Use of Multiple Algorithms in Image Content Searches

J. R. Parker
Digital Media Laboratory
Department of Computer Science
University of Calgary
+1 403 220 6784
parker@cpsc.ucalgary.ca

Brad Behm
Computer Science Department
University of Calgary
2500 University Dr. N.W.
Calgary,Alberta,Canada T2N 1N4
1 403 220 6784

Abstract

Statistical image analysis is simple to implement, but rarely achieves high levels of significance in image similarity searches except on contrived image collections. Semantic object recognition is, at this time, expensive to implement, especially for the wide variety of objects that humans are interested in, but would probably give accurate and relevant results in an image retrieval system. It is proposed that simple geometric features and measure of grey level and color can be combined to form a similarity-based retrieval system that is both efficient and effective.

Keywords

image similarity, image indexing, content based search

1. Introduction

The ability to search a collection of image data for images that are similar to a specific target has been to object of a great deal of study. In spite of this, most similarity searches available to the general user are based on text searches of captions, or use manually classified images. To be able to conduct a web search for images 'that look like this one' is especially desired by students and researchers in the humanities and social sciences, who have less recourse generally to other options than do the scientists and engineers.

In March of 2003, the Digital Media Laboratory became involved in a large research project related to the creation of an educational object repository. Educational objects range in complexity from simple text objects to highly complex interactive multimedia systems, but the ability to search the repository for desired objects was essential, more so as the repository grows in size and complexity. The Media Lab has extensive experience with images in general, and computer vision in particular, and it was thought that we could create a prototype image search engine that could be used in a practical context to access objects across the internet.

How do we tell if one image, stored as a large set of RGB pixels, similar to another? We need a way to compare images against each other in an objective manner. Looking at the problem simply, it is obvious that two different images will never be identical, even if they represent the same scene - the random, or 'noise, component of each would be unlikely to agree, if nothing else were different.

Our approach was simple, but consistent with our standard mode of operation. We use a collection of methods that we know have potential to work in this context, some of which have been evaluated at some time in the recent past by more than one researcher. We then use all of these methods in an algorithm fusion mode, having each algorithm arrive at a decision and then merging the decisions into a single one. This approach has worked in the past for handprinted symbol recognition and signature verification, among other tasks. If the individual methods are simple and quick, then the overall method will be also, and the success rates will be high, higher than any of the individual success rates of the component algorithms.

2. Related Work

We have not found any previous work on the use of multiple algorithms in this context. There are, of course, a large collection of individual similarity algorithms published, especially in the past six or seven years.

Probably the best single source of information is del Bimbo [1], in which dozens of algorithms of various types are described and compared. This work groups methods into four basic types: color similarity, shape similarity, texture, and spatial relationships. Texture similarity is much like to color similarity in principle, and is usually expensive to implement, so it will not be pursued at length. Spatial relationships will also not be pursued here because it was anticipated that, again, any practical implementation would be slow.

2.1. Color

Color is a practical and effective feature that can be used for similarity searches [10,12], and there are many ways to use color. We have been interested in histogram based techniques for other applications (E.G. [8]), and were therefore intrigued by the use of color histograms described in multiple sources [3,5,13].

Of course, some images do not have color, and so it may turn out to be useful to apply the color histogram methods to simple grey levels. In addition, color images can be converted into grey scale without the loss of region or shape information, in most cases, and similarity search of these images should be possible. After all, we can watch black and white television and recognize most of the objects without difficulty.

2.2. Shape Similarity

Shape similarity methods have a very strong relationship to traditional object recognition techniques in computer vision. They generally require a *segmentation* step, the separation of a potential object region from the background. This can be quite a difficult problem, but is possibly the most important and difficult stage of processing. After segmentation, objects in the image have been distinguished from the background, and shape measurements can be applied to each object.

It will be critical that highly successful image retrieval systems in the future will use advanced segmentation methods.

3. A Multiple Algorithm System

Our multiple algorithm system will use a selection of color based and simple shape based techniques. Given a limited time frame for experimentation and implementation, it was decided to use methods that did not require a segmentation step, or that use a very fast and simple one.

We have selected five algorithms for the implementation, although other methods were tried and discarded for various reasons. For each algorithm we implemented five methods of defining regions on the image, and compared these against each other using each similarity algorithm.

Each similarity algorithm uses a simple measure of an image property. This becomes a similarity algorithm on an image by computing the measure for all images in the database and for the query image I. The image in the database having a measure m_i most like that of the query image I_i is said to be most similar to I.

The five measures we used were: grey sigma, edge density, boolean edge density, edge direction, and color histograms.

3.1. Grey Sigma

This is best described as a simple texture metric. It measures the intensity variation across a region by calculating the standard deviation of the intensity values of all of the pixels. If the image is color, then the pixels are converted into grey values using the HSI conversion [1,6] or by simply averaging the R,G, and B values.

3.2. Edge Density

This is found by first using a standard edge detector (E.G. Sobel[6]) to enhance the pixels that belong to edges and boundaries. The result is a set of pixels whose values are in proportion to their residence on al edge; pixels far from an edge are 0, those near and edge increase to a maximum value.

The edge density measure consists of the mean pixel value of the edge enhanced image.

3.3. Boolean Edge Density

This is closely related to the edge density method above. After the edge detector has been applied to the image, the image is thresholded so that what could be called edge pixels are white (1) and non-edge pixels are black. The measure returns the proportion of white (edge) pixels in the region.

3.4. Edge Direction

Some edge detectors, including the Sobel edge detector, operate over a small (3x3) image region. This allows a crude estimate of edge direction to be made. In particular, for a typical 3x3 region in an image:

$$s_y = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \qquad \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} = s_x$$

The direction associated with the pixels in the region is an estimate of the gradient:

$$\theta = \mathrm{atan}\left(\frac{S_y}{S_x}\right)$$

The edge direction metric computes an overall estimate of the direction of the edges in a region by calculating a resultant vector over all pixels, and using the direction of that resultant.

3.5. Hue and Intensity Histograms

We use a color histogram technique described in [13] which has the good sense to disregard achromatic information which is often included as noise in traditional color histogram techniques. It does this by calculating σ, the standard deviation of the R,G,B components of a color pixel and normalizing to the range [0,1].

The chrominance of a pixel is determined using the function

$$\mu(\sigma) = \begin{cases} 0 & \text{if } 0 \le \sigma < a \\ 2\left(\frac{\sigma - a}{b - a}\right)^2 & \text{if } a \le \sigma < \frac{a+b}{2} \\ 1 - 2\left(\frac{\sigma - a}{b - a}\right)^2 & \text{if } \frac{a+b}{2} \le \sigma < b \\ 1 & \text{if } b \le \sigma < 1 \end{cases}$$

where a and b are constants between 0 and 1, where a<b. In the experiments described here, a=0.05 and b = 0.8 after some empirical trials.

These values were computed and used to construct a color histogram with 16 bins. An intensity histogram was also created, having only 4 bins.

4. Regions

Past experimentation by ourselves and others (E.G. [9]) that statistical measures based on images as a whole are often less successful than the same measurements based on subdivisions of the same image. When using regions, one of the five defined measurements is made on each defined region and collected into a large set of measures. We defined five distinct ways to break up an image into regions: overall, rectangular, angular, circular, and hybrid.

4.1. Overall

This is the *null* or *trivial* region, the entire image considered as a single region. This corresponds with the usual global techniques for image analysis and recognition. This is used, for example, in [13].

4.2. Rectangular

This is a first step towards regionalization of an image, and is simple to implement because an image is rectangular, and so are the regions. For the experiments described here, the image is broken into five vertical and horizontal parts. Features are then extracted from each of the 25 regions in the grid.

For example, if the image is 250 x 500, then each region is 50x100 pixels. there is no overlap between the regions.

4.3. Angular

Angular regions [9] are wedge-shaped regions radiating from the geometric center of the image. We use an angular differential of 45 degrees, creating eight angular regions.

4.4. Circular

Circular regions (Also [9]) consist of concentric circles or rings beginning at the geometric center of the image, as nearly as possible. For the experiments here five rings were used, so that the radius of the last ring is equal to the maximum of the largest row and column index.

4.5. Hybrid

Hybrid regions (again from [9]) are a combination of angular and circular regions, as defined above. Both the concentric rings seen in circular regions and the radial segments of the angular regions are superimposed. In the experiments defined here, there were 8 angular regions and 4 circular ones, for a total of 32 regions.

5. Experiments

Our experimental database contains 782 images in 8 classes. Seven of the classes had 100 images, while the final one had only 82. The *accuracy* A for each class is calculated as $A = 100\frac{c}{qn}$, where ***c*** is the number of correct (in-class) retrievals and ***n*** is the number of images in that class [4]. The symbol ***q*** represents the number of results that were returned. This requires some explanation.

When a search engine is given a query, the resulting responses are ranked according to relevance, and are returned on a web page. The number of responses on the page is ***q***, frequently 10 or so. In the results presented here ***q***=30.

There are many ways of reporting success in this kind of enterprise. We are suggesting that success is the percentage of relevant responses on the first page of a typical query. This is certainly a measure of success that would be quickly

Table 1: Region/Feature Accuracies for some classes

	Grey Sigma	Edge Direction	Edge Density	Boolean edge density	Hue Histogram	Intensity Histogram
			Accuracy for class ‘beach’ (%)			
Overall	12.8	18.3	9.6	11.6	23.7	18.5
Rectangular	13.2	20.7	9.6	10.2	22.1	12.7
Angular	12.7	15.1	9.6	9.6	23.7	14.9
Circular	20.0	12.5	13.1	15.2	23.4	20.1
Hybrid	22.1	21.4	5.8	14.3	25.5	12.4
			Accuracy for class ‘horses’ (%)			
Overall	20.3	22.3	13.7	12.9	79.3	34.0
Rectangular	47.7	11.1	44.8	38.7	90.8	49.5
Angular	23.9	16.9	25.6	24.7	85.9	43.7
Circular	41.0	20.3	26.6	25.3	85.9	38.6
Hybrid	42.2	13.3	29.0	37.8	86.8	45.8
			Accuracy for class ‘dinosaur’ (%)			
Overall	11.2	48.0	25.1	38.6	24.1	97.0
Rectangular	41.7	49.4	54.3	69.5	51.2	100
Angular	16.6	45.0	34.3	48.0	26.4	99.6
Circular	70.9	42.7	74.3	82.1	29.2	99.1
Hybrid	53.9	57.2	0.0	22.7	57.9	98.8
			Accuracy for class ‘flower’ (%)			
Overall	11.2	14.8	23.7	17.5	38.4	49.7
Rectangular	43.8	28.9	56.5	47.7	40.9	59.8
Angular	20.2	6.7	51.1	42.8	46.9	55.7
Circular	34.5	9.7	27.3	22.6	32.6	54.8
Hybrid	53.8	26.8	73.1	69.1	44.3	61.1

understood by anyone who uses web search engines frequently. Out of the ten responses reported on the first page of a response to a query, how many of them are really a match? When asked this question of text based queries, the average person would accept 3 successes, which they think of as typical.

Given this measure of success, the overall system was initially tested on the 782 images at our disposal. Eight tables, one for each class, is necessary to convey all of the information resulting from the trials. Each image is queried against the database, and a table of success percentage with similarity algorithms occupying columns and region drawing methods as rows. Four of the tables are shown collected as Table 1, and it is plain to see that there is a significant variability in success among the classes tested. It is also plain that the method and region scheme that works best for one class does not necessarily work the best for some other.

What is wanted is a scheme that works best for all classes.

6. Algorithm Fusion

Each algorithm/region combination has been applied to each image, looking in a database of all other images excepting the one being searched. this means that a similarity value has been calculated for all images in the database. these can be sorted into a ranked list for each algorithm, in which the first image is the best (highest similarity value, most likely match). We can use these ranked lists as a means to vote for the best match. The method used to do this is called a *Borda count*[2,7].

The problem encountered when attempting to merge ranked responses is as follows: given M rankings, each having N choices, which choice has the largest degree of support? For example, consider the following 3 classifier/4 class problem [11]:

C1: a b c d   C2: c a b d   C3: b d c a

This case has no majority winner; a, b and c each get one first place vote. The *Borda count* is an ancient scheme for resolving this kind of situation, in which each alternative is given a number of points depending on where in the ranking it has been placed. A selection is given no points for placing last, one point for placing next to last, and so on, up to **R**-1 points for placing first. In other words, the number of points (the weight) given to a selection is the number of classes below it in the ranking. However, consider the following 5 classifier/3 class problem:

C1: a b c C2: a b c   C3: a b c   C4: b c a   C5: b c a

The Borda counts are **a=6**, **b=7**, **c=2**, which selects b as the winner. However, a simple majority of the first place votes would have selected **a**. This presents a conflict with the simple majority rule.

Behind the Borda count is the presumption that the second most likely classification is relatively near, in terms of likelihood or preference, to the best classification; its rank is only one away it. Consider a four-candidate vote and the result A B C D. The sum of the ranks is 6 (in general N(N-1)/2 for N candidates). Treating these as scores, A gets 3 and B gets 2; the difference (1) is 1/6 of the total, the same as the difference between B and C, and the difference between C and D. In other words, a Borda count assumes that the distance between each candidate, once sorted, is the same; a presumption of uniformity.

Other voting methods were tried (simple majority, weighted Borda, etc.) but the simple Borda count appeared to provide the most robust solution. The overall results, using this algorithm combination methods, were as follows:

beach	28.78%
horses	86.37%
dinosaur	98.97%
elephant	39.30%
flower	81.67%
architecture	40.43%
bus	58.78%
mountain	26.90%

Overall 56.95% (13343 out of 23430) beach 28.78%

This means that, in a web search having ten results per page, the first page would have 5-6 correct (directly relevant) matches to the query, on the average. This is better than our informal poll suggests is acceptable, and better than the same poll is being achieved now on text-based queries.

Table 2: Final results - Comparison Between Methods

Image Class	Correct retrievals (Rao[9]) %	Correct Retrievals (Tico[13]) %	Correct Retrievals (This work) %
Beach	27.7	25.6	28.8
Horses	89.0	68.3	86.4
Dinosaur	42.0	72.6	99.0
Elephant	20.0	24.7	39.3
Flower	46.4	51.3	81.7
Architecture	27.0	24.2	30.4
Bus	36.0	33.7	58.8
Mountain	26.0	19.9	26.9
TOTAL	39.5	40.4	57.0

7. Comparison Against the Literature

We performed a formal comparison of our multiple algorithm method for image similarity search against two published methods: that of Rao[9] and that of Tico[13]. Both methods were implemented by us using the original papers as the correct description of the method; this means that there is a chance that the program we used to generate the results is somewhat different from the one used by the original authors.

The results were computed in the same was as for the previous experiment, and are tabulated in Table 2.

8.

9. Conclusions

We have described a small collection of fast methods for determining similarity between images, and have evaluated them on a set of 782 images. These algorithms were then combined, using a rank-based voting scheme, to produce a multiple algorithm system that gives overall results that are significantly better than two of the methods found in the literature.

The prototype system is completely functional, and we hope to have it installed in a publicly assessable system within the next few months.

10. ACKNOWLEDGMENTS

We thank Michael Magee and the folks at eduSource, and the Alberta Government STEP program for funding this work.

11. REFERENCES

[1] A. del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann, San Francisco, 1999.

[2] Jean-Charles de Borda, *Memoire sur les Elections au Scrutin*, **Histoire de l'Academie Royale des Sciences**, Paris, 1781.

[3] J. Hafner, et. al., *Efficient Color Histogram Indexing for Quadratic Form Distance Functions*, **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 19(7), Pp. 729-736, 1995.

[4] H. Muller, S. Marchand-Mailler, and T. Pun, *The Truth About Corel - Evaluation in Image Retrieval*, **Proc. CIVR 2002**, Pp. 38-49, Springer-Verlag, 2002.

[5] W. Niblack, R. Barber, W. Equitz, M.Flickner, E. Glassman, D. Petkovic, P. Yanker, *The QBIC Project: Querying Images by Content Using Color, Texture, and Shape*, **SPIE 1908, Storage and Retrieval for Image and Video Databases**, Feb. 1993

[6] J. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons Ltd., New York, 1996.

[7] J.R. Parker, *Voting Methods for Multiple Autonomous Agents*, **Proc. ANZIIS'96**, Perth, Australia, 1996.

[8] Parker, J.R., *Histogram Methods For Scientific Curve Classification*, **SPIE Vision Geometry VI**, San Diego, July 28-29, 1997.

[9] A. Rao, R. K. Srihari, and Z. Zhang, *Spatial color histograms for content-based image retrieval*, **Proceedings**

of IEEE International Conference on Tools with Artificial Intelligence, pages 183--186, November 1999.

[10] T. Seidl and .H. Kriegel, *Adaptable Similarity Search In Large Image Databases*, in **Content Based Image and Video Retrieval**, R. Veltkamp, H. Burkhardt and H-P. Kriegel eds., Kluwer, Publishers, 2001.

[11] P.D. Straffin, Jr., *Topics in the Theory of Voting*, Birkhauser, Boston, 1980.

[12] M. Swain and D. Ballard, *Color Indexing*, **International Journal of Computer Vision**, Vol. 7, Pp. 11-32, 1991.

[13] M. Tico, T. Haverinen, and P. Kuosmanen. *A method of color histogram creation for image retrieval*, In **Proceedings of NORSIG**, 2000.