ICIP, Rochester, NY, Sept 2002

Feature Space Warping: An Approach to Relevance Feedback

Hoon Yul Bang and Tsuhan Chen

Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
{hbang, tsuhan}@andrew.cmu.edu

ABSTRACT*

Relevance feedback has been shown to be an effective tool to enhance content-based information retrieval (CBIR) systems. We propose a new approach to relevance feedback by warping the database's feature space, or shifting the objects' data points in a controlled manner responding to user feedback. We demonstrate that given consistent feedback, the performance of the retrieval system can be significantly increased.

1. INTRODUCTION

While increasing amounts of multimedia are available for consumer perusal, the average human's attention span remains constant. Indeed, human attention becomes more and more valuable, magnifying the need for more intelligent multimedia retrieval systems. Currently many content-based information retrieval systems rely on low-level computable features such as color, shape and texture. Because high-level descriptions are more intuitive to human users, infusing high-level meaning is an area of active research.

A considerable disadvantage to low-level features is their inability to describe high-level similarity, or semantic relationships among objects. Semantically similar objects may have completely different feature values; therefore, searching among the low-level feature space often results in erroneous results.

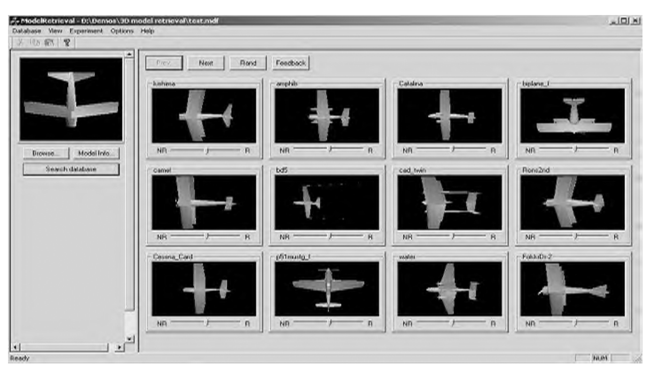
One method to add high-level information to a query is through relevance feedback. Noteworthy approaches to relevance feedback include the PicHunter system, which employs a Bayesian framework, where the system selects the image with the highest probability of relevance, given the relative distance to positive and negative feedback [1]. Other schemes employ algorithms that stretch and shrink dimensions of the feature space to bring relevant objects closer together [2,3,4]. In addition to globally adjusting the feature space, Mindreader attempts to refine the user's query, based on the Rocchio equation [3]. Similar systems such as the MARS CBIR system also have been successful [5]. Still other systems employ archetypal models as representatives for high-level concepts or clusters, such as support vectors, prototypes, and examples [6,7,4]. Another noteworthy approach employs the expectation-maximization algorithm to estimate the high-level similarity and dissimilarity of images given relevance feedback [8].

However, these approaches rely on a feature space that may already inaccurately portray the objects in a high-level perspective. For example; reweighting of relevant features may bring some relevant objects closer, but it still depends on a potentially inaccurate foundation because it must use flawed feature points. In the worst-case scenario, the user may never find the object desired, even though it exists in the database. A noteworthy implementation is the usage of smart user interface to bring users' perception into the retrieval system [9]. While it may be possible to use a small number of dimensions to approximate a "desktop" of human perception, more dimensions might be needed to accurately and permanently describe high-level features.

In this paper, the chosen environment for circumventing inaccurate feature data is described in Section 2. The proposed algorithm, feature space warping, is written in Section 3; Section 4 illustrates performance with closed-database queries, Section 5 explains open-database performance, and Section 6 concludes the paper.

2. RETRIEVAL OF 3D MODELS

The database considered is a collection of 1750 three-dimensional triangular-mesh models in the Virtual Reality Markup Language (VRML) format such as architectural



* Work supported in part by NSF CAREER Award.

structures, people, airplanes, trees, and geometric objects. The database internally represents each model as a set of ten size-invariant features including the aspect ratio, the surface area to volume ratio, moments and Fourier coefficients, as detailed in [10]. When given a query, the retrieval system calculates the query model's location in the feature space, and returns models in the database with small Euclidean distance to the query. Figure 1 shows an example database:

$$U = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \ldots\}$$

Given a query vector $\mathbf{q}$, the system returns $\mathbf{p}_1 \cdots \mathbf{p}_4$. The vector $\mathbf{p}_5$ is not returned.

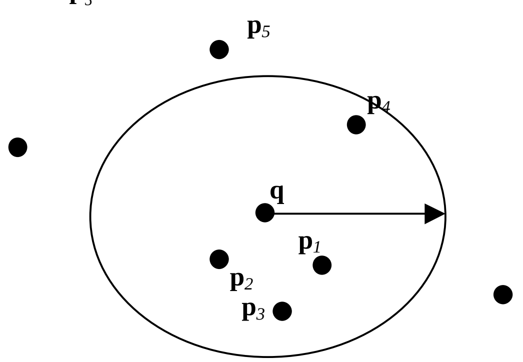


Figure 1. Simple diagram of internal database representation.

However, the inability of the selected features to accurately describe distinguishable characteristics results in poor retrieval performance. For instance, a jet airplane and a book may share similar aspect ratio values, and the same jet airplane may have the same similar surface area to volume ratio as a thin rectangular prism. On the other hand, all airplanes may have very different aspect ratios and surface areas. As a result, a simple nearest neighbor retrieval algorithm is frequently inconclusive. However, a system that is able to incorporate high-level semantic information can make generalizations that cannot be produced from a low-level feature space.

Relevance feedback gives the ability for the user to help the system by providing relevance data with respect to the query. In receiving feedback, the system can refine its knowledge of the database to perform better on future queries.

3. FEATURE SPACE WARPING

Once the user indicates which retrieved models are relevant and which models are irrelevant, the proposed approach to implementing relevance feedback involves shifting relevant models closer to the query and irrelevant models farther away. Furthermore, other models may also receive such feedback indirectly by moving according to their distances from the relevant/irrelevant models as shown in Figure 2, which illustrates the movement of points in the feature space in response to query $\mathbf{q}$, positive feedback $\mathbf{f}_p$, and negative feedback $\mathbf{f}_n$.

Models $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ also move closer to the query in decreasing strength with respect to their distance to $\mathbf{f}_p$. Similarly, models $\mathbf{p}_4$ and $\mathbf{p}_5$ move farther away from the query in decreasing strength with respect to their distance to $\mathbf{f}_n$.

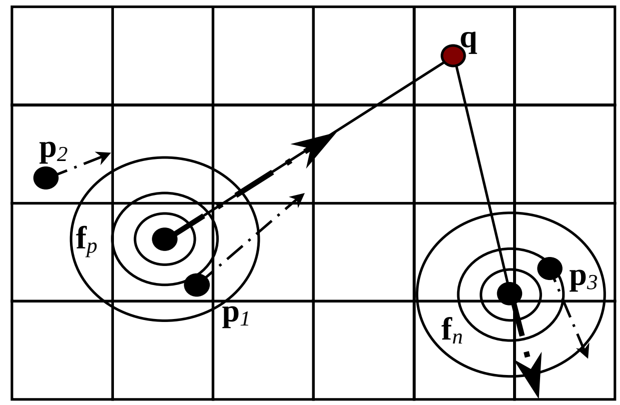


Figure 2. Feature Space Warping.

Specifically, for all models $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_j, \ldots, \mathbf{f}_M$ that are given feedback, each model $\mathbf{p}_i$ that is not given any feedback can be moved toward or away from the query model by an amount as described in Figure 3. The vector from the model $\mathbf{p}_i$ to the query $\mathbf{q}$, $\mathbf{v}_{iq}$, and the vector from $\mathbf{p}_i$ to the model given feedback, $\mathbf{v}_{ij}$, are needed to calculate $\mathbf{v}_{pi}$. A scalar quantity u_i is given from the user as the strength of positive or negative feedback. In the current implementation, u_i may take values within [-2, 2]. The movement for $\mathbf{p}_i$, $\mathbf{v}_{pi}$, is determined as follows:

$$v_{pi} = \left[\gamma \sum_{j=1}^{M} u_i \exp\left(-c\left|v_{ij}\right|\right) \right] v_{iq}$$

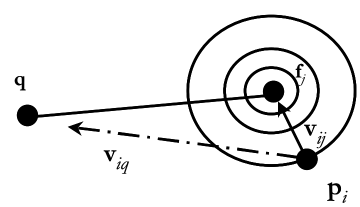


Figure 3. Warping mechanics.

Note that for one feedback session, each model $\mathbf{p}_i$ in the database receives a movement contribution from each model $\mathbf{f}_j$ labeled as relevant or irrelevant. The closer the model $\mathbf{p}_i$ to a relevant/irrelevant model, the more contribution it receives from that model. Furthermore, the coefficients γ and c are chosen globally such that the final coefficient multiplying $\mathbf{v}_{iq}$ is less than one.

It is interesting to note that once the database is given feedback, the models' representation in the feature space has been altered due to the system's incorporation of higher-level information. A consequence of this process is that the features themselves have supplemented meaning. The coordinates in the feature space no longer represent the low-level values of the surface-to-volume ratio, Fourier coefficients, etc. Because of this consequence our method would work effectively in retrieval systems with feature spaces that do not already make high-level, or semantic generalizations.

4. TEST RESULTS

In order to gauge performance of the relevance feedback implementation, we compile ground truth data containing classification information for each 3D model in the database. Then, we use each object in the database as a query, and from each query we count the number of relevant objects in the top twelve results. The sum over all the queries from the database is the performance score. After the initial score is computed, the database undergoes an iteration of feedback from the user. For each queried object in the database, the top T retrieved results are given feedback. In Figure 4, the performance score is then reevaluated. A total of four iterations are completed. The performance during four iterations of feedback is averaged. Performance increases with c, until a threshold of 6π is reached. Figure 5 shows the average improvement versus γ. For the VRML database, $\gamma = 0.3$ and $c = 6\pi$ was found to be optimal. In Figure 6, we show the database performance after each iteration of feedback, for tuned parameters $\gamma = 0.3$ and $c = 6\pi$. These parameters were found to give the best average performance. Slightly better performance was found for $T = 12$, $\gamma = 0.2$, $c = 5\pi$, $\#\,\text{iterations} = 3\,\&\,4$, suggesting that with enough consistent feedback, (1) the influence of warps on the feature space can be increased, while (2) the magnitude of the warp can be decreased, giving even more improvement with consistent feedback. An increased number of retrieved objects given feedback per query, T, also improves performance. For comparison, the standard Rocchio query refinement algorithm was implemented according to the equation:

$$\mathbf{z}' = \alpha\mathbf{z} + \beta\boldsymbol{\mu}_r + \gamma\boldsymbol{\mu}_i$$

where $\mathbf{z}$ is the old query vector, $\mathbf{z}'$ is the new query vector, $\boldsymbol{\mu}_r$ is the mean of the relevant objects, $\boldsymbol{\mu}_i$ is the mean of the irrelevant objects, and the parameters α, β, γ are tuned for optimum performance. For this database, Rocchio's performance was improved to 8634 relevant models from 7869 relevant models, for an increase of 9.7%. In Figure 6, average database performance was improved from a performance score of 7869 to 15046 relevant models, for an increase of 91%. Testing four feedback iterations on the whole database takes 3-6 minutes on an Athlon 1400 MHz running Windows 98.

Figure 4. Precision Improvement Due to 4 Feedback Iterations; gamma = .3

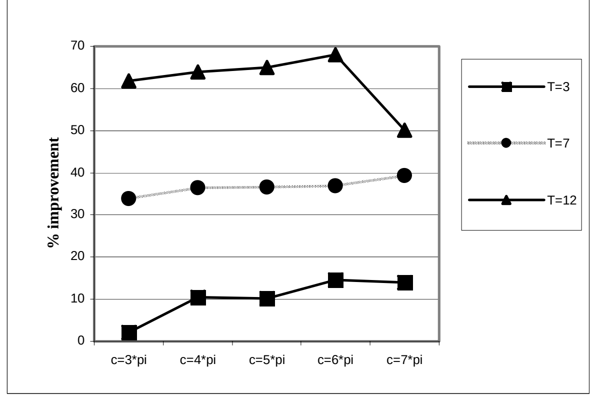


Figure 5. Precision Improvement Due to Feedback (T averaged out), Varying Gamma

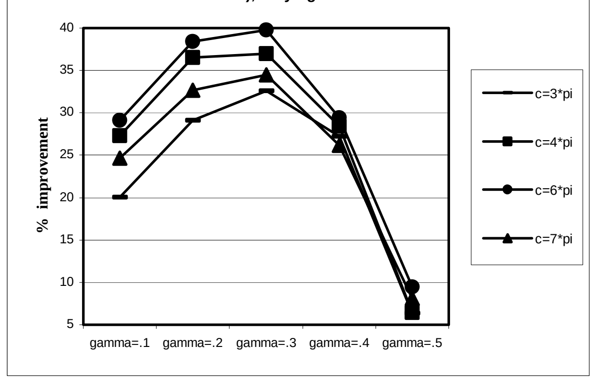
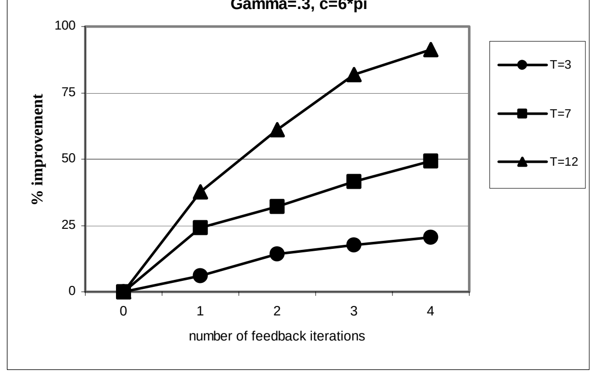


Figure 6. Precision Increase Due to Feedback with Gamma=.3, c=6*pi

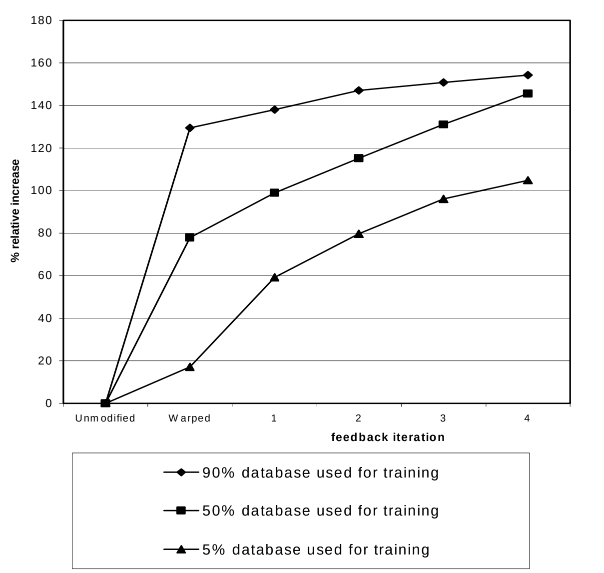


5. OPEN DATABASE RESULTS

Previously we have discussed feature space warping in reference to a database where queries are from within the database. Now we simulate the effect of new models being introduced to the database by withholding a portion of the 1750 models as queries.

To bring a new model **A** from the original feature space into the database's warped feature space, each feedback, or each warp, must be recorded. The feature vector of **A** undergoes the transformations described in Figure 3 in the correct order. In this sense, feature space warping helps classify never-seen-before models by moving them according to models already given feedback. Although such recording seems computationally arduous, bringing ~1000 vectors into the warped feature space after ~10^5 warps ran for less than one minute on the same computation environment as described earlier. Figure 7 shows the performance of feature warping as a classifier given training from 90%, 50%, and 5% of the whole database.

Figure 7. Precision After Feedback Iteration



6. CONCLUSION

We have proposed a method for implementing relevance feedback via feature space warping. Given consistent feedback through ground truth data, this method is shown to increase the number of relevant objects retrieved by the system. Performance is shown to rely heavily on the amount of feedback and the parameters γ and c, which control the magnitude and influence of warping, respectively. Face space warping is shown to outperform other methods such as the Rocchio's refinement.

7. REFERENCES

[1] Ingemar J. Cox, et. al. "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments", *IEEE Trans. On Image Processing*, pp. 20-37, vol. 9, no. 1, Jan 2000.

[2] Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval", *IEEE Trans. On Circuits and Systems for Video Technology*, pp. 644-655, vol. 8, no. 5, Sep. 1998.

[3] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Query Database through Multiple Examples", Proceeding of the 24th VLDB Conference, New York, 1998.

[4] Selim Aksoy, Robert M. Haralick, Faouzi A. Cheikh, Moncef Gabbouj. "A Weighted Distance Approach to Relevance Feedback". *Pattern Recognition, 2000. Proceedings. 15th* pp. 812-815.

[5] K. Porkaew and S. Mehrotra. "Query Refinement for Content Based Multimedia Retrieval in MARS". *Multimedia Computing and Systems, 1999.* pp. 747-751.

[6] Qi Tian, Pengyu Hong, Thomas S. Huang, "Update Relevant Image Weights for Content-based Image Retrieval Using Support Vector Machines", *Multimedia and Expo, 2000. ICME 2000.* pp. 1199-1202, Vol. 2, 2000.

[7] Pengyu Hong, Qi Tian, Thomas S. Huang. "Incorporate Support Vector Machines to Content-based Image Retrieval with Relevant Feedback". *Image Processing, 2000. Proceedings.* pp. 750-753.

[8] J. Yoon and N. Jayant. "Relevance Feedback for Semantics Based Information Retrieval". *IEEE ICIP Proceedings*, 2001. pp. 42-45.

[9] S. Santini, A. Gupta, R. Jain. "Emergent Semantics through Interaction in Image Databases". *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, issue 3, May-June 2001, pp. 337-351.

[10] C. Zhang and T. Chen, "Efficient Feature Extraction for 2D/3D Objects in Mesh Representation", *ICIP 2001*, Greece, October 2001.