Classifying Images by Canny Mask Segmentation and Laplacian Sigmas in Content Based Image Retrieval System

Ensaf ALZurqa
Dept. of IT, Faculty of Engineering and IT, Taiz University, Taiz, Yemen
desaflzurqa@gmail.com

Abstract - Content-based image retrieval draws many of its methods from the field of image processing and is regarded by some as a subset of that field. It differs from these fields principally through its emphasis on the retrieval of images with desired characteristics from a collection of significant size. "Content-based" means that the search will analyze the actual contents of the image. The term 'content' in this context might refer colors, shapes, textures, or any other information that can be derived from the image itself. Without the ability to examine image content, searches must rely on metadata such as captions or keywords, which may be laborious or expensive to produce. In this paper, we work on specific "eggs" by using different laplacian sigmas and canny mask, to retrieve the images from the data base that are related to the original one. The main objectives of our work are classifying an image as belonging to a specific category and retrieving images that are similar to database by showing statistical results of the different sigmas values in our CBIR system.

Keywords: Canny Masks, Laplacian Sigmas, Content Based Image Retrieval System, Segmentation Analysis, Image Enhancement and Adjustment .

1. Introduction

The earliest use of the term content-based image retrieval in the literature seems to have been by Kato [1992], to describe his experiments into automatic retrieval of images from a database by colour and shape feature. The term has since been widely used to describe the process of retrieving desired images from a large collection on the basis of features (such as colour, texture and shape) that can be automatically extracted from the images themselves. The features used for retrieval can be either primitive or semantic, but the extraction process must be predominantly automatic. Retrieval of images by manually-assigned keywords is definitely not CBIR as the term is generally understood even if the keywords describe image content.

CBIR operates on a totally different principle from keyword indexing. Primitive features characterizing image content, such as colour, texture, and shape, are computed for both stored and query images, and used to identify (say) the 20 stored images most closely matching the query. Semantic features such as the type of object present in the image are harder to extract, though this remains an active research topic. Video retrieval is a topic of increasing importance – here, CBIR techniques are also used to break up long videos into individual shots, extract still keyframes summarizing the content of each shot, and search for video clips containing specified types of movement. Three commercial CBIR systems are now available – IBM’s QBIC, Virage’s VIR Image Engine, and Excalibur’s Image RetrievalWare. In addition, demonstration versions of numerous experimental systems can be viewed on the Web, including MIT’s Photobook, Columbia University’s WebSEEk, and Carnegie-Mellon University’s Informedia. CBIR systems are beginning to find a foothold in the marketplace; prime application areas include crime prevention (fingerprint and face recognition), intellectual property (trademark registration), journalism and advertising (video asset management) and Web searching. Both the Alta Vista and Yahoo! Search engines now have CBIR facilities, courtesy of Virage and Excalibur respectively. The effectiveness of all current CBIR systems is inherently limited by the fact that they can operate only at the primitive feature level. None of them can search effectively for, say, a photo of a dog – though some semantic queries can be handled by specifying them in terms of primitives. A beach scene, for example, can be retrieved by specifying large areas of blue at the top of the image, and yellow at the bottom. There is evidence that combining primitive image features with text keywords or hyperlinks can overcome some of these problems, though little is known about how such features can best be combined for retrieval.

In this paper, we work on different images' content in; to retrieve the images from the data base that are related to the original one. The main objective is retrieving images that are similar in shape, texture and/or color to an image, that is done using different Laplacian sigmas and canny mask segmentation. We will discuss in details
which technique is the best for our application, the deciding factor is applying the next steps until segmentation. The processing goes as we illustrate in following sections. Before processing the images, it needs to be enhanced in order to brighten eggs’ color, also images captured contains some noise that need to be removed or at least reduced for further correct processing on the image. After the image was enhanced and filtered, each egg must be extracted separately for further processing, so the next step here is detecting edges of eggs in each image, once the edges are detected from background, eggs are segmented (separated from one another) to make them ready for further processing. Extracting feature vectors needs image adjustments and preprocessing procedures. So, the rest of the paper is organized as follows: In Section 2, related works to CBIR are introduced. The preprocessing procedure (enhancement procedures) steps are illustrated in Section 3. In Section 4, image adjustment functions are presented as parts of feature extraction step. We present our segmentation work by canny mask, in section 5. In Section 6, we introduce the evaluation results for our work. Finally, in Section 7 concluded and future work is highlighted.

2. Related work

Research and development issues in CBIR cover a range of topics, many shared with mainstream image processing and information retrieval [1-8]. Some of the most important are: understanding image users’ needs and information-seeking behaviour, identification of suitable ways of describing image content, extracting such features from raw images, providing compact storage for large image databases, matching query and stored images in a way that reflects human similarity judgements and efficiently accessing stored images by content, providing usable human interfaces to CBIR systems Key research issues in video retrieval include: automatic shot and scene detection, ways of combining video, text and sound for retrieval amid effective presentation of search output for the user. Enser [1995] reviews methods for providing subject access to pictorial data, developing a four-category framework to classify different approaches. He discusses the strengths and limitations both of conventional methods based on linguistic cues for both indexing and search, and experimental systems using visual cues for one or both of these. His conclusions are that, while there are serious limitations in current text-based techniques for subject access to image data, significant research advances will be needed before visually-based methods are adequate for this task. He also notes, as does Cawkell [1993] in an earlier study, that more dialogue between researchers into image analysis and information retrieval is needed. Aigrain et al [1996] discuss the main principles of automatic image similarity matching for database retrieval, emphasizing the difficulty of expressing this in terms of automatically generated features. They review a selection of current techniques for both still image retrieval and video data management, including video parsing, shot detection, keyframe extraction and video skimming. They conclude that the field is expanding rapidly, but that many major research challenges remain, including the difficulty of expressing semantic information in terms of primitive image features, and the need for significantly improved user interfaces. CBIR techniques are likely to be of most use in restricted subject domains, and where synergies with other types of data (particularly text and speech) can be exploited. Eakins [1996] proposes a framework for image retrieval, classifying image queries into a series of levels, and discussing the extent to which advances in technology are likely to meet users’ needs at each level. His conclusion is that automatic CBIR techniques can already address many of users’ requirements at level 1, and will be capable of making a significant contribution at level 2 if current research ideas can be successfully exploited. They are however most unlikely to make any impact at level 3 in the foreseeable future. Idiris and Panchanathan [1997a] provide an in-depth review of CBIR technology, explaining the principles behind techniques for colour, texture, shape and spatial indexing and retrieval in some detail. They also discuss the issues involved in video segmentation, motion detection and retrieval techniques for compressed images. De Marsici et al [1997] also review current CBIR technology, providing a useful feature-by-feature comparison of 20 experimental and commercial systems. In addition to these reviews of the literature, a survey of “non-text information retrieval” was carried out in 1995 on behalf of the European Commission by staff from GMD (Gesellschaft für Mathematik und Datenverarbeitung), Darmstadt and Université Joseph Fourier de Grenoble [Berrut et al, 1995]. This reviewed current indexing practice in a number of European image, video and sound archives, surveyed the current research literature, and assessed the likely future impact of recent research and development on electronic publishing. The survey found that all current operational image archives used text-based indexing methods, which were perceived to have a number of shortcomings. In particular, indexing vocabularies were not felt to be adequate for non-text material. Despite this, users seemed generally satisfied with existing systems. The report concluded that standard information retrieval techniques were appropriate for managing collections of non-text data, though the adoption of intelligent text retrieval techniques such as the inference-based methods developed in the INQUERY project [Turtle and Croft, 1991] could be beneficial. In section 3, the first step of preprocessing procedures that are used in our works to enhance the images is illustrated.
3. Enhancement Procedures

The preprocessing procedure steps, that are used before the image analysis stage, are: Image Enhancement, Image adjustment, Noise Removal and Segmentation.

Image Enhancement has many enhancement techniques, so we have put under study six different enhancement techniques. We will discuss in details which technique is the best for our application in such that the deciding factor is applying on the other stages until segmentation.

The cumulative histogram technique is one of enhancement techniques that is not suitable for our images since when applying the next steps, the final results were not sharp enough and contained a lot of noise as shown in figure (1).

Applying the linear point operation to increase the contrast of the images gave us very good results in the enlarged images (zoomed x400). Different values of $\alpha >1$ were tried to increase the contrast level. the optimal value of $\alpha$ is 1.9 which was obtained by trail and error. The higher values caused very high contrast that spoiled the image as shown in figure (2).

But the nonlinear technique gave better results to the images zoomed x100 than the linear technique. The second type of image was more advantageous than the first type. The optimum values for $\alpha$’s were found by trial and error to be 0.9 and 0.7 for the first and second techniques respectively. Figures (3) and (4) show the results of the nonlinear techniques.

When we test Rescaling technique on images, we find that it makes the mean of the image equals 0 and it’s variance equal 1. Results of this technique are show in figure (5).

To correct the non-uniform illumination in the preprocessed images, we first find a coarse estimate of the background illumination by determining the minimum of each 33-by-32 block in the image. A surf plot of the coarse estimate is shown In Figure (6).

The coarse estimate is then expanded in size so that it is the same size as the original image. A bicubic interpolation is used to ensure that the data is smooth. Then computed illumination is then subtracted from the original image to correct nonuniformity. Figure (7) shows the final output of this technique.

After finishing image enhancement, we adjust the image. In our work, we used an adjustment function that is considered the second step in preprocessing procedure. This adjustment function is explained in the section 4.

4. Image Adjustment Function

In our work, we used an adjustment function to rescale the image so that it covers the entire dynamic range ([0,1]). It gave good results except when it was used after the rescaling technique discussed previously. The adjusted images are shown in figure (8).
In our sample images, we found some images that have noise, which causes some problems in the programming. These problems were:
- This noise could be detected as a part of an egg
- It may connect two or more eggs and this leads to detect them as one egg
- Noise may be detected as an egg (this problem is solved in the last section)

To remove these types of noise, as an important step in preprocessing procedure, we used the median filter. The median filter is an effective filter, when the noise pattern consists of strong, spike-like components and in the characteristic to be preserved is edge sharpness. We are using a filter of size 3 X 3, which gave us the best results by trial and error. The result is shown in figure (9).

As we mentioned in section 3, that the segmentation is one of the preprocessing steps which are used in image analysis stages. So in section 5, we introduce the segmentation analysis problems and then in subsection 5.1, we explain canny mask segmentation that is used in our work.

5. Segmentation analysis

To segment objects from each other, First we scan the output image from the previous step to get the starting point, we add this point to the position matrix (this matrix contains the positions of the contour of each object), then we loop to find the remaining points of the contour. We take the first position we find without checking whether the other positions are valid or not. Every time we find a position we check whether it is already in the position matrix or not, if it is we stop to avoid entering an endless loop. Then we use the matrix of positions to get the segmented object from the original image by calculating the width and height of the object then putting it in a matrix with these dimensions, and then clearing it from the original image.

This primary technique caused two problems:
- The first problem was that the next position to be tested is out of the range of the image, i.e. the object to be segmented is at the border of the image. To solve this problem we pad the image with a white border before scanning it.
- The second problem appears in the following figure (16)

Figure (16): chopped parasite egg

This problem occurred because a position was reached that was already in the position matrix, but it was not the starting point of the contour this caused the search for more positions to stop causing figure (17) to be segmented as a separate object.

Figure (17): a part of an egg detected as an egg

To solve this problem, we put a condition that if the position under test is in the position matrix, we check if it is the first position (i.e. the staring point) then we stop, otherwise we check the other positions. This solution caused entering an endless loop, when the position under test is in the position matrix but it is not the starting point, as figure (18) shows.

Figure (18): noise that caused the program to enter into an endless loop

So we save the position found in some variables but not in the position matrix again, if no other new position is found we go back to the position saved and search around it for a new position. But sometimes there is more than one position around the position under test all of them are already in the position matrix, this caused the problem entering an endless loop in an object such as the one in figure (19). The program kept looping around the second egg because of the pixel that joins between them.

Figure (19): two eggs joint together with one pixel
To solve this, we use a function that returns the best position from the positions surrounding the position under test when all of them are already in the position matrix. Best position meaning a position has a valid next position around it. If no best position if found we use the last one found and try around it, this sometimes causes entering an endless loop, to avoid this we defined a maximum number of repetitions if exceeded we break using only the positions saved to define the new object segmented.

We detect noise by two consecutive ways:
- First we check the size of the segmented object by comparing it to the maximum and minimum expected values of the height and width of the parasite, if it is within range we assume that it is a valid egg of a parasite. Figure (20) shows detected objects that are out of range.

- But sometimes some matrices containing noise are within range, so we pad the matrix to the maximum expected values of height and width, then we calculate the ratio of colour that is not white to all colours including the white background. If this ratio is less than some threshold we assume that this matrix contains noise.

![Figure (20): two examples of noise](image)

![Figure (21): noise detected as an egg](image)

Figure (21) shows a detected object that is within range but the ratio is less than the threshold value.

After noise removal, we need to perform the last step in preprocessing phase ‘segmentation’ for two reasons:
- First we need to separate the eggs from the background.
- In images containing more than one egg, we need to get each egg alone.

The image used in detecting objects from background is the output image from the previous steps, in these images the objects to be segmented from the background differ greatly in contrast from the background, changes in the background can be detected by operators that calculate the gradient of an image. In the following subsection 5.1, we illustrate why we choose canny mask in our segmentation work and then explained the mask itself.

5.1 Canny Mask Segmentation

We have several ways to sement the objects from background and calculate the gradient of an image: The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. In the Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. To find edges by looking for zero crossings after filtering the image, we use a Laplacian of Gaussian filter. The zero-cross method finds edges by looking for zero crossings after filtering the image with a filter we specify. And finally, the Canny method finds edges by looking for local maxima of the gradient of the image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be "fooled" by noise, and more likely to detect true weak edges.

These techniques were tried on different images. We concluded that applying the sobel mask on images enhanced using the linear technique gave the best results in images zoomed x400, while applying the canny mask after the correction of non-uniform illumination technique gave best results on images zoomed x100. Figure (11) shows the output of the two masks.

![Figure (11): a) output of the canny mask b) output of the sobel mask](image)

The binary gradient mask shows the lines of high contrast in the image, these lines do not quite delineate the outline of the object of interest, compared to the original image the appears gaps in the lines surrounding the objects in the gradient mask, these gaps will disappear if the output image is dilated using linear structuring elements.

The binary gradient mask is dilated using the vertical structuring element (S2) followed by the horizontal structuring element (S3), the output is shown in figure (12).

![Figure (12): the outputs of the two masks after dilation](image)
The dilated gradient mask show the outline of the eggs quite nicely, but there are still holes in the interiors of the eggs, to fill these holes we use a filling function, the output is shown in figure (13). Then we smooth the object by eroding the image twice with the diamond structuring element (S4), the output is shown in figure (14). After smoothing objects, we use the original image to get the colours of the segmented object as shown in figure (15).

After we show how canny mask work and why it is the best for our work, we test different Laplacian sigmas and so the statistical results of using them.

In section 6, we will show some statistical results of the different techniques we used in the different stages of our system. We tested five different Laplacian sigma. So we can choose any value of sigma.

### 6.2 Hu-moment Sigma

The following results occurs: We found that the best value of sigma that successfully classifies all parasite types is 0.1.

### 6.3 Flusser-moment Sigma

The following results occurs: We found that the best value of sigma that successfully classifies all parasite types is 0.1.

### 6.4 Co-occurrence Sigma

The following results occurs: We found that the worst value of sigma that successfully classifies all parasite types is 0.1.
6.5 Histogram Sigma

The following results occur: From all the previous results, so we conclude that the combining between the Sobel mask and the linear technique gives the best results, those results were also sharp and the eggs were not distorted, it helped in very good segmentation.

| Table 5 |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Monet   | 0.1     | 0.2     | 0.3     | 0.4     | 0.5     | 0.6     | 0.7     | 0.8     | 0.9     |
|         | %       | %       | %       | %       | %       | %       | %       | %       | %       |
| Parmisano | %       | 96.2    | 97.4    | 97.8    | 98.0    | 97.5    | 96.6    | 97.9    | 98.8    |

7. Conclusion

Content-based image retrieval (CBIR), is the application of computer vision to the image retrieval problem, that is, the problem of searching for digital images in large databases. CBIR draws many of its methods from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field. It differs from the field of image processing and computer vision, and is regarded by some as a subset of that field.

The extent to which CBIR technology is currently in routine use is clearly has great impact on the more general applications of image searching. However, the main drawback of current CBIR systems is more fundamental. It is that the only retrieval cues they can exploit are primitive features. Hence current CBIR systems are likely to be of significant use only for applications at level 1.

Retrieval does not always yield images that have clearly the same feature, unless the database contains many images with a dominant one. Searching by laplacian sigmas and using canny mask give often suprising results. Apparently the features used for matching are not the most effective ones.

There are the notions of precision (the ratio of relevant images to the total number of images retrieved) and recall (the percentage of relevant images among all possible relevant images) as we see in our work. We conclude from all the previous results for ratio of relevant images, that the combining between the cumulative histogram technique, the canny mask and different sigmas give the best results, but these results were based on the noise ratio.

References


