Multi-Modal Browsing of Images in Web Documents

Francine Chen[a] Ullas Gargi[b] Les Niles[a] Hinrich Schütze[a]

[a]Xerox Palo Alto Research Center, 3333 Coyote Hill Rd, Palo Alto, CA USA

[b]Dept. of Computer Science and Eng., Penn State University, University Park, PA USA

ABSTRACT

In this paper, we describe a system for performing browsing and retrieval on a collection of web images and associated text on an HTML page. Browsing is combined with retrieval to help a user locate interesting portions of the corpus, without the need to formulate a query well matched to the corpus. Multi-modal information, in the form of text surrounding an image and some simple image features, is used in this process. Using the system, a user progressively narrows a collection to a small number of elements of interest, similar to the Scatter/Gather system[1] developed for text browsing. We have extended the Scatter/Gather method to use multi-modal features. With the use of multiple features, some collection elements may have unknown or undefined values for some features; we present a method for incorporating these elements into the result set. This method also provides a way to handle the case when a search is narrowed to a part of the space near a boundary between two clusters. A number of examples illustrating our system are provided.

Keywords: multi-modal information access, image/document browsing and retrieval, clustering, web documents

1. INTRODUCTION

Much of the research in information retrieval has focused on retrieving text documents based on their textual content or on retrieving image documents based on their visual features. Recently, there has been some research on the use multi-modal features for retrieval.[2,11,12] We are investigating an approach to document browsing and retrieval in which a user iteratively narrows their search using both the image and text associated with the image, and possibly other types of information related to the document, such as usage. We refer to disparate types of information such as text, image features and usage as modalities.

In this paper, we present a method of information access to a collection of web images and associated text on an HTML page. Our method permits the use of multi-modal information, such as text and image features, for performing browsing and retrieval of images and their associated documents or document regions. In our approach, we use text features derived from the text surrounding or associated with an image, which often provides an indication of its content, together with image features. The novelty of our approach lies in the way it makes text and image features transparent to users, enabling them to successively narrow down their search to the images of interest. This is particularly useful when a user has difficulty in formulating a query well matched to the corpus, especially when working with an unfamiliar or heterogeneous corpus, such as the web, where the vocabulary used in the corpus or the image descriptors are unknown.

Our work can be thought of as an extension to image browsing. An ideal image browsing system would allow a user to browse images that may or may not have descriptive annotative text and use both text or image features. Users may wish to browse through image collections based either on their semantic content ("what does the image show?") or their visual content ("what does the image look like?"). Image retrieval systems are often based on manual keyword annotation or on matching of image features, since automatically annotating images with semantic information is currently an impossible task. Even so, a manually labeled image collection cannot include all the possible semantic significances that an image might have.

Current image retrieval systems commonly display a random selection of images (e.g., QBIC,[3] Virage[14]) or allow an initial text query as a starting point (e.g, QBIC,[3] Smith and Chang[11]). In the latter case, a set of images with that associated text is returned. The user selects the image most similar to what they are looking for, a search using the selected image as the query is performed and the most similar images are displayed. This process is repeated as the user finds images closer to what is desired. In some systems, the user can directly specify image features such as color distribution and can also specify weights on different features, such as color histograms, texture, and shape.[3] In web pages, text such as URLs may also provide clues to the content of the image. Current image retrieval technology

also allows the use of URL, alt tags, and hyperlink text to index images on the web (e.g., Dunlop[2]; Smith[11]). One approach also attempts to determine for each word surrounding an image caption whether it is likely to be a caption word[8] and then match caption words to "visual foci" or regions of images (such as the foreground).[4] The Webseek image search engine[11] and MARS-2[10] allow for relevance feedback on images by marking them as positive or negative exemplars.

Using multi-modal features, our system permits quick initial focusing of the set of elements of interest, and then organization and expansion to include similar elements, some of which may have incomplete feature sets or occur in another cluster. One difficulty in the use of multiple features in search and browsing is combining information from the different features. This is commonly handled in image retrieval tasks by having weights associated with each feature that can be set by the user. In contrast to current image search systems, in our method of browsing and retrieval, a user employs different multi-modal features to progressively narrow a collection to a small subset of images of interest, with associated text, *without* weighting the different features. Each feature is used one at a time to either refine or enlarge the set of images. The image features are used independently of text features to create multiple clusterings in the different modalities that the user can navigate, using text (e.g., section headings, body text, abstract, title, "alt" tags in image anchors) and image features to refine the images in the collection.

Although the use of clustering in image retrieval is not new, it has usually been used for pre–processing, either to aid a human during the database population stage,[7] or to cluster the images offline so that distance searches during queries are performed within clusters.[5] In our work, we use iterative clustering and selection of cluster subsets to help a user identify images of interest. Clustering is used for interactive searching and presentation, and relevance feedback is implicit in the user's choice of clusters. Because the user is dealing with clusters, not individual images, the feedback step is also easier to perform. Our work is most similar to Scatter/Gather which was developed by Cutting et al.[1] for text browsing. Scatter/Gather iteratively refines a search by "scattering" a collection into a small number of clusters, and then a user "gathers" clusters of interest for "scattering" again. We have extended the Scatter/Gather paradigm to multiple modalities and have added an "expand" function so that elements from outside the working set can be incorporated into the working set.

In practice, an initial text query is used to find candidate images of interest. Some of the returned clusters containing images of interest are then identified by the user for further consideration. By expanding based on similarity of one image feature, the system then finds and presents image clusters that are similar to those represented by the initially selected clusters, but without associated text or with text not similar enough to the user-specified query. Thus the expand function permits relevant images that are absent in the original set as a result of the text query to be identified and included. The expand function can also identify for consideration elements that are near the feature space of interest, but that are — due to the partitioning at an earlier step — in another cluster.

2. CLUSTERING AND GATHERING SUBCOLLECTIONS

A preprocessing step is used to precompute information needed during browsing and to provide the initial organization of the data. A set of features, possibly from different modalities, is precomputed for each document image and stored as vectors. The text features include the words of text surrounding and associated with each image, the URL of the image, alt tags, and hyperlink text. The image features include a color histogram and a measure of color complexity. The documents are initially clustered into groups based on the text features.

To search for images, a user begins by entering a text query. A hypothetical session is illustrated in Fig. 1 where: a node represents the data in a cluster; the solid arrows represent the scattering or gathering of data in a node; and the dashed lines represent movement of a subset of data in a node to another node, as in the expand function. The precomputed text clusters are ranked by similarity to the query terms using the cosine distance and the highest ranking clusters are displayed by representative text (see Fig. 1a). The user then selects the clusters that are most similar to their interest. This may include all or a subset of clusters (see Fig. 1b). One of two operations is then performed: 1) The elements in the selected clusters are reclustered based on a selected feature (see Fig. 1c) or 2) The selected clusters are expanded to new similar (dashed lines in Fig. 1d) clusters based on a selected feature. The new clusters are displayed as representative text or images, depending on whether the selected feature is derived from text or image data. The selected feature may be any of the precomputed features. By reclustering, the user can refine the set of images. By expanding, images similar in the specified feature, possibly with missing values in other features, can be brought into the set of images for consideration.

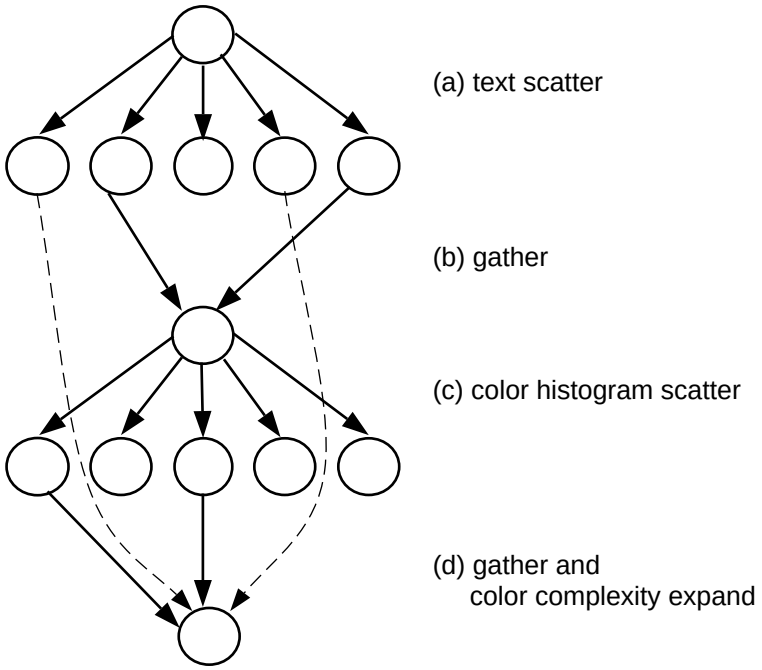

Figure 1. Hypothetical session of scattering and gathering collection elements in different modalities.

Clustering is performed using a standard k-means clustering algorithm with a preset number of clusters. In the preprocessing step, the number of clusters is larger than the number of clusters presented to the user. This is because only a subset of clusters will be presented in response to the initial text string query. In our case with an initial text query, we cluster to 20 clusters and return the 5 most similar clusters. The clusters selected by the user for gathering are then reclustered, where the number of clusters is equal to the number of clusters to be displayed, again 5 in our case. Each subsequent gather and scatter operation results in 5 clusters. As each operation is performed, cluster results are stored. This permits backing up and is also needed by the expand operation.

3. EXPANDING SUBCOLLECTIONS

The expand function addresses a problem with progressively narrowing a search based on different features: images with missing values will be eliminated from consideration. For example, some documents contain images with no associated text, or text unrelated to the contents of the image. In other cases, the text surrounding the image has no relevance to the semantic content of the image. Another problem with progressively narrowing a search is that the search may be narrowed to a part of the space near a boundary between two clusters.

The expand operation adds images or clusters to the current set, based on similarity in one feature dimension. Because only one feature is examined at a time, the distance metric can be different for each feature. For example, we use the cosine distance for text similarity and the normalized histogram intersection as an indication of histogram similarity. The expand operation is performed in one of two ways. The first method insures that the elements of the current clusters remain and the set size is increased by adding to the current working set some elements that are close to the working set based on the selected feature. The mean of the selected feature for the current working set is computed, and then those elements, represented as vectors, in the entire database that are closest to this mean are added. This is most appropriate for text features. A second method is to add elements that are close to each displayed representative in the working set. This is more applicable to the image features, in which the clusters are represented by selected images instead of a compilation of the elements used to represent text. However, if the text is represented by selected documents, this method of expansion would also be appropriate.

Referring to the example in Fig. 1, expansion is performed by identifying the most similar clusters based on the color complexity feature. In this way, images with no relevant text are identified if they are similar, in this case based on the complexity feature, to images with relevant associated text. For example, the terms in some URLs are not informative (e.g. the terms in the URL http://parcweb/project/anyproject/pics/fig1.tif are: parcweb,

project, anyproject, pics, fig1 and tif). By first identifying images that are associated with terms of interest and then expanding to images similar in another feature, such as the color complexity feature, a larger number of images can be identified without starting the search over or requiring the use of weights.

4. FEATURES AND DISTANCE METRICS

The system currently uses three simple features. Two of the features are image-based; the third is text-based. We chose these features because we wanted simple, understandable features that would illustrate our method for combining image and text modalities in information access. Because each feature is used separately, the most suitable distance metric can be applied to each feature. In the future, we would like to enlarge the set of features to include features that other researchers have found most useful, such as the use of local color histograms for different image regions, segmentation and texture features.

4.1. Text Feature

The text feature is a *term vector*, where the elements of the vector represent terms used to represent "documents" and the terms are derived from text surrounding an image, image URL, and page URL. Currently, we limit the scope of the surrounding text to 800 characters preceeding or following the image location. If a horizontal rule, heading or another image occurs prior to the limit being reached, the scope ends at the rule, heading or image. A stop-list is used to prevent indexing of common terms with little content, such as prepositions and conjunctions. Currently, the terms from the different sources are combined into one term vector. An alternative would have been to separate the terms from the different text sources. A single term vector was used in part because the amount of text associated with an image is fairly small, in comparison to the amount of text in normal documents. The vector similarity is computed using the cosine distance:

$$d(t_1, t_2) = \frac{\sum_i t_1(i) t_2(i)}{\sqrt{\sum_i t_1^2(i) \sum_j t_2^2(j)}},$$

where t_1 and t_2 represent the term vectors from the two documents for which the similarity is to be computed.

4.2. Color Histogram Feature

A single color histogram is used as the color feature. The feature space is converted to HSV, and two bits are assigned to each dimension. The histogram is normalized so the bin values sum to one for each image. The distance between histograms is computed similarly to the *intersection measure* by Swain and Ballard,[13] but with normalization by the largest bin value:

$$1.0 - \frac{\sum_i \min(h_1(i), h_2(i))}{\sum_i \max(h_1(i), h_2(i))},$$

where h_1 and h_2 represent the normalized color histograms for the two images. Thus the distance is symmetric with respect to the two images. A symmetric distance is needed in our framework because we are computing distances between an image and another image or centroid for clustering purposes, rather than retrieval purposes.

4.3. Complexity Feature

The complexity feature attempts to capture a coarse semantic distinction that humans might make between images: that between simple logos and cartoons at the one extreme, which are composed of a relatively small number of colors with regions of high color homogeneity, and outdoor photographs on the other, which are composed of a relatively large number of colors with fine shading. The feature is derived from the run-length of the colors. In particular, run-lengths of the "same" color are identified in the x and y directions. A histogram is computed for each direction, where the bins represent the percentage of the image width or image height a run spans in the x or y direction, respectively. The count in each bin is the number of pixels in the image belonging to that particular run-length. Another way to interpret this is that the value added to a bin for each run is weighted by the length of the run, giving greater weight to longer runs. The total number of elements in a histogram is the number of pixels in the image. With the distance metric used, there is no need to normalize the sum of the bins.

The distance metric between two vectors, v_1 and v_2, is the average of the similarity between each pair of histograms:

$$d(v_1, v_2) = .5\frac{\sum_i x_1(i)x_2(i)}{\sqrt{\sum_i x_1^2(i)\sum_j x_2^2(j)}} + .5\frac{\sum_i y_1(i)y_2(i)}{\sqrt{\sum_i y_1^2(i)\sum_j y_2^2(j)}},$$

where the similarity is computed using the cosine distance, x_1 and x_2 represent the x-run-length histograms, and y_1 and y_2 represent the y-run-length histograms for the two images.

5. REPRESENTING CLUSTERS

When using a clustering scheme such as Scatter/Gather, it is necessary to display or represent the clusters to the human user during a browsing session. A text cluster can be represented in a number of ways, the most common being the selection of a set of words that are in some way most representative of the cluster, and displaying them. In our work, clusters based on text features are represented by high frequency content words. When image clusters need to be so represented, it is less meaningful to choose image features that are common to the cluster members and display them, since these will not, in general, have semantic meaning to the user. Some systems display a collection of images in a two dimensional space using multi-dimensional scaling (e.g., Rubner *et al.*,[9] Marks *et al.*[6]). To display the clusters more quickly, we select a small number of representatives from each cluster and display only those representatives. The representatives are comprised of: 1) the three images closest to the centroid of the cluster and 2) three images representative of subregions of the cluster. The three subregion representatives are computed by removing the three most central images, computing three subclusters, and using the image closest to the centroid of each subcluster. This representation provides a sense of the cluster centroid and the range of images in the cluster. The representative images could also have been placed on a 2-D display using multi-dimensional scaling, but for the examples in this paper, we display the representatives in a row of three "centroid" images or three "subcluster" images (e.g., see Fig. 4). This permits very similar images, such as thumbnails and multiple copies of originals, to be more readily identified.

6. EXAMPLES

In our current work, we have used a collection of web documents containing 2310 images as our corpus. Web documents contain many of the same types of "meta-information" that can be found in scanned images of documents and can be used to infer the content of a document or the components in a document. By working with web documents, the issues involved with identifying components and layout in an image are minimized, while permitting development of techniques for using meta-data in the retrieval process. In the future, we would like to extend this work to collections of scanned documents.

To prevent the corpus from being dominated by "uninteresting" images such as logos and icons that are so ubiquitous on the Web, we applied some simple, and somewhat arbitrary, criteria that images must satisfy to be included in the corpus. (Note that it was not necessary, nor a goal of this work, to include *all* images of any particular class, only to assemble an interesting corpus from what's available on the Web, so we intentionally set a high reject threshold.) An image was required to have height and width of at least 50 pixels, and to contain at least 10,000 total pixels. An image was also required to pass some color-content-based tests: that no more than 90% of the image be composed of as few as 8 colors, no more than 95% of the image be composed of as few as 16 colors, and that the RGB colorspace covariance matrix of the image's pixels be non-singular. Qualitatively, these criteria ensure that the images are not simple line drawings, and contain enough variety of color content to be well-differentiable by the color features described above. We did not screen for multiple versions of the "same" image, so the corpus does contain identical images, as well as an image and a thumbnail of the image.

We present three sample sessions illustrating the use of "scattering" and "gathering" in different modalities. The first example illustrates the use of the text feature to first narrow the collection and then use of an image feature to organize the results. The user starts by typing in the text query "ancient cathedral". A snapshot of the screen displaying five returned text clusters is shown in the left half of Fig. 2. These clusters are the clusters closest to the query terms. The most frequent content terms in each cluster are displayed to represent each cluster. The user can scroll each text window to view additional representative terms for a text cluster. The user decides to scatter the first text cluster containing the terms "ancient" and "cathedral" again based on text. A snapshot of the screen

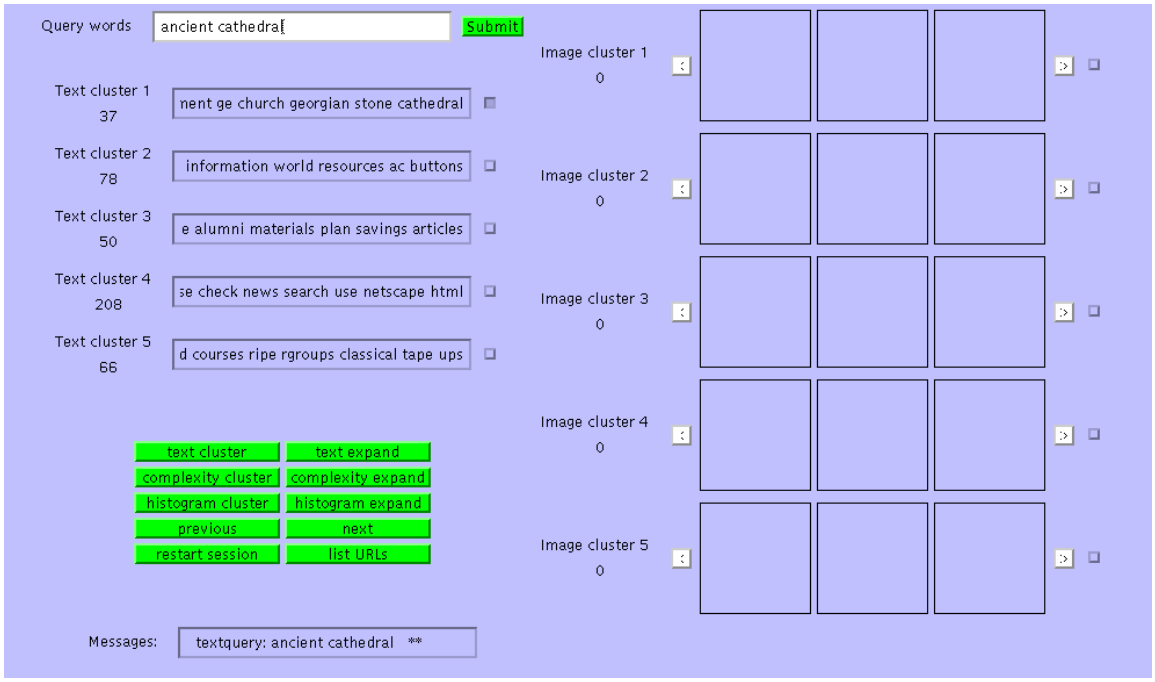

Figure 2. Text clusters returned in response to the query "ancient cathedral".

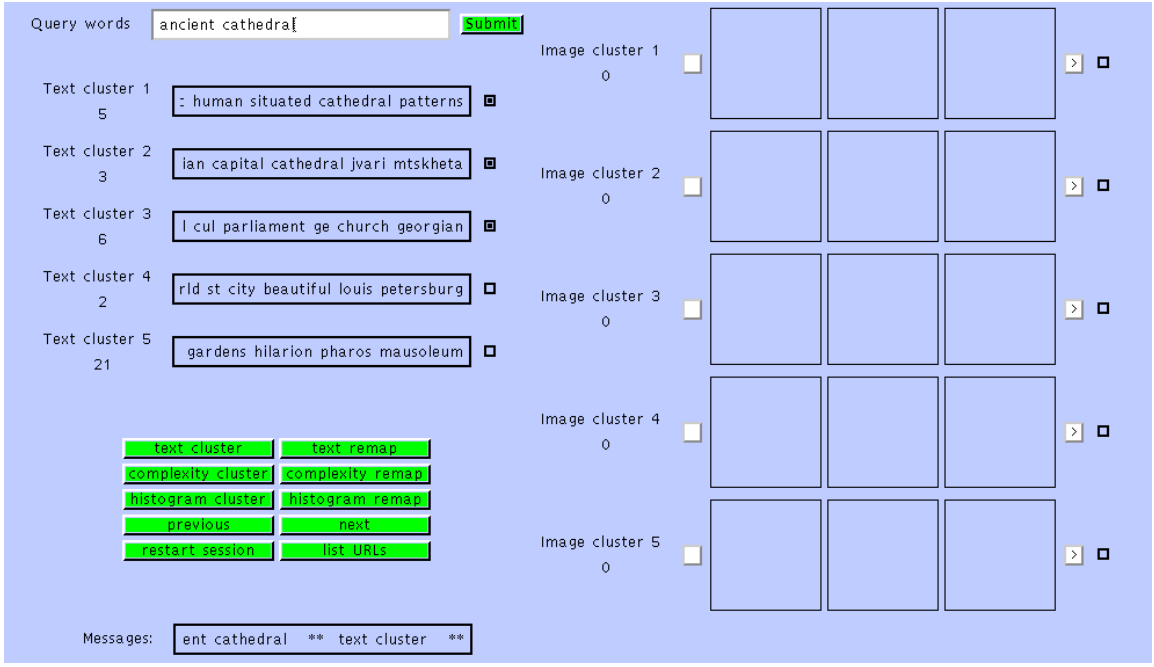

Figure 3. Text clusters returned after scattering Text cluster 1 in Fig. 2

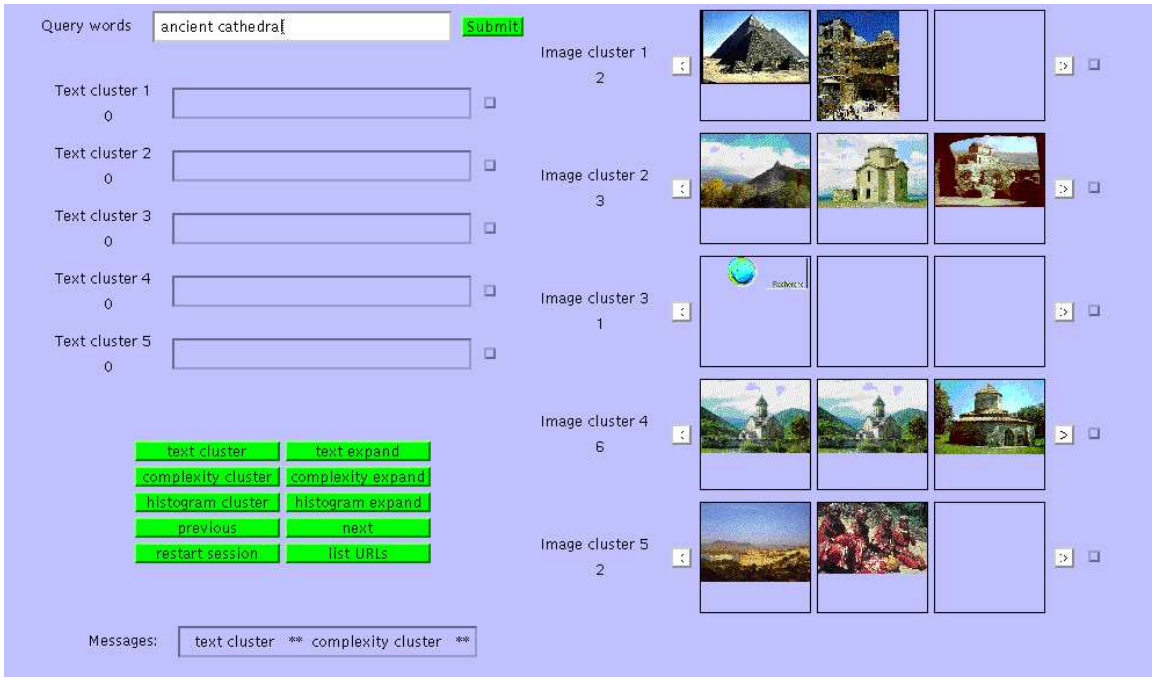

Figure 4. Image clusters returned after clustering based on the complexity feature.

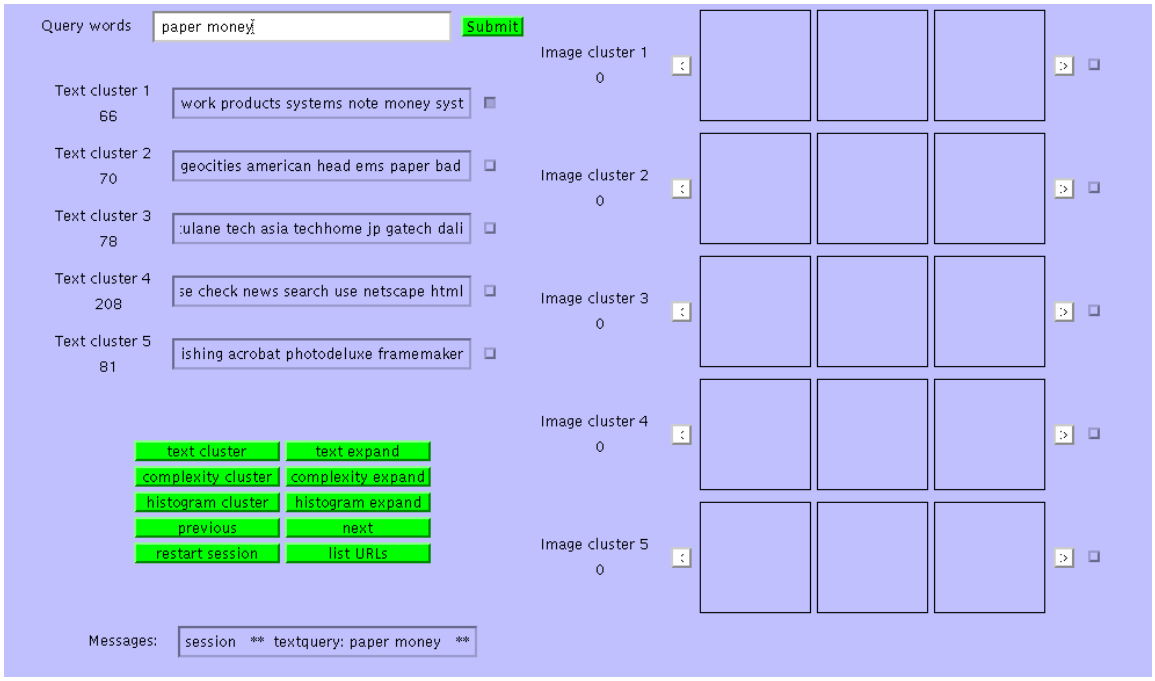

Figure 5. Text clusters returned in response to the query "paper money".

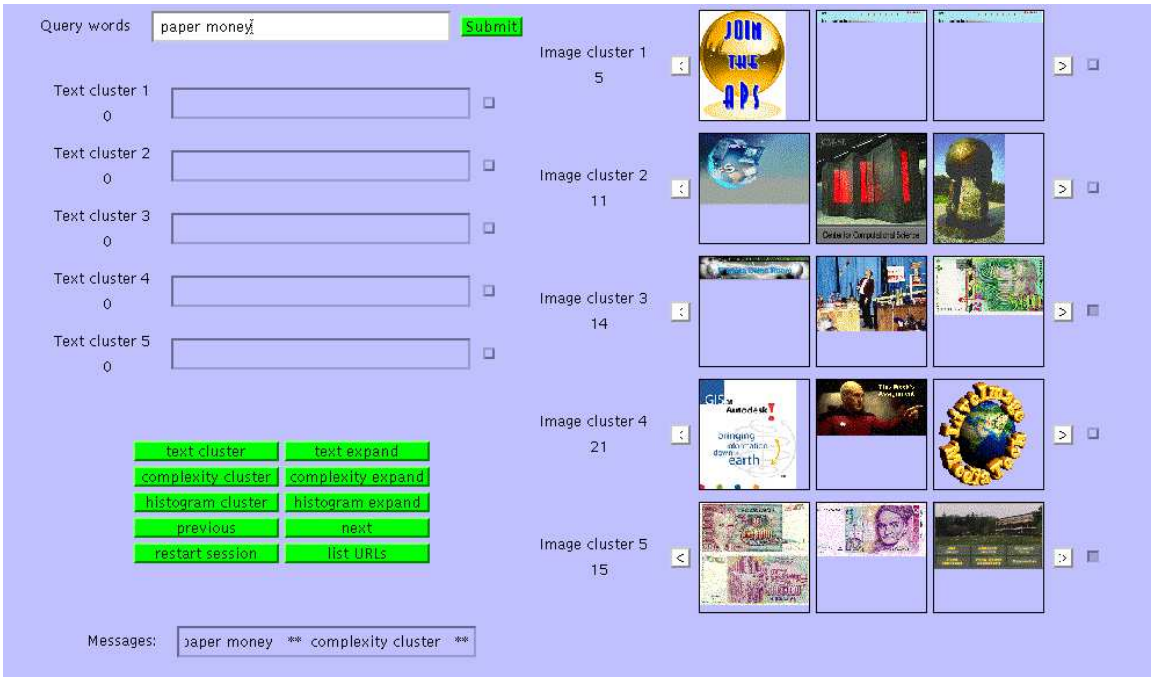

Figure 6. Image clusters returned after clustering Text cluster 1 in Fig. 5 based on the complexity feature.

displaying the five resulting text clusters is shown on the left half of Fig. 3. The user selects the three clusters that contain the terms "ancient," "cathedral" and "church" to gather and selects complexity as the feature for scattering.

A snapshot of the screen after clustering based on the image complexity is shown in Fig. 4. The representative images closest to the centroid are displayed. By clicking on the arrows next to each image cluster, the user can move between the centroid and subcluster representative views. Image clusters 1, 2 and 4 contain images primarily of "ancient" buildings and monuments, including old churches and cathedrals. Image cluster 3 contains a logo and Image cluster 5 appears to contain miscellaneous items.

In the second example, our hypothetical user is trying to find a number of images of paper money in our corpus. An initial query of "paper money" is given and the text clusters are shown in Fig. 5. The first text cluster contains the word "money" as well as the word "note". This cluster looks promising so the user selects it. Text cluster 2 contains the word "paper", but the surrounding words do not indicate that the desired sense of the word paper is being used, so this cluster is not selected. Since money is printed in many colors, the color complexity measure is more appropriate to use initially as an image feature. Text cluster 1 is scattered based on the color complexity feature and the clusters are shown in Fig. 6. Image clusters 3 and 5 contain images of paper money, so they are gathered and then scattered based on the color histogram feature this time. The resulting image clusters are shown in Fig. 7. Image cluster 2 contains 14 images, and the central representatives are all images of paper money. This cluster is scattered again based on the histogram feature and we note that it contains many images of paper money, as shown in Fig. 8. Some of the images appear to be duplicates, but in this case they are actually a thumbnail and the full-size image. Examination of the sub-cluster representatives reveals some images in the subclusters that do not contain money, but which have similar colors to the money images.

This example illustrates the use of different features in serial combination to selectively narrow the set of images to a set of interest. Scattering is used to help organize a larger collection into smaller subsets. Gathering permits different collections to be combined and reorganized together.

In the final example, the user is searching for pyramids and types in the query "pyramid egypt". The returned text clusters are shown in Fig. 9. The user selects the first text cluster to be scattered based on the complexity feature, and representative images from the resulting image clusters are shown in Fig. 10. The user notes that there are outdoor scenes with stone in image clusters 2 and 4 and selects those for further clustering based on the color

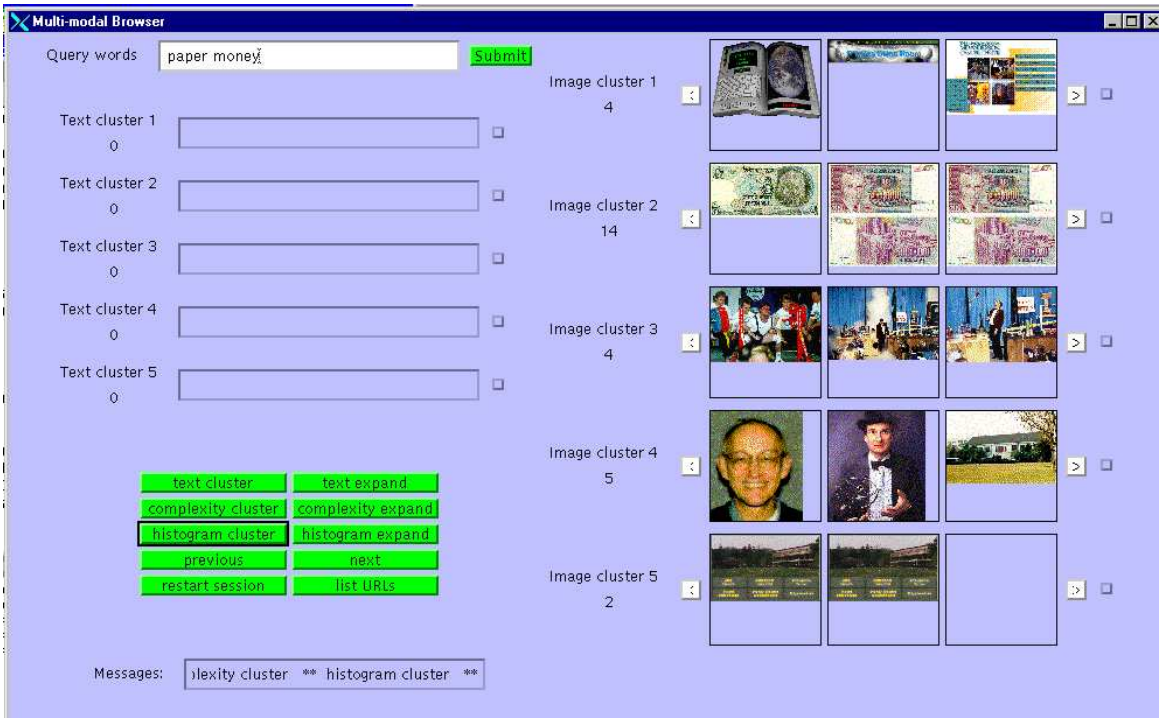

Figure 7. Image clusters returned after clustering Image clusters 3 and 5 in Fig. 6 based on the color histogram feature.

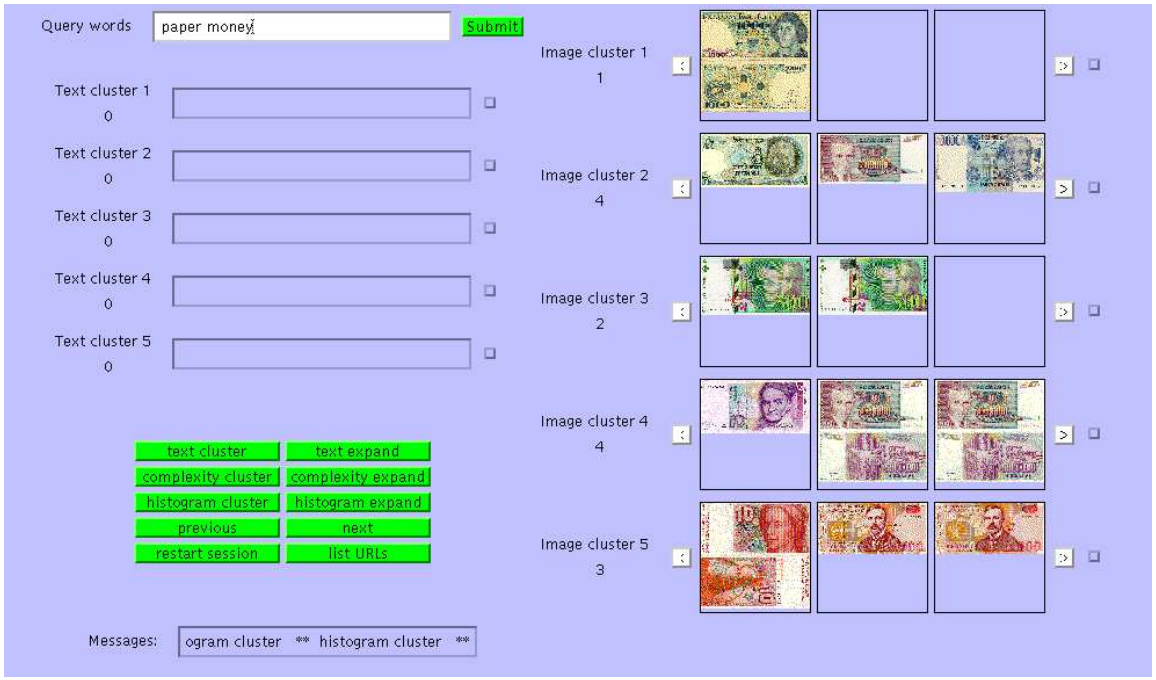

Figure 8. Image clusters returned after clustering Image cluster 2 in Fig. 7 based on the color histogram feature.

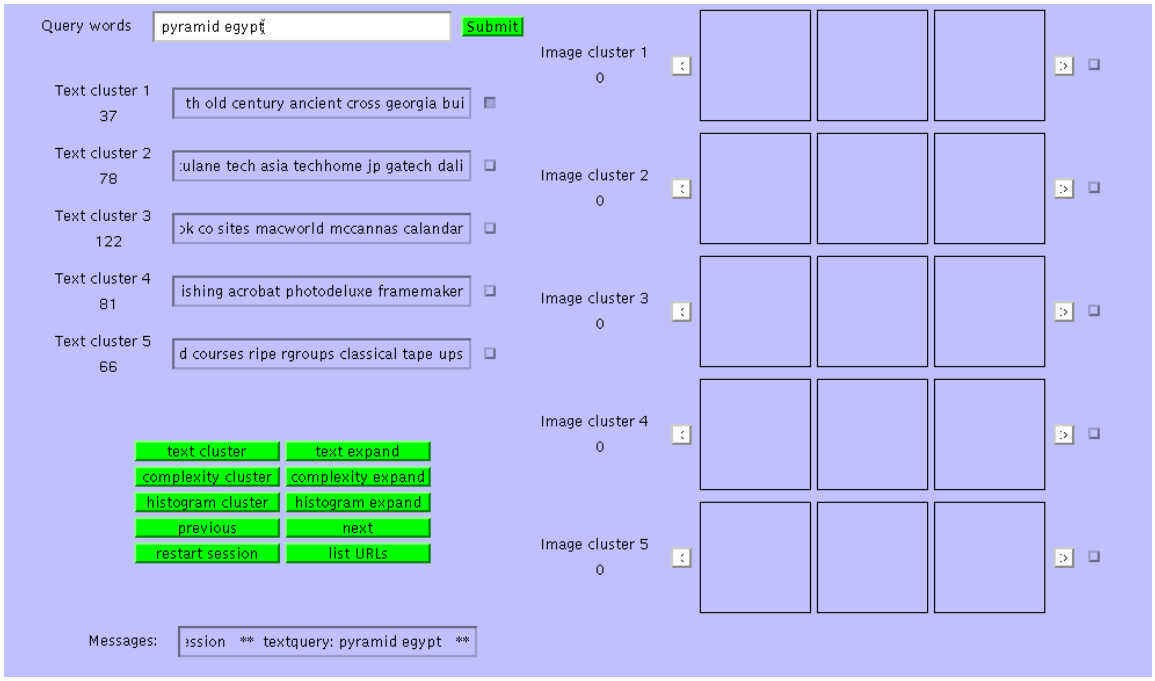

Figure 9. Text clusters returned in response to the query "pyramid egypt".

Figure 10. Image clusters returned after clustering based on the complexity feature.

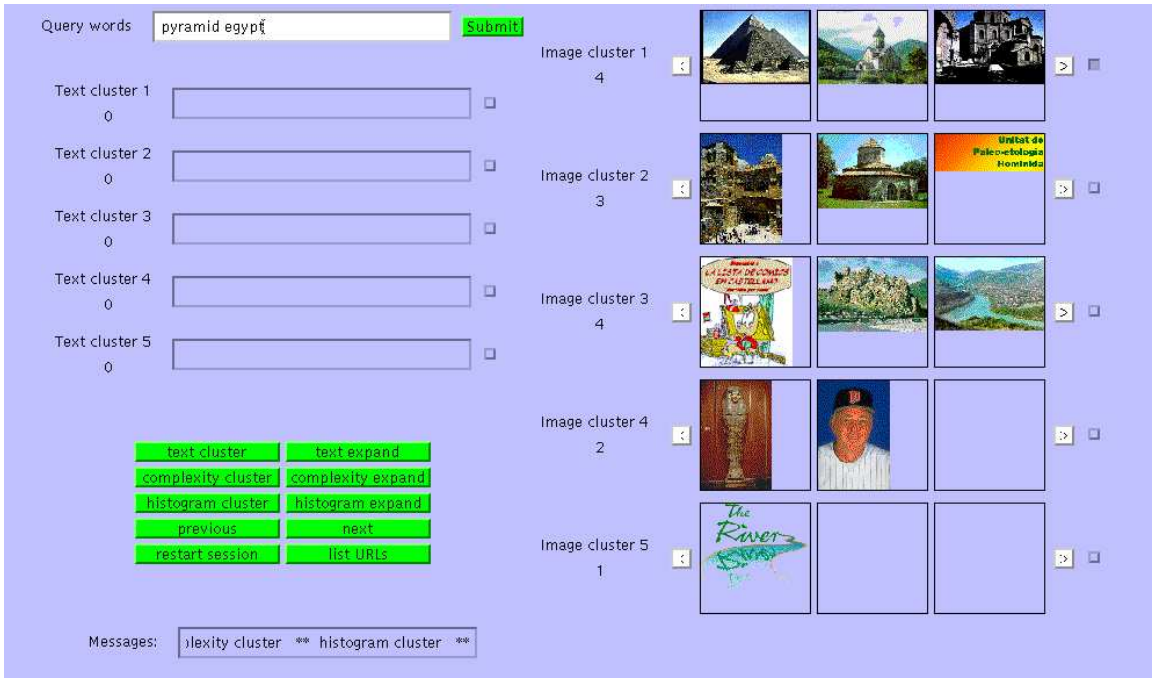

Figure 11. Image clusters returned after clustering based on the color histogram feature.

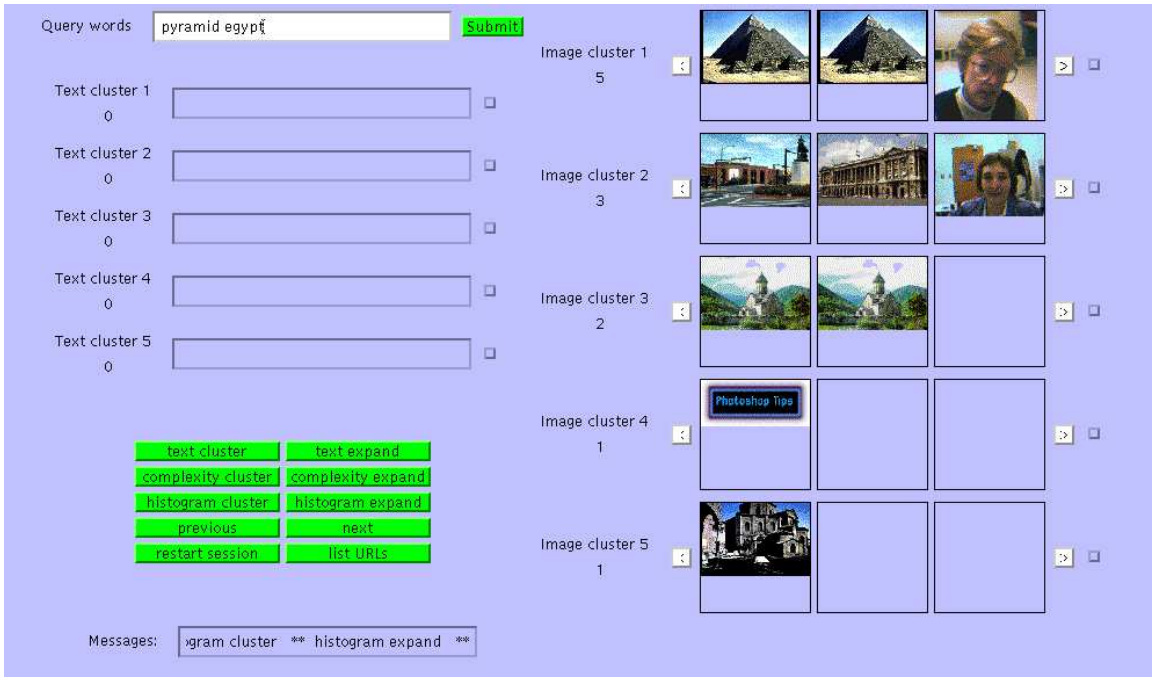

Figure 12. Text clusters returned after expanding the set of images in Fig. 11 and clustering the result based on the color histogram feature.

histogram feature. The resulting image clusters are shown in Fig. 11. Image cluster 1 contains four images, and the first image is of pyramids.

When Image cluster 1 is expanded to include similar images based on the color histogram feature, another image of a pyramid is identified, as shown in Fig. 12. This image occurs on a web page without any text and with a non-informative URL, and so it was retrieved on the basis of the color histogram feature.

In this example, the text query was used to reduce the size of the image collection, and the reduced collection was organized for presentation based on the image complexity feature. Additional images were obtained that were similar in the color histogram feature dimension.

In these examples, features in different modalities are used serially to help a user browse a set of images with associated text, using techniques of "scattering" and "gathering" subsets of elements in the corpus. A session begins with a text query to start with a more focussed initial set than the entire corpus. Clusters which are observed to contain one or more interesting elements can then be scattered to examine their content.

7. SUMMARY AND FUTURE WORK

We have developed a system for browsing a collection utilizing multiple modalities. Through an iterative process of "gathering" clusters and "scattering" the elements to examine the clusters, a user can find groups of images of interest. The expand function permits identification of elements in a collection that may be missing a value in one or more dimensions but are similar to other elements in some dimension.

In the future, we plan to enlarge the number of features and to investigate the utility of using the text features separately. The text feature can be enlarged by creating separate feature vectors for each term source (e.g., image URL, surrounding text, page URL), as described in the features section. An additional direction is to determine good methods for selecting subsets of features to combine at each step.

REFERENCES

1. D. Cutting, D.R. Karger, J.O. Pedersen and J.W. Tukey, "Scatter/Gather: A cluster-based approach to browsing large document collections," *Proceedings of the 15th Annual International SIGIR Conference*, pp. 318-329,1992.
2. M. D. Dunlop. *Multimedia Information Retrieval.* Ph.D. Thesis. Computing Science Department, University of Glasgow, Report 1991/R21, 1991.
3. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems,* 3, pp. 231-262, 1994.
4. E. J. Guglielmo and N. C. Rowe. "Natural language retrieval of images based on descriptive captions," *ACM Transactions on Information Systems,* 14, 3, May 1996, 237-267.
5. B.S. Manjunath and W.Y. Ma, " Browsing Large Satellite and Aerial Photographs," *Proceedings of the 1996 IEEE International Conference on Image Processing* Part 2 (of 3) 2 1996.
6. J. Marks, B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kan, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, S. Shieber, "Design Galleries: A general approach to setting parameters for computer graphics and animation", *SIGGRAPH97,* pp. 389-400, 1997.
7. T.P. Minka and R.W. Picard. "Interactive Learning With A 'Society of Models'," *Pattern Recognition,* 30, pp 565-581, 1997.
8. N. C. Rowe and B. Frew. "Automatic caption localization for photographs on World Wide Web pages," *Information Processing and Management,* 34, 1, 95-107, 1998.
9. Y. Rubner, L. J. Guibas, and C. Tomasi. "The earth mover's distance, multi-dimensional scaling, and color-based image retrieval," *Proceedings of the ARPA Image Understanding Workshop,* New Orleans, LA, 1997.
10. Y. Rui, T.S. Huang, and S. Mehrotra, "Relevance feedback techniques in interactive content-based image retrieval," *Proc. SPIE* 3312, pp. 25-36, 1998.
11. J.R. Smith and S.F. Chang, "An Image and Video Search Engine for the World-Wide Web," *Proc. SPIE* 3022, pp. 84-95, 1997.
12. R.K. Srihari and Z. Zhang, "A Multimedia Image Annotation, Indexing, and Retrieval System," *Proc. SIGIR Conference on Research and Development in Information Retrieval WWW Workshop*, pp. 29-45, 1998.
13. M.J. Swain and D.H. Ballard, "ColorIndexing," *Intl. Journal of Computer Vision,* 7, No. 1, pp. 11-32, 1991.
14. Arun Hampapur, Amarnath Gupta, Bradley Horowitz, Chiao-Fe Shu, Charles Fuller, Jeffrey R. Bach, Monika Gorkani and Ramesh Jain, "Virage Video Engine," *Proc. SPIE,* 3022, pp. 188-198, 1997.