CAMEL: Concept Annotated iMagE Libraries

Apostol (Paul) Natsev[a*†] Atul Chadha[b†] Basuki Soetarman[c]
Jeffrey Scott Vitter[a]

[a] Department of Computer Science, Duke University, P. O. Box 90129, Durham, NC 27708.
Email: {natsev, jsv}@cs.duke.edu.
[b] eJadoo, Inc., 2548 Minuet Drive, San Jose, CA 95131.
Email: atul@ejadoo.com
[c] Enterprise Information Portal, IBM Silicon Valley Laboratory, San Jose, CA 95141.
Email: basuki@us.ibm.com.

ABSTRACT

The problem of content-based image searching has received considerable attention in the last few years. Thousands of images are now available on the internet, and many important applications require searching of images in domains such as E-commerce, medical imaging, weather prediction, satellite imagery, and so on. Yet, content-based image querying is still largely unestablished as a mainstream field, nor is it widely used by search engines. We believe that two of the major hurdles for this poor acceptance are poor retrieval quality and usability.

In this paper, we introduce the CAMEL system—an acronym for Concept Annotated iMagE Libraries—as an effort to address both of the above problems. The CAMEL system provides and easy-to-use, and yet powerful, text-only query interface, which allows users to search for images based on *visual concepts*, identified by specifying relevant keywords. Conceptually, CAMEL annotates images with the visual concepts that are relevant to them. In practice, CAMEL defines visual concepts by looking at sample images off-line and extracting their relevant visual features. Once defined, such visual concepts can be used to search for relevant images on the fly, using content-based search methods. The visual concepts are stored in a Concept Library and are represented by an associated set of wavelet features, which in our implementation were extracted by the WALRUS image querying system. Even though the CAMEL framework applies independently of the underlying query engine, for our prototype we have chosen WALRUS as a back-end, due to its ability to extract and query with image region features.

CAMEL improves retrieval quality because it allows experts to build very accurate representations of visual concepts that can be used even by novice users. At the same time, CAMEL improves usability by supporting the familiar text-only interface currently used by most search engines on the web. Both improvements represent a departure from traditional approaches to improving image query systems—instead of focusing on **query execution**, we emphasize **query specification** by allowing simpler and yet more precise query specification.

Keywords: CAMEL, WALRUS, concepts, content-based query, images, multimedia

1. INTRODUCTION

The proliferation of digital images on the internet, and in domain-specific applications, has made the problem of searching among these images increasingly important in recent years. Image querying by content—or searching for visually similar images/regions to a query image/region—has important applications for internet search engines, E-commerce, medical diagnosis, weather prediction, agriculture and environmental change tracking, insurance, entertainment, and the petroleum industry, among others. These applications require content-based querying and retrieval of images in diverse image domains such as internet, medical, satellite, seismic, etc. Despite the need for content-based image search and the abundance of image search application domains, however, the field is still not well established or widely popular.

We believe that two major hurdles for making the image search field more mature and well accepted are its inconsistent retrieval quality and poor usability. With respect to quality, most image search engines return just a few relevant images with a significant amount of noise. This is largely due to the fact that content-based image searching is difficult to begin with, and correspondingly, most research up to date has been focused on that problem. The usability problem, however, has hardly

[*] CONTACT AUTHOR.
[†] The work was done while the author was at the Database Technology Institute, IBM Silicon Valley Laboratory.

been addressed so far, even though for some applications it is the single most deciding factor for acceptance or rejection of the technology. For example, for internet search engines, it is very hard to find and specify an appropriate query image. The problem of finding relevant images is compounded by the problem of finding an appropriate query image. The typical solutions are to provide some random images until something remotely similar is found, and then to refine the search. An alternative is to manually categorize some images to be used as query images. Both approaches, though, are inefficient, and inconvenient to the user and the database administrator.

In this paper, we propose the CAMEL system that tries to address both of the hurdles described above, with emphasis being placed on usability, which has mostly been ignored to this date despite of its importance. CAMEL stands for Concept Annotated iMagE Libraries and provides a natural approach to the usability problem. The idea is to allow users to search for images by simply specifying the keyword terms they are interested in, without requiring manual annotation of all images beforehand, and without losing the power of content-based image search. For example, given a large collection of patients' X-ray images, a doctor may want to search for images that contain "lung cancer" in order to compare previous cases to the current patient. The familiar text-only query interface is clearly a significant improvement in terms of usability. The challenge is how how to preserve the ability to efficiently search for relevant images based on their visual content, and how to avoid the subjective and labor-intensive process of manual annotation of all images. CAMEL addresses that challenge, as well as the retrieval quality problem, by providing semi-automatic, high quality objective annotation of images with visual concepts.

The remainder of the paper is organized as follows. Section 1.1 provides some background on content-based image searching and describes existing approaches to that problem. Section 1.2 gives a general overview of our proposed approach and points out some of its advantages. The features of the back-end image query system chosen for our prototype are described in Section 2. In Section 3, we discuss the similarity model adopted by our framework and in Section 4, we describe the architecture of the CAMEL system in more detail, along with a high-level description of the WALRUS components. Finally, Section 5 illustrates our implementation and some sample results, and Section 6 concludes with a summary of our contributions and future work.

1.1. Previous work

While work on improving the usability of image search applications is fairly scarce, the problem of improving the retrieval quality of image search systems has received considerable attention in the literature. The widely accepted approach to solving this problem is typically to extract a *feature vector* (or a *signature*) from every image, and to map each image essentially into a d-dimensional point P, where the coordinates of P correspond to the feature vector or a low-dimensional approximation of it. All of the images in the database are therefore mapped to points in some d-dimensional space, and the points are indexed using a multi-dimensional index, such as an R-Tree. During querying, the point corresponding to the query image is used to search the index for neighboring points that are within a small distance away from the query point, with respect to some metric such as Euclidean distance, for example. The retrieved points correspond to the similar images, which may be filtered out further by performing finer similarity computation. The approaches in the literature differ mainly in the types of features being extracted, the mapping of images to a metric space, the distance function used for similarity computation, and the type of indexing method being used.

The early image query systems, such as IBM's QBIC,[1–3] the Virage system[4] by Virage Inc., and the Photobook system[5,6] from the MIT Media Lab, used separate indexes for color histograms, shape, and texture features. The task of appropriately weighing the features relative to each other was not easy, however, and was left for the user, thus hindering usability significantly. Also, the fact that for each feature type, only a single feature vector was extracted for the entire image lead to inconsistent retrieval quality since often one signature per image is not sufficient. Jacobs *et al.*[7] were the first to use wavelet features that captured color, texture and shape simultaneously. Their system improved usability by eliminating the need to weigh the different features, and provided good performance due to the excellent approximation and dimensionality reduction properties of wavelets. The WBIIS system developed by Wang *et al.*[8] improved on the wavelet method of Jacobs *et al.*[7] by using Daubechies' wavelets, a better distance metric, and a three-step search process. Still, though, all of these methods extract only one signature per image and perform similarity comparisons at the granularity of the entire image.

John Smith considered image query systems that integrate spatial and feature information, both at the image and region levels[9–11] . Region extraction was achieved by reverse-mapping region features from a static library of image patterns. His approach allowed the user to specify the spatial location of regions, both in absolute terms as well as relative to each other, but the pipeline also allowed for false dismissals. Still, his Ph.D. thesis was a very complete treatment of region-based image querying, and addressed issues such as scale-, rotation-, and translation-invariance of image regions.

The WALRUS system[12,13] used an alternative approach for region-based querying and placed emphasis on achieving scale- and translation-invariance of image objects. The system computed wavelet signatures for hundreds, or even thousands, of sliding windows per image. The signatures were subsequently clustered in order to identify a set of regions for each image. Thus, the system extracted a variable number of wavelet region features per image, depending on the complexity of the image. By considering windows of different sizes and with different positions in each image, the system was able to achieve excellent retrieval quality. It also employed a more intuitive similarity model independent of scale or location of image regions. For further discussion of the WALRUS system, see Section 2.

1.2. Proposed approach

The CAMEL approach for improving both retrieval quality and usability is a departure from the traditional approaches to content-based image search. Instead of concentrating on the actual search itself, and trying to improve the **query execution**, CAMEL focuses on the **query specification**. Usability is improved through a simplified query interface, while retrieval performance is improved due to more expressive representation of the query term. The key idea is to split the query process in two stages: concept cataloging phase (done off-line by an expert) and actual searching (done on-line by user). The two phases are illustrated in Figure 1. Note that we do not describe a separate indexing phase because it is done solely for speedup purposes and is necessary from a pure functionality perspective.

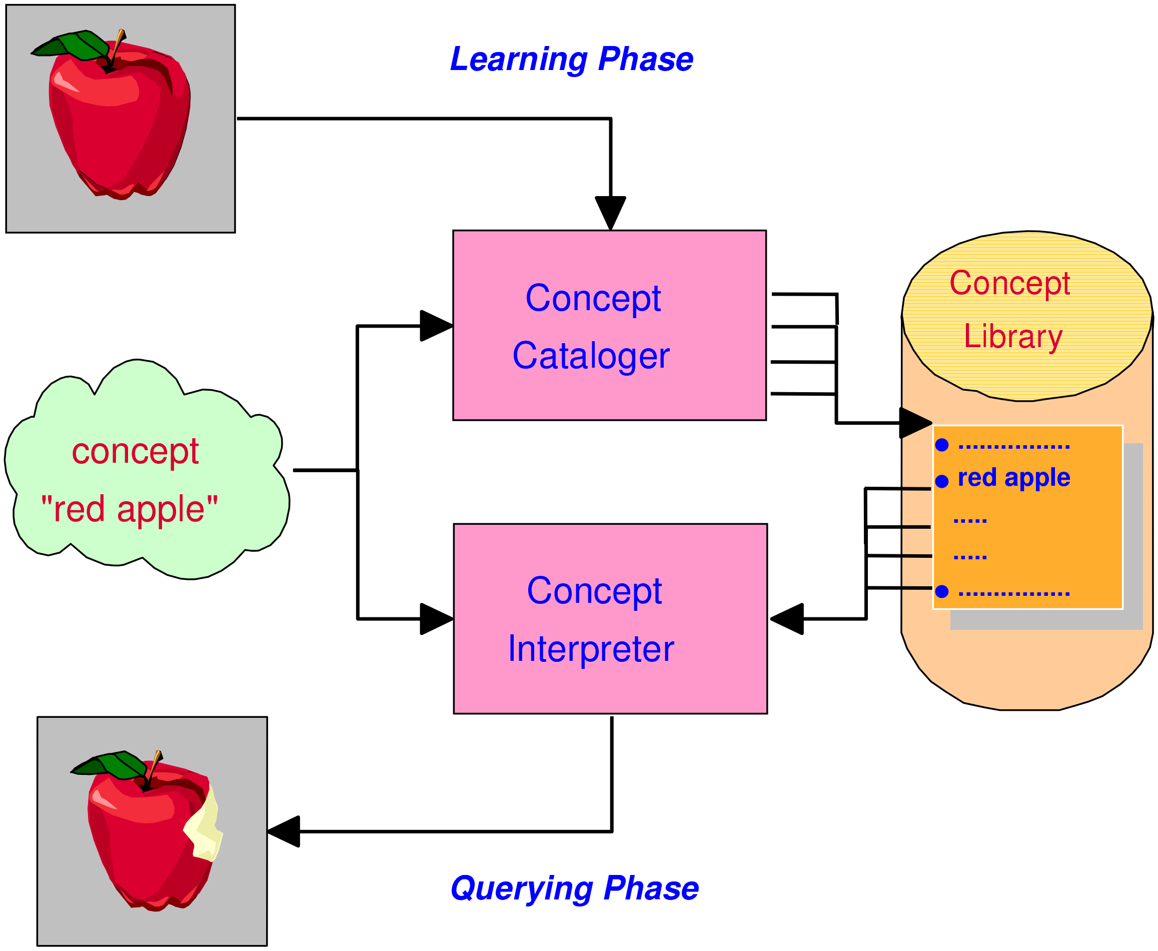


Figure 1. CAMEL system overview

The concept cataloging, or learning, phase is used to define visual concepts and build a concept library. Concepts are defined by specifying one or more sample images that are used by the system to automatically extract relevant semantic information about the concept and to associate it with the given concept. The association is performed by the *Concept Cataloger* and is stored in a *Concept Library* for future use. The Concept Library is a module for persistent storage of concepts and it can be maintained and used by many people. Note that even though the Concept Library should be built by an expert in order to maximize performance, it can later be modified by users, either by inserting new concepts or by refining existing ones. Also, such Concept Libraries can come as optional domain-specific plug-in modules, similarly, for example, to IBM's viaVoice speech recognition product that can be shipped with specific add-on modules for recognizing domain-specific terminology.

The querying phase simply takes a keyword phrase, and maps it to a visual concept (our implementation uses the identity mapping but more sophisticated approaches, including the use of multiple keywords, are discussed in the future work section). The *Concept Interpreter* then looks up the representation of the visual concept from the Concept Library and uses it to query the database for images containing the relevant concept. Even though conceptually this approach annotates images with concepts, in reality, it is significantly different from manual annotation of images. First of all, once a concept is defined, it can be used to search for relevant images without having annotated each and every one of them. Second, the concept definition is more objective than a person's annotation because a concept is represented with visual features, as opposed to limited keyword descriptions. Therefore, the image search in CAMEL is based on image content (as opposed to a simple text search among keyword-annotated images), and the final similarity score for each image match is more intuitive. Third, the extraction of the appropriate semantic information from each image is automatic, and the concept annotation is implicit. In contrast, the manual annotation approach requires that each image be explicitly annotated before it can be considered in a keyword query.

2. WALRUS FEATURES

The high-level approach that we described in the previous section requires a back-end image query system to perform the actual image search and possibly to construct the visual concept representation from one or more sample images. Even though the approach is general enough, and is independent of the underlying query engine, there are some considerations that make certain query engines more suitable than others. In the following, we consider the major factors for selecting a back-end query system and we motivate our choice.

The most fundamental question relating to our approach is the nature of the concepts' representation in the Concept Library. The simplest and the most similar to the manual annotation approach is to use text representation. However, in that case, we would lose the ability to query by content, and in addition to the manual labor required, the approach would suffer from lack of expressivity for the concept representation. In the spirit of the old saying that a picture equals a thousand words, we believe that search based on image content is a much better and more objective approach. The first such approach that comes to mind is to simply store the image IDs of some representative images for each visual concept. However, although this approach would help with the usability problem, it would be inefficient in terms of performance, both with respect to retrieval quality and running time. Since ultimately the search is based on image features rather than the images themselves, storing image IDs would mean that for each query, several images would have to be fetched from disk and processed for feature extraction. This would be redundant computation and can be saved simply by doing it off-line and storing the relevant image features for each concept, rather than image IDs. For retrieval quality considerations, storing only the relevant image features for each concept will be a more accurate representation of the concept, compared to the entire image. This is due to the fact that each image contains some noise in addition to the relevant content, and without pre-processing the image to extract only the relevant information, this noise would deteriorate the quality of the concept representations, and therefore the overall retrieval quality. We therefore opted to represent concepts by relevant image features only, and we try to maximize the amount of computation done in the off-line phase so that there is minimal overhead in the online phase. That way we achieve both CPU time savings and increased retrieval quality due to higher precision of the concept representations.

The next question that we need to answer is what features exactly we should store in the concept representations. One of the most commonly used image features is the color histogram, which encodes statistical information about the distribution of colors in the picture. Color histograms, however, do not capture any shape or texture information, so by themselves they are not sufficient to represent concepts. Combining them with other independent features for shape and texture is not easy and usually places the burden on the user to specify weights for the different features. Since this goes against our effort to improve usability, we use wavelet features instead. Wavelets provide very compact representations, offer excellent approximation properties, and capture color, shape, and texture in a single unified framework.[7,8,13,12] Ideally, we would also want to be able to extract such features for image regions, rather than the entire images, and we would like to generate a variable number of regions based on the complexity of the image content. The region feature extraction and matching at the sub-image level are important because in order to achieve the most accurate possible concept representations, we need to have the highest possible granularity for image features. By being able to extract image regions and their features, we can describe concepts more precisely and we can eliminate noise from sample images more effectively. Since the WALRUS system[12] meets all of the above requirements, and was available to us, we chose it as a back-end to the CAMEL system.

In order to extract regions for an image, WALRUS considers sliding windows of varying sizes and then clusters them based on the proximity of their wavelet signatures. Thus, the system generates a variable number of feature vectors for each image—one per image region, where the number of image regions depends on the complexity of the image. Furthermore, by performing similarity matching at the sub-image granularity, WALRUS's similarity model can detect similarity in cases where some regions in an image are scaled or translated versions of regions from another image. The performance of the WALRUS system is very competitive in practice and achieves very good retrieval quality. All of these properties are naturally inherited by the CAMEL system.

3. SIMILARITY MODEL

The image similarity model that we use in the CAMEL framework is roughly the same as the WALRUS model but with slight modification to accommodate the notion of visual concepts. The WALRUS similarity model is illustrated in Figure 2. The formal definitions appear in Natsev *et al.*[12] Informally, the similarity between two images is defined as the fraction of matched image area vs. the total image area in both images. For simplicity, we assume here that the matched area is simply the union of all matched regions in the images.[1] Image regions are said to be similar if their wavelet signatures are within a certain ϵ distance from each other. Scale and position of regions are ignored, as shown in Figure 2, where translated and scaled versions of query regions are matched in the target image. According to the above similarity model, the similarity score between the two images should be around 0.5 to 0.6 because the area of matched regions comprises about 50% to 60% of the total area in the two images.

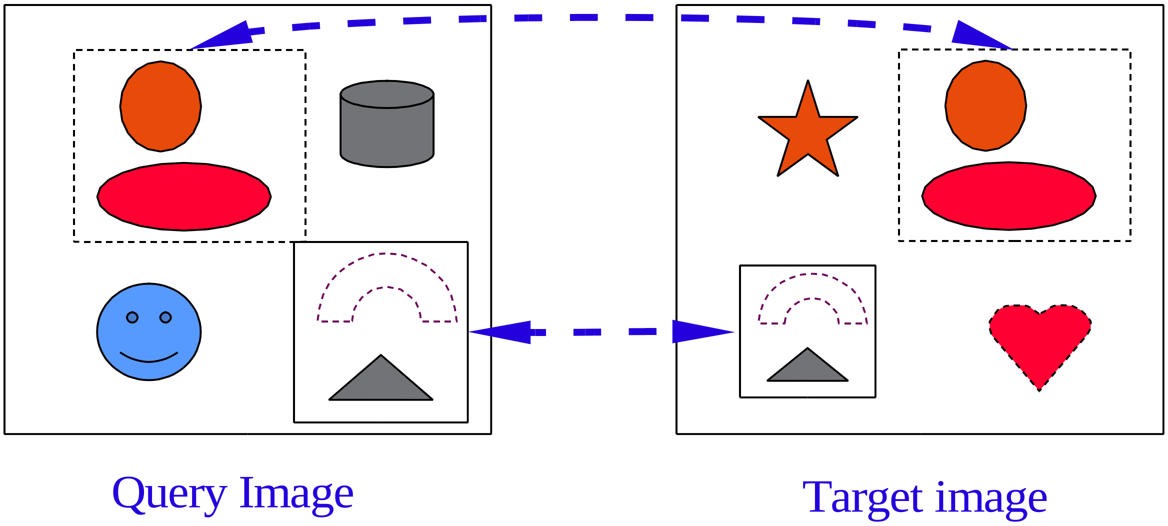


Figure 2. Image similarity model

For the CAMEL system, we need to consider the fact that the query object is not an image but rather a visual concept, consisting of region signatures extracted from one or more images. Since the feature vectors stored in a concept's representation correspond to actual regions from some sample images, the region matching definition still applies and we could identify pairs of matching regions from the query concept and the target image. However, since the query regions no longer come from a single image, and we do not store position information with each region, we cannot simply compute the union of all matched query regions in order to calculate the total matched query area. Instead, we define the similarity between a query concept and target image to be the weighted average between a query match score and a target match score. The target match score is defined as before to be the fraction of matched area vs total image area in the target image. The query match score, however, is

[1] An alternative definition is discussed in Natsev *et al.*,[12] where the system enforces the restriction that each image region of the query image can match no more than one target image region and vice versa.

defined as the fraction of matched query sliding windows vs the total number of query sliding windows. Recall that the image regions are extracted by clustering the wavelet signatures of sliding windows of the image so each image region represents a number of sliding windows. By adding up the numbers of sliding windows for all matched regions, and dividing it by the total number of sliding windows, we arrive at a sensible way of computing the query match score, even though the query concepts are represented by regions extracted from multiple images. We formally define the similarity model between a concept and a candidate image by modifying the following definitions from the original WALRUS model[12]:

DEFINITION 3.1. **(Similar region pair set)** *For a query concept Q and a target image T, the set of ordered region pairs $\{(Q_1, T_1), \ldots, (Q_l, T_l)\}$ is referred to as a similar region pair set for Q and T if Q_i is similar to T_i and for every $i \neq j$, $Q_i \neq Q_j$ and $T_i \neq T_j$.*

DEFINITION 3.2. **(Image similarity)** *A query concept Q is said to be similar to a target image T if there exists a similar region pair set for Q and T $\{(Q_1, T_1), \ldots, (Q_l, T_l)\}$, such that:*

$$W \cdot \frac{card(\cup_{i=1}^{l}(Q_i))}{card(Q)} + (1 - W) \cdot \frac{area(\cup_{i=1}^{l}(T_i))}{area(T)} \geq \xi$$

In the above definition, $card(\cup_{i=1}^{l}(Q_i))$ is the number of sliding windows in Q represented by regions $Q_1, \ldots, Q_l$. The term $area(\cup_{i=1}^{l}(T_i))$ represents the number of pixels in T covered by regions $T_1, \ldots, T_l$ considered together. The weight W can be static or can be dynamically manipulated by the user according to the application needs. Alternative definitions are also possible, similarly to the variations suggested by the WALRUS paper.[12]

4. SYSTEM ARCHITECTURE

Our CAMEL prototype is built on top of the WALRUS system for image indexing and retrieval. Figure 3 illustrates the architecture of the CAMEL system along with that of the WALRUS system. The WALRUS system is comprised of three major components: region feature extraction, region matching, and image matching. In addition, the CAMEL system adds the Concept Cataloger and the Concept Interpreter components, as well as the Concept Library storage module.

The basic operations previously supported by the WALRUS system were indexing an image and querying with an image. The image indexing phase involves extracting region signatures and inserting them into an R^*-tree index, while the querying operation involves region feature extraction, probing of the R^*-tree index to find matching regions to the extracted query image regions, and finally using the matched region pairs to compute a final similarity score for ranking of each candidate image match. The region feature extraction procedure consists of decomposing the image into sliding windows, computing a Haar wavelet signature for each sliding window, and then clustering all window signatures in order to identify regions with homogeneous wavelet signatures. For a detailed description of the WALRUS system and its components see.[12]

The new functionality added by the CAMEL system includes the concept cataloging phase and the querying by concept phase. In the concept cataloging phase, the system takes as an input a concept specification (e.g., concept "apple") and a sample image containing the specified concept (i.e., an image of an apple). The system then uses the WALRUS region feature extraction module to get all regions from the sample image. The regions are classified as either *important*, *unimportant*, or *neutral*, according to a significance score evaluated for each region. The significance score uses a heuristic formula that takes into account the image area covered by the region (i.e., the span of the region) and the number of sliding windows represented by the region (i.e., the cardinality of the region). The final significance score for each region is a fractional number between 0 and 1, with 1 representing absolutely critical regions, and 0 representing noise and region outliers. Regions with significance score higher than a certain threshold are considered *important*, and are always inserted into the Concept Library as part of the representation of the specified concept. If there is an existing region feature in the concept's representation that is close enough to a new important region, then the two region clusters are merged, and the adjusted centroid (or bounding box) is used as a representative feature for the new cluster. If the new region signature is not close to any of the existing representative signatures, then it is inserted as a new representative signature for the given concept. If on the other hand, the significance score for the region is below the threshold but there is an existing signature similar enough to the new signature, then the new signature is deemed *neutral* and is merged with the existing region in order to refine its signature. All other signatures are considered *unimportant* and are discarded. The above mechanism allows for a more selective procedure for defining a visual concept's representation and it tries to filter out noise so that the resulting concept representation is pure. This procedure and the fact that region signatures from multiple images can be used to refine the concept, improves the quality of the concept representation and lead to better retrieval quality. Improvements in the heuristic scoring formula and the threshold estimation functionality are still open problems.

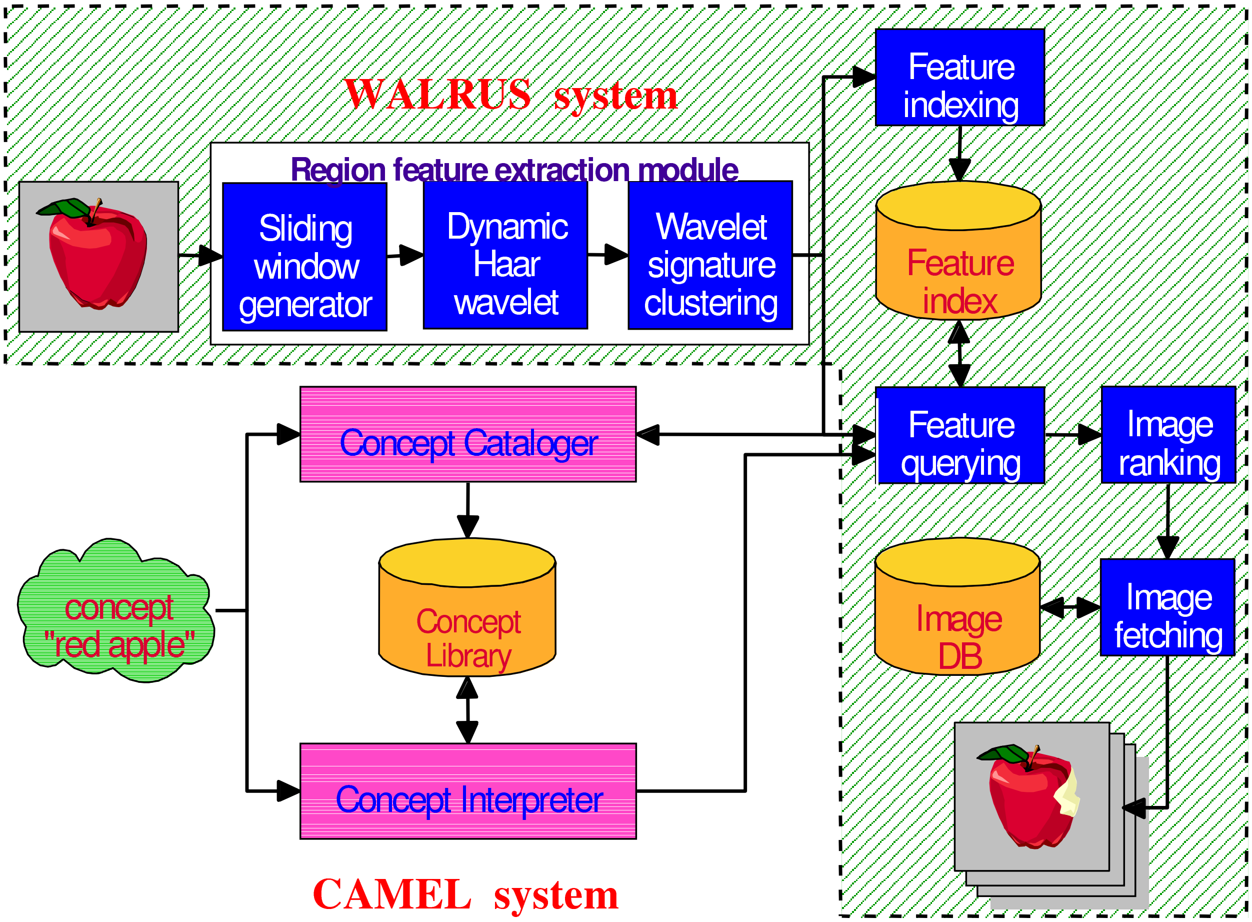


Figure 3. Architecture of the CAMEL system

The last piece of new functionality includes the Concept Interpreter and the querying by concept phase. During an actual query by concept, the system takes as input only the concept, and uses the Concept Interpreter to look up its representation from the Concept Library. The returned set of concept features are then used to probe the R^*-tree index to find matching image regions, and the modified similarity scoring is used to rank potential image matches relevant to the specified concept. The pseudo-code for the above four basic operations is listed in Figures 4 and 5.

5. IMPLEMENTATION

In this section we show some sample queries of the CAMEL system, and compare its retrieval quality with the WALRUS system. The results in this section are not meant to be an extensive study on performance but rather to appear simply as a proof of concept, and to demonstrate that usability can be improved without necessarily sacrificing performance.

In terms of retrieval quality, we show that a refined concept representation can achieve better performance than the WALRUS system, which by itself has very good retrieval quality. Due to space and time constraints we do not report comparisons with image query systems other than WALRUS, although CAMEL did perform favorably with respect to QBIC, and a comparison of WALRUS with an alternative wavelet approach can be found in the WALRUS paper.[12] The running time and space considerations of CAMEL are comparable to that of the underlying image query engine, or WALRUS in our case. We do have some extra space requirements to store the Concept Library but they are generally offset by the fact that the data stored there consists of precomputed feature vectors and therefore saves CPU time. Also, it could be argued that without the Concept Library, a search engine would need to store actual images on the client side or transmit them through the network before every query, which is somewhat equivalent to keeping that information in a Concept Library. Also, our experience has been that the concept representations in the Concept Library are very compact due to the region filtering and noise elimination performed in

```
Index image operation:

1. Extract region features from input image:
   (a) Generate sliding windows of different sizes
   (b) Dynamically compute Haar wavelet for all sliding windows
   (c) Cluster wavelet signatures for all sliding windows
   (d) Use each cluster's centroid (or bounding box) as a region feature
2. For each region signature, do:
   (a) Insert the region signature into $R^*$-tree index

Query by image operation:

1. Extract region features from input image $Q$:
   (a) Generate sliding windows of different sizes
   (b) Dynamically compute Haar wavelet for all sliding windows
   (c) Cluster wavelet signatures for all sliding windows
   (d) Use each cluster's centroid (or bounding box) as a region feature
2. For each query region signature $R_Q$, do:
   (a) Probe the $R^*$-tree index to find similar signatures
3. For each returned region match $R_T$ from target image $T$, do:
   (a) Add $T$ to the set of candidate matches $M$, if it's not already present
   (b) Add $R_T$ to the union $UT(T)$ of matched target regions in image $T$
   (c) Add $R_Q$ to the union $UQ(T)$ of matched query regions by image $T$
4. For each target image $T$ from $M$, do:
   (a) Use $UT(T)$ and $UQ(T)$ to compute similarity score $S(T)$ for target image $T$
5. Sort candidate images by similarity score
6. Pipe ranked list of image matches to the output
```

Figure 4. Pseudo-code for WALRUS functionality

the concept cataloging phase. We can often store more expressive concept representations derived from several images with roughly the same number of region features as generated from a single image in the WALRUS system.

Our implementation of the CAMEL prototype system is built on top of a re-implemented library of WALRUS back-end functionality. As in the original WALRUS implementation, we used the BIRCH pre-clustering phase for clustering of signatures.[14] We have also used the R^*-tree instantiation of the GiST package[1] as a disk-based index for the feature vectors, and the ImageMagick library[2] for reading various image formats and converting between different color spaces. The Concept Library was implemented as three database tables populated and queried through the Call Level Interface of IBM's DB2 v. 6.1 database. The data set we used for querying is the one used in the WALRUS paper,[12] and consists of approximately 10000 JPEG images with sizes 85×128, 128×85, or 96×128. The query response time was in the order of 10 seconds. The R^*-tree index was slightly bigger than the total size of the images since, on the average, we store about 25 feature vectors per image. The image database we used contained relatively small images (see above) and was therefore very compact. We don't expect the side of the index to be necessarily larger than the original database in other application scenarios, where the images are large.

Figure 6 shows a sample query result of the WALRUS system. The query image is in the top left corner and has an ID of 885. The clustering and querying parameters used in the WALRUS system were as specified in the original paper,[12] and the

[1] Available at http://epoch.cs.berkeley.edu:8000/gist/libgistv1.

[2] Available at http://www.wizards.dupont.com/cristy/ImageMagick.html

Catalog concept operation:

```
1. Extract region features from input sample image:
   (a) Generate sliding windows of different sizes
   (b) Dynamically compute Haar wavelet for all sliding windows
   (c) Cluster wavelet signatures for all sliding windows
   (d) Use each cluster's centroid (or bounding box) as a region feature
2. For each region signature, $R$, do:
   (a) $RegionProcessed$ = False
   (b) Compute significance score for that region, $SS(R)$
   (c) For each existing region feature $E$ in given concept's representation, do:
       i. If $dist(E, R) \leq \tau$, then
          A. Merge $R$ into $E$
          B. Use new cluster's centroid (or bounding box) as a feature representative for new region $E$
          C. $RegionProcessed$ = True
          D. Break
   (d) If $RegionProcessed$ = False && $SS(R) > \sigma$, then
       i. Insert $R$ as new region for given concept
   (e) Else discard region $R$
```

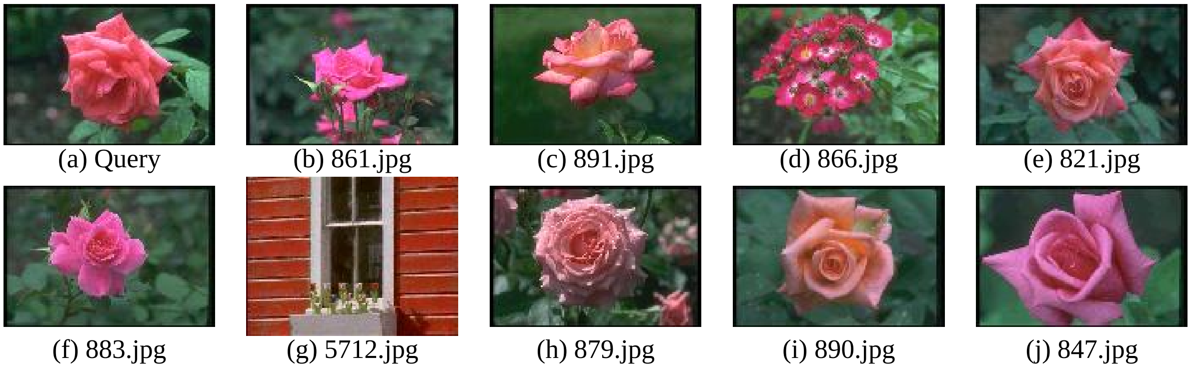
Query by concept operation:

```
1. Lookup concept's region features from the Concept Library
2. For each region signature $R_Q$, do:
   (a) Probe the $R^*$-tree index to find similar signatures
3. For each returned region match $R_T$ from image $T$, do:
   (a) Add $T$ to the set of candidate matches $M$, if it's not already present
   (b) Add $R_T$ to the union $UT(T)$ of matched target regions in image $T$
   (c) Add cardinality of $R_Q$ to the cardinality $CQ(T)$ of matched query concept features by image $T$
4. For each target image $T$ from $M$, do:
   (a) Use $UT(T)$ and $CQ(T)$ to compute similarity score $S(T)$ for target image $T$
5. Sort candidate images by similarity score
6. Pipe ranked list of image matches to the output
```

Figure 5. Pseudo-code for CAMEL functionality

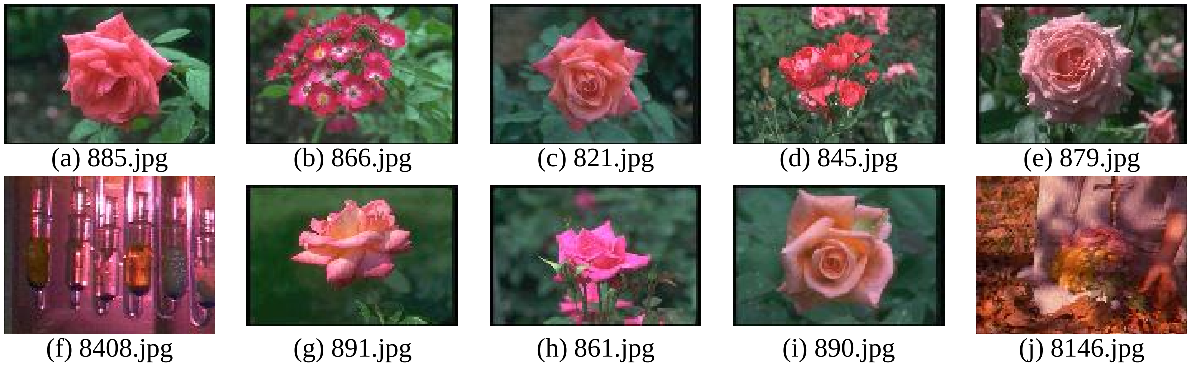
wavelet transform was performed in the YCC color space. The top 10 images (9 if we don't count the query image) are listed from left to right, top to bottom, in order of decreasing similarity. With the exception of image 5712, all of the returned images are indeed visually similar to the query. We should note that for all three sample queries presented in this paper, the top 20 images appearing immediately after the ones reported here were typically visually similar to the query, with some exceptions. For technical and printing considerations we only show the top 10 results from each query.

Figure 7 illustrates a sample query result in the CAMEL system. The concept used for querying was defined using only a single image (with ID 885) as an example. We adjusted the noise filter so that regions with span below 50% of the image area are discarded as noise. Due to noise filtering effects, the CAMEL system stores only 9 out of the 11 regions generated from image 885.jpg. From the results, we can observe that the quality is slightly worse than the corresponding WALRUS query—this can be attributed to the modified similarity model which can only consider cardinality ratio as a matching score for the query


(a) Query (b) 861.jpg (c) 891.jpg (d) 866.jpg (e) 821.jpg

(f) 883.jpg (g) 5712.jpg (h) 879.jpg (i) 890.jpg (j) 847.jpg

Figure 6. Images retrieved by WALRUS. The query image has ID 866 and generates 11 regions.

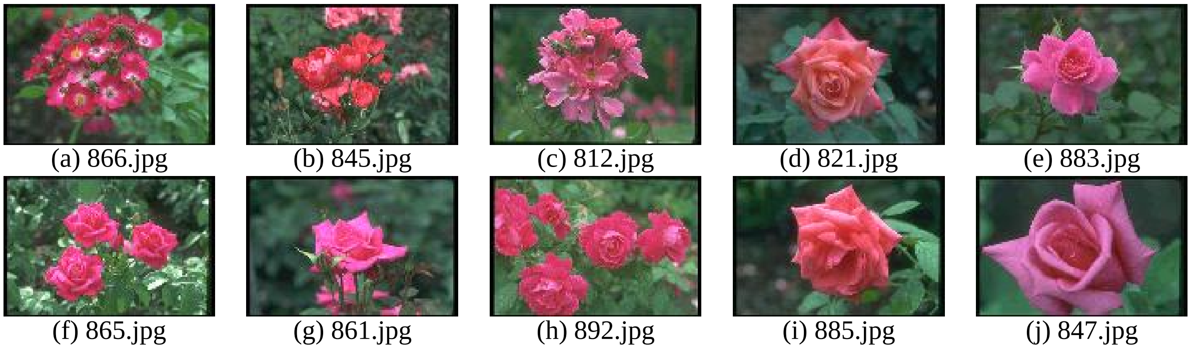

(a) 885.jpg (b) 866.jpg (c) 821.jpg (d) 845.jpg (e) 879.jpg

(f) 8408.jpg (g) 891.jpg (h) 861.jpg (i) 890.jpg (j) 8146.jpg

Figure 7. Images retrieved by CAMEL for a *flower* concept query represented by 9 regions extracted from image 885.jpg.

concept, as opposed to the more expressive matched area fraction used for images. However, the quality is still excellent and considering the fact that the concept was defined using a single image, the query results are still promising.

The potential for improved retrieval quality is more clearly established in Figure 8, where the *flower* concept is refined by considering 4 sample pictures, rather than just one. In other words, we took four examples of flower pictures, extracted their regions, merged the similar ones, and discarded the outliers. This results in a slightly bigger concept representation but produces the best results out of the three by returning relevant images in all of the top spots. In addition, the images returned in the third query are more diverse than the previous two query results in that they include pictures of single flowers (e.g., 812, 821, 883, 861, 885, and 847), as well as groups of a small number of flowers (e.g., 865 and 892), and also flowers appearing as a bunch (e.g., 866 and 845). That effect is not clearly expressed in the WALRUS query results or in the initial concept query, which shows that the retrieved answers can be manipulated by careful refinement of the visual concept so that the results are tailored towards a specific application. This capability alone is an important advantage of the CAMEL system.


(a) 866.jpg (b) 845.jpg (c) 812.jpg (d) 821.jpg (e) 883.jpg

(f) 865.jpg (g) 861.jpg (h) 892.jpg (i) 885.jpg (j) 847.jpg

Figure 8. Images retrieved by CAMEL using a refined *flower* representation consisting of 25 regions from images 885.jpg, 865.jpg, 866.jpg, and 892.jpg.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a new direction for improving current image query systems, namely focusing on the query specification part as a gateway to better performance and usability. The CAMEL system that we have proposed, coupled with the WALRUS system as a back-end, performs image matching at the sub-image level and uses an intuitive similarity model that is robust with respect to scale and position. The wavelet region features extracted by the WALRUS system capture color, texture and shape, and enable high quality concept representations to be automatically constructed from sample images. The separation of querying from feature selection provides a way for experts to build domain-specific concept libraries that users can refine even further. As a result of combining semantic information from multiple images into a single concept, the queries become more expressive and more representative of what users are really looking for, and therefore users observe increased retrieval quality. The flexible query specification mechanism of CAMEL allows novice users to query through a familiar text-only interface, without the need for intensive manual annotation labor and while allowing experts to utilize the powerful search features of WALRUS. Thus, CAMEL successfully combines the power of content-based querying with the simplicity of keyword querying.

As future work we would like to consider semantic hierarchies of visual concepts, so that lower level concepts (e.g., “pine” and “oak” are grouped under higher level categories (e.g., “tree”). This extension is motivated by the fact that the current “flat” architecture makes it hard to define more general concepts, such as “animals”, for example, since the images that fall in such categories may be very different from each other. We would also like to investigate ways for automatic taxonomy and classification of images, based on such concept hierarchies.

Another improvement that we are currently considering is the introduction of spatial constraints in the query engine. For example, we may want to preserve the scale and position of two objects relative to each other, even though the absolute scale and position do not matter. As an example, if we are looking for pictures of a mother with a young daughter, we might want to impose the restriction that the mother is roughly twice as tall as the daughter, and also that they are near each other. The proximity constraint has an equivalent analogy in text searching, where the locations of the keywords in the document are taken into account when ranking the documents. Thus, if the keywords appear in the same sentence or paragraph, the corresponding document is ranked higher.

Finally, another way of improving usability even more would be to investigate more sophisticated mappings from keywords to visual concepts, as well as Boolean combinations of keywords. In the current implementation, this mapping is the identity mapping and each keyword represents a unique concept. An alternative would be to use a thesaurus and compute synonym equivalence classes that represent visual concepts (using for example WordNet[15]). During a query, the system would then transform each keyword into the canonical representation of an appropriate equivalence class (perhaps with the help of the user), so that the user can find relevant matches even if they didn’t phrase the query precisely. This has already been investigated in the context of text search and is currently available, for example, in IBM’s DB2 Text Extender.

REFERENCES

1. C. Faloutsos *et al.*, “Efficient and effective querying by image content,” *Journal of Intelligent Information Systems* **3**, pp. 231–262, 1994.
2. W. Niblack *et al.*, “The QBIC project: Query image by content using color, texture and shape,” in *Storage and Retrieval for Image and Video Databases*, pp. 173–187, SPIE, (San Jose), 1993.
3. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, B. Dom, Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, “Query by image and video content: the QBIC system,” *IEEE Computer* **28**(9), pp. 23–32, 1995.
4. A. Gupta and R. Jain, “Visual information retrieval,” *Communications of the ACM* **40**(5), pp. 69–79, 1997.
5. R. W. Picard and T. Kabir, “Finding similar patterns in large image databases,” in *IEEE ICASSP*, vol. V, pp. 161–164, (Minneapolis), 1993.
6. A. Pentland, R. W. Picard, and S. Sclaroff, “Photobook: Content-based manipulation of image databases,” in *SPIE Storage and Retrieval Image and Video Databases II*, (San Jose), 1995.
7. C. E. Jacobs, A. Finkelstein, and D. H. Salesin, “Fast multiresolution image querying,” in *Proc. of SIGGRAPH 95*, Annual Conference Series, pp. 277–286, August 1995. Available at http://www.cs.washington.edu/research/projects/grail2/www/pub/abstracts.html.
8. J. Z. Wang, G. Wiederhold, O. Firschein, and S. X. Wei, “Content-based image indexing and searching using Daubechies’ wavelets,” *Intl. Journal of Digital Libraries (IJODL)* **1**(4), pp. 311–328, 1998. Available at http://www-db.stanford.edu/∼zwang/project/imsearch/IJODL97/.

9. J. R. Smith and S.-F. Chang, “Integrated spatial and feature image query,” *Multimedia Systems* **7**(2), pp. 129–140, 1999.
10. J. R. Smith and S.-F. Chang, “Querying by color regions using the VisualSEEk content-based visual query system,” in *Intelligent Multimedia Information Retrieval*, T. M. Maybury, ed., IJCAI, 1997.
11. J. R. Smith, *Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis*. PhD thesis, Graduate School of Arts and Sciences, Columbia University, Feb. 1997. Available at http://www.ctr.columbia.edu/∼jrsmith/publications.html.
12. A. Natsev, R. Rastogi, and K. Shim, “WALRUS: A similarity retrieval algorithm for image databases,” in *Proc. 1999 ACM SIGMOD International Conference on Management of Data*, pp. 395–406, (Philadelphia, PA), May 1999.
13. A. Natsev, R. Rastogi, and K. Shim, “WALRUS: A similarity matching algorithm for image databases,” tech. rep., Bell Laboratories, Murray Hill, 1998.
14. T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An efficient data clustering method for very large databases,” in *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 103–114, (Montreal, Canada), June 1996.
15. C. Fellbaum, ed., *WordNet: An Electronic Lexical Database*, no. ISBN 0-262-06197-X, MIT Press, 1998. Software can be downloaded from http://www.cogsci.princeton.edu/ wn/.