Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 860743, 10 pages
doi:10.1155/2008/860743

Research Article

Unsupervised Video Shot Detection Using Clustering Ensemble with a Color Global Scale-Invariant Feature Transform Descriptor

Yuchou Chang,[1] D. J. Lee,[1] Yi Hong,[2] and James Archibald[1]

[1] *Electrical and Computer Engineering Department, Brigham Young University, Provo, UT 84602, USA*
[2] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong*

Correspondence should be addressed to D. J. Lee, djlee@ee.byu.edu

Received 1 August 2007; Revised 30 October 2007; Accepted 22 November 2007

Recommended by Alain Tremeau

Scale-invariant feature transform (SIFT) transforms a grayscale image into scale-invariant coordinates of local features that are invariant to image scale, rotation, and changing viewpoints. Because of its scale-invariant properties, SIFT has been successfully used for object recognition and content-based image retrieval. The biggest drawback of SIFT is that it uses only grayscale information and misses important visual information regarding color. In this paper, we present the development of a novel color feature extraction algorithm that addresses this problem, and we also propose a new clustering strategy using clustering ensembles for video shot detection. Based on Fibonacci lattice-quantization, we develop a novel color global scale-invariant feature transform (CGSIFT) for better description of color contents in video frames for video shot detection. CGSIFT first quantizes a color image, representing it with a small number of color indices, and then uses SIFT to extract features from the quantized color index image. We also develop a new space description method using small image regions to represent global color features as the second step of CGSIFT. Clustering ensembles focusing on knowledge reuse are then applied to obtain better clustering results than using single clustering methods for video shot detection. Evaluation of the proposed feature extraction algorithm and the new clustering strategy using clustering ensembles reveals very promising results for video shot detection.

1. INTRODUCTION

The recent rapid growth of multimedia databases and the increasing demand to provide online access to these databases have brought content-based video retrieval (CBVR) to the attention of many researchers. Because manual indexing of archived videos is infeasible due to prohibitively high labor costs, automatic video retrieval is essential to the online accessing of multimedia databases. Generally, video content can be represented by a hierarchical tree which contains shots, scenes, and events [1]. A continuous video bitstream is segmented into a series of cascaded video shots, which are the basis for constructing high-level scenes and events with semantic meanings. Hence, shot detection [2], the identification of a continuously recorded sequence of image frames, is critical for semantic analysis of video content.

Shot detection can generally be categorized into five classes: pixel-based, histogram-based, feature-based, statistics-based, and transform-based methods [2]. In this paper, we focus on clustering-based shot detection [3–14] which can be considered as a combination of feature-based and statistics-based methods. Different clustering algorithms such as hierarchical clustering [4, 10], k-means [5, 13], self-organizing map (SOM) [7], fuzzy c-means [8, 11], co-occurrence matrix [9], information-theoretic coclustering [12], and other clustering methods [3, 6, 14] have been used for shot detection in recent years.

Berkhin [15] classified clustering algorithms into 8 groups, for example, hierarchical methods, partitioning methods, grid-based methods, constraint-based clustering, and so forth. Generally, clustering-based shot detection methods use just a single clustering algorithm to categorize frames into corresponding shots. Each clustering method has its own advantages and disadvantages that result in different performance over different data sets, so no single method is consistently the best. Considering the success of clustering

ensembles [16–19] in machine learning in recent years, we propose a novel clustering strategy using clustering ensembles for shot detection.

Features that help the user or machine judge if a particular frame is contained within a shot are critical for shot detection. Many visual features have been proposed for describing the content of the image [24]. Scale-invariant feature transform (SIFT) has been shown to be the most robust, invariant descriptor of local features [20–23]. However, SIFT operates on grayscale images rather than the color images that make up the vast majority of recorded videos. SIFT uses a one-dimensional (1D) vector of scalar values for each pixel as a local feature descriptor and cannot be extended to operate on color images which generally consist of three-dimensional (3D) vector values. The main difficulty of applying SIFT to color images is that no color space is able to use 1D scalar values to represent colors. Although there are many color space conversion methods that transform 3D RGB color values to other color spaces such as HSV and CIE Lab, the transformed color spaces still represent colors in 3D.

In order to use SIFT for color video shot detection, each color video frame must be converted into color indices to represent a small set of important colors present in the frame. SIFT can then be applied to the color indices which are treated as gray-level values in grayscale images for feature extraction. We adopt a very powerful color quantization method called Fibonacci lattice-quantization [25] to quantize color information and generate a palette of color indices for SIFT. Based on this approach, we propose a novel color feature descriptor using the global context of the video frame. This new color feature descriptor, based on SIFT, is called the color global scale-invariant feature transform (CGSIFT) descriptor. We then apply clustering ensembles to the new CGSIFT descriptor to detect shots in color video.

The rest of this paper is organized as follows. Section 2 describes background work related to SIFT and clustering ensembles. Section 3 introduces the new CGSIFT for color feature extraction based on SIFT. Shot detection structure based on clustering ensembles is presented in Section 4. Section 5 discusses processing time and storage space analysis to illustrate the advantages of the proposed method. Experimental results are presented in Section 6 to evaluate the performance of the proposed method based on the new feature descriptor and clustering ensembles. Section 7 concludes this work.

2. RELEVANT WORK

2.1. Scale-invariant feature transform

SIFT is a computer vision algorithm that extracts distinctive features from an image. It was originally used for object recognition [20, 22] and later applied to content-based image retrieval [23]. Features extracted by SIFT are invariant to image scale, rotation, and changing viewpoints. The algorithm transforms a grayscale image into scale-invariant coordinates of local features, which are the keypoints of the image. Each keypoint is represented by a 128-dimension vector. SIFT consists of 4 steps [20]: scale-space extrema detection, keypoint localization, orientation assignment, and keypoint descriptor assignment.

However, as previously noted, SIFT features are generally derived from grayscale images. With the development and advancements in multimedia technologies, the bulk of video data of interest is in color. Color images contain more visual information than grayscale. For SIFT feature extraction, video data must be converted to grayscale, causing important visual information to be lost. In order to describe color video contents as accurately as possible, we use a quantization method based on Fibonacci lattices [25] to convert the color image into color indices for SIFT. Furthermore, because SIFT extracts only local features and cannot describe global context for visual content analysis, a new feature-extraction algorithm designed to address the color video shot detection problem would be very useful. We propose such a technique: color global scale-invariant feature transform (CGSIFT).

2.2. Clustering ensemble

Methods based on clustering ensembles have been shown to be effective in improving the robustness and stability of clustering algorithms [16–19]. Classical clustering ensemble methods take multiple clusters into consideration by employing the following steps. First, a population of clusters is obtained by executing different clustering algorithms on the same data set. Second, an ensemble committee is constructed from all resulting clusters. Third, a consensus function is adopted to combine all clusters of the ensemble committee to obtain the final clusters.

Figure 1 shows the framework of a classical clustering ensemble method. By leveraging the consensus across multiple clusters, clustering ensembles give a generic knowledge framework for combining multiple clusters. Two factors crucial to the success of any clustering ensemble are the following:

(i) the construction of an accurate and diverse ensemble committee of diverse clusters;
(ii) the design of an appropriate consensus function to combine the results of the ensemble committee.

Strehl and Ghosh [16] introduced the clustering ensemble problem and provided three effective and efficient algorithms to address the problem: cluster-based similarity partitioning algorithm (CSPA), hypergraph partitioning algorithm (HGPA), and meta-clustering algorithm (MCLA). In order to benefit from the clustering ensemble approach, objects should be represented using different features. The number and/or location of initial cluster centers in iterative algorithms such as k-means can be varied. The order of data presentation in on-line methods such as BIRCH [27] can be varied. A portfolio of very different clustering algorithms can be jointly used. The experiments of Strehl and Ghosh show that clustering ensembles can be used to develop robust, superlinear clustering algorithms and to dramatically improve sets of subspace clusterings for different research domains.

Topchy et al. [17] extended clustering ensemble research in several regards. They introduced a unified representation for multiple clusterings and formulated the corresponding

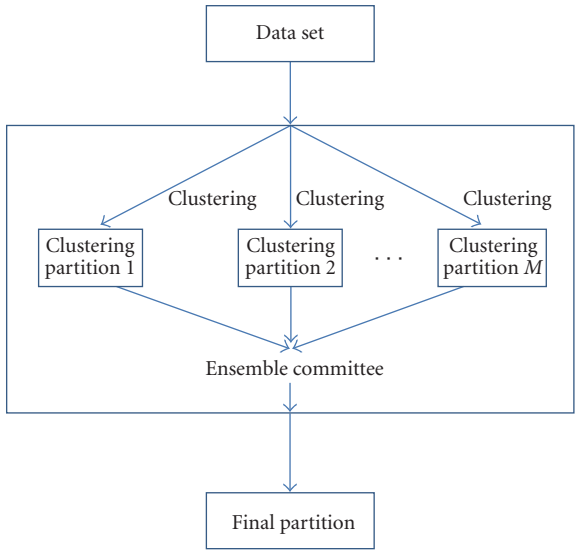


Figure 1: Framework of classical clustering ensemble.

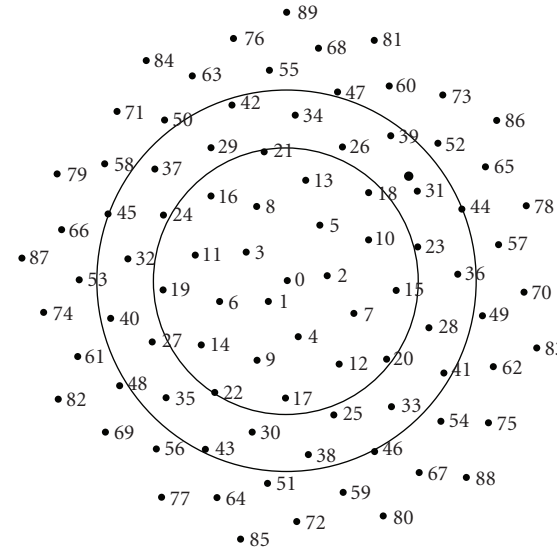


Figure 2: Points of the Fibonacci lattice in a complex plane.

categorical clustering problem. They proposed a probabilistic model of the consensus function using a finite mixture of multinomial distributions in a space of clusterings. They also demonstrated the efficiency of combining partitions generated by weak clustering algorithms that use data projections and random data splits.

Fred and Jain [18], based on the idea of evidence accumulation, considered that each partition is viewed as an independent evidence of data organization. Individual data partitions are combined based on a voting mechanism to generate a new $n \times n$ similarity matrix for n patterns. The final data partition of these n patterns is obtained by applying a hierarchical agglomerative clustering algorithm on this matrix. Kuncheva and Vetrov [19] used standard k-means that started from a random initialization to evaluate the stability of a clustering ensemble. From their experimental results they concluded that ensembles are generally more stable than single component clustering.

Clustering ensembles have demonstrated stable and accurate clustering results through a large number of experiments on real and synthetic data in the literature. We employ them here to group color video shots based on the features detected by our CGSIFT algorithm.

3. FEATURE EXTRACTION USING CGSIFT

3.1. *Retain color information by Fibonacci lattice- quantization*

24-bit color images have three color components: red, green, and blue, which are combined to generate over 16 million unique colors. Compared to a 256 grayscale image, a color image can convey much more visual information, providing the human perceptual system with much more details about the scene. However, not all 16 million colors are distinguishable by humans, particularly if colors are very similar.

Color quantization [26] is a sampling process of 3D color spaces (RGB, CIE Lab, HSV, etc.) to form a subset of colors known as the palette which are then used to represent the original color image. Color quantization is particularly convenient for color image compression, transmission, and display. Unlike most color quantization methods that generate a color palette with three separate color components for each color in the selected subset, quantization using Fibonacci lattices denotes colors using single scalar values. These scalar values can be used to denote visual "distance" between their corresponding colors. However, traditional color quantization algorithms such as uniform [29], median cut [29], and Octree [30] use palette indices only to point to the stored, quantized 3D color values. Attributes of this new quantization method are very useful for our application: we use Fibonacci lattice-quantization to convert colors into 256 scalar color indices and then use these indices to construct SIFT.

The Fibonacci lattice sampling scheme proposed in [25] provides a uniform quantization of CIE Lab color space and a way to establish a partial order relation on the set of points. For each different L value in CIE Lab color space, a complex plane in polar coordinates is used to define a spiral lattice as a convenient means for sampling. The following set of points in the (a, b) plane constitutes a spiral lattice:

$$z_n = n^{\delta} e^{j2\pi \cdot n\tau}, \quad \tau, \delta \in R, \ n \in Z. \tag{1}$$

Figure 2 shows an example of the spiral, Fibonacci lattice for $\tau = (\sqrt{5} - 1)/2$ and $\delta = 1/2$. Each point z_n is identified by its index n. Parameters τ and δ determine the axial distribution and the radial distribution of the points, respectively. If there exist N_L luminance (L) values and N_p colors in the corresponding (a, b) plane, for each color in the palette, the corresponding symbol is determined by adding its chrominance index n to a multiple of its luminance index i:

$$q = n + N_p \cdot i. \tag{2}$$

Consequently, the L, a, and b values for any color from the palette can be reconstructed from its symbol q. For a pixel p, with color components L_p, a_p, and b_p, the process of determining the closest palette point starts with finding the closest luminance level L_S from the N_L levels available in the palette. The luminance level L_S determines an (a, b) plane and one of the points z_n, $0 \le n \le N_p$, in that plane is the minimum mean square error (MSE) solution. The exact solution, q, is the point whose squared distance to the origin is the closest to $r_p^2 = a_p^2 + b_p^2$.

These L values can approximately denote the luminance levels of the image. Since the (a, b) plane is not circular, there will be points in the Fibonacci lattice whose colors are not valid in RGB color space. Thus, we label all these points as "range invalid." The points are given by $z_n = S\sqrt{n}e^{j(2\pi n\tau+\alpha_0)}$, where $\tau = (\sqrt{5}-1)/2$, $\alpha_0 = 0.05$, and $S = 1.5$. For a 400×300 image shown in Figure 3(a) having 43963 colors, the L component is quantized into 12 user-selected values (0, 10, 20, 30, 40, 50, 65, 70, 76, 85, 94, and 100). These L values and $N_p = 60$ points on each plane are used to construct the palette. Therefore, the size of palette is $12 \times 60 = 720$.

Figure 3(b) shows the quantized image with 106 colors in the palette. Each pixel is labeled by the one-dimensional symbol q, which not only is the index of an entry in the palette, but also represents the color information to some extent. Compared with Figure 3(c) of a 256 grayscale image, the red car and green trees are much easier to distinguish in the quantized image (Figure 3(b)) despite the grayscale frame having more levels (256) than the frame quantized by Fibonacci lattices (106). Easily distinguished colors can appear very similar in a grayscale image. Because human perception contrast in quantized images can be measured by the distance between the q symbols of two colors, it is more accurate to construct SIFT based on color indices to a palette constructed by Fibonacci lattice-quantization than using 256 levels of grayscale.

Using this attribute of Fibonacci lattice-quantization, we can retain color and visual contrast information in constructing accurate SIFT features from color video frames. According to (3), we perform a normalization process on quantized frames to obtain SIFT keypoint descriptors:

$$I_N(x, y) = \frac{q(x, y) - q_{\min}}{q_{\max} - q_{\min}} \times 255. \tag{3}$$

In the equation, $I_N(x, y)$ is the normalized value at the current position (x, y) in the image, $q_{\max}$ and $q_{\min}$ are maximum and minimum symbol values within the image, and $q(x, y)$ is the current pixel symbol value. After this normalization process, pixel symbol values are normalized to be between 0 and 255 and treated as gray-level values. The procedures in [20] can then be applied to this constructed grayscale image to obtain keypoint descriptors.

3.2. Join global context information into color SIFT

In order to extend local SIFT features to global features which can better describe the contents of the whole frame, we partition the image frame into symmetric regions to extract new global features. Assume that, after performing SIFT based on Fibonacci lattice-quantization, one image has N_I keypoints, each of which is a 128-dimension vector. We construct a template shown in Figure 4 to gather position information for constructing CGSIFT. This template consists of 24 distinct regions that increase in size as their distance from the center of the image increases. Generally, objects in the center of an image attract more attention than surrounding objects, which are often considered to be background or other trivial details. For example, in Figure 3, the vehicles are the main focus in the frame, and the trees and ground are background and relatively unimportant. Hence, smaller regions in the center part tend to describe more important contents, and larger regions on the periphery tend to depict less important details.

We give each region an order label to distinguish the partitions. The eight regions nearest the center are labeled as 1 to 8, the eight intermediate regions are 9 to 16, and outermost regions are 17 to 24. In each region, a mean color value is calculated based on the symbol q of each pixel within the region as follows:

$$V_{\text{ColorMean\_}i} = \frac{\sum_{i=1}^{\text{Num}P_i} q(x, y)}{\text{Num}P_i}, \quad i = 1, 2, \ldots, 24. \tag{4}$$

In (4), NumP_i is the number of pixels in region i, and $q(x, y)$ is the symbol q within the current region i. In a similar manner, we calculate the color variance in each region:

$$V_{\text{ColorVar\_}i} = \frac{\sum_{i=1}^{\text{Num}P_i} \left(q(x, y) - V_{\text{ColorMean\_}i}\right)^2}{\text{Num}P_i}, \quad i = 1, 2, \ldots, 24. \tag{5}$$

The third component of CGSIFT is the number of keypoints in each region $V_{\text{NumKeypoints\_}i}$, $i = 1, 2, \ldots, 24$. Since keypoints can reflect the salient information within the image, if one region has a higher number of keypoints, it should naturally be considered as a more important part of the image frame. The next two components of CGSIFT are the mean and variance of the orientation of keypoints in the region which are calculated by the original SIFT. These two components are calculated according to (6) and (7), respectively:

$$V_{\text{OrientationMean\_}i} = \frac{\sum_{i=1}^{\text{NumKey}_i} o(x, y)}{\text{NumKey}_i}, \quad i = 1, 2, \ldots, 24. \tag{6}$$

NumKey$_i$ is the number of keypoints in region i, and $o(x, y)$ is the orientation of the keypoint within current region i. Variances of the keypoints in each region are obtained as follows:

$$V_{\text{OrientationVar\_}i} = \frac{\sum_{i=1}^{\text{NumKey}_i} \left(o(x, y) - V_{\text{OrientationMean\_}i}\right)^2}{\text{NumKey}_i}, \quad i = 1, 2, \ldots, 24. \tag{7}$$

(a) Original frame (b) Quantized image (c) Grayscale image

Figure 3: (a) Original frame, (b) color quantized result using Fibonacci lattices, (c) corresponding gray frame.

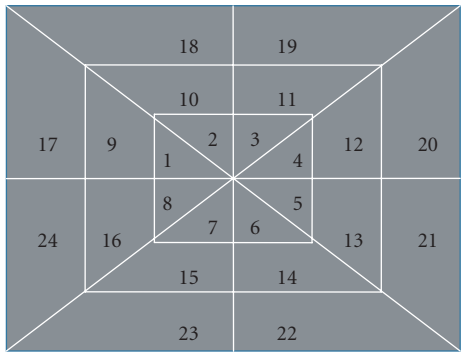


Figure 4: A new space description template for constructing CGSIFT.

These five components of the CGSIFT ($V_{\text{ColorMean\_}i}$, $V_{\text{ColorVar\_}i}$, $V_{\text{NumKeypoints\_}i}$, $V_{\text{OrientationMean\_}i}$, and $V_{\text{OrientationVar\_}i}$) are used to construct a $5 \times 24 = 120$-dimension feature vector of CGSIFT. Thus, CGSIFT combines the color, salient points, and orientation information simultaneously, resulting in more robust operation than can be obtained using single local SIFT grayscale feature. Moreover, CGSIFT can be used as the basis for color video shot detection.

4. VIDEO SHOT DETECTION USING CLUSTERING ENSEMBLES

As noted in Section 1, many different clustering methods have been used for shot detection. We use a novel clustering strategy with clustering ensemble for shot detection. Instead of using a single clustering method, clustering ensemble focuses on knowledge reuse [16] of the existing clustering groups so as to achieve a reasonable and accurate final partition result. k-means is a popular clustering method used widely in the literature since 1967. We choose k-means [28] as the basic clustering method to create clustering ensembles because of its simplicity and efficiency. The k-means algorithm attempts to minimize total intracluster variance as follows:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \text{Dist}(x_j, \mu_i), \tag{8}$$

where there are k clusters S_i, $i = 1, 2, \ldots, k$, μ_i is the centroid of each cluster S_i, and $\text{Dist}(x_j, \mu_i)$ is a chosen distance measure between a data point x_j and the cluster centroid μ_i. $\text{Dist}(x_j, \mu_i)$ can be Manhattan distance, Euclidean distance, or Hamming distance.

In order to depict the essential CGSIFT feature distribution as accurately as possible, we adopt random initial clustering centroids which generate different results depending on the initial centroids selected. The procedure of using a k-means single-clustering algorithm for processing a color frame consists of the following steps.

(1) Determine the numbers of clusters $K_1, K_2, \ldots, K_M$ for M k-means clusterings to form M clustering results on CGSIFT features of a set of frames.
(2) For each single k-means clustering, randomly select K_i, $i = 1, 2, \ldots, M$, CGSIFT features of M frames as the initial clustering centroids.
(3) Assign each frame to the group that has the closest centroid based on the Euclidean distance measure.
(4) After all frames have been assigned to a group, recalculate the positions of the current clustering K_i, $i = 1, 2, \ldots, M$, centroids.
(5) Repeat steps (3) and (4) until the centroids no longer move, then go to step (6).
(6) Repeat steps (2), (3), (4), and (5) until M separate k-means clusterings have been created.

Using the clustering groups generated by the repeated application of the k-means single-clustering method, the ensemble committee is constructed for the next ensemble step. We use the clustering-based similarity partition algorithm (CSPA) [16] as the consensus function to yield a combined clustering. (Complete details about CSPA can be found in [16].) The combined clustering is used as the final partition of the video shots.

5. PROCESSING TIME AND STORAGE SPACE ANALYSIS

The proposed shot detection algorithm is composed of two parts: feature extraction and clustering. Because Fibonacci lattice-quantization generates 1D scalar values rather than 3D vector values, it saves storage space. For example, for any 12-bit color palette (4096 colors) storing $R, G,$ and B values for each color, it needs 12 kilobytes of data for the palette.

Using a Fibonacci palette, fewer than 50 bytes are needed [25], because it is not required to store real color values. For processing time complexity, since it is not necessary to search 3D color values in the palette like traditional color quantization methods, Fibonacci lattice-quantization only uses a scalar value to reduce the searching time to assign color to each pixel.

Feature extraction is carried out on partitioned symmetric regions and five components of the feature are obtained by processing each pixel five times or less, so its processing time is less than $O(5 \times n^2)$. Compared to an image histogram [31], a classical and efficient feature in information retrieval with processing time complexity $O(n^2)$, the proposed feature extraction algorithm has the same order of magnitude ($O(n^2)$ for an $n \times n$ image) of computation. After the feature extraction process, each color image is represented by a 120-dimension vector of single-precision floating point numbers, requiring just 120×32 bits = 0.47 kilobytes storage space. However, for a frame or color image of 400×300, it will take up $400 \times 300 \times 24$ bits = 351.6 kilobytes. Compared to the original color frame storage requirement, feature-based image denotation reduces memory or disk space significantly, especially for large color video databases.

The group calculation of clustering ensemble is the most time-consuming portion of the proposed algorithm, especially when executed sequentially. However, parallel computing [32] can be applied to run each single clustering on a different processing unit at the same time, thus reducing the overall processing time for the clustering ensemble. To achieve parallel processing, the clustering ensemble could be implemented in hardware such as a field programmable gate array (FPGA), technology that has been used to accelerate image retrieval in recent years [31, 33, 34]. Another option is to use a graphics processing unit (GPU) for the computation. GPUs are known for their capability of processing large amounts of data in parallel. Implementation of the proposed algorithm for real-time applications is the next step of our research, and the details are beyond the scope of this paper.

Through the analysis of time and space complexities mentioned above, we can see that our CGSIFT feature extraction algorithm reduces computation time and storage space requirements to some extent and maintains more acceptable resource usage than histogram approaches. As for clustering ensemble computation, we propose constructive methods to lower its computational time while maintaining its high accuracy. Our complexity analysis did not include SIFT because it is a very robust local feature extraction algorithm that has been thoroughly analyzed in many studies described in the literature.

6. EXPERIMENTAL RESULTS

6.1. Test videos and ground truth

We used five videos, "campus," "autos," "Hoover Dam," "Colorado," and "Grand Canyon" to test CGSIFT, the proposed feature extraction algorithm, and the new clustering strategy. First, we used the "campus" and "autos" videos to test clustering accuracy via clustering ensembles relative to the original k-means single-clustering method. Then, in order to avoid the bias from the better clustering strategy we proposed in this paper, we applied the same k-means clustering to the proposed CGSIFT and the traditional SIFT for comparison. Finally, we used recall and precision rates as measures to test the performance of the proposed approach on the "Hoover Dam," "Colorado," and "Grand Canyon" videos and compare it with that of other clustering-based methods.

At the outset, the "campus" and "autos" videos were manually segmented into separate shots to establish the ground truth for comparison. Each video has a total of 100 frames. The "campus" footage contains 10 separate shots with abrupt changes, and each shot contains exactly 10 frames; "autos" contains 5 video shots with abrupt changes, each of which contains 20 frames. The key frames of both videos are shown in Figure 5.

6.2. Single clustering versus clustering ensembles and CGSIFT versus SIFT

Since we manually determined the number of shots in each video, we set the final partition number for both the clustering ensemble and k-means methods to 10 and 5 for "campus" and "autos," respectively. We used 10 groups of k-means single-clustering with different initial clustering centroids to construct the ensemble committee.

For each componential k-means single-clustering, we set 12 cluster centroids for the "campus" video and 8 cluster centroids for the "autos" video. We repeated $M = 10$ k-means single-clusterings for both of them to form 10 clustering results. After obtaining individual results from each of these 10 single-clusterings on 100 frames of "campus" and "autos," at the clustering ensemble stage, we set the number of centroids in the final partition to 10 and 5 for the two videos, respectively. CSPA was used to obtain final 10 and 5 partitions. For the comparative k-means clustering algorithm, we directly set its number of cluster centroids to be 10 and 5 at the beginning.

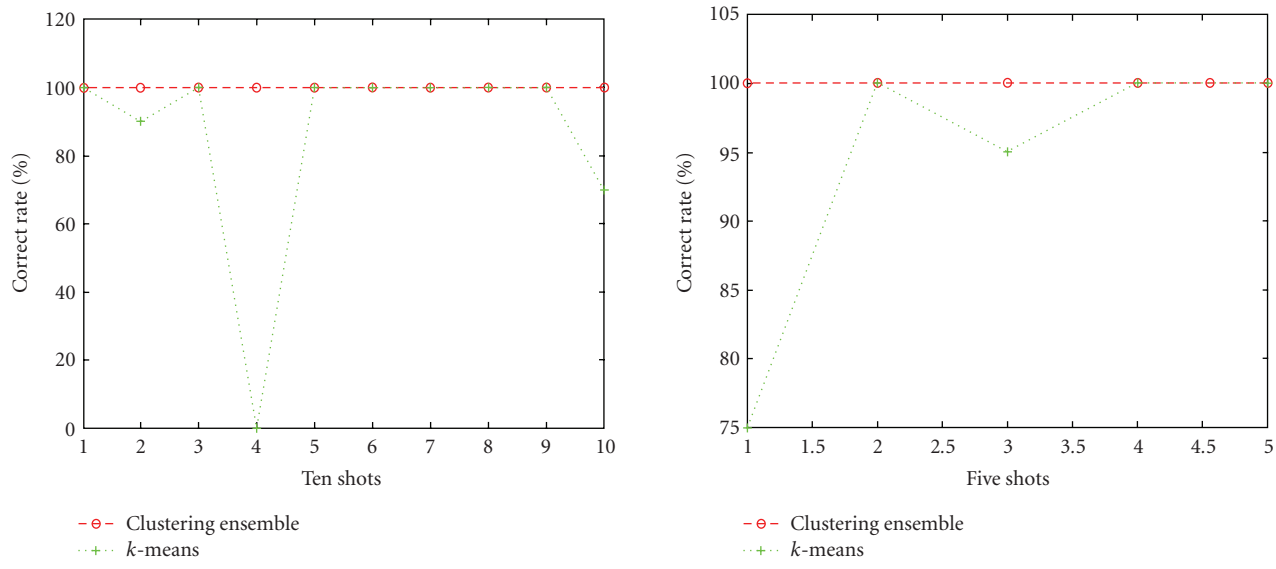
Figure 6 shows that the approach employing the clustering ensemble outperforms k-means single clustering. Figure 6(a) shows that, for 10 abruptly changed shots in "campus," the clustering ensemble partitioned all 10 video shots correctly. However, k-means wrongly partitions all frames of shot 4 as shot 3, resulting in 0% accuracy for that shot. Furthermore, for shots 2 and 10 respectively, only 90%, and 70% of the frames are correctly labeled. As shown in Figure 6(b), the clustering ensemble successfully grouped all frames of five video shots of the "autos" video into the correct shots. In contrast, k-means was unable to cluster any frames of shot 1 into the correct shot, and it could not correctly classify all frames in shot 3.

When SIFT is applied to the grayscale image, multiple keypoints are generated, each of which has a 128-dimension vector. We used the average value of these 128-dimension vectors to compare the CGSIFT performance via k-means clustering. As shown in Figure 7, although shot 4 of CGSIFT in video "campus" had 0% accuracy, the overall performance was still much better than SIFT. In processing the "autos" video, CGSIFT was clearly better than SIFT. Taken

(a) 10 key frames in video "campus"

(b) 5 key frames in video "autos"

FIGURE 5: The key frames of abrupt change shots of the videos (a) "campus" and (b) "autos."



(a) Video "campus" video shot detection result

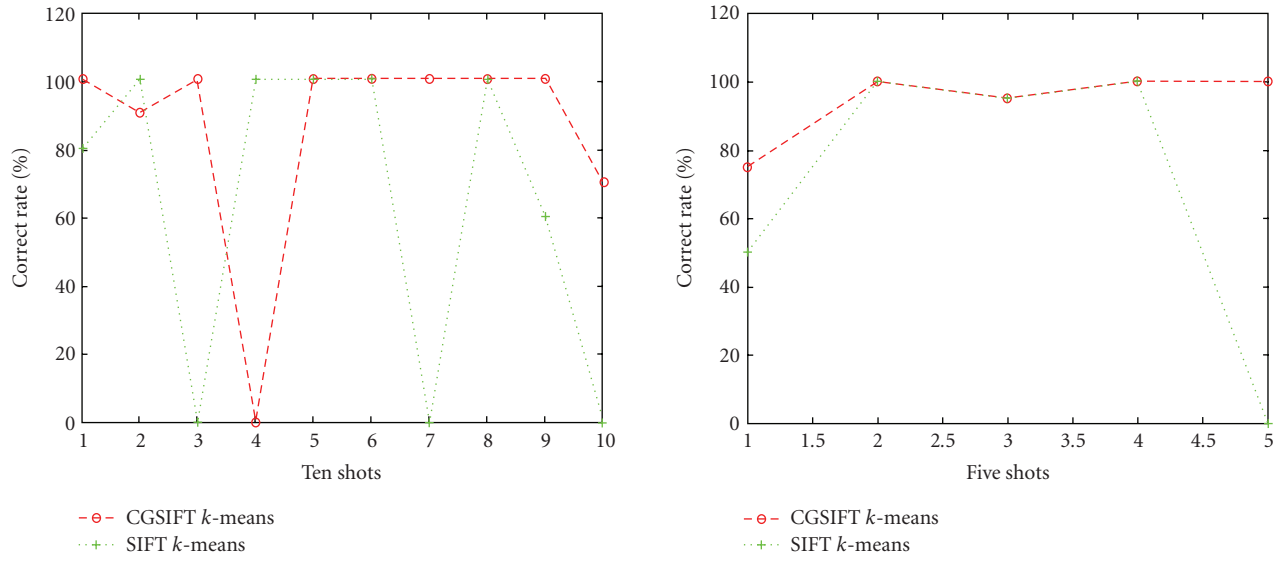
(b) Video "autos" video shot detection result

FIGURE 6: Performance comparison between clustering ensemble and k-means clustering.

in combination, the graphs in Figure 7 show that CGSIFT is a significant improvement over SIFT for the two test videos. This improvement is the result of CGSIFT considering color and space relationships simultaneously—SIFT describes only local contents in grayscale.

6.3. TRECVID

The TRECVID evaluation tools [35] were developed in conjunction with a text retrieval conference (TREC), organized to encourage research in information retrieval by providing a large test collection, uniform scoring procedures, and a forum for organizations interested in comparing their results. In order to evaluate the robustness of the proposed feature extraction and clustering ensemble algorithms for color video shot detection, we compared the proposed framework to fuzzy c-means [11] and SOM-based [7] shot detection methods. Because the main focus of this paper is the extraction of robust features and the application of a novel clustering strategy on *unsupervised* shot detection problem, we chose clustering-based shot detection methods for comparison instead of shot change-based detection algorithms. Unlike clustering-based shot detection algorithms, the latter consider the time and sequence information.



(a) Video "campus" video shot detection result

(b) Video "autos" video shot detection result

Figure 7: Performance comparison between CGSIFT and SIFT based on k-means clustering.

Table 1: Video data used for comparison among algorithms.

Video	No. of frames	No. of shots
Hoover Dam	540	27
Colorado	200	10
Grand Canyon	600	30

To compare performance, we used three videos ("Hoover Dam," "Colorado," and "Grand Canyon") from the open video project associated with TRECVID 2001. Because our algorithm is intended to construct a robust cut detection between shots, we manually removed gradual transition frames to form the ground truth. Table 1 shows summary information for the three selected videos.

We used recall and precision as performance metrics [36]. They are defined in the equations below:

$$\text{recall} = \frac{D}{D + D_M}, \qquad \text{precision} = \frac{D}{D + D_F}. \tag{9}$$

In the equations, D is the number of shot transitions correctly detected by the algorithm. D_M is the number of missed detections, and D_F is the number of false detections (the transitions that should have been detected but were not).

Similar to "campus" and "autos," we set the number of cluster centroids in each componential k-means single-clustering to be 33, 12, and 35 for "Hoover Dam," "Colorado," and "Grand Canyon," respectively, and the final partition numbers to be 27, 10, and 30. Using clustering ensemble and k-means clustering-based CGSIFT, we obtained the performance comparison in Table 2. It can be seen that recall and precision measures are better for the proposed clustering ensemble method than for fuzzy c-means, SOM, and k-means clustering using SIFT.

From Table 2, we can see that the proposed algorithm outperforms all other three methods. We note that the SOM-based method [7] used 6 features in MPEG-7 to detect the shots. Because we considered only the visual and not the audio content of the video in this paper, we used only five features: motion activity, color moments, color layout, edge histogram, and homogeneous texture for SOM. Its performance is worse than that of the proposed algorithm. Although five visual features were used to describe video content, each feature focused on just a single aspect of the content. Our CGSIFT feature obtained a better description. Furthermore, because fuzzy c-means [11] uses only a histogram—a feature which does not incorporate spatial relationship information—it is not as robust as the clustering ensemble approach. Its performance was the worst of the selected algorithms. Finally, the performance of k-means using SIFT feature was also worse than that of the proposed algorithm. This comparison indicates that the proposed method using CGSIFT feature and clustering ensemble is more efficient than the method using the original SIFT feature and k-means.

Existing video shot segmentation can be classified into two categories: shot change detection approach and clustering approach. The former measures the difference of adjacent frames to judge whether the difference is significant enough to detect the cut. On the other hand, the latter (clustering) approach needs prior knowledge of the number of clusters to group frames into corresponding sets. Both have their advantages and disadvantages. Because our research work focuses on robust color video frame feature extraction and a novel unsupervised learning method, we only selected clustering-based methods for comparison. In order to discriminate shots having similar visual content in the clustering process,

Table 2: Performance evaluation of clustering ensemble, fuzzy *c*-means, SOM, and *k*-means on SIFT using "Hoover Dam," "Colorado," and "Grand Canyon" videos.

Video	Clustering ensemble		Fuzzy *c*-means		SOM		*k*-means on SIFT	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Hoover Dam	100.0%	100.0%	65.4%	16.4%	92.3%	78.8%	88.5%	66.7%
Colorado	100.0%	100.0%	66.7%	17.0%	88.9%	90.0%	100.0%	60.0%
Grand Canyon	96.6%	90.6%	66.7%	13.8%	89.7%	82.9%	93.1%	58.0%

some constraints such as temporal changes and sequence number could be added.

7. CONCLUSIONS

We have presented a color feature extraction algorithm and clustering ensemble approach for video shot detection. First, considering that the single color index value of Fibonacci lattice-quantization can more accurately represent color than can grayscale, we use this quantization method to preprocess color frames of the video. Then, according to the template reflecting spatial relationships, CGSIFT is constructed which contains color and salient point information to provide color global features. A clustering ensemble is used to group different frames into their corresponding shots so as to detect the boundaries of the video shots. Experiments show that the proposed video shot detection strategy has better performance than the strategy using *k*-means single-clustering and the SIFT descriptor.

In our future work, we will address the challenge of creating descriptors that incorporate color, space, and texture simultaneously, ideally resulting in further increases in performance and more robust operation. Furthermore, we will address the problem of joining constraint information with traditional clustering ensembles.

REFERENCES

[1] Y. Li and C. J. Kuo, *Video Content Analysis Using Multimodal Information: For Movie Content Extraction Indexing and Representation*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[2] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos," *Multimedia Systems*, vol. 7, no. 5, pp. 359–368, 1999.

[3] W. Tavanapong and J. Zhou, "Shot clustering techniques for story browsing," *IEEE Transactions on Multimedia*, vol. 6, no. 4, pp. 517–527, 2004.

[4] C. W. Ngo, T. C. Pong, and H. J. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 446–458, 2002.

[5] H. Lu, Y. P. Tan, X. Xue, and L. Wu, "Shot boundary detection using unsupervised clustering and hypothesis testing," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 2, pp. 932–936, Chengdu, China, June 2004.

[6] X. D. Luan, Y. X. Xie, L. D. Wu, J. Wen, and S. Y. Lao, "AnchorClu: an anchor person shot detection method based on clustering, international conference on parallel and distributed computing ," in *Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT '05)*, pp. 840–844, 2005.

[7] M. Koskela and A. F. Smeaton, "Clustering-based analysis of semantic concept models for video shots," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, vol. 2006, pp. 45–48, Toronto, ON, Canada, July 2006.

[8] J. Xiao, H. Chen, and Z. Sun, "Unsupervised video segmentation method based on feature distance," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision (ICARCV '04)*, vol. 2, pp. 1078–1082, Kunming, China, December 2004.

[9] H. Okamoto, Y. Yasuqi, N. Babaquchi, and Y. Kitahashi, "Video clustering using spatio-temporal image with fixed length," in *Proceedings of the International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 53–56, Lusanne, Switzerland, August 2002.

[10] Z. Lei, L. D. Wu, S. Y. Lao, G. Wang, and C. Wang, "A new video retrieval approach based on clustering," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1733–1738, Shanghai, China, August 2004.

[11] C. C. Lo and S. J. Wang, "Video segmentation using a histogram-based fuzzy c-means clustering algorithm," in *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 920–923, Melbourne, Australia, December 2002.

[12] P. Wang, R. Cai, and S. Q. Yang, "Improving classification of video shots using information-theoretic co-clustering," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS '05)*, vol. 2, pp. 964–967, May 2005.

[13] H. C. Lee, C. W. Lee, and S. D. Kim, "Abrupt shot change detection using an unsupervised clustering of multiple features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '00)*, vol. 4, pp. 2015–2018, Istanbul, Turkey, June 2000.

[14] C. J. Fu, G. H. Li, and J. T. Wu, "Video hierarchical structure mining," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 3, pp. 2150–2154, Guilin, China, June 2006.

[15] P. Berkhin, "Survey of clustering mining techniques," Tech. Rep., Accrue Software, San Jose, Calif, USA, 2002.

[16] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, no. 3, pp. 583–617, 2003.

[17] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[18] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[19] L. I. Kuncheva and D. P. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1798–1808, 2006.

[20] D. G. Lowe, "Distinctive image feature from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 506–513, Washington, DC, USA, June 2004.

[22] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, vol. 2, pp. 1150–1157, Kerkyra, Greece, September 1999.

[23] L. Ledwich and S. Williams, "Reduced SIFT feature for image retrieval and indoor localisation," in *Proceedings of the Australasian Conference on Robotics and Automation (ACRA '04)*, Canberra, Australia, 2004.

[24] T. Deselaers, "Features for image retrieval," M.S. thesis, Human Language Technology and Pattern Recognition Group, RWTH, Aachen University, Aachen, Germany, 2003.

[25] A. Mojsilović and E. Soljanin, "Color quantization and processing by fibonacci lattices," *IEEE Transactions on Image Processing*, vol. 10, no. 11, pp. 1712–1725, 2001.

[26] A. K. Jain, *Fundamentals of Digital Image Processing*, Information and System Sciences Series, Prentice Hall, Upper Saddle River, NJ, USA, 1989.

[27] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the ACM International Conference on Management of Data (SIGMOND '96)*, vol. 25, pp. 103–114, Montreal, Canada, June 1996.

[28] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[29] P. Heckbert, "Color image quantization for frame buffer display," in *Proceedings of the ACM Conference on Computer Graphics and Interactive Techniques*, vol. 16, pp. 297–307, Boston, Mass, USA, July 1982.

[30] M. Gervautz and W. Purgathofer, "A simple method for color quantization: octree quantization," in *New Trends in Computer Graphics*, pp. 219–231, Springer, Berlin, Germany, 1988.

[31] L. Kotoulas and I. Andreadis, "Colour histogram content-based image retrieval and hardware implementation," *IEE Proceedings on Circuits, Devices and Systems*, vol. 150, no. 5, pp. 387–393, 2003.

[32] A. Grama, A. Gupta, G. Karypis, and V. Kumar, *Introduction to Parallel Computing*, Addison-Wesley, Reading, Mass, USA, 2nd edition, 2003.

[33] K. Nakano and E. Takamichi, "An image retrieval system using FPGAs," in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '03)*, pp. 370–373, January 2003.

[34] A. Noumsi, S. Derrien, and P. Quinton, "Acceleration of a content-based image-retrieval application on the RDISK cluster," in *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS '06)*, Miami, Fla, USA, April 2006.

[35] http://www-nlpir.nist.gov/projects/t01v/.

[36] C. Cotsaces, N. Nikolaidis, and I. Pitas, "Shot detection and condensed representation—a review," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 28–37, 2006.