Efficient Image Retrieval Based On Texture Features Using Concept of Histogram

Neelima Sadinen

Received: 13 April 2012 Accepted: 1 May 2012 Published: 15 May 2012

Abstract

Image retrieval is fast growing research oriented area now days. As information retrieval plays a major role in transmitting knowledge both in the forms of text and images, image retrieval got a major focus. In this paper we integrated the Histogram Intersection measure method to compare the query image with database images; by this approach we can measure over-all similarity between images, by incorporating all local properties of the texture histograms of the images through which we proved that our approach in retrieving the image is accurate.

Index terms— Histogram, image, texture, database, retrieval

o date, image and video storage and retrieval systems have typically relied on human supplied textual annotations to enable indexing and searches. The text-based indexes for large image and video archives are time consuming to create. They necessitate that each image and video scene is analyzed manually by a domain expert so the contents can be described textually. The language-based descriptions, however, can never capture the visual content sufficiently. For example, a description of the overall semantic content of an image does not include an enumeration of all the objects and their characteristics, which may be of interest later. A content mismatch occurs when the information that the domain expert ascertains from an image differs from the information that the user is interested in. A content mismatch is catastrophic in the sense that little can be done to approximate or recover the omitted annotations. In addition, a language mismatch can occur when the user and the domain expert use different languages or phrases. Because text-based matching provides only hit-or-miss type searching, when the user does not specify the right keywords the desired images are unreachable without examining the entire collection. The prime requirement for Retrieval systems is to be able to display images relating to a named query image. The text indexing is often limited, tedious and subjective for describing image content. So there is increasing interest in the use of CBIR techniques. The problems with text-based access to images have prompted increasing interest in the development of image based solutions. This is more often referred to as Content Based Image Retrieval (CBIR) as shown in Fig. 1. Content Based Image Retrieval relies on the characterization of primitive features such as color, shape and texture that can be automatically extracted from the images themselves. Queries to CBIR system are most often expressed as visual exemplars of the type of the image or image attributed being sought. For Example user may submit a sketch, click on the texture pallet, or select a particular shape of interest. This system then identifies those stored images with a high degree of similarity to the requested feature.

Digital imaging has become the standard for all image acquisition devices. So there is an increasing need for data storage and retrieval. With lakhs of images added to the image database, not many images are annotated with proper description. So many relevant images go unmatched. The most widely accepted content-based image retrieval techniques cannot address the problems with all images, which are highly specialized .Our approach Histogram based Image Retrieval using Texture Feature retrieves the relevant images based on the texture property. We also provide an interface where the user can give a query image as an input. The texture feature is automatically extracted from the query image and is compared to the images in the database retrieving the matching images.

The goal of Content-Based Image Retrieval (CBIR) systems is to operate on collections of images and, in response to visual queries, extract relevant image. The application potential of CBIR for fast and effective image retrieval is enormous, expanding the use of computer technology to a management tool. CBIR operates on the principle of retrieving stored images from a collection by comparing features automatically extracted from the images themselves. The commonest features used are mathematical measures of color, texture or shape. A typical system allows users to formulate queries by submitting an example of the type of image being sought, though some offer alternatives such as selection from a palette or sketch input. The system then identifies those stored images whose feature values match those of the query most closely, and displays thumbnails of these images on the screen. Several methods for retrieving images on the basis of color similarity have been described in the literature, but most are variations on the same basic idea. Each image added to the collection is analyzed to compute a color histogram, which shows the proportion of pixels of each color within the image. The color histogram for each image is then stores in then stored in the database. At each time, the user can either specify the desired proportion of each color (&75% olive green and 25% red, for example), or submit an example image from which a color histogram is calculated. Either way, the matching process then retrieves those, which a color histogram is calculated. Either way, the matching process then retrieves those images whose color histograms match those of the query most closely.

1 b) Texture Retrieval

The ability to match on texture similarity can often be useful in distinguishing between areas of images with similar color (such as blue sky and sea or green leaves and grass). A variety of techniques has been used for measuring texture similarity; the best established rely on comparing values of what are known as second-order statistics calculated from query and stored images. Essentially, these calculate the relative brightness of selected pairs of pixels from each image. From these it is possible to calculate measures of image texture such as the degree of contrast, coarseness, directionality and regularity or periodically, directionality and randomness.

Texture queries can be formulated in a similar manner to color queries, by selecting examples of desires texture a palette, or by supplying an example query image. The system then retrieves images with texture measures most similar in value to the query. c) Shape Retrieval Two major steps are involves in shape feature extraction. They are object segmentation and shape representation.

Object segmentation: Segmentation is very important to Image Retrieval. Both the shape feature and the layout feature depend on good segmentation allow fast and efficient searching for information of a user's need.

2 Shape Representation

In image retrieval, depending on the applications, some requires the shape representation to be invariant to translation, rotation, and scaling. In general, the shape representations can be divided into two categories, boundary-based and region-based. The former uses only the outer boundary of the shape while the latter uses the entire shape region.

Texture is one of the crucial primitives in human vision and texture features have been used to identify contents of images. Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity. Texture contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. One crucial distinction between color and texture features is that color is a point, or pixel, property, whereas texture is a localneighbourhood property. As a result, it does not make sense to discuss the texture content at pixel level without considering the neighbourhood.

The texture is a property inherent to the surface. Various parameters or textural characteristics describe it. They are:

The Granularity which can be rough or fine The Evenness which can be more or less good The Linearity The directivity The repetitiveness The contrast The order The connectivity The other characteristics like color, size, and shape also must be considered. The Methodologies used for analysis of the texture are as follows a) Texture Spectrum Method

The basic concept of texture spectrum method was introduced by H1 and Wang. The texture can be extracted from the neighborhood of 3 X 3 window which constitute the smallest unit called 'texture unit'. The neighborhood of 3 X 3 consists of nine elements respectively as V={ V1 , V2 , V3 , V4 , V0 ,V5 , V6 ,V7 , V8 } where V0 is the central pixel value and V1?.V8 are the values of neighboring pixels within the window. The corresponding texture unit for this window is then a set containing eight elements surrounding the central pixel, represented as: TU = { E1 , E2 , E3 , E4 , E0 ,E5 , E6 ,E7 , E8 } Where Ei is defined as: Ei = 0 if Vi < V0 1 if Vi =V0 2 if Vi >V0 And the element E1 occupies the corresponding V1 pixel. Since each of the eight element of the texture units has any one of three values (0, 1, or 2)NTU = ? Ei * 3 (i -1) [For i=1 to 8]

Where NTU is the texture unit value. The occurrence distribution of texture unit is called the texture spectrum (TS). Each unit represents the local texture information of 3X3 pixels, and hence statistics of all the texture units in an image represent the complete texture aspect of entire image. b) Cross Diagonal Texture Spectrum AL-Jan obi (2001) has proposed a crossdiagonal texture matrix technique. In this method the eight neighboring pixels of 3 X 3 widows is broken up into two groups of four elements each at cross and diagonal positions. These groups are named as Cross Texture Unit (CTU) and Diagonal Texture Unit (DTU) respectively. Each of the four elements of these units is assigned a value (0, 1, and 2) depending on the gray level difference of the corresponding pixel with that of the central pixel of 3X3 window. These texture units have values from 0 to 80 (34, i.e 81 possible values).

Cross Texture Unit (CTU) and Diagonal Texture Unit (DTU) can be defined as:

3 March

Formation of cross diagonal texture units d) Texture Spectrum with Thershold

The texture spectrum method with threshold is intended to make difference between the values of neighborhood matrix which are very close to the cental pixel value and those the rest.In this method the texture unit matix is represented as: TU = { E1 , E2 , E3 , E4 , E0 ,E5 , E6 ,E7 , E8 } Where Ei is defined as:Ei = 0 if Vi < =(V0 + t)1 if Vi > (V0 + t)

Where t is the threshold value. NTU = ? Ei * 2 (i -1) [For i=1 to 8] The texture unit value can range between (0-254).

4 e) Reduced Texture Unit

In this method the range of texture unit values are (0,1).As the range is decreased the memory required to compute texture unit value also reduces.In this method TU = { E1 , E2 , E3 , E4 , E0 ,E5 , E6 ,E7 , E8 } Where Ei is defined as: Ei = 0 if Vi < = V0 1 if Vi > V0 Where t is the threshold value.

N RTU = ? Ei * 2 (i -1) [For i=1 to 8] The texture unit value can range between (0-254).

5 f) Splitting Texture Unit Matrix into Rows and Columns

In this approach the texture unit matrix is split into 3 separate rows/columns. Texture unit value is calculated separately for each row/column. Later all the 3 texture unit values are added to get a single texture unit value. By doing this the texture unit value can be limited to 42.Thus memory and computation time can be saved. The texure unit value is calculated separately for each texture unit matrix (j) as:N TUj = ? Eji * 2 (i -1) [For i=1 to 3]

The final texture unit value is evaluated as:N TU = ? N TUj [For j=1 to 3]

The texture unit value can range between (0-42).

To overcome the disadvantages of Euclidean distance we taken histogram intersection measure. The histogram intersection was investigated for color image retrieval by swain and Ballard. Their objective was to find known objects within images using color histograms. When the object (q) size is less than the image (t) size, and the histograms are not normalized, then |hq|<=|ht|. The intersection of histograms hq and ht is given by: Where |h| = h[m] [for m=0 to M-1].The above equation is not a valid distance metric since it is not symmetric hq,t not equal to dt,q. However that equation can be modified to produce a true distance metric by making it symmetric in hq and ht as follows:

Alternatively when the histograms are normalized such that |hq|=|ht|, both equations are true distance metrics.When |hq|=|ht| that D1(q,t)=dq,t and the Histogram Intersection is given by Class Diagram models class structure and contents using design elements such as classes, packages and objects as shown in Fig. **??**. It also displays relationships such as containment, inheritance, associations and others. In this work we experimented with the ideas of Histogram based Image Retrieval using Texture Feature system with different methods of extracting texture feature. We incorporated the Histogram Intersection measure method to compare the query image with database images. A measure of the over-all similarity between images, defined by our approach, incorporates all local properties of the texture histograms of the images. We proved that our approach is well suited to retrieve best possible results. There are several improvements that can be taken as future work for this project. Our system considers only the texture feature of the image. Consideration of other features like shape, location can help for a better retrieval of images. The database of images can be of even more images. [1] [2]

1

Figure 1: Fig. 1 :

2

Figure 2: Fig. 2 :

3 I

Figure 3: Fig. 3 :

NTRODUCTION

Figure 4: Global

N

Figure 5:

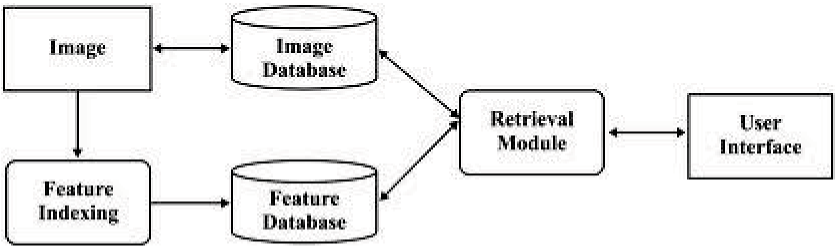

Figure 6:

56 II.

Figure 7: Fig. 5 :Fig. 6 :

79101112 B

Figure 8: Fig. 7 :Fig. 9 :Fig. 10 :Fig. 11 :Fig. 12 :

CKGROUND &

Figure 9:

[Jain and Dubes ()] *Algorithms for clustering data*, K Jain , R Dubes . 1988. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[Comaniciu et al. (1998)] 'Bimodal system for interactive indexing and retrieval of pathology images'. P Comaniciu , D Meer , E Foran , Medl . *Workshop on Applications of Computer Vision*, (Princeton, NJ) Oct.1998. p. .

[Kak and Pavlopoulou (2002)] 'Content-Based Image Retrieval from Large Medical Databases'. C Kak , Pavlopoulou . *3D Data Processing*, (Visualization, Transmission, Padova, Italy) June 2002.

[Lehmann et al. (2000)] 'Content-Based Image Retrieval in Medical Applications: A Novel Multi-Step Approach'. T Lehmann , B Wein , J Dahmen , J Bredno , F Vogelsang , M Kohnen . *International Society for Optical Engineering*, Feb. 2000. 3972 p. .

[Clausi and Jernigan (2000)] 'Designing Gabor filters for optimal texture separability'. M Clausi , Jernigan . *Pattern Recognition* Jan. 2000. 33 p. .

[Korn et al. ()] 'Fast and e.ective retrieval of medical tumor shapes'. F Korn , N Sidiropoulos , C Faloustos , E Siegel , Z Protopapas . *IEEE Transactions on Knowledge and Data Engineering* 1998. 10 (6) p. .

[Chu et al. ()] 'Knowledge-based image retrieval with spatial and temporal constructs'. W Chu , C Hsu , A Cardenas , R Aira . *IEEE Transactions on Knowledge and Data Engineering* 1998. 10 (6) p. .

[Bovik et al. ()] 'Multichannel texture analysis using localized spatial .lters'. M Bovik , W Clark , Geisler . *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1990. 12 p. .

[Huang et al. ()] 'Multimedia Analysis and Retrieval System (MARS) project'. T S Huang , S Mehrotra , K Ramchandran . *Proceedings of 33rd Annual Clinic on Library Application of Data processing -Digital Image Access and Retrieval*, (33rd Annual Clinic on Library Application of Data processing -Digital Image Access and RetrievalUrbana/Champaign, Illinois) 1996. p. .

[Ma and Manjunath (1997)] 'Netra: a toolbox for navigating large image databases'. W Y Ma , B S Manjunath . *International Conference on Image Processing (ICIP)*, Oct. 1997. p. .

[Haralick et al. ()] 'Textural features for image classi.cation'. R Haralick , K Shanmugam , I Dinstein . *IEEE Transactions Systems on Man and Cybernetics*, 1973. 3 p. .

[Bach et al. (1996)] 'Virage image search engine: An open framework for imagemanagement'. J Bach , C Fuller , A Gupta , A Hampapur , B Horowitz , R Humphrey , R Jain . *SPIE Storage and Retrieval for Imageand Video Databases*, (SanDiego/La Jolla, CA) Jan. 1996. 2670 p. .