Content-Based Image Retrieval

Malek Adjouadi[1], Marie L. Lucas[1], Emanuel L. Pozo[1], Hien Nguyen[1], Kevin Maynard[2], Stacey Thomas[1], Armando Barreto[1], Scott Graham[1], Naphtali Rishe[1]

Florida International University [1]

University of Illinois at Chicago[2]

Abstract

With today's large increase in digital images and automatically generated imagery, such as videos and stills generated from surveillance equipment, the need for efficient image retrieval and indexing has become fundamental. Since text-based information retrieval has been shown to perform very poorly when searching through images, research has been active in the field of content-based image retrieval (CBIR). CBIR systems make use of the properties of images in order to compare them and extract content by matching the query image. Comparing features – such as color, texture, and shape – allows for better retrieval accuracy; however, the algorithms used are still very limited.

This paper will provide a survey of CBIR systems and explain the fundamental properties and techniques used in these systems. First, the history of CBIR systems will be discussed together with some typical CBIR systems. After this, the paper will touch on text-based information retrieval and explain why it does not work for searching through collections of images. The latter portion of this document will provides an overview of a typical CBIR system and the main techniques involved in querying such a system. Finally, image features and indexing schemes will be described.

1. History

Information retrieval has been in existence for quite a while. It is formally defined as the process of searching through a collection of documents for desired information. The phrase information retrieval was believed to be coined by Calvin Mooers in his 1948 M.I.T master thesis. Since then, text based information retrieval, which refers to searching collections of text, has been used tremendously, both in academia and in large scale commercial applications such as Google and Yahoo [12]. Even though this method of searching through data works, there is an ever increasing need to adapt IR to today's growing multimedia market, that is, searching through collections of videos and images based on their content and not based on semantic values associated with them. Content based image retrieval systems attempt to accomplish this challenging feat.

The roots of CBIR began in the field of computer vision and can be traced to 1992, when it was used by T. Kato to describe experiments that dealt with the retrieval of images from a large database based on their syntactic features, such as colors, textures, and shapes [12]. Since that time many techniques and algorithms have been used to accomplish this method of IR, based on traditional statistical and probabilistic methods. There have been many CBIR systems developed, running in different environments. Some have been desktop GUI applications, while others have taken advantage of the internet and are web-based applications.

Some of the most popular systems include QBIC, Photobook, and NETRA [13]. For example, the first commercial system, QBIC (Query by Image Content), was developed by IBM at the Almaden research laboratory, and was used to query large multimedia and image databases based on the content of the images stored in those databases [7]. The QBIC system supports many low-level features, such as average color, color histograms, textures, and shapes. This large scale application has lead to major improvements in CBIR retrieval techniques and has played a vital role in the development of large scale CBIR systems [13].

2. Why Is Text-Based Information Retrieval not Efficient for Multimedia Content

Traditionally, images are annotated by text; meta-data is associated to each image and then used to index images. Once this is in place, they can be searched by traditional textual methods. These methods work for simple data collections; however, with large databases they become impractical, since annotating images requires an automatic process which does not guarantee precise meta-data naming of the image collection. These, therefore, result in the need for human annotation which is practically impossible because of the amount of data in question. In addition, human annotation can be subjective, therefore resulting in inaccurate naming conventions.

The ideal solution to this problem is to search a collection of images based on their content. This is why content-based information retrieval systems are used. This approach searches images based on color, texture coordinates and object shapes. These systems are far from being used on a massive scale; however, they have tremendous potential and can in theory eliminate the tedious task of annotating images. Of course, it is a very challenging undertaking and has become the topic of research for many computer science laboratories around the world.

3. Models and Features

The goal of a CBIR system is to allow users to find and retrieve images that satisfy the matching criteria of a query efficiently [8]. Such a system retrieves images that are stored in a collection by automatically extracting and comparing the features of these images themselves [8].

Features refer to characteristics which describe the content of an image and are mainly separated into two categories: general purpose features and domain specific features [13]. The simplest form of visual feature is derived directly from the pixel values of images. However, this primitive form of feature is very sensitive to noise and not invariant to transformations, such as translation, rotation, etc. [13]. Due to this fact, pixel values are not often used as a basis for CBIR systems. Instead, most systems make use of general purpose features like color, shape, and texture. These features are much more informative and contain sufficient discriminating power to judge whether or not images are similar [13]. Another important aspect of these general purpose features is that they are more invariant to spatial transformations and minor changes related to the lighting conditions of particular images [13]. The second type of features is high-level and deals with various application domains. These features are much harder to compare, and require more advanced and intricate implementation techniques. For example, facial CBIR systems, which make use of these high-level features, have to be able to distinguish between faces. They have to allow face recognition in order to differentiate between, for example, an image of Bill Clinton and other images of men his age. Techniques that permit such systems to exist are widely studied, however, they remain very complex and most existing systems still only make use of low-level general purpose features.

Typically, CBIR systems make use of the vector space model, and each image stored is associated with a feature vector which holds the unique characteristics of the image as described above. A quantified similarity measure is used to compare the feature vectors of two images and find their similarities. This is done under the assumption that the two images are close together in the feature space. Otherwise, if they are not close, and therefore not visually similar, they do not get compared [15]. One issue is that the distances between feature vectors are often Euclidian distances and, therefore, only provide similarity measures based on the distances of feature vectors in vector space, not on relative relevancies of those feature vectors [15]. As a result, these distance measures sometimes hurt the retrieval performance of CBIR systems, and probabilistic approaches need to be applied in order to gain retrieval accuracy.

Existing probabilistic approaches include using binary classification to classify color feature vectors as relevant or irrelevant [14], using the classical quadratic logistic regression model to set image feature vectors as relevant or not, or using a Bayesian classifier to measure the similarity degree between two images [15]. The latter approach of applying Bayesian inference, along with relevance feedback, allows CBIR systems to learn from the user what choices are relevant. It permits the system to adapt and the results to be more precise. Once the results of a query are returned, the probability of an image being marked as relevant by the user is computed with regards to the total set of images retrieved. Using such probabilistic measures allows for highly customizable metrics to calculate relevance, since the system learns directly from the user. Most systems, therefore, combine vectors with probabilities in order to become more optimized.

4. Query Methods

Unlike textual information retrieval, content-based image retrieval presents many challenges to researchers, as there exists a gap between low-level image features and high-level semantic image contents. In an attempt to reduce this gap, different techniques have been developed. One of the most commonly used methods is called relevance feedback. This method, originally created for textual information retrieval, consists of allowing the user to rate content as relevant or not. Another important technique is to use query by example, in which the user provides an example images or sketches that the system will base its search upon. Other techniques, including semantic retrieval and content comparisons have also played an important part in the research done to refine content-based image queries. These methods are often used in conjunction with one another to allow for better retrieval accuracy and will be discussed in the following sections.

Relevance Feedback

Although, in traditional document retrieval, relevance feedback has not been one of the main areas of studies, it has become very important for image retrieval. This phenomenon occurred in part due to the fact that the retrieval accuracy for the general CBIR algorithm is so low that directly applying the relevance feedback framework developed for classical textual information retrieval is able to improve the accuracy significantly [2].

There are two approaches to traditional relevance feedback: query point movement and re-weighting [2]. Query point movement tries to improve the estimate of the "ideal query point," while re-weighting tries to enhance the importance of the dimensions of a feature that helps with relevance while decreasing the one of a feature that would hinder retrieving relevant images [2]. Both of these approaches in essence modify the vector space model used for documents by replacing keywords with low-level features. The problem with that is that low-level features are not enough to take into account semantic content, since two images could be very similar in meaning but have very different low-level features (i.e. background, shape, color, etc.). In order to deal with this problem, some systems tried to incorporate a correlation matrix, while others made use of hidden annotation through learning process [3].

When using relevance feedback, many factors have to be taken into account. For example, traditional relevance feedback approaches only memorize the feedback during the time of a query session. It might be a good choice, however, to allow a system to learn continuously and have a memory. The type of learning method used is also very important. Query point movement and re-weighting are both simple learning methods; however, Bayesian learning has been shown to be more advantageous [9]. Bayesian learning and its many variations (i.e. Naïve Bayesian, Bayesian inference network, Monte-Carlo methods, etc.) consist in using probabilities to predict what users expect to have as results, and systems such as *Pic-Hunter* [3] have proven to incorporate one of these adaptive mechanisms very effectively.

Query by Example

Most current CBIR systems rely heavily on input from the user. A good CBIR system will take into account feedback and adapt accordingly to return more relevant image results to the user. This information presented to the system could be provided in several ways; however, a very effective way for the system to return precise query results is to query by example.

Query by example is a technique that requires the user to provide the CBIR system with sample images with which the system will then base its searches on. There are two variations of this process; querying the database based on a single image, or using a collection of images to base comparisons against.

A single example query uses a single image, and is therefore not as complex as using a group of images to base the query on. In this method, image features are stored in distance metrics, which provide a means of measuring similarity between two images. The problem with this is that the features are static and the weights associated to the features remain static, resulting in less precise query matches. On the other hand, there are systems which employ techniques which take as input a collection of images and allow the user to categorize images into relevant, non-relevant, or relevant groups. The system then searches through the features within the assigned groups and computes range distances which are then used to adjust the weight metric dynamically. The result is a changing weight metric which returns more precise results to the user as well as reduces the time of the match algorithm. This time reduction is accomplished by matching feature sets from a group of images rather than having a one-to-one comparison between two images. A system such as this one is shown in [4].

After submitting the image, or images, to the system, the searching and comparison algorithms are performed. These techniques vary from one CBIR system to another, but the method of providing the system with an example image, or images, is standard across systems. There are several ways for the system to require these sample images. One way to provide the system with example images is to have the user upload sample images from a personal collection. In that case, a means of uploading images to the system is provided and the uploaded images are compared against others to return other relevant images. The system can also start of by presenting a collection of initial images for the user to choose from. Another possibility is to allow the user to provide a rough sketch of the image based on shapes and blobs of colors, and using that as a basis to search the image collection [12]. This method is much more involved since sketches vary tremendously amongst people. A sample web based application is demonstrated at *http://labs.systemone.at/retrievr/*.

The query by example technique has been proven to be effective for searching through image content; however, there still exist many challenges. The main problem of query by example is the fact that, although users can choose relevant images to base their searches on, not all of the image content is indeed relevant. Take for example a query based on the selection of an image that contains the Sears Tower, the object relevant to the user is the Sears Tower, yet other parts of the image such as other buildings, backgrounds, and cars are also considered in the query. This means that the effectiveness of the query depends largely on the image submitted. There is one technique that attempts to tackle this problem and it is called noise free queries. Where noise refers to any objects, shapes, or parts of the image that the user is not interested in [10].

A noise free query model is one that allows the user to select subsections of the image by means of a frame or some type of contour. These selected objects are then used to specify spatial and scaling constraints amongst grouped objects, or it can use the contour to disassociate objects all together. These techniques are explained in [10].

Although there are different ways for approaching a query by example model, they try to accomplish the same goal, which is to compare features extracted from the sample image and return the set of matches that is the most relevant to that image based on some feature set.

The comparison of the example image against other images is done by means of feature comparisons, and this varies across CBIR systems. Some systems might attach more weight, for instance, to the shapes associated within them, as opposed to others who might give greater importance to pixel matches between images. These features are then extracted from the images and stored in feature vectors. The vectors are then used by a comparison algorithm to check for similarities amongst various images. Either way, a good CBIR system should take into account comparing images based on many features, while associating higher weight to one or more of these features.

Semantic Retrieval

Semantic retrieval is one of the most challenging query method used in CBIR systems. The ideal CBIR system would allow for a user to enter a textual query of the form “The Eiffel Tower,” and have images pertaining to the search string returned. However, since CBIR systems are in question, these types of systems would not match the query to meta-data or tags associated to the images, but would match it against the content of the image itself [12]. Therefore, when typing “The Eiffel Tower,” the system has to have a sophisticated level of intelligence in order to know what the Eiffel Tower is. This needs to be taught somehow to the system by means of query by example, or some other technique in order for it to adapt and acquire more intelligence. Furthermore, an indexing system needs to be in place to handle these types of semantic query demands. Therefore, many systems incorporate a limited amount of semantic retrieval, since it requires extensive research on the behalf of the designers in order to make it effective at all.

To illustrate the difficulty of semantic retrieval take, for example, a user that is searching for images of dogs. The search query, “dogs,” is fed to the system for querying the database. At this point, the system is expected to return images of dogs; however, not all dogs are alike; a Saint Bernard is very different from a British Bulldog. The system must have the intelligence to already have classified these two breeds into the “dogs” category for the query to be effective. If not, then some results would be omitted and recall would be lost. Furthermore, a user would have to know specific breeds of dogs in order to get more precise matches.

Other Query Methods

Although the previous approaches are the most commonly used in CBIR systems, there are other ways of querying an image database. Some of these strategies involve specifying the proportion of desired colors in the image or searching based on an object provided in a sample image. All the query methods make use of one common thing however, and that is using the features of images as a basis of comparison. Another integral part of any CBIR system is indexing the features collected from images in an efficient manner, so that searching for images in a content-driven database is efficient in terms of time.

5. Indexing

While indexing data structures for small databases is not usually necessary, applications with large image databases depend considerably on indexing techniques to improve their performance. Therefore, indexing is one of the most important aspects of a CBIR system, and much research must be done in order to come up with the most efficient indexing scheme for a particular system. An index is supposed to facilitate searching through large collections of data in an efficient manner. In a classic database management system, the values of search keys are sorted together and then used to locate a record quickly [1].

CBIR systems, consequently, must have the same goal and the ability to return results to the users in the same way as a traditional DBMS. This, however, becomes a daunting task, since indexing images is not at all similar to indexing text [1]. The approach to indexing an image collection is, therefore, automatically extracting features from the images such as its colors, shapes, and textures and storing those values in an N-dimensional feature vector. These vectors can then be indexed using similarities in the feature values.

An efficient index scheme in any CBIR system must be able to satisfy different types of queries, especially three main types. The first type, range queries, asks the system to return results contained inside or that intersect with the feature vector space. Since these regions could actually be points, if the hit is extremely accurate then these queries are also referred to as point queries [1]. The second type of query is referred to as the nearest neighbor query, which requires the system to match queries falling the closest in similarity to a vector space region. Of course, the system must be equipped with the intelligence to know what "similar" is [1]. Finally, the third type of query the CBIR system must be able to suit is called special join query, and corresponds to returning pairs of data that are similar. One example use for this type of query would be to remove redundant images from a data collection [1].

Traditionally, collections of images have been indexed by human beings by means of naming conventions and meta-data associated with images themselves. This is a laborious, involved process, requiring the system to tolerate human error and subjectivity. Furthermore, imagine indexing snapshots of screen captures taken by surveillance cameras. This would be an even more challenging task for human indexing. Many systems still try to apply document techniques for CBIR. Researchers often use a method called Tree-Based Indexing in an attempt to better index their multimedia collections [16]. Some use distance-based techniques such as SS-Trees, M-Trees, and VP-Trees, while others use KDB-Trees and R-Trees, combining the tree-based method and image features into one indexing scheme [16]. This more efficient form of indexing is similar to the one previously mentioned, which relies on automatically indexing images based on their content, and is regularly used to improve system performance. It employs multidimensional techniques to index image structures to sub databases for faster searching and retrieving and can be separated into subtypes, such as color, texture, or shape. Most systems use color and texture features, while few make use of shape or layout features [18]. Content-Based indexing, however, does come with disadvantages; namely, efficiency in the sense that, although it would be ideal to automatically index a collection based on its content, it is still not optimized enough to do this in a timely manner.

5.1. Color

Extracting color features from images and indexing them by these values is commonplace amongst CBIR systems. The reason for this is that it is a relatively inexpensive task to perform. Color moment and color-correlogram are example techniques used to characterize image colors. A color-correlogram is a 3 dimensional histogram used to characterize the color distribution of the pixel and the spatial correlation of color pairs [17].

5.2. Texture

Indexing images based on their textures is also very common practice and shown to be effective. The most prevalent texture measures are: Tamura texture features, wavelet transform, and Haralick's gray level co-occurrence features. Like many features, good texture features always have high dimensions; therefore, some systems try to reduce the images' feature dimensions prior to indexing them in order to have efficient indexing and high retrieval rate [19].

5.3. Shape

Shape features are not as commonly used as color and texture features in CBIR system implementation. However, it is possible to use it since shape features are extracted from an area object or region of an image. This technique is divided into two categories: boundary based and region based features. Boundary-based features use only the outer boundary of the shape. The most successful boundary-based features are Fourier descriptors, which uses the Fourier transform operator to get a continuous boundary as the shape feature [18]. On the other hand, region-based features use the interior of the shape region. This approach has shown to be more difficult than simple boundary based shape features, since regions inside an image are not easy to detect. It also becomes more confusing if the system using such a technique incorporates the use of relevance feedback.

5.4. Spatial Layout

Spatial layout is the technique that encodes the relative position, or an absolute position of image regions. It encodes the composition of objects, or regions in the image. For instance, consider an image of a cat sitting atop a chair. Spatial attributes can be represented in different ways, such as 2D-strings, and symbolic images. 2D-strings represent the spatial relationships of regions in horizontal and vertical directions using two strings. Symbolic images group objects by relevance and then labeled using symbols. This technique is represented as edges in a weighted graph [17]. The limitations in precise algorithms inhibit the use of indexing images according to their spatial layout.

6. Conclusion

This paper provided a brief historical overview of CBIR as well as some popular CBIR systems currently in use. Furthermore, CBIR features, such as colors or shapes were reviewed, together with probabilistic methods used to categorize these features. Query methods, such as relevance feedback and query by example, were also surveyed to finally present an overview of indexing techniques.

Content-based image retrieval systems have progressed and come a long way. Research is constantly being done in the field and many advances in indexing, sorting, and querying have been done; however, CBIR systems are still far from where they need to be. Just like traditional information retrieval which incorporates different techniques for doing things so do CBIR systems. However, the difference is that in textual retrieval there is some notion of what works and what does not, whereas, in CBIR systems, this is still to be completely determined. It is, for example, suggested that, while most current content-based image retrieval systems work with low level features, the next generation of systems should focus and operate at a higher semantic level [18].

One important thing to note is that the field of CBIR is just a subsection of multimedia based content storage and retrieval, which also involves indexing and retrieving collections of videos and interactive feeds. Video processing is still far away from being understood and implemented by a system, and there is not a fully effective query language for searching image and video databases [11]. In spite of everything, some keep on attempting to apply textual techniques to content-based retrieval and most of them do not do it efficiently. Multimedia is stored as blobs of data, which is highly inefficient and there are few large scale multimedia databases available for users and developers. As a result, the more collections of images and videos grow, the more the need is for a full featured multimedia database that would manage this ever evolving information in an optimized manner.

7. Acknowledgements

This research was supported in part by NSF grants HRD-0317692, CNS-0220562, CNS-0320956, and CNS-0426125, and NATO grant SST.NR.CLG:G980822.

8. References

[1] L. Brown and L. Gruenwald, "Tree-Based Indexes for Image Data," *Journal of Visual Communication and Image Representation*, vol. 9, issue 4, Dec. 1998, pp.300-314.

[2] G. Caenen and E.J. Pauwels "Regression Models for Relevance Feedback in CBIR," *CWI Tech Report*, June 2001.

[3] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathornas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychological Experiments," *IEEE Transactions on Image Processing*, vol. 9, issue 1, Jan. 2000, pp. 20-37.

[4] N. Hiransakolwong, K.A. Hua, S. Koompairojn, K. Vu, S.-D. Lang, "An adaptive distance computation technique for image retrieval systems," *SAC*, 2005, pp.1195-1199.

[5] http://labs.systemone.at/retrievr/

[6] E.J. Pauwels and G. Caenen, "Regression Models for Relevance Feedback and Feature Selection in Content-Based Image Retrieval," 2001.

[7] http://www.qbic.almaden.ibm.com/

[8] M.L. Shyu, S.-C. Chen, M. Chen, C. Zhang, and C.-M. Shu, "MMM: A Scholastic Mechanism for Image Database Queries," 2002.

[9] Z. Su and H. Zhang, "Relevance Feedback in CBIR," *Sixth Working Conference on Visual Database Systems: Visual and Multimedia Information Management*, vol. 216, 2002, pp.21-35.

[10] K. Vu, K.A. Hua, and N. Jiang, "Improving image Retrieval Effectiveness in Query-by-Example Environment" *2003 ACM symposium on Applied computing*, pp.774 – 781.

[11] T. Westerveld, A.P. de Vries, A. van Ballegooij, F. de Jong, and D. Hiemstra, "A Probabilistic Multimedia Retrieval Model and Its Evaluation," *Journal on Applied Signal Processing*, 2003, vol. 2, pp. 186-198.

[12] http://www.wikipedia.com

[13] C.C. Yang, "Content-Based Image Retrieval: A Comparison Between Query by Example and Image Browsing Map Approaches," *Journal of Information Science*, Mar. 2004, vol. 30, pp. 254-267.

[14] J. Peng, B. Bhanu, and S. Qing, "Learning Feature Relevance and Similarity Metrics in Image Databases," Workshop on Content-Based Access of Image and Video Libraries," 2004, pp. 14-18.

[15] R. Ksantini, D. Ziou, B. Colin, and F. Dubeau, "Logistic Regression Models for a Fast CBIR Method Based on Feature Selection," IJCAI, vol. 7, pp.2790-2796.

[16] K. Chatterjee and S. Chen, "Affinity Hybrid Tree: An Indexing Technique for Content-Based Image Retrieval in Multimedia Database", 2002.

[17] N. Chen "A Survey of Indexing and Retrieval of Multimodal Documents: Text and Images," 2006.

[18] R. Veltkamp, and M. Tanase "Content-Based Image Retrieval Systems: A Survey," 2002.

[19] M. Park, J. S. Jin, L. S. Wilson, "Hierarchical Indexing Images Using Weighted Low Dimensional Texture Features," 2002.