (PCA) [2] which places the objects into a canonical coordinate frame. This normalization step serves to provide invariance with respect to transformations such as rotation, translation, scaling, or reflection of the objects. After normalization, a set of d features (real values) is extracted from the object, assigning each feature value to a coordinate in a vector $x \in \mathbb{R}^d$. Once all 3D objects of the database are transformed, the problem is reduced to the nearest neighbor search in $\mathbb{R}^d$ using any Minkowski (l_p) norm.

There exist a variety of 3D FV extraction methods, derived from different aspects of 3D objects. Many of these methods focus on geometric properties, while others rely on 2D projections of the object. For example, the *depth buffer* FV [3] characterizes 3D objects using an image-based approach. The objects are first PCA-normalized and scaled into the axis-parallel unit cube. Then, six grey-scale depth images are rendered using parallel projection, each two for one of the principal axes. Each pixel encodes the distance from the viewing plane (defined to be the respective side of the unit cube) to the object in an 8 bit grey value. These images correspond to the concept of z- or depth-buffers in computer graphics. After rendering, the 6 images are transformed using the standard 2D discrete Fourier transform, and the magnitudes of certain k first low-frequency coefficients of each image contribute to the depth buffer FV of dimensionality $6k$.

For research purposes, we have implemented 15 different 3D FVs from the literature which we call by the following keywords: *Depth buffer* [3], *voxel* [3], *complex* [4], *rays with spherical harmonics* [4][5], *silhouette* [3], *3DDFT* [6], *shading* [4], *ray-based* [7], *rotational invariant* [8], *harmonics 3D* [9], *shape distribution* with d_2 [10], *cords* [11], *moments* [11], *shape spectrum* [12], and *volume* [3]. These FVs can describe the global shape of any 3D object. However, as we learned from our experiments, the effectiveness of a given FV cannot be assessed for the general case, as it depends on the specific *type* of 3D object that one wants to search. For example, we have observed that the best effectiveness for "car models" is achieved using the depth buffer FV, but the best effectiveness for "sea animal models" is achieved using the silhouette FV. In this work, we attempt to improve the effectiveness of a similarity search system for a general 3D object database, where no restrictions are imposed on the 3D models, as they are allowed to represent any type of object.

We introduce a heuristic for the a priori estimation of individual FV performance. We then define two methods that use this estimator to improve the effectiveness of the retrieval system: The first one makes use of the heuristic to *select* a good FV given a query object from the pool of available FVs. The other one uses it to *combine* FVs with weights based on the estimator value. Figure 1 shows a proof-of-concept taken from our retrieval system of how retrieval effectiveness can benefit from an appropriate combination of FVs: The query object (left column) is a Formula-1 racing car. The first row shows the objects retrieved using the *depth buffer*

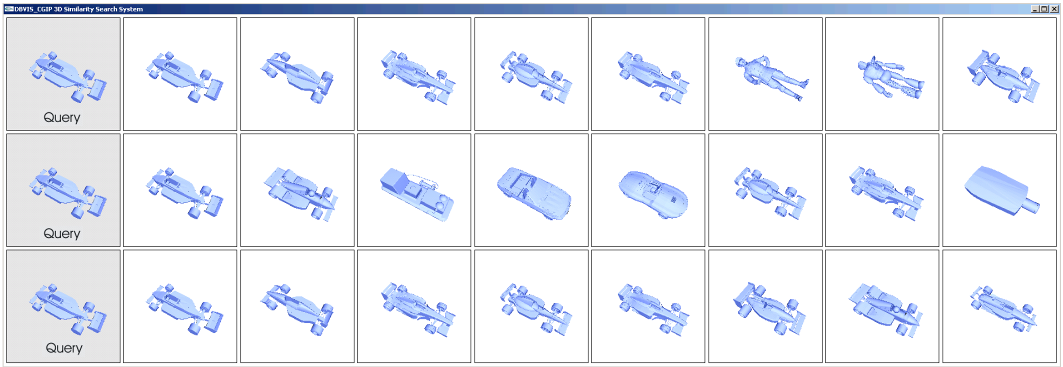

Fig. 1. Example of three similarity queries (with the same query object) using depth buffer, silhouette, and a combination of both FVs.

FV, in ascending distance from the query point. The second row shows the objects retrieved using the *silhouette* FV. The third row shows the objects retrieved using an unweighted combination of both FVs. With this combination, only relevant objects are retrieved on the first eight ranks, while the answer sets using a single FV also include some non-relevant objects.

III. Proposed methods using entropy impurity

We use an estimator based on the *entropy impurity* [13] for determining the best FVs to use. The entropy impurity is a well known measure used in the context of decision tree induction, where it measures the "impurity" of a node N of the tree w.r.t. the elements assigned to N. If these elements all have the same class label, then the impurity is 0, otherwise it is a positive value that increases up to a maximum when all classes are equally represented. Other impurity measures are the *Gini* and the *misclassification impurity* [13], but we experimentally obtained the best results using entropy impurity.

A. First method: Query-dependent selection of best FV

One way to improve the effectiveness of the 3D retrieval system is to try to select the best suited FV for a query object q. In a general 3D object retrieval scenario, we have observed that for different query objects different FVs have the best effectiveness. Hence, given a set of FVs, we would like to select the best one for performing the similarity search for q. Our hypothesis is that a good FV is expected to have a certain level of *coherence* in the answer set, i.e., we expect to retrieve similar objects at the first positions of the ranking list.

Let U be the universe of valid 3D objects. Let $T \subset U$ be a finite set of *training objects*, where $\omega_j \subseteq T,\ 1 \leq j \leq N$, is a *class* of objects (i.e., all objects in ω_j are considered similar), and $T = \biguplus \omega_j$. Let $q \in U$ be a query object. Given a FV f, a *ranking* R_f^q is a list of objects from T sorted in ascending order by the distances between q and every object in T with respect to f. Also, $P_k(\omega_j, R_f^q)$ denotes the fraction of objects at the first k positions of R_f^q that are in class ω_j.

Definition 1: The k-entropy impurity of a FV f with respect to q is defined as

$$i(f,q,k) = -\sum_{j=1}^{N} \begin{cases} P_k(\omega_j, R_f^q)\log_2(P_k(\omega_j, R_f^q)) & \text{if } P_k() > 0 \\ 0 & \text{otherwise} \end{cases}$$

If the k objects are in the same class, the impurity is 0; otherwise it is positive, with the greatest value occurring when the different classes are equally likely and the number of classes covered by the k objects is maximal. We use our previously defined k-entropy impurity to measure the degree of coherence of each FV.

Definition 2: Let $F = \{f_1, \ldots, f_M\}$ be a set of M FVs. The k-entropy impurity selection function is defined as

$$EntImpSelection(F,q,k) = \arg \min_{1 \leq \ell \leq M} \{i(f_\ell, q, k)\}.$$

The FV that minimizes the k-entropy impurity for q is selected. In case of ties, the best FV according to a precomputed ranking of FVs (cf. Section IV) is selected.

B. Second method: Query-dependent combination of FVs

Another way to improve the effectiveness is using a *combination of FVs*. The problem is to determine which FVs to combine, as inclusion of FVs irrelevant to q can harm the overall effectiveness of the search system. We propose to use the k-entropy impurity to weigh each FV in the combination, giving more weight to those FVs with lower entropy impurity. Let d_ℓ be the distance function using FV ℓ, and let $dmax_\ell$ be the maximum distance between q and any object of the database using FV f_ℓ.

Definition 3: The k-entropy impurity weighted distance between a query object q and an object $o \in U$ is defined as

$$\delta_k(q,o) = \sum_{\ell=1}^{M} \frac{1}{1 + i(f_\ell, q, k)} \frac{d_\ell(q,o)}{dmax_\ell},$$

We use $\delta_k(q,o)$ to produce the combined ranking list.

IV. Experimental results

Our database consists of 1,838 3D objects collected from the Internet. From this set, 292 objects were classified into 17 different model classes (e.g., cars, planes), and the rest of them were left as unclassified. The classified objects were used as queries, and those objects which belong to the same model class as a given query q were considered the objects "relevant" to q. We used the l_1 norm to perform the similarity queries.

From our set of 15 implemented FVs, we experimentally determined the best average dimensionality of each FV (we

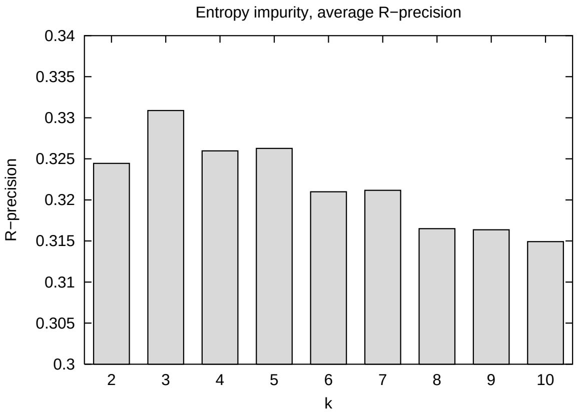

Fig. 2. Entropy impurity, average R-precision varying parameter k.

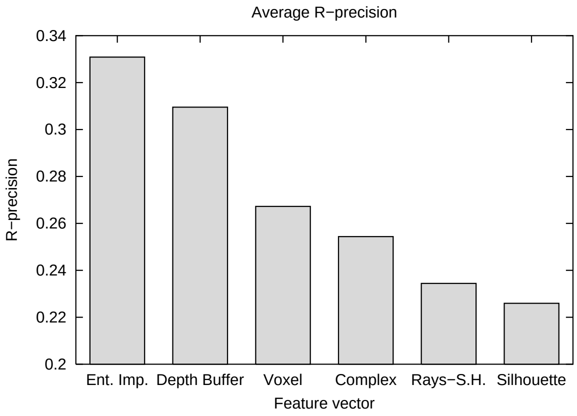

Fig. 3. Average R-precision, all feature vectors.

tested from 6 up to 512 dimensions), and we then compared the effectiveness scores between them. We then selected the best five FVs (with their best dimensionality) to focus our study with entropy impurity: Depth buffer (366-d), voxel (343-d), complex (196-d), rays with spherical harmonics (105-d), and silhouette (375-d). This ordered list of FVs also serves to resolve ties that may occur when using our selection criterion.

To evaluate our selection technique, we partitioned the set of classified objects in order to perform cross-validation [14]. We randomly partitioned the classified set of objects in two halves. One half was used as the training set T. The other half was used as the query set Q. For computing the effectiveness scores, the objects of T were not considered to be part of the database. We repeated this procedure s times, and we averaged the results to obtain final scores. We experimentally found that $s = 100$ gives us stable results.

We use *precision vs. recall figures*, a standard evaluation technique for retrieval systems [15], for comparing the effectiveness of our algorithms. *Precision* (P) is the fraction of the retrieved objects which are relevant to a given query, and *recall* (R) is the fraction of the relevant objects which have been retrieved from the database. If R is the set of relevant objects to the query, A is the set of objects retrieved, and R_A is the set of relevant objects in the result set, then $P = |R_A|/|A|$ and $R = |R_A|/|R|$. We also use the *R-precision* measure [15], which is defined as the precision when retrieving exactly the number of objects relevant to the query. The R-precision gives a single number to rate the effectiveness of a retrieval system.

Figure 2 shows the average R-precision with the entropy impurity selection test (method *A*) while varying parameter k from 2 to 10. The best effectiveness score is achieved with $k = 3$, but the scores with $2 \leq k \leq 5$ are very similar. This result suggests that it is not necessary to search the optimum k for each similarity query, and that any k from 2 to 5 is equally good. For $k > 7$, the effectiveness starts to decrease.

Figure 3 compares the average R-precision of all individual FVs and the 3-entropy impurity selection technique. The improvement in effectiveness between the best single FV (depth buffer) and the selection technique is about 7%, which is significant in terms of quality of the retrieved answer. It is comparable with the effectiveness improvement between two consecutive FVs in the list.

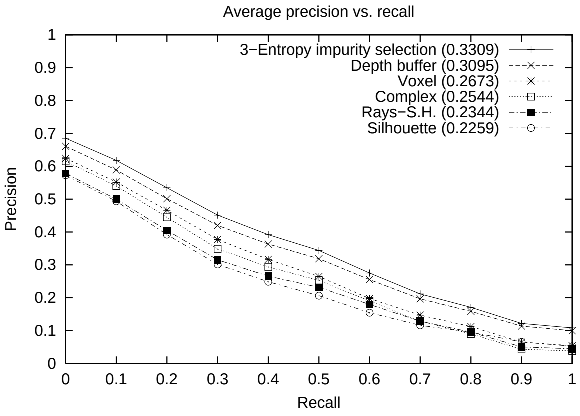

Fig. 4. Average precision vs. recall, all feature vectors.

Figure 4 shows the precision vs. recall figures for all individual FVs and the selection technique. The average R-precision values are also indicated for each curve. The 3-entropy impurity selection has better precision for all recall levels compared with the best single FV, which means that our method is more effective than any of the studied FVs.

Now, we present experimental results of the proposed combination technique with k-entropy impurity (method *B*). Figure 5 shows precision vs. recall curves for the best single FV and the combination method using 3-entropy impurity. One can observe a large effectiveness improvement of 29% in terms of R-precision using the combination technique, which is greater than any improvement between the single FVs used in these experiments. We obtained similar experimental results with $2 \leq k \leq 10$, which also suggests that it is not necessary to search for an optimum k value for each query.

Figure 6 presents a summary of the average R-precision values obtained with the proposed techniques, and compares them with the optimal selection score, i.e., for each query object the FV with the best performance w.r.t. the given query was used. It shows that the effectiveness obtained by our combination method is pretty close to the optimal selection. We also tested the combination method using all of the

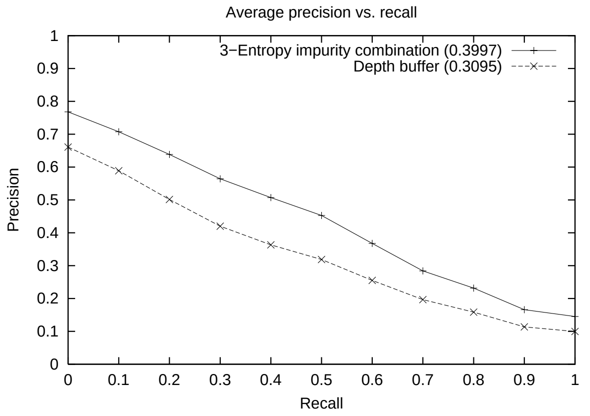

Fig. 5. Average precision vs. recall, combination with entropy impurity.

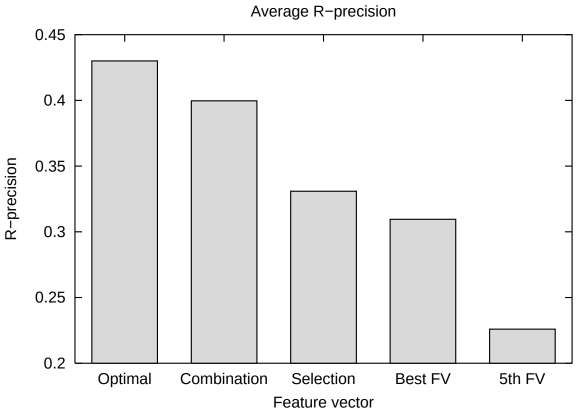

Fig. 6. Comparison of the effectiveness obtained with the proposed methods based on entropy impurity.

15 implemented FVs. We obtained a slightly better result (0.4073% in R-precision) than with the combination using 5 FVs. However, this improvement is obtained at the expense of higher CPU cost, because in that case we have to compute 15 rankings instead of 5.

V. Conclusions and Future Work

In this paper, we have proposed two techniques based on the entropy impurity concept in order to improve the effectiveness of a 3D object similarity search system. The first technique proposes to use the k-entropy impurity to estimate the most appropriate FV to perform the similarity search, given a query object. The second technique proposes to use a combination of FVs for performing the similarity search, weighting each FV with a value inversely proportional to the k-entropy impurity.

Our experimental results show that both techniques allow us to significantly improve the effectiveness of the search system, especially with the combination technique. We experimentally found that it is possible to improve the effectiveness by almost 30%, in terms of R-precision, using the combination technique with a small set of good FVs (5 in our experiments). This method allows us to dynamically and automatically set the weights for each FV depending on the query object. Our results show that retrieval systems may profit from automatic feature selection techniques. More FVs will be available in the future to query for 3D objects, but with high probability no single method will dominate for the general case.

We plan to apply our selection technique to multimedia retrieval systems supporting other formats besides 3D objects. An open problem is the automatic selection of a FV or combination of FVs for searching for a given query object in a database consisting entirely of unclassified objects. In this case, the proposed methods cannot be directly applied.

Focusing on the efficiency side, future work will involve finding an index structure that efficiently handles similarity search based on combination techniques. This step involves the design of an index for vector-set-represented objects.

Acknowledgments

This work was partially funded by the Deutsche Forschungsgemeinschaft (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] C. Faloutsos, *Searching Multimedia Databases by Content*. Kluwer Academic Publishers, 1996.

[2] D. Vranić, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics," in *Proc. IEEE Workshop on Multimedia Signal Processing*, 2001, pp. 293–298.

[3] M. Heczko, D. Keim, D. Saupe, and D. Vranić, "Methods for similarity search in 3D databases," *Datenbank-Spektrum*, vol. 2, no. 2, pp. 54–63, 2002, in German.

[4] D. Vranić and D. Saupe, "Description of 3D shape using a complex function on the sphere," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2002, pp. 177–180.

[5] ——, "3D model retrieval with spherical harmonics and moments," in *Proc. 23rd DAGM-Symposium*, ser. LNCS 2191. Springer, 2001, pp. 392–397.

[6] ——, "3D shape descriptor based on 3D Fourier transform," in *Proc. EURASIP Conf. on Digital Signal Processing for Multimedia Communications and Services*. Comenius University, 2001, pp. 271–274.

[7] ——, "3D model retrieval," in *Proc. 16th Spring Conf. on Computer Graphics (SCCG'00)*. Comenius University, 2000, pp. 89–93.

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

[9] 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*, vol. 22, no. 1, pp. 83–105, 2003.

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Trans. on Graphics*, vol. 21, no. 4, pp. 807–832, 2002.

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

[12] T. Zaharia and F. Prêteux, "3D shape-based retrieval within the MPEG-7 framework," in *Proc. SPIE Conf. on Nonlinear Image Processing and Pattern Analysis XII*, vol. 4304, 2001, pp. 133–145.

[13] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley-Interscience, 2001.

[14] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Morgan Kauffman, 2001.

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