COLOR IMAGE RETRIEVAL USING THE DATASIEVE

Balaji K. Iyer

Naval Air Warfare Center - Weapons Division
Point Mugu, CA 93042, USA
balaji.iyer@navy.mil

Malcolm D. Macleod

QinetiQ
Malvern, WR14 3PS, UK
m.macleod@signal.qinetiq.com

ABSTRACT

With the increased use of on-line resources as well as the availability of digitized images, the need for efficient image retrieval algorithms is high. Research is focusing on systems that use the properties of the regions of an image in their retrieval process so that the search mechanism yields images that are more closely related to the query image. In this paper, we use the datasieve, a recursive nonlinear multiscale morphological operation, to give a scale-space decomposition of the input image. The datasieve uses efficient graph-based algorithms, and can typically compute an entire scale-space decomposition in less than half a second. We show that the 'seed nodes' obtained from the datasieve scale-tree provide a representative sample of the regions in the image. Image matching is then carried out by comparing the seed node features of the query image with the seed node features of the images in the database. The results presented show the method to be promising.

1. INTRODUCTION

Image retrieval systems attempt to match an input query image with a database of images, and retrieve all matching images. Most image databases lack proper annotation, and thus, a text-based search is generally not possible. Some image retrieval systems perform their search based on global properties of the image (such as a color histogram of the image) while other systems perform a partial search based on the properties of particular regions within the image [1, 2, 3]. The search process consists of collecting and storing these features of all the images in the database beforehand, and then searching this feature database to find matches to a given query image.

Based on earlier work on region segmentation [4, 5], we decided to investigate the use of the datasieve for extracting image region features. The datasieve is a recursive non-linear multiscale morphological operation which outputs a scale-space decomposition of an input signal [6, 7, 8]. Datasieves process data by their size (area) as well as by their value (e.g. grey scale intensity). They operate by processing extrema pertaining to scale. Thus, the datasieve decomposes an image into regions of various sizes. The datasieve works on efficient graph-based algorithms [6], and typically computes an entire scale-space decomposition for a 512-by-512 image in less than half a second on a 1.8 GHz Pentium IV processor.

The datasieve is based upon a chosen elementary morphological operation, which must be idempotent. Idempotency implies that after the operation is performed, further repetitions of the same operation do not modify the output, i.e. a root signal is achieved after a single pass of the operation. This means that open, close, open-close, and close-open morphological filters may be used, but not dilations or erosions alone, as they are not idempotent [7].

Thanks to Lavanya B. Iyer, Veeco Instruments Inc., for developing the image retrieval software incorporating the method described in this paper.

Among the various datasieves, three of the most commonly used, which have proven to be extremely robust and good at extracting image details, are the M-sieve, N-sieve, and R-sieve [8, 9]. In our work, we chose to use the M-sieve, defined as follows.

M-sieve: The M-sieve is a recursive series of opening operations followed by closing operations, at increasing scale. Given a function f at scale 0, and the M operator representing an opening followed by a closing, the M-sieve at scale m (denoted by M^m) is given by equation (1):

$$M^m(f) = \gamma_m(\psi_m(M^{m-1}(f))) \qquad (1)$$

where ψ_m and γ_m are opening and closing operations at scale m, and $M^0(f) \equiv f$.

Similarly, the N-sieve is a recursive series of closing operations followed by opening operations, at increasing scale, and the R-sieve is a series of recursive median filtering operations at increasing scale. In this paper, all sieve operations are performed on the luminance component, as defined in equation (2), of the color image.

A scale-tree of an image is used to represent the containment of one region within another. Such a scale-tree can be constructed efficiently as the datasieve processes the successive scales in the image.

Datasieves have been used in a wide range of applications in the signal processing field such as speech recognition [10], pattern recognition [11], and image segmentation [4, 5, 7, 12]. In [5], we showed how the datasieve-based scale-tree can be used to obtain the features corresponding to the regions in an image by means of what are referred to as 'seed nodes'. In this paper, we extend this idea further and show that the region features obtained from the seed nodes can be used for a content-based image retrieval.

2. REGION FEATURES FROM THE SEED NODE

The scale-tree constructed by processing an image with a datasieve represents the containment of regions in the image within each other. The nodes of the scale-tree represent the regions. However, for most complex images, the regions represented by this scale-tree are not necessarily the same as what a human eye considers a region [5]. In [4], a segmentation method was developed to use the datasieve scale-tree to segment a grey-scale image and yield regions that more closely correspond to the human perception of regions. Extending the use of the datasieve for color image segmentation, the concept of seed nodes was developed in [5]. This provided a set of nodes that corresponded to all regions in the image. While the definition of the seed node in [5] is good in yielding a small, but sufficient set of nodes that represent the regions in the image, extensive processing of the scale-tree is required to yield these seed nodes. To avoid having to perform this processing for each image of the database, as well as for the query image, we apply a subset of the seed node definition in [5]. We define the seed node to be all nodes of the scale-tree that have no children. This yields a larger set of seed nodes, but is a much faster approach. To reduce the number of seed nodes obtained, we first sieve the image to remove regions of smaller scales (i.e. noise). In this manner, the sieve can be used as a lowpass filter to remove regions of area less than a specified scale.

For a given database of images, the seed nodes for each image are obtained and the color features of these seed nodes are calculated. We calculate the color features of each seed node, as measured using the *Yuv* color space. Using the *Yuv* color space gives a perceptible difference from one color to another based on its chrominance and luminance. The conversion from *RGB* to *Yuv* values is done by using equations (2), (3), and (4).

$$Y = 0.3R + 0.6G + 0.1B \tag{2}$$

$$u = 0.5(B - Y) \tag{3}$$

$$v = 0.624(R - Y) \tag{4}$$

Next, we cluster the seed nodes by a K-means clustering algorithm, as was done in [5], and store the average *Yuv* values of the clusters as our feature set. This reduces the number of data points in our set.

When a query image is presented, the seed nodes for the query image can be obtained, the average *Yuv* values of the seed node clusters can be calculated, and the closest matches to the database can be found.

3. IMAGE MATCHING USING THE HAUSDORFF DISTANCE

The color features of the query image need to be compared with the color features of each image in the database to find the closest matches. When comparing sets of data (where the features of each image form one set) to find the closest matching sets, the distance should take into account the proximity of all the points of the sets. A distance measure known as the Hausdorff distance does this.

Given two sets A and B, the directed Hausdorff distance from set A to set B is denoted by $h(A, B)$. From each point in A, the Euclidean distance d to the closest point in B is found, and then the maximum of these distances is taken as defined in equation (5).

$$h(A, B) = \max_{a \in A}\{\min_{b \in B}\{d(a, b)\}\} \tag{5}$$

The directed Hausdorff distance from B to A is denoted by $h(B, A)$. From each point in B, the Euclidean distance d to the closest point in A is found, and then the maximum of these distances is taken as defined in equation (6).

$$h(B, A) = \max_{b \in B}\{\min_{a \in A}\{d(b, a)\}\} \tag{6}$$

The Hausdorff distance between sets A and B is then given by equation (7).

$$H(A, B) = \max\{h(A, B), h(B, A)\} \tag{7}$$

When using the Hausdorff distance method, outliers can easily spoil the results by greatly increasing the distance measure. To take care of outliers, we modify the Hausdorff calculation as shown in equations (8), (9), and (10).

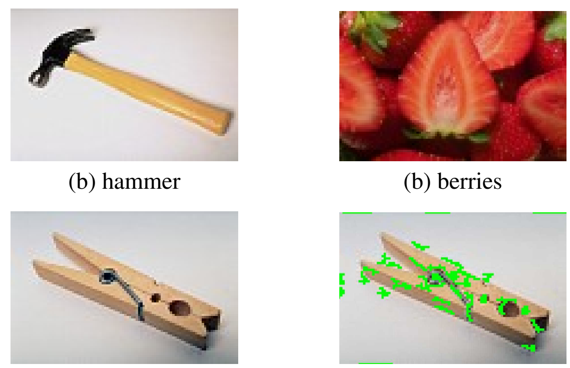
$$h(A, B) = \Sigma_{a \in A}\{\min_{b \in B}\{d(a, b)\}\} \tag{8}$$

$$h(B, A) = \Sigma_{b \in B}\{\min_{a \in A}\{d(b, a)\}\} \tag{9}$$

$$H(A, B) = h(A, B) + h(B, A) \tag{10}$$

A disadvantage of the Hausdorff distance method is its high execution time. For a set of M points in set A, and a set of N points in set B, the distance d must be calculated $2 * M * N$ times. When a sieved image yields about 1000 seed nodes, the resulting calculation is too much for a large database of images. Therefore, our procedure of sieving the image and clustering the seed nodes is beneficial in decreasing the computation time. Clustering the seed nodes can result in a reduction of data by upto a factor of 50.

To illustrate the method, we applied it to the sample images in figure 1 of a hammer, a peg, and some strawberries (courtesy Corel). The hammer is black and wooden-colored, with a light background. The peg is very similar in its color properties. The strawberry is primarily red and green, with some black coloring in-between the strawberry pieces. The images are first sieve filtered to scale 10 to remove regions of area less than 10 pixels.



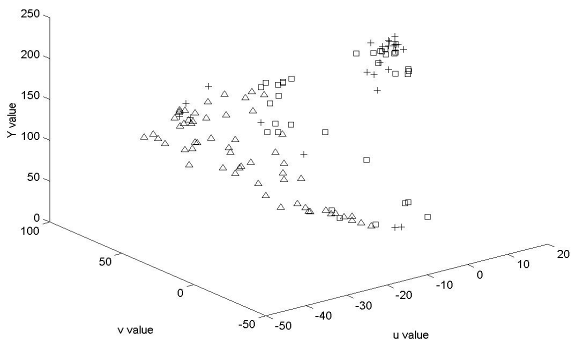
(b) hammer (b) berries

(c) peg (d) peg seed nodes

Fig. 1. Color images of hammer (wood/black), peg (wood/black), and strawberries (red/green) images, and seed nodes of peg.

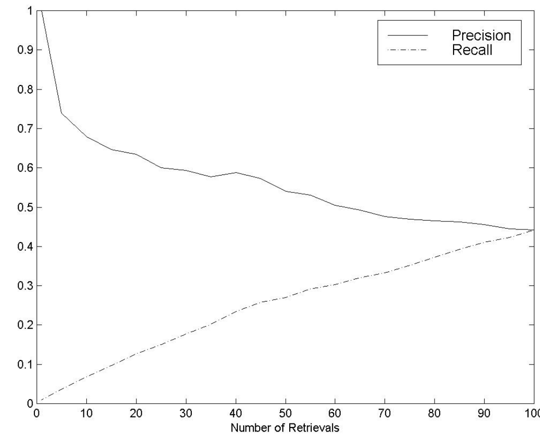
The datasieve operation was then run on the scale-10 filtered images and the seed nodes were obtained from the resulting scale-tree. Figure 1 shows the seed nodes of the peg image. Various regions of the peg image such as the background, the surface of the peg, and the darker shaded regions of the peg are identified by these seed nodes. The seed node locations of each of these three images in the *Yuv* color space are shown in figure 2. From the distribution of these seed nodes, it can be seen that the seed nodes of the hammer and peg images are in close proximity to each other (due to their similar color characteristics). The seed nodes of the strawberry image, however, mostly occupy a different region of the color space, due to the red-green color properties of that image, although there are of course overlapping seed nodes pertaining to the black regions of all three images Also, some outliers do exist.

Nevertheless, the Hausdorff distance between the sets of data (i.e. the average *Yuv* values of the clustered seed nodes) pertaining to the hammer and peg images indicates a closer match than the



Seed node locations

Fig. 2. Seed node locations in *Yuv* color space, after sieving to scale 10, of hammer (denoted by $+$ sign), peg (denoted by $\square$ sign), and strawberries (denoted by $\triangle$ sign) images.



Precision and Recall

Fig. 3. Retrieval precision and recall averaged over ten queries (one from each image category).

larger Hausdorff distances between the sets of data pertaining to the hammer and the strawberry images or the peg and the strawberry images.

4. RESULTS

This method was applied to the *Simplicity* [2] test database of 1000 images (from Corel). These images fall into ten general categories of 100 images each: horses, flowers, elephants, buses, beaches, mountains and glaciers, food, buildings, dinosaurs, and Africa people and villages. The discussed method yields good results as illustrated by figure 4. This figure shows the six closest matches for the horse, elephant, and flower queries. The first image in each set is the query image (as well as the best match).

An evaluation of retrieval accuracy can be measured by the precision and recall as defined in equations (11) and (12) where K is the number of retrievals, C_K is the number of relevant matches among all the K retrievals, and M is the total number of relevant matches in the database [3].

$$\text{Precision}(K) = C_K/K \tag{11}$$

$$\text{Recall}(K) = C_K/M \tag{12}$$

Figure 3 shows the retrieval precision and recall averaged over ten queries (one from each image category). When compared to the method of [2], which was also evaluated on the *Simplicity* database, the performance of our method is equally good. In [2], the evaluation was carried out using every single image of the database as a query image, giving results which are therefore averages over one hundred query images for each category. The average precision and recall over all ten image categories were about 47%. Our method yields corresponding average precision and recall values of about 44%.

Because the retrieval method we have described is purely color-based, images of other categories with similar colors may be retrieved. For example, the query using the elephant image in figure 4 shows an example of an image of a different category (buildings), but with similar colors, being retrieved.

5. DISCUSSION AND CONCLUSIONS

We have shown that the color information contained in the seed nodes obtained from the datasieve-based scale-tree can be used to develop a good content-based image retrieval algorithm. This method yields good results for retrieving images purely based on the color properties of the regions of the image. A way in which the method described above might be enhanced is to use the areas of the regions formed by clustering the seed nodes, because there is a significant range of region sizes (areas) which results. For example, the ratio of areas occupied by different colors might be useful [3]. This remains as future work.

Many image retrieval algorithms work on a database that is indexed (i.e. pre-sorted based on the feature vectors), so that the retrieval times are fast - generally within a second. Other retrieval algorithms perform the search based on a single selected region [3] or a selected set of regions [1]. Our method matches based on all regions of the image. Furthermore, our method requires very minimal pre-processing of the database images, since only the seed nodes need to be calculated and clustered. It might therefore be useful for system cross-cueing applications, where images coming across a system need to be rapidly analyzed to locate matching/non-matching properties. The Hausdorff measure we use provides robustness, but is computationally intensive, requiring approximately twice the computation of a quadratic-type distance measure such as that in [3]. Nevertheless, the method in its current form works well for smaller databases, yielding results in less than 10 seconds (for a database of about a thousand images) using a C++ implementation running on a Pentium IV, 1.8 GHz processor. Developing a faster version for larger databases, while retaining the benefits of the method, is a subject for future work.

6. REFERENCES

[1] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik, "Blobworld: Image segmentation using expectation-

Fig. 4. Image retrieval results for query horse, elephant, and flower color images (image size 256-by-384 pixels). First image is the query image as well as the closest match.

maximization and its application to image querying," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, August 2002.

[2] James Z. Wang, Jia Li, and Gio Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, September 2001.

[3] Yining Deng and B. S. Manjunath, "An efficient low-dimensional color indexing scheme for region-based image retrieval," 1999, vol. 6, pp. 3017–3020.

[4] B. Iyer and M.D. Macleod, "Region segmentation based on a datasieve scale-tree region growing," in *Proceedings JCIS 2000*, 2000, vol. 2, pp. 313–316.

[5] Balaji K. Iyer and Malcolm D. Macleod, "Multiscale region segmentation of images using nonlinear methods," in *Visual Communications and Image Processing*, 2000, vol. 4067, pp. 1116–1125.

[6] J.A. Bangham, R.W. Harvey, P.D. Ling, and R.V. Aldridge, "Morphological scale-space preserving transforms in many dimensions," *Journal of Electronic Imaging*, vol. 5, no. 3, pp. 283–299, 1996.

[7] Javier Ruiz Hidalgo, "The representation of images using scale trees," M.S. thesis, School of Information Systems, University of East Anglia, 1998.

[8] J. Andrew Bangham, Javier Ruiz Hidalgo, and Richard Harvey, "Robust morphological scale-space trees," 1998, pp. 133–139.

[9] J.A. Bangham, P. Ling, and R. Young, "Multiscale recursive medians, scale-space and sieve transforms with an inverse," *IEEE Transactions in Image Processing*, vol. 5, pp. 1043–1047, 1996.

[10] Iain Mathews, J. Andrew Bangham, Richard Harvey, and Stephen Cox, "Nonlinear scale decomposition based features for visual speech recognition," 1998, pp. 303–306.

[11] C.J. Pye and J.A. Bangham, "2D pattern recognition using 1D sieves based on alternating sequential, root median, and self-dual filters," in *Signal Processing VII*, 1994, pp. 856–859.

[12] J. Andrew Bangham and Javier Ruiz Hidalgo, "The segmentation of images using scale-space trees," *Proceedings British Machine Vision Conference - BMVC*, vol. 1, pp. 33–43, 1998.