IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. VI (Nov – Dec. 2014), PP 29-33
www.iosrjournals.org

Analysis of Image and Video Using Color, Texture and Shape Features for Object Identification

Kumar Rajpurohit[1], Rutu Shah[2], Sandeep Baranwal[3], Shashank Mutgi[4]
[1234]*(Computer Department, Vishwakarma Institute of Information Technology, Pune University, India)*

Abstract: *The recent developments in cognitive sciences and Artificial Intelligence has made possible the process of automation in object detection and recognition using various low level visual features used to represent a standard image of an object. Our approach uses analysing the methods used by most intelligent living entities, humans. Humans use the sense of sight and intuition to recognise objects. This paper highlights the methodologies that can be used to identify objects using details such as colour, texture and shape of an image to infer the object.*

Keywords: *CBVR, Edge Detection, Feature Extraction, Object Identification, Video Segmentation*

I. Introduction

Identification and recognition of real world objects is a distinctive and important trait in humans and animals that provides essential knowledge and data about the object's structure and its behaviour. Most of the knowledge of an object is retrieved through this data. Furthermore the main percept or sense that humans or animals take into account is the image provided by sight.

The logical correctness of the well-known proverb "A picture is worth a thousand words" is undeniably applicable in this situation. Incorporating this feature in machines requires study of methodologies utilized by animals and humans i.e. analysing an image using its low level features.

An image is represented in digital form using pixels. Pixels are the smallest element of an image that are used to store information about a particular point in an image. We can define an object as a group of interrelated and similar pixels. Our methodology explores ways to recognize an object in an image by identifying separating these interrelated pixels from all other pixels in image. After separating these pixels we can infer using the knowledge that is already stored in some system to recognize the object.

Knowledge of an object can be represented using 3 basic low level features. These features are colour, texture and shape. These features helps to represent the content of an object without the need of any external notations attached to the image describing its contents.

Content Based Video Retrieval (CBVR) can be considered as an extension of Content Based Image Retrieval (CBIR) with the added functionalities to index and analyse the frames of a video which holds a great deal of importance in CBVR. This paper discusses and describes some of the techniques used to identify and recognize objects from frames using CBIR features.

The process of CBVR mainly consists of following stages namely: Video analysis and segmentation, Key frame extraction, feature extraction and video indexing for effective browsing through the video content.

In the following, Section (II) illustrates the behavior of segmentation of video using shot boundary detection. Section (III) highlights the algorithms used for edge detection and key frame extraction and Section (IV) the object identification and recognition algorithms used using colour and texture features.

II. Video Segmentation

The A video can be described as a sequence of images with a time axis to refer that image at a particular moment of time. Video segmentation is to segment moving objects in video sequences. Video segmentation initially segments the first image frame as the image frame into some moving objects and then it tracks the evolution of the moving objects in the subsequent image frames. After segmenting objects in each image frame, these segmented objects have many applications, such as surveillance, object manipulation, scene composition, and video retrieval. Video is created by taking a set of shots and composing them together using specified composition operators. Extracting structure primitives is the task of video segmentation that involves the detecting of temporal boundaries between scenes and between shots. Temporal analysis of a video shows that a video sequence can be divided into the following levels of granularity [1].

1. Frame level: At this level each image frame of the video is treated as an individual component.
2. Shot level: A shot is defined as a set of continuous frames having the same background obtained from the same camera but divided on the basis of its shot boundaries.

3. Scene level: A scene is a combination of many shots having the same semantic meaning as taken from the same video sequence.
4. Video level: It is the complete element representing the video sequence itself as an entire individual element.

The following diagram illustrates the behavior of the hierarchy of the different levels of video representation

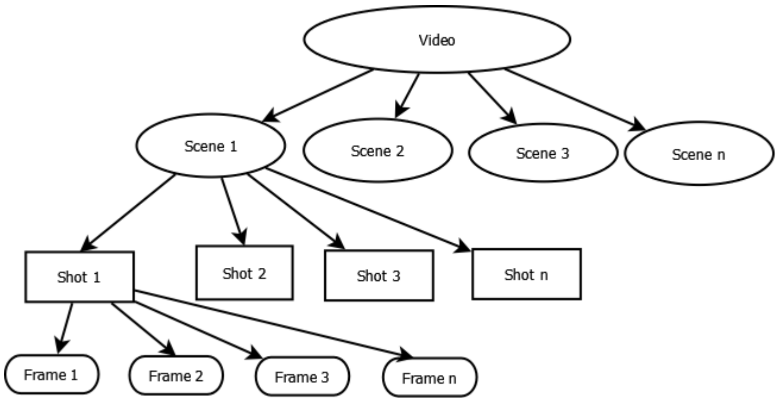


Fig: 1. Video Segmentation

III. Shot Boundary Detection

The shot boundary detection method used in this paper is based on the difference between color histograms of frames belonging to a video sequence. This difference is computed as:

$$HistDif[i] = \sum_{i=1}^{M} \left| h_i(i) - h_{i-1}(j) \right| \qquad (1)$$

Where h_i is the color histogram and M is the number of bins of i^{th} frame of the video sequence. Previous approaches only compare peaks in the histogram difference graph with a previously obtained threshold value. Differences that reach above the threshold value represent detected shot cuts. Figure 2 shows the result of computing the color histogram difference for a given video sequence. In the figure a peak appears when a large discontinuity occurs between histograms. These peaks are usually associated to an abrupt transition. However, from all appearing peaks in figure 2, a real shot cut is represented only with the peak appearing at the last few frames. All other peaks are caused by the intensive object movement in front of the camera. It is evident that selecting the threshold value is a problem of its own. Selecting too high threshold value increases the number of missed shot cuts. Using a lower threshold results in increasing the number of false alarms. A way to eliminate the peaks caused by the camera or objects motion, has to be derived. Figure 2 show that an abrupt scene transition produces only one peak value within a period of time. Therefore, we consider a sliding window of size 2n+1 along the axis that covers frame transitions HistDif [i-n], . . . HistDif [i+n].

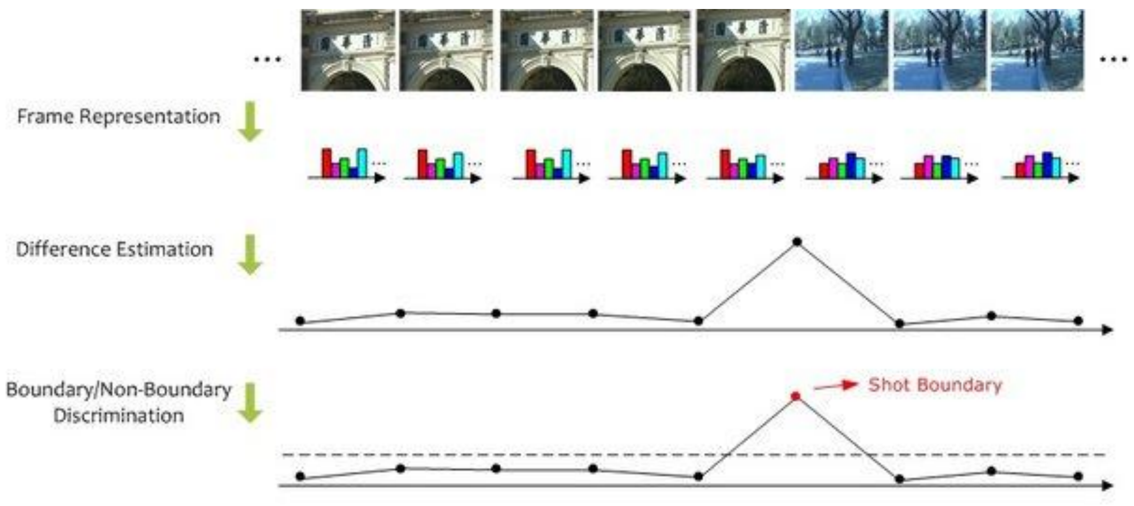


Fig: 2. Color histogram difference

Next we compute the local mean-ratio within the sliding window, for each frame:

$$M_i = \frac{\sum_{j=i-n, j\neq i}^{j=i+n} HistDif_j}{2n} \qquad (2)$$

Then we map the color histogram difference curve into the local mean-ratio space. The histogram difference value at frame i is now equal to its original value HistDif[i] divided by the mean M_i of the appropriate sliding window:

$$HistDif^*[i] = \frac{M_i}{HistDif[i]} \qquad (3)$$

It is evident that false alarms appearing in the original histogram difference graph are eliminated. The only peak that appears in this curve is the actual shot cut at last few frames. The chosen value of 5 for the half length of the sliding window 'n' is empirically derived from various experiments. Choosing a greater value than 5 increases the danger of including two shot cuts in a single sliding window.

IV. Key Frame extraction

Key frame extraction is the process of selecting an individual key-frame to represent an entire shot. This method reduces the amount of storage required to store the information acquired from a shot by simply selecting a frame that represents most of the activity or most number of objects in that shot sequence.

Usually key-frames are selected as the first and last frame of a shot sequence. This technique mostly elaborates all of the contents present in the frames. Hence if we have a video sequence V consisting of M number of scenes then we can get 2*M number of key-frames representing the video sequence V.

V. Edge Detection

The purpose of edge detection is to significantly reduce the amount of data in an image with respect to the further processing of that image for object identification or recognition. Over the years there has been a lot of research in the field of edge detection, but some of these algorithms stand out as compared to the others.

1. The Canny Edge Detection Algorithm:

John F Canny proposed the Canny edge detection algorithm in 1986 [2], and even though this algorithm is quite old it is still used as one of the standard edge detection algorithms in today's world.
The algorithm can be mainly described as it runs in the following 5 steps,

1) Smoothing:
It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the image is first smoothed by applying a Gaussian filter.

2) Finding Gradients:
The Canny algorithm basically finds edges where the grayscale intensity of the image changes the most. These areas are found by determining gradients of the image.

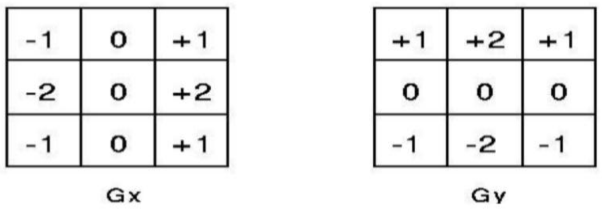
3) Non maxima suppression:
The purpose of this step is to convert the "blurred " edges into "sharp " ones by rounding off the gradient direction to the one nearest to 45^o .

4) Double thresholding:
The edge-pixels that are left after non maxima suppression are marked by their respective strengths by considering 2 threshods .One determining max and other min threshold so the edge pixels having greater values than max are accepted and lower than min are rejected.

5) Edge tracking by hysteresis:
Strong edges are interpreted as "certain edges", and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges.

2. Sobel Edge Detection:

Sobel method is another edge detection algorithm. The Sobel edge detector use two masks with 3x3 sizes, one estimating the gradient in the x-direction and the other estimating the gradient in the y-direction. The mask is slid over the image, manipulating a square of pixels at a time. The algorithm calculates the gradient of the image intensity at each point, and then gives the direction to increase the image intensity at each point from light to dark. Edges areas represent strong intensity contrasts which are darker or brighter. Sobel algorithms work using a mathematical procedure called convolution and commonly analyze derivatives or second derivatives of the digital numbers over space. We implement the Sobel method for edges detection, which is based on a 3 by 3 array that is moved over the main image.



-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Fig.3: Sobel masks

VI. Texture

Texture Feature Extraction:

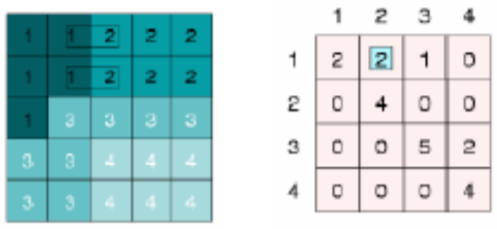
We can define texture as the different patterns observed on objects. We can use texture features to identify the similarity between 2 objects by comparing their statistical features such as coarseness, likeliness, roughness and regularity

1. Edge Histogram

The distribution of edges is a good texture signature that is useful for image to image matching even when the underlying texture is not homogeneous. The edge histogram descriptor represents the spatial distribution of five types of edges (four directional edges and one non directional). It consists of local histograms of these edge directions, which may optionally be aggregated into global or semi-global histograms. The image is divided in 4x4 non-overlapping sub-images where the relative frequencies of five different edge types (vertical, horizontal,450, 1350, non-directional) are calculated by using 2x2-sized edge detectors for the luminance of the pixels. The descriptor is obtained with a nonlinear mapping of the relative frequencies to discrete values.

2. Gray co-occurrences matrix

This is one of the most important texture feature extraction method. Julesz found through his research on human visual perception of texture that no texture pair can be differentiated if there is a large class of textures, if their second order statistics match[2].



1	1	2	2	2
1	1	2	2	2
1	3	3	3	3
3	3	4	4	4
3	3	4	4	4

	1	2	3	4
1	2	2	1	0
2	0	4	0	0
3	0	0	5	2
4	0	0	0	4

Fig.3 GLC Matrix

VII. Color Feature Extraction

Color is one of the most widely and most data rich feature of an image capable of representing huge amounts of knowledge.

1. Color histograms

A color histogram represents the distribution of colors in an image, through a set of bins, where each histogram bin corresponds to a color in the quantized color space. A color histogram for a given image is represented by a vector:

H={H[0],H[1],H[2], H[3],…., H[i],… H[n]}

Where i is the color bin in the color histogram and H[i] represents the number of pixels of color i in the image, and n is the total number of bins used in color histogram. Typically, each pixel in an image will be assigned to a bin of a color histogram. Accordingly in the color histogram of an image, the value of each bin gives the number of pixels that has the same corresponding color. In order to compare images of different sizes, color histograms should be normalized. The normalized color histogram H' is given as:

Where H' [i] = $\frac{H[i]}{p}$, p is the total number of pixels of an image.

VIII. Conclusion

In this paper we discussed and studied a few techniques and described the working of those techniques in relation to Content Based Video and Image Retrieval systems. We also discussed the basic steps followed in order to get a complete idea of image and video processing systems. We presented different algorithms and their working in brief and the different low level features which can be used in order to extract and identify objects from image and video sequences.

References

Journal Papers:

[1] Shweta Ghodeswar , B.B. Meshram , Content Based Video Retrieval, International Journal of UbiComp (IJU), Vol.3, No.2, April 2012.

[2] Canny Edge Detection 09gr820 March 23, 2009.

[3] Ivica Dimitrovisci, Suzana Loskovska, Gorgi Kakasevki and Inam Chorbev, Video Content Based Retrieval System, The International Conference on “Computer as a tool”, September 2012.

[4] Dr.S.Vijayarani, Mrs.M.Vinupriya, Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Issue 8, October 2013.