www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 7271-7279

A Survey on Content Based Video Retrieval

Mr. Amit Fegade[1], Prof. Vipul Dalal[2]

[1]Alamuri Ratnamala Institute of Engineering and Technology, Shahapur, Mumbai University, India
amit.fegade121@gmail.com

[2]Vidyalankar Institute of Technology, Wadala, Mumbai University, India
vipul.dalal@vit.edu.in

Abstract**:** ***Content based video retrieval has a wide spectrum of promising applications, motivating the interests of the researchers worldwide. This paper represents an overview of the general strategies used in visual content-based video retrieval. It focuses on the different methods for video structure analysis, including shot segmentation, key frame extraction, scene segmentation, feature extraction, video annotation, and video retrieval method. This work helps the upcoming researchers in the field of video retrieval to get the idea about different techniques and methods available for the video retrieval.***

Keywords: Feature extraction, shot detection, Scene segmentation, Video retrieval, Video annotation, Video structure analysis,

1. INTRODUCTION

The challenges behind the design and implementation of the content based video browsing; indexing and retrieval systems have attracted researchers from much compliance. It is widely accepted that successful solution to the problem of understanding and indexing the videos requires combination of information from different sources such as images, audio, text, speech etc. Videos have the following characteristics: 1) much richer content than individual images; 2) huge amount of raw data; and 3) very little prior structure. These characteristics make the indexing and retrieval of videos quite difficult. In the past, video databases have been relatively small, and indexing and retrieval have been based on keywords annotated manually. More recently, these databases have become much larger and content based video indexing and retrieval is required, based on the automatic analysis of videos with the minimum of human participation. Content based video retrieval has a wide range of applications such as quick video browsing, analysis of visual electronics commerce, remote instructions, digital museums, news video analysis [1], intelligent management of the web videos and video surveillance. A video may have an auditory channel as well as a visual channel. The available information from videos includes the following [2], [3]: 1) video metadata, which are tagged texts embedded in videos, usually including title, summary, date, actors, producer, broadcast duration, file size, video format, copy-right, etc. 2) audio information from the auditory channel. 3) Transcripts: Speech transcripts can be obtained by speech recognition and caption texts can be read using optical character recognition techniques. 4) Visual information contained in the images themselves from the visual channel. In this paper, we focus on the visual contents of the videos and give a survey on visual content-based visual retrieval and indexing.

2. VIDEO INDEXING

The process of building indexes for videos normally involves the following three main steps:

2.1 Video Parsing:

It consists of temporal segmentation of the video contents into smaller units. Video parsing methods extract structural information from the video by detecting temporal boundaries and identifying significant segments, called *shots*.

2.2 Abstraction:

It consists of extracting the representative set of video data from the video. The most widely used video abstractions are: the “highlight” sequence (A shorter frame sequence extracted from the shot) and the key frame (images extracted from the video shot). The result of video abstraction forms the basis for the video indexing and browsing.

2.3 Content Analysis:

It consists of extracting visual features from key frames. Several techniques used for image feature extraction can be used but, they are usually extended to extraction of features that are specific to video sequences, corresponding to the notion of object motion, events & actions.

3. VIDEO PARSING

Similarly to organizing a long text into smaller units, such as paragraph, sentences, words and letters, a long video sequence must be organized into smaller and more manageable components, upon which indexes can be built. The process of breaking a video into smaller units is known as video parsing. These components are usually organized in a hierarchical way, with 5 levels, in decreasing degree of granularity: video, scene, group, shot and key frame. The basic unit is called as a shot. It is defined as a sequence of frames recorded contiguously and representing a continuous action in time or space. The most representative frame of a shot is called a key frame. A *scene* or *sequence* is formally defined as a collection of semantically related and temporally adjacent shots, depicting and conveying a high-level concept or story. A video *group* is an intermediate entity between the physical shots and semantic scenes and serves as a bridge between the two. Examples of groups are temporally adjacent shots and visually similar shots [4]. In the following sections we present few algorithms and techniques for video parsing at a shot level and boundary detection at a scene level.

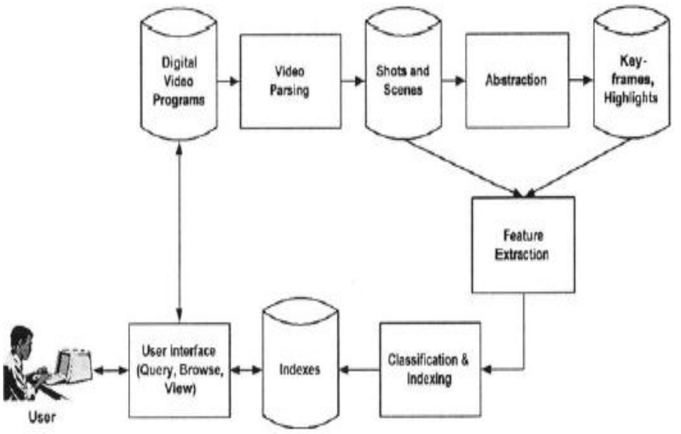

Figure 1: .Content based video retrieval system

3.1. Shot Boundary Detection:

Shot detection is the process of detecting boundaries between two consecutive shots, so that a sequence of frames belonging to a shot will be grouped together [5]. There are different types of boundaries between shots. The simplest one is the *cut*, an abrupt change between the last frame of a shot and the first frame of a subsequent shot. Gradual boundaries are harder to detect. Examples include: dissolves, wipes, fade-ins, and fade-outs. A *fade* is a "gradual means of closing or starting a shot, often used as a transitional device when one scene closes with the image disappearing (a fade-out) and the next scene comes into view as the image grows stronger and stronger (a fade-in) [6]." A *dissolve* is "a transition between two shots whereby the first gradually fades out as the second gradually fades in with some overlap between the two [7]." A wipe is a transition "in which the new shot gradually appears while pushing or 'wiping' off the old [7]." An additional level of difficulty is imposed by camera operations such as panning (the process of moving a camera horizontally around a fixed axis) and zooming (the apparent movement either toward or away from a subject). A robust shot boundary detection algorithm should be able to detect all these different types of boundaries with accuracy.

The basis for detecting shot boundaries is the detection of significant changes in contents on consecutive frames lying on either side of a boundary. Automatic shot boundary detection techniques can be classified into seven main groups:

3.1.1. Pixel-based:

The easiest way to detect a shot boundary is to count the number of pixels that change in value more than some threshold. This total is compared against a second threshold to determine if a shot boundary has been found [8]. The major problems with this approach are its sensitivity to camera movement and noise. Examples of pixel-based shot detection techniques can be found in [9], [10], [11], [12], [13].

3.1.2. Statistics-based:

Statistical methods expand on the idea of pixel differences by breaking the images into regions and comparing statistical measures of the pixels in those regions [8]. For example, Kasturi and Jain [14] use intensity statistics (mean and standard deviation) as shot boundary detection measures. This method is reasonably robust to noise, but slow and prone to generate many false positives (i.e., changes not caused by a shot boundary) [8].

3.1.3. Histogram-based:

The most popular metric for sharp transition detection is the difference between histograms of two consecutive frames. In its simplest form, the gray level or color histograms of two consecutive frames are computed: if the bin-wise difference between the two histograms is above a threshold, a shot boundary is said to be found. Several variants of the basic idea have been proposed in the literature. Nagasaka and Tanaka [15] proposed breaking the images into 16 regions, using a x2-test on color histograms of those regions, and discarding the eight largest differences to reduce the effects of object motion and noise. Swanberg, Shu, and Jain [16] used gray level histogram differences in regions; weighted by how likely the region was to change in the video sequence. Their results were good because their test video (CNN Headline News) had a very regular spatial structure. Zhang, Kankanhalli, and Smoliar [13] compared pixel differences, statistical differences and several different histogram methods and concluded that the histogram methods were a good trade-off between accuracy and speed. They also noted, however, that the basic algorithm did not perform too well for gradual transitions as it did for abrupt cuts. In order to overcome these limitations, they proposed the *twin-comparison algorithm*, which uses two comparisons: one looks at the difference between consecutive frames to detect sharp cuts, and the other looks at accumulated difference over a sequence of frames to detection gradual transitions. This algorithm also applies a global motion analysis to filter out sequences of frames involving global or large moving objects, which may confuse the gradual transition detection. Additional examples of histogram-based shot detection techniques include [17], [18], [19], [20], [21].

3.1.4. Transformed-based:

Transform-based techniques use the compressed Discrete Cosine Transform (DCT) coefficients present in an MPEG stream as the boundary measure. The first comparison metric based on DCT for partitioning JPEG compressed videos was developed by Arman and colleagues [22] and extended to MPEG by Zhang et al. [21]. In this algorithm, a subset of the blocks in each frame and a subset of the DCT coefficients for each block were used as a vector representation for each frame and the difference metric between frames is defined by content correlation in terms of a normalized inner product between the vector representations of two - not necessarily consecutive - frames. Yeo and Liu [23] have observed that for detecting shots boundaries, DC components of DCTs of video frames provide sufficient information. Based on the definition of DCT, this is equivalent to a low-resolution version of the frames, averaged over 8x8 non-overlap blocks. This observation has led to yet another method for video segmentation in which, instead of comparing histograms of pixel values, histograms of DCT-DC coefficients of frames are compared. DCT-based metrics can be directly applied to JPEG video, where every frame is intra-coded. In MPEG, however, DCT-based metrics can be applied only in comparing I-frames. Since only a small portion of frames in MPEG are I-frames, this significantly reduces the amount of processing required to compute differences at the expense of a loss of temporal resolution between I-frames, which typically introduces a large fraction of false positives and requires additional processing [18], [23], [21].

3.1.5. Edge-based:

Zabih, Miller, and Mai [12] proposed an edge-based algorithm that works as follows. Consecutive frames are aligned to reduce the effects of camera motion and the number and position of edges in the edge detected images is recorded. The percentage of edges that enter and exit between the two frames is then computed. Shot boundaries are detected by looking for large edge change percentages. Dissolves and fades are identified by looking at the relative values of the entering and exiting edge percentages. They concluded that their method was more accurate at detecting cuts than histogram-based techniques.

3.1.6. Motion-based:

Zhang, Kankanhalli, and Smoliar [21] used motion vectors determined from block matching to detect whether or not a shot was a zoom or a pan. Shahraray [24] used the motion vectors extracted as part of a region-based pixel difference computation to decide if there is a large amount of camera or object motion in a shot. Because shots with camera motion can be incorrectly classified as gradual transitions, detecting zooms and pans increases the accuracy of a shot boundary detection algorithm. Other examples of motion-based shot detection can be found in [25], [26]. Motion vector information can also be obtained from MPEG compressed video streams. However, the block matching performed as part of MPEG encoding selects vectors based on compression efficiency and thus often selects inappropriate vectors for image processing purposes [8].

3.1.7. Other approaches:

Recent work in shot boundary detection include the use of clustering and post filtering [27], which achieves reasonably high accuracy without producing many false positives, and the combination of image, audio, and motion information [28].

Several studies [8], [29], [30] have compared shot boundary detection algorithms, and have concluded that histogram-based algorithms and MPEG compression domain feature-based techniques exhibit the best performance both from accuracy as well as speed vantage points.

3.2. Scene Boundary Detection:

The automatic detection of *semantic* boundaries (as opposed to *physical* boundaries) within a video program is a much more challenging task and the subject of ongoing research. Part of the problem lies in the fact that scenes and stories are semantic entities that are inherently subjective and lack universal definition or rigid structure. Moreover, there is no obvious direct mapping between these concepts and the raw video contents. Its solution requires a higher level of content analysis. Two different strategies have been used to solve the problem of automatic scene detection: one based on film production rules, the other based on *a priori* program models. Examples of the former include the work of Aigrain et al. [31] using filming rules (e.g., transition effects, shot repetition, appearance of music in the soundtrack) to detect local (temporal) clues of macroscopic change and the research results of Yeung et al. [32] in which a *time-constrained clustering* approach is proposed, under the rationale that semantically related contents tend to be localized in time. *A priori* model-based algorithms rely on specific structural models for programs whose temporal structures is usually very rigid and predictable, such as news and sports [33, 34, 35].

Scene segmentation is also called as story unit segmentation. In general, a scene is nothing but a group of contiguous shots that are consistent with a certain subject. Scenes have higher level semantics than shots. Scenes are identified by grouping the successive shots into a meaning-full semantic unit with similar content. The grouping may be based on information from audio track, texts or images in the video. According to shot representation, scene segmentation methods can be classified into three categories: key frame based audio and visual information integration-based, and background-based.

3.2.1. Key frame based approach:

This approach represents each video shot by a set of key frames from which features are extracted. Temporally close shots are grouped into a scene. An author in [36] compute similarities between the shots using block matching of key frames, then similar shots are linked together and scenes are identified by connecting the overlapping links. Ngo et al. [37] extract and analyze the motion trajectories encoded in the temporal slices of image volumes Scene changes can be identified by measuring the similarities of the key frames in the neighboring shots. A limitation of this approach is that key frames cannot effectively represent the dynamic content of the shots.

3.2.2. Audio and vision integration based approach:

This method selects a shot boundary where the visual as well as audio contents change simultaneously as a scene boundary. The limitation of this approach is that it is difficult to determine the relation between visual shots and audio segments.

3.2.3. Background based approach:

Background based approach segments the scenes under the assumption that shots belonging to the same scene often have similar backgrounds. An author, Chen et al. [38] uses a mosaic technique to reconstruct the background of each video frame. Then, the texture and color distributions of all the background images in a shot are estimated to determine the shot similarity and the rules of filmmaking are used to guide the shot grouping process. The limitation of this method is the assumption that shots in the same scene have similar backgrounds: sometimes the backgrounds in shots in a scene can be different.

According to the processing method, scene segmentation approaches can be divided into four categories: splitting-based, merging based, shot boundary classification based, and statistical model based.

(a) Splitting based approach:

This method splits the entire video into separate coherent scenes using a top-down style. For example, Rasheed and Shah [39] construct a shot similarity graph for a video and partition the graph using normalized cuts. The sub graphs represent individual scenes in the video.

(b) Merging based approach:

This method gradually merges similar shots in a bottom-up style to form a scene. Rasheed and Shah [40] proposed a two-pass scene segmentation algorithm. In the first pass, over segmentation of scenes is carried out using backward shot coherence and in the second pass, the over segmented scenes are detected using motion analysis and then merged.

(c) Shot boundary classification based approach:

In this method, features of shot boundaries are extracted and then used to classify shot boundaries into scene boundaries and non-scene boundaries. An author, Goela et al. [41] presents a genre independent method to detect scene boundaries in broadcast videos. In that method, scene segmentation is based on a classification with the two classes of "scene change" and "non-scene change." An SVM had been used to classify the shot boundaries. Hand labeled video scene boundaries from a variety of broadcast genres are used to generate positive and negative training samples for the SVM. The common point in the splitting-based, merging-based, and statistical model-based approaches is that the similarities between different shots are used to combine similar shots into scenes. This is intuitive and simple. However, in these methods, shots are usually represented by a set of key frames, which often fail to represent the dynamic contents of the shots. As a result, two shots are regarded as similar, if their key frames are in the same environment rather than if they are visually similar. The shot boundary classification-based approach takes advantage of the local information about shot boundaries. It ensures that algorithms with low computational complexities are easy to obtain. However, lack of global information about shots inevitably reduces the accuracy of scene segmentation.

(d) Statistical model based:

This method constructs the statistical models of shots to segment scenes. Zhai and Shah [42] use the stochastic Monte Carlo sampling to simulate the generation of scenes. The scene boundaries are updated by merging, diffusing and splitting the scene boundaries.

4. VIDEO ABSTRACTION

Video abstraction is the process of extracting a presentation of visual information about the landscape or structure of a video program, in a way that is more economical than, yet representative of, the original video. There are two main approaches to video abstraction: *key-frames* and *"highlight" sequences.*

4.1. Key Frame extraction:

A key-frame is the still image extracted from the video data that best represents the contents of a shot in an abstract manner. There are great redundancies among the frames in the same shot; therefore, certain frames that best reflect the shot contents are selected as key to succinctly represent the shot. The extracted key frames should contain as much salient content of the shot as possible and avoid as much redundancy as possible. The features used for key frame extraction include colors (particularly the color histogram), edges, shapes, optical flow, MPEG-7 motion descriptors such as temporal motion intensity and spatial distribution of motion activity, MPEG discrete cosine coefficient and motion vectors, camera activity, and features derived from image variations caused by camera motion. Current approaches to extract key frames are classified into six categories: sequential comparison-based, global comparison-based, reference frame-based, clustering-based, curve simplification-based, and object/event-based.

4.1.1. Sequential comparison between frames:

In these algorithms, frames subsequent to a previously extracted key frame are sequentially compared with the key frame until a frame which is very different from the key frame is obtained. This frame is selected as the next key frame. The merits of the sequential comparison-based algorithms include their simplicity, intuitiveness, low computational complexity, and adaptation of the number of key frames to the length of the shot. The limitations of these algorithms include the following. 1) The key frames represent local properties of the shot rather than the global properties. b) The irregular distribution and uncontrolled number of key frames make these algorithms unsuitable for applications that need an even distribution or a fixed number of key frames. c) Redundancy can occur when there are contents appearing repeatedly in the same shot.

4.1.2. Global comparison between frames:

The algorithms based on global differences between frames in a shot distribute key frames by minimizing a predefined objective function that depends on the application. In general, the objective function has one of the following four forms.

(a) Evan temporal variance:

These algorithms select key frames in a shot such that the shot segments, each of which is represented by a key frame, have equal temporal variance. The objective function can be chosen as the sum of differences between temporal variances of all the segments. The temporal variance in a segment can be approximated by the cumulative change of contents across consecutive frames in the segment or by the difference between the first and last frames in the segment.

(b) Maximum coverage:

These algorithms extract key frames by maximizing their representation coverage, which is the number of frames that the key frames can represent. If the number of key frames is not fixed, then these algorithms minimize the number of key frames subject to a predefined fidelity criterion; alternatively, if the number of key frames is fixed, the algorithms maximize the number of frames that the key frames can represent.

(c) Minimum correlation:

These algorithms extract key frames to minimize the sum of correlations between key frames (especially successive key frames), making key frames as uncorrelated with each other as possible.

(d) Minimum reconstruction error:

These algorithms extract key frames to minimize the sum of the differences between each frame and its corresponding predicted frame reconstructed from the set of key frames using interpolation. These algorithms are useful for certain applications, such as animation.

The merits of the aforesaid global comparison-based algorithms include the following. 1) The key frames reflect the global characteristics of the shot. 2) The number of key frames is controllable. 3) The set of key frames is more concise and less redundant than that produced by the sequential comparison-based algorithms. The limitation of the global comparison-based algorithms is that they are more computationally expensive than the sequential comparison-based algorithms.

4.1.3. Reference frame:

These algorithms generate a reference frame and then extract key frames by comparing the frames in the shot with the reference frame. The merit of the reference frame-based algorithms is that they are easy to understand and implement. The limitation of these algorithms is that they depend on the reference frame: If the reference frame does not adequately represent the shot, some salient contents in the shot may be missing from the key frames.

4.1.4. Clustering:

These algorithms cluster frames and then choose frames closest to the cluster centers as the key frames. The merits of the clustering-based algorithms are that they can use generic clustering algorithms, and the global characteristics of a video can be reflected in the extracted key frames. The limitations of these algorithms are as follows: First, they are dependent on the clustering results, but successful acquisition of semantic meaningful clusters is very difficult, especially for large data, and second, the sequential nature of the video cannot be naturally utilized: Usually, clumsy tricks are used to ensure that adjacent frames are likely to be assigned to the same cluster.

4.1.5. Curve simplification:

These algorithms represent each frame in a shot as a point in the feature space. The points are linked in the sequential order to form a trajectory curve and then searched to find a set of points which best represent the shape of the curve. The merit of the curve simplification-based algorithms is that the sequential information is kept during the key frame extraction. Their limitation is that optimization of the best representation of the curve has a high computational complexity.

4.1.6. Objects/Events:

These algorithms jointly consider key frame extraction and object/event detection in order to ensure that the extracted key frames contain information about objects or events. The merit of the object/event-based algorithms is that the extracted key frames are semantically important, reflecting objects or the motion patterns of objects. The limitation of these algorithms is that object/event detection strongly relies on heuristic rules specified according to the application. As a result, these algorithms are efficient only when the experimental settings are carefully chosen.

Because of the subjectivity of the key frame definition, there is no uniform evaluation method for key frame extraction. In general, the error rate and the video compression ratio are used as measures to evaluate the result of key frame extraction. Key frames giving low error rates and high compression rates are preferred. In general, a low error rate is associated with a low compression rate. The error rate depends on the parameters in the key frame extraction algorithms. Examples of these parameters are the thresholds in sequential comparison-based, global comparison-based, reference frame-based, and clustering-based algorithms, as well as the parameters to fit the curve in the curve simplification-based algorithms. Users choose the parameters according to the error rate that can be tolerated. These algorithms represent each frame in a shot as a point in the feature space. The points are linked in the sequential order to form a trajectory curve and then searched to find a set of points which best represent the shape of the curve. The merit of the curve simplification-based algorithms is that the sequential information is kept during the key frame extraction. Their limitation is that optimization of the best representation of the curve has a high computational complexity.

4.2. Highlight sequence:

This approach - also known as *video skimming* or *video summaries*- aims at abstracting a long video sequence into a much shorter (summary) sequence, with a fair perception of the video contents. A successful approach is to utilize information from multiple sources (e.g., shot boundaries, human faces, camera and object motions, sound, speech, and text). Researchers working on documents with textual transcriptions have suggested producing video abstracts by first abstracting the text using classical text skimming techniques and then looking for the corresponding parts in the video sequence. A successful application of this type of approach has been the informed project, in which text and visual content information are merged to identify video sequences that highlight the important contents of the video. The extension of this skimming approach from documentary video programs to other videos with a soundtrack containing more than just speech remains an open research topic.

5. FEATURE EXTRACTION

The extraction of content primitives (referred to as "metadata" in the scope of the emerging MPEG-7 standard) from video programs is a required step that allows video shots to be classified, indexed, and subsequently retrieved. Since shots are usually considered the smallest indexing unit in a video database, content representation of video is also usually based on shot features. There are two types of features: those associated with key-frames only which are static by nature and those associated with the frame sequence that compose a shot which may include the representation of temporal variation of any given feature and motion information associated with the shot or some of its constituent objects. Representing shot contents at an object level through the detection and encoding of motion information of dominant objects in the shot is a new and attractive technique, because much of the object information is available in MPEG-4 video streams.

5.1. Static key frame features:

The key frames of a video represent the characteristics of the video to some extent. Traditional image retrieval techniques can be applied to key frames for achieving a video retrieval. The static key frame features are useful for video indexing &retrieval and are mainly classified as texture-based, color-based and shape-based.

5.1.1. Color-based features:

Color-based features include color histograms, color moments, color correlograms, a mixture of Gaussian models, etc. The exaction of color-based features depends on color spaces such as RGB, HSV, YCbCr and normalized r-g, YUV, and HVC. The choice of color space depends on the applications. Color features can be extracted from the entire image or from image blocks into which the entire image is partitioned. Color-based features are the most effective image features for video indexing and retrieval. In particular, color histogram and color moments are simple but efficient descriptors. The limitation of color-based features is that they do not directly describe texture, shape, etc., and are, thus, ineffective for the applications in which texture or shape is important.

5.1.2. Texture-based features:

Texture-based features are object surface-owned intrinsic visual features that are independent of color or intensity and reflect homogenous phenomena in images. They contain crucial information about the organization of object surfaces, as well as their correlations with the surrounding environment. Texture features in common use include Tamura features, simultaneous autoregressive models, orientation features, wavelet transformation-based texture features, co-occurrence matrices, etc. The merit of texture-based features is that they can be effectively applied to applications in which texture information is salient in videos. However, these features are unavailable in non texture video images.

5.1.3. Shaped-based features:

Shape-based features that describe object shapes in the image can be extracted from object regions. A common approach is to detect edges in images and then describe the distribution of the edges using a histogram.

Shape-based features are effective for applications in which shape information is salient in videos. However, they are much more difficult to extract than color- or texture-based features.

5.2. Object features:

Object features include the dominant color, texture, size etc. of the image regions corresponding to the objects. These features can be used to retrieve videos likely to contain similar objects. The limitation of object-based features is that identification of objects in videos is difficult and time-consuming. Current algorithms focus on identifying specific types of objects, such as faces, rather than various objects in various scenes.

5.3. Motion-based features:

Motion is the essential characteristic distinguishing dynamic videos from still images. Motion information represents the visual content with temporal variation. Motion features are closer to semantic concepts than static key frame features and object features. Video motion includes background motion caused by camera motion and foreground motion caused by moving objects. Thus, motion-based features for video retrieval can be divided into two categories: camera-based and object-based. For camera-based features, different camera motions, such as "zooming in or out," "panning left or right," and "tilting up or down," are estimated and used for video indexing. Video retrieval using only camera-based features has the limitation that they cannot describe motions of key objects. Object-based motion features have attracted much more interesting recent work. Object-based motion features can be further classified into statistics-based, trajectory-based, and objects' spatial relationships-based.

5.3.1. Statistics-based:

Statistical features of the motions of points in frames in a video are extracted to model the distribution of global or local motions in the video. The merit of statistics-based features is that their extraction has low computational complexity. The limitation of these features is they cannot represent object actions accurately and cannot characterize the relations between objects.

5.3.2. Trajectory-based:

The merit of trajectory-based features is that they can describe object actions. The limitation of these features is that their extraction depends on correct object segmentation and tracking and automatic recording of trajectories, all of which are still very challenging tasks.

5.3.3. Object's relationship based:

These features describe spatial relationships between objects. The merit of object's relationship-based features is that they can intuitively represent relationships between multiple objects in the temporal domain. The limitation of these features is that it is difficult to label each object and its position.

6. VIDEO ANNOTATION

Video annotation is the allocation of video shots or video segments to different predefined semantic concepts, such as person, car, sky and people walking. Video annotation and video classification share similar methodologies: First, low- level features are extracted, and then certain classifiers are trained and employed to map the features to the concept/category labels. Corresponding to the fact that a video may be annotated with multiple concepts, the methods for video annotation can be classified as isolated concept-based annotation, context-based annotation, and integrated-based annotation

6.1. Isolated concept based annotation:

This annotation method trains a statistical detector for each of the concepts in a visual lexicon, and the isolated binary classifiers are used individually and independently to detect multiple semantic concepts correlations between the concepts are not considered. The limitation of isolated concept-based annotation is that the associations between the different concepts are not modeled.

6.2. Context-based annotation:

The task of context-based annotation is to refine the detection results of the individual binary classifiers or infer higher level concepts from detected lower level concepts using a context-based concept fusion strategy. The limitation of context-based annotation is that the improvement of contextual correlations to individual detections is not always stable because the detection errors of the individual classifiers can propagate to the fusion step, and partitioning of the training samples into two parts for individual detections and conceptual fusion, respectively, causes that there are no sufficient samples for the conceptual fusion because of usual complexity of the correlations between the concepts.

6.3. Integration-based annotation:

This annotation method simultaneously models both the individual concepts and their correlations: The learning and optimization are done simultaneously. The entire set of samples is used simultaneously to model the individual concepts and their correlations. The limitation of the integration-based annotation is its high computational complexity.

7. QUERY AND VIDEO RETRIEVAL

Once video indices are obtained, content-based video retrieval can be performed. On receiving a query, a similarity measure method is used, based on the indices, to search for the candidate videos in accordance with the query. The retrieval results are optimized by relevance feedback, etc. In the following, we review query types, similarity matching, and relevance feedback.

7.1. Query types:

Non-semantic based video query types include query by example, query by sketch, and query by objects. Semantic-based video query types include query by keywords and query by natural language.

7.1.1. Query by Example:

This query extracts low-level features from given example video or image and similar videos are retrieved by measuring feature similarity. The static features of key frames are suitable for query by example, as the key frames extracted from the example video or image can be matched with the stored key frames.

7.1.2. Query by Keywords:

It represents the user's query by a set of keywords. It is the simplest and most direct query type, and it captures the semantics of videos to some extent. Keywords can refer to video metadata, visual concepts, transcripts, etc.

7.1.3. Query by Sketch:

This type of query allows users to draw sketches to represent the videos they are looking for. First, features are extracted from the sketches and then they are matched to the features of the stored videos.

7.1.4. Query by Object:

This query allows users to provide an image of object. Then, the retrieval system finds and retrieves all occurrences of the object in the video database. In contrast with query by example and query by sketch, the search results of query by objects are the locations of the query object in the videos.

7.1.5. Query by Natural Language:

It is the most natural and convenient way of making a query. It uses semantic word similarity to retrieve the most relevant videos and rank them, given a search query specified in the natural language (English). The most difficult part of a natural language interface is the parsing of natural language and the acquisition of accurate semantics.

7.1.6. Combination-based Query:

This query combines different types of queries such as text-based queries and video example-based queries. The combination-based query is adaptable to multimodal search.

7.2. Similarity measurements:

Video similarity measurement plays an important role in content-based video retrieval. Methods to measure video similarities can be classified into feature matching, text matching, ontology-based matching, and combination-based matching. The choice of method depends on the query type.

7.2.1. Feature matching:

The most direct measure of similarity between two videos is the average distance between the features of the corresponding frames. Query by example usually uses low-level feature matching to find relevant videos. However, video similarity can be considered in different levels of resolution or granularity. According to different user' demands, static features of key frames, object features, and motion features all can be used to measure video similarity. The merit of feature matching is that the video similarity can be conveniently measured in the feature space. Its limitation is that semantic similarity cannot be represented because of the gap between sets of feature vectors and the semantic categories familiar to people.

7.2.2. Text matching:

Matching the name of each concept with query terms is the simplest way of finding the videos that satisfy the query. It normalizes both the descriptions of concepts and the query text and then computes the similarity between the query text and the text descriptions of concepts by using a vector space model. Finally, the concepts with the highest similarity are selected. The merits of the text-matching approach are its intuitiveness and simplicity of implementation. The limitation of this approach is that all related concepts must be explicitly included in the query text in order to obtain satisfactory search results.

7.2.3. Ontology-based matching:

This approach achieves similarity matching using the ontology between semantic concepts or semantic relations between keywords. Query descriptions are enriched from knowledge sources, such as ontology of concepts or keywords. The limitation of this approach is that irrelevant concepts are also likely to be brought in, perhaps leading to unexpected deterioration of search results.

7.2.4. Combination-based matching:

This approach "leverages semantic concepts by learning the combination strategies from a training collection. It is useful for combination-based queries that are adaptable to multimodal searches. The merits of the combination-based matching approach are that concept weights can be automatically determined and hidden semantic concepts can be handled to some extent. The limitation of this approach is that it is difficult to learn query combination models.

7.3. Relevance Feedback:

In relevance feedback, the videos obtained in reply to a search query are ranked either by the user or automatically. This ranking is used to refine further searches. The refinement methods include query point optimization, feature weight adjustment, and information embedding. Relevance feedback bridges the gap between semantic notions of search relevance and the low-level representation of video content. Relevance feedback also reflects user's preferences by taking into account user feedback on the previously searched results. Like relevance feedback for image retrieval, relevance feedback for video retrieval can be divided into three categories: explicit, implicit, and pseudo feedback.

7.3.1. Explicit relevance feedback:

This feedback asks the user to actively select relevant videos from the previously retrieved videos. The merit of explicit feedback is that it can obtain better results than implicit feedback or the pseudo feedback discussed later as it uses the user feedback directly. Its limitation is that it needs more user interaction, which requires more user patience and cooperation.

7.3.2. Implicit relevance feedback:

This feedback refines retrieval results by utilizing click-through data obtained by the search engine as the user clicks on the videos in the presented ranking. The merit of implicit feedback is that it does not require the conscious cooperation of the user, making it more acceptable, available, and practicable than explicit feedback. The limitation of implicit feedback is that the information gathered from the user is less accurate than in explicit feedback.

7.3.3. Pseudo relevance feedback:

This feedback selects positive and negative samples from the previous retrieval results without the participation of the user. The positive samples are the ones near to the query sample in the feature space, and the negative samples are far from the query sample. This way, the user's feedback is simulated. These samples are returned to the system for the second search. The merit of pseudo relevance feedback is the substantial reduction in user interaction. It is limited in applications because of the semantic gap between low-level and high-level features: the similarities of low-level features obtained from different videos do not always coincide with the similarities between the videos defined by the user.

CONCLUSION

We have presented a review on recent developments in visual content-based video indexing and retrieval. The state of the art of existing approaches in each major issue has been described with the focus on the following tasks: video structure analysis including shot boundary detection, key frame extraction and scene segmentation, features extraction of static key frames, objects and motions, video annotation, query type and video retrieval methods, video search including interface, similarity measure and relevance feedback.

REFERENCES

[1] Y. X. Peng and C.W. Ngo, "Hot event detection and summarization by graph modeling and matching," in Proc. Int. Conf. Image Video Retrieval, Singapore, pp. 257–266.

[2] A. F. Smeaton, "Techniques used and open challenges to the analysis, indexing and retrieval of digital video," Inform. Syst., vol. 32, no. 4, pp. 545–559.

[3] Y. Y. Chung, W. K. J. Chin, X. Chen, D. Y. Shi, E. Choi, and F. Chen, "Content-based video retrieval system using wavelet transform," World Sci. Eng. Acad. Soc. Trans. Circuits Syst., vol. 6, no. 2, pp. 259–265.

[4] Y. Rui and T.S. Huang. Unified framework for video browsing and retrieval. In A. Bovik, editor Handbook of Image and Video Processing chapter 9.2.Academic Press, San Diego.

[5] H.-J Zhang. Content-based video browsing and retrieval. In B. Furht, editor, Handbook of Internet and Multimedia Systems and Applications, chapter 712. CRC Press, Boca Raton.

[6] I. Konigsberg. The Complete Film Dictionary - 2nd ed. Penguin, New York, 1997.

[7] I. Konigsberg. The Complete Film Dictionary - 2nd ed. Penguin, New York.

[8] J.S. Boreczky and L.A. Rowe. Comparison of video shot boundary detection techniques. In Proceedings of the (SPIE) Conference on Storage and Retrieval for Image and Video Databases, volume 2670, pages 170-179.

[9] A. Hampapur, R. Jain, and T. E Weymouth. Digital video segmentation. In Proceedings of the second A CM international conference on Multimedia '94, pages 357-364, San Francisco, CA.

[10] A. Hampapur, R. Jain, and T. E. Weymouth. Production model based digital video segmentation. In B. Furht, editor, Multimedia Tools and Applications, chapter 4. Kluwer Academic Publishers, Boston.

[11] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-search for video appearances. In E. Knuth and I.!\I. Wegener, editors, Visual Database Systems 2, pages 113-127. Elsevier Science Publishers, Amstredam.

[12] R. Zabih, K. 11,1ai, and J. i\Iiller. A robust method for detecting cuts and dissolves in video sequences. In Proceedings of the ACM 3rd Internationl Multimedia Conference, pages 189-200.

[13] H.-J. Zhang, A. Kankanhalli, and S.W Smoliar. Automatic partitioning of full-motion video. ACM Multimedia Systems, 1(1):10-28.

[14] R. Kasturi and R. Jain. Dynamic visions. In Proceedings of Computer Vision: Principles, Washington, 1991. IEEE Computers Society Prees.

[15] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-search for video appearances. In E. Knuth and I.!\I. Wegener, editors, Visual Database Systems 2, pages 113-127. Elsevier Science Publishers, Amstredam.

[16] D. Swanberg, C.F. Shu, and R Jain. Knowledge guided parsing in video databases. In Storage and Retrieval for Image and Video Databases, *SPIE*, volume 1908, pages 13-25.

[17] P. Aigrain and P. Joly. The automatic real-time analysis of film editing and transition effects and its applications. Computer and Graphics, 18(1):93-103.

[18] 1. K. Sethi and N. V Patel. A statistical approach to scene change detection. In Proc. of the IS€3T /SPIE Symposium on Electronic Imaging Science and Technology, Conference on Storage and Retrieval for Image and Video Databases III, San Jose.

[19] N. Vasconcelos and A. Lippman. Statistical models of video structure for content analysis and characterization. IEEE Transactions on Image Processing,9(1):3-19.

[20] H.-J. Zhang, C.Y Low, S.W. Smoliar, and D. Zhong. Video parsing, retrieval and browsing: An integrated and content-based solution. In Proceedings of A CM conference on Multimedia, pages 15-24, San Fransisco

[21] H.-J. Zhang et al. Video parsing using compressed data. In Proceedings of SPIE conference on Image and Video Processing 2, pages 142-149.

[22] F. Arman, A. Hsu, and M.-y' Chiu. Feature management for large video databases. In Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases I, pages 2-12.

[23] B. L. Yeo and B. Liu. Rapid scene analysis on compressed videos. IEEE Trans.Circuits Syst. Video Technology, 5(6):533-544.

[24] B. Shahraray. Scene change detection and content-based sampling of video sequences. In Proceedings of SPIE Conference on Digital Video Compression:Algorithm and Technologies, volume 2419, pages 2-13, San Jose.

[25] J. Meng, F. Juan, and S.-F. Chang. Scene change detection in a mpeg compressed video sequence. In Symposium on Electronic Imaging: Science f3 Technology, volume 2417, pages 1-12, San Jose.

[26] B.-L. Yeo. Efficient Processing of Compressed Images and Video. PhD thesis, Princeton University.

[27] M.R. Naphade, R. Mehrotra, A.M. Fermant, J. Warnick, T.S. Huang, and A.M. Tekalp. A high-performance shot boundary detection algorithm using multiple cues. In Proceedings of International Conference on Image Processing, volume 1, pages 884-887, 1998.

[28] J.S. Boreczky and L.D. Wilcox. A Hidden Markov Model framework for video segmentation using audio and image features. In Proceedings of International Conference on Acoustics, Speech and Signal Processing, volume 6, pages 3741-3744

[29] A. Dailianas, R. Allen, and P. England. Comparison of automatic video segmentation algorithms. In Proceedings of SPIE Photonics East, volume 2615, pages 2-16, Philadelphia, October 1995.

[30] R.M. Ford, C. Robson, D. Temple, and M. Gerlach. Metrics for scene change detection in digital video sequences. In Proceedings of International Conference on Mult~media Systems, pages 610-611.

[31] P. Aigrain, P. Joly, and V. Longueville. Medium knowledge-based macrosegmentation of video into sequences. In }'1. Maybury, editor, Intelligent Multimedia Information Retrieval, pages 159-173

[32] M. Yeung, B.-L. Yeo, and B. Liu. Extracting story units from long programs for video browsing and navigation. In Proceedings of the International Conference on Multimedia Computing and Systems, pages 296-305.

[33] Y. Gong. Automatic parsing of TV soccer programs. In Proceedings of 2nd International Conference on Multimedia Computing and Systems, pages 167-174.

[34] D. Swanberg, C.F. Shu, and R Jain. Knowledge guided parsing in video databases. In Storage and Retrieval for Image and Video Databases, *SPIE,* volume 1908, pages 13-25.

[35] Y. Zhuang, Y. Rui, T.S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In Proceedings of the IEEE International Conference on Image Processing, volume 1, pages 866-870, Chicago.

[36] A. Hanjalic, R. L. Lagendijk, and J. Biemond, “Automated high-level movie segmentation for advanced video-retrieval systems,” IEEE Trans.Circuits Syst. Video Technol., vol. 9, no. 4, pp. 580–588.

[37] C.-W. Ngo, T.-C. Pong, H.-J. Zhang, and R. T. Chin, “Motion-based video representation for scene change detection,” Int. J. Comput. Vis.,vol. 50, no. 2, pp. 127–142, 2002.

[38] L.-H. Chen, Y.-C. Lai, and H.-Y. M. Liao, “Movie scene segmentation using background information,” Pattern Recognit., vol. 41, no. 3, pp. 1056–1065, Mar. 2008.

[39] Z. Rasheed and M. Shah, “Detection and representation of scenes in videos,” IEEE Trans. Multimedia, vol. 7, no. 6, pp. 1097–1105, Dec.2005

[40] Z. Rasheed and M. Shah, “Scene detection in Hollywood movies and TV shows,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., Jun. 2003,vol. 2, pp. 343–350.

[41] N. Goela, K. Wilson, F. Niu, A. Divakaran, and I. Otsuka, “An SVM framework for genre-independent scene change detection,” in Proc. IEEE Int. Conf. Multimedia Expo., vol. 3, New York, Jul. 2007, pp. 532–535.

[42] Y. Zhai and M. Shah, “Video scene segmentation using Markov chainMonte Carlo,” *IEEE Trans.Multimedia,* vol. 8, no. 4, pp. 686–697, Aug.2006.