Action Scene Extraction from Movie using Finite State Machine

Mr. Maulik P. Patel[1] , Mr. Anirudha Singh[2], Mr. Mehul Amin[3]

Dept. of Electronics and Communication Engineering,
Dr. Jivraj Mehta Institute of Technology, Nr. Sankara Eye Hospital, NH No.8,
Mogar,
Anand, Gujarat, Pin: 388340-INDIA

mpp_1515@yahoo.com[1], aniranu1984@gmail.com[2], amin.mehul@yahoo.com[3]

Abstract— **To attract the viewers into paying to see full movie, the creation of movie trailers is an essential part of movie industry. Action scene is the main factor of a movie trailer. In this paper, we propose an automatic action scene extraction algorithm using finite state machine based on analyzing audio-visual features. The input video is first decomposed into shot. Then audio features compute and applied to support vector machine in order to determine whether it is speech, silence or music. Extracted Audiovisual features are to feed into finite state machine in order to determine action scene classification from others scene.**

Keywords— *vedio-audio content analysis; shot change detection, finite state machine; support vector machine;*

I. Introduction

In last few years, there is amazing increased use of digital video documents. Viewer may have interest in some specific parts rather than the whole video. But due large amount of video data, the efficient retrieving the parts user wants to have is difficult to get fast enough. Effective abstraction tools must be available, so that the users could retrieve the interested parts quickly and with more accuracy. In film industry, trailer of a movie is made such that the people get attract and take interest into it. Particularly, if action scenes are there, people get excited. Thus, automatic action scene extraction has much importance for the filmmakers. Also methods and techniques for automatically separate out video documents based on the contents are very important for video browsing and organization.

In recent years there is not much efforts have been devoted to extract the action scenes from video. Early video database systems segment video into shots, and extract key frames from each shot to represent it. M. M. Yeung and B. L. Yeo [1] proposed a cluster based scene detection. They made N clusters, one for each shot and stopped when the difference between 2 clusters is above a predefined threshold. Then they merge the most similar pair of clusters together and same process was repeated. Finally they classify the fight scenes and non fight scenes Frameworks [2], analyze the structure of the movie scene and classify scenes into more specific categories. Yun Zhai, Zeeshan Rasheed, and Mubarak Shah propose that the Finite State Machines (FSM) are suitable for detecting and classifying scenes and demonstrate their usage for three types of movie scenes; conversation, suspense and action. Their framework utilizes the structural information of the scenes together with the low and mid-level features. Low level features of video including motion and audio energy and a mid-level feature, face detection, are used in their approach. The transitions of the FSMs are determined by the features of each shot in the scene.

In [3], Liang-Hua Chen, Chih-Wen Su, Chi-Feng Weng and Hong-Yuan Mark Liao propose an automatic action scene detection algorithm based on the analysis of high -level video structure. The input video is first decomposed into a number of basic components called shots. Then, shots are grouped into semantic related scenes by taking into account the visual characteristics and temporal dynamics of video. Based on the filmmaking characteristics of action scene, some features of the scene are extracted to feed into the support vector machine for classification.

L. Chen and M. T. Ozsu [4] proposed a rule based model to extract simple action scenes. They analyze video editing rules and the temporal appearance patterns of shots in action scenes of video. Based on it, they deduced a set of rules to recognize action scenes. Based on these rules, a finite state machine is designed to extract dialog or action scenes from videos automatically. L. Chen, S. J. Rizvi, and M. T. Ozsu [5] proposed a rule based model along with audio clues inserted into FSM modem in order to achieve higher accuracy. In [4], they are only concerned on visual features. This model is extended in [5] by incorporating audio features. In this paper, we have find out audio features such as Zero crossing rate(ZCR), Average energy, Silence ratio, Pitch ratio and visual

II. SYSTEM OVERVIEW

An overall system diagram for our approach is presented in figure 1. The input to the system is a movie file, while the output is an action scene. All of the analysis is undertaken using AVI video and PCM encoded WAV audio having sampling rate 8000 KHz. The first step is the shot change detection using color histogram [6]. The second step is the extraction of shot features, consequent to signal-based, low-level features.

The sub-shot audio features were chosen in order to classify the audio into a number of relevant categories such as speech, music or silence. These are useful for action scene detection. The features extracted are: silence ratio, high zero crossing rate ratio, Average energy and pitch ratio. Similarly, the motion features were chosen in order to accurately represent the amount and type of movement at any one time in the movie. Given the shot features, a shot-level feature vector is then created.

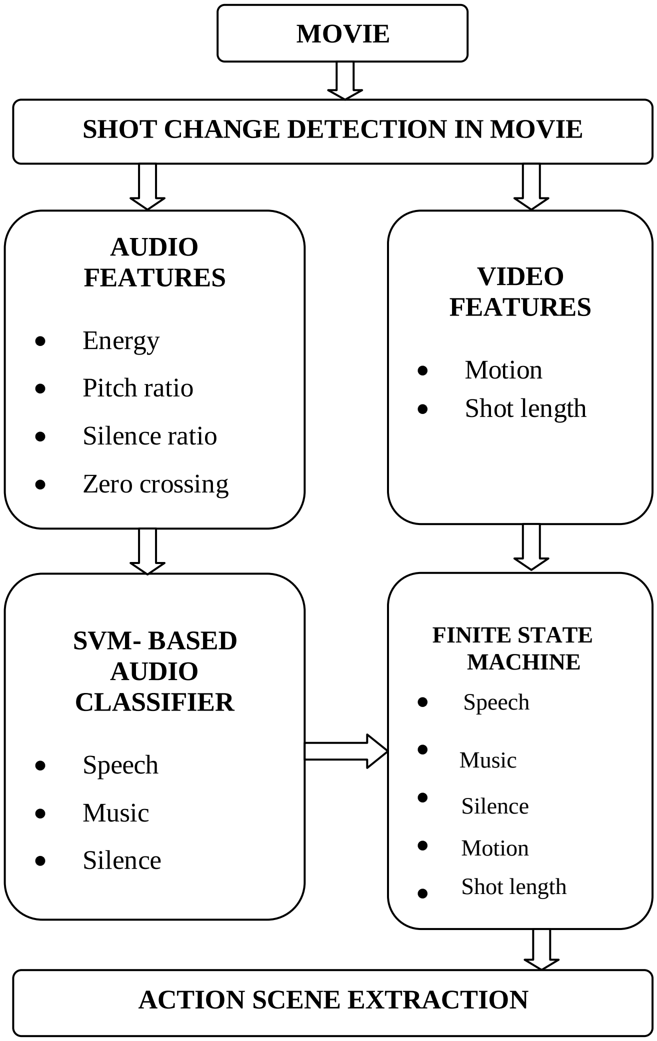

Fig.1 system block diagram

The sub-shot audio features are used as a basis for audio classification. Four audio classes are created: speech, music silence and other audio. Using a support vector machine (SVM) based classification method, with the sub-shot audio features as inputs and output is any one class. A more detailed explanation of this process can be found in [7]. Using the shot boundary information, the shot length can be calculated. This gives an indication of the speed of editing.

The shot feature vector contains all of the audio, motion and shot length for each shot. This is feed into finite state machine. However if short length and high motion pattern occurred then shot feed into action shot categories and also action scene contain high audio energy and low silence. That means it contain music.

III. SHOT CHANGE DETECTION

There are so many techniques to detect shot boundaries such as edge detection, shot boundary detection using macroblocks, wipe change detection, background tracking technique, color histogram etc. The color histogram technique is fast, very efficient and less complication. In this method frame-to-frame similarities based on colors is computed. After computing the inter-frame similarities, a threshold can be used to indicate shot boundaries.

The algorithm for shot detection is shown in figure 2. The video frame is first divided into R, G and B component and also divided frame into four blocks and takes histogram for each block for every component. Color histogram comparison ($d_{r,g,b}(f_i,fj)$) is calculated by histogram comparison of each color space of adjacent two frames f_i and f_j. It is defined as (1)

$$d_{r,g.b}(f_i,f_j) = \left(|H_i^r(k) - H_i^r(k)| + |H_i^g(k) - H_i^g(k)| + |H_i^b(k) - H_i^b(k)|\right) \quad (1)$$

Now, color histogram for each block is defined as (2)

$$d(f_i,f_j) = \sum_{bl=1}^{m} DP(f_{i,}f_j,bl) \qquad (2)$$

Where m is the number of block and frame difference of block is defined as (3)

$$DP(f_i,f_j,bl) = |H_i(k,bl) - H_i(k,bl)| \qquad (3)$$

$H_i(k,bl)$ is the histogram distribution of k position of the frame (f_i) block(bl) and m is the number of total blocks. However the d(fi,fj) of shot which contain high motion is very high. So, we subtract the previous difference from next one if this difference is greater than predefined threshold than shot boundary are detected. This algorithm is almost 100% accurate.

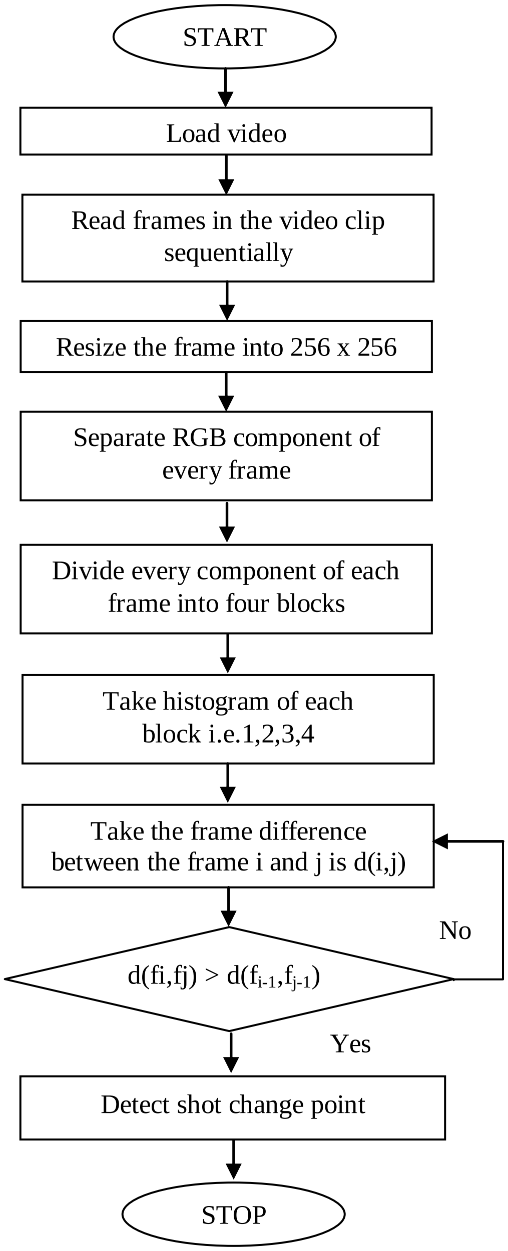
START
Load video
Read frames in the video clip sequentially
Resize the frame into 256 x 256
Separate RGB component of every frame
Divide every component of each frame into four blocks
Take histogram of each block i.e.1,2,3,4
Take the frame difference between the frame i and j is d(i,j)
No
d(fi,fj) > d(f_{i-1},f_{j-1})
Yes
Detect shot change point
STOP

Fig. 2 Flowchart of shot change detection

IV. AUDIO-VISUAL FEATURES EXTRACTION

In order to capture the characteristics of audio data, we select four audio features from two domains: average energy, silence ratio, zero crossing rate and pitched ratio and to visual features.

Average energy

Let x_i (n), n = 1, ..., N the audio samples of the i^{th} frame, of length N. Then, for each frame i the energy is calculated according to the equation:

$$E(i) = \frac{1}{N}\sum_{n=1}^{N} | x_i \ (n) |^2 \qquad (4)$$

This simple feature can be used for detecting silent periods in audio signals, but also for discriminating between audio classes. The Average energy of music is higher than in speech and silence.

Silence ratio

Silence Ratio (SR) is defined as the ratio of the amount of silence in an audio piece to the length of the piece. SR is a useful statistical feature for audio classification; it is usually used to differentiate music from speech. Normally speech has higher SR than music. We divide an audio of shot into 50 samples and if energy of this window is higher than predefined value than this window is consider as a silence frame.

Zero crossing rate

Zero Crossing Rate (ZCR) is the rate of sign-changes of a signal, i.e., the number of times the signal changes from positive to negative or back, per time unit. It is defined according to the equation:

$$Z(i) = \frac{1}{2N}\sum_{n=1}^{N} | \mathrm{sgn}[x_i \ (n)] - \mathrm{sgn}[x_i \ (n-1)] | \qquad (5)$$

where sgn (.) is the sign function. This feature is actually a measure of noisiness of the signal. Therefore, it can be used for discriminating noisy environmental sounds, e.g., rain. Furthermore, in speech signals, the σ^2 / μ ratio of the ZCR sequence is high, since speech contains unvoiced (noisy) and voiced parts and therefore the ZCR values have abrupt changes. On the other hand, music, being largely tonal in nature, does not show abrupt changes of the ZCR.

Pitched ratio

If frequency is high then pitch is greater. If frame is not belongs to silence frame than find out pitch of that frame and ratio of pitched frame to the total length of audio clip is known as pitch ratio. The pitched ratio of music is highest and silence has lowest pitched ratio. So, these feature also useful to classify the music from speech and silence.

Motion intensity

Motion is a visual feature which is essential to capture temporal variation of video. It also reveals the correlations between frame sequences within a video scene. To characterize the degree of motion within a scene, the average motion intensity is computed. First divide frame into number of block and take pixel-wise difference between subsequent frames in order to detect the number of block change.

Overview of Support Vector Machine

SVM classify the data between two classes that mean it gives output yes or no. Because of that if we want to be categorized n type of scene we required $\binom{n}{2}$ SVM machines. We will take simplest case: linear machines trained on separable data Again label the training data $\{x_i, y_i\}$, i = 1, .,n $y_i \in \{-1, 1\}$, xi $\in R^d$. Suppose we have some hyper plane which separates the positive from the negative examples (a separating hyperplane). The points x which lie on the hyper plane satisfy w.x + b = 0, where w is normal to the hyper plane, $\frac{|b|}{\|w\|}$ is the perpendicular distance from the hyper plane to the origin, and $\|w\|$ is the Euclidean norm of w. Let d+ (d−) is the shortest distance from the separating hyperplane to the closest positive (negative) example. Define the margin of a separating hyper plane to be d+ + d−. For the linearly separable case, the support vector algorithm simply looks for the separating hyperplane with largest margin. This can be formulated as follows: suppose that all the training data satisfy the following constraints:

$$x_i .w + b \geq +1 \text{ for } y_i = +1 \quad (6)$$

$$x_i .w + b \leq -1 \text{ for } y_i = -1 \quad (7)$$

These can be combined into one set of inequalities:

$$y_i (x_i .w + b) - 1 \geq 0 \;\; \forall_i \quad (8)$$

Now consider the points for which the equality in Eq. (6) holds (requiring that there exists such a point is equivalent to choosing a scale for w and b). These points lie on the hyperplane H1: $x_i .w + b = 1$ with normal w and perpendicular distance from the origin $\frac{|1-b|}{\|w\|}$. Similarities, the points for which the equality in Eq. (7) holds lie on the hyper plane H2: $x_i .w + b = -1$, with normal w perpendicular distance from the origin$\frac{|-1-b|}{\|w\|}$. Hence distance d+ = d− = $\frac{1}{\|w\|}$. and the margin is simply $\frac{2}{\|w\|}$. Note that H1 and H2 are parallel (they have the same normal) and that no training points fall between them.

Thus we can find the pair of hyperplanes which gives the maximum margin by minimizing $\|w\|^2$, subject to constraints (8). Those training points for which the equality in Eq. (8) holds (i.e. those which wind up lying on one of the hyperplanes H1, H2), and whose removal would change the solution found, are called support vectors; they are indicated in Figure 3 by the extra circles.

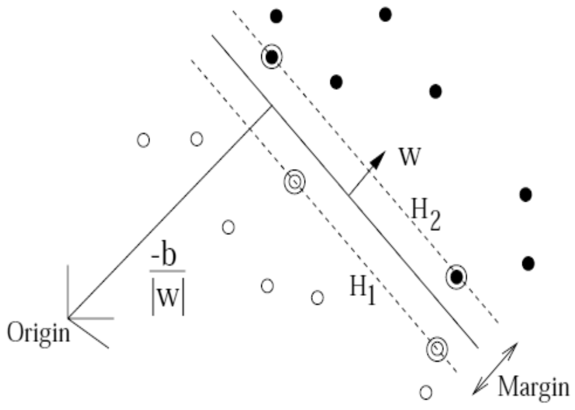

Fig. 3 Linear separating hyperplane.

If dataset is nonlinear then how the above methods can be generalized to classify nonlinear dataset. First notice that the only way in which the data appears in the training problem, is in the form of dot products, x_i . x_j. Now suppose we first mapped the data to some other (possibly infinite dimensional) Euclidean space H as shown in figure 4, using a mapping which we will call Φ:

$$\Phi: R_d \rightarrow H.$$

Then of course the training algorithm would only depend on the data through dot products in H, i.e. on functions of the form $\Phi(x_i). \Phi(x_j)$. Now if there were a “kernel function” K such that $K(x_i, x_j) = \Phi(x_i). \Phi(x_j)$, we would only need to use K in the training algorithm.

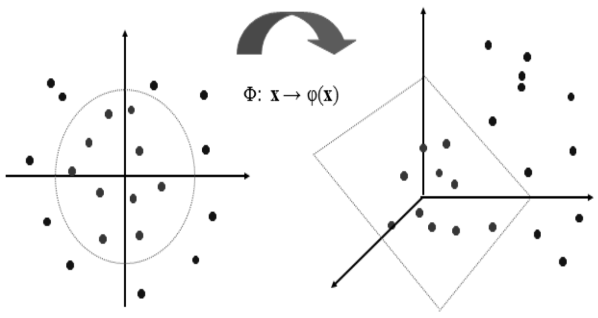

Fig. 4 Convert input space to feature space

Two kernel functions are frequently used:

1. Polynomial kernel:

$$K(X, S_i) = (X^T.S_i + 1)^d \quad (9)$$

Where S_i are support vectors which are determined from training data and d = 1, 2,.... is the degree of the polynomial.

2. Gaussian Radial Basis Function (RBF) kernel:

$$K\,(X, S_i) = e^{\|X-S_i\| / \sigma^2} \qquad (10)$$

Where $\sigma > 0$ is defined to be the global basis function width.

The more detailed explanation on support vector machine of audio classification is in [7,8] and detailed expiation of support vector machine is in [9].

V. Finite state machine for action scene detection

The shot in the action scene has a short length with high motion and also sound has a high energy and low silence. That means the sound is mostly music during action. Assume that there is more than five shot in action scene.

As shown in fig state S is start state and state A is acceptance state. If action shot is detected then it is fall into state 1. Five continuous action shot occurred then this pattern is accepted as an action sequence pattern. In action scene some of the shot like actor jump from high building, running behind someone contain typically long shot. So, if that type of shot occurred which has long shot length or low motion then it is fallen into intermediate stage. However, continuous non-action shot occurred then it reached to start state and action sequence pattern is finished.

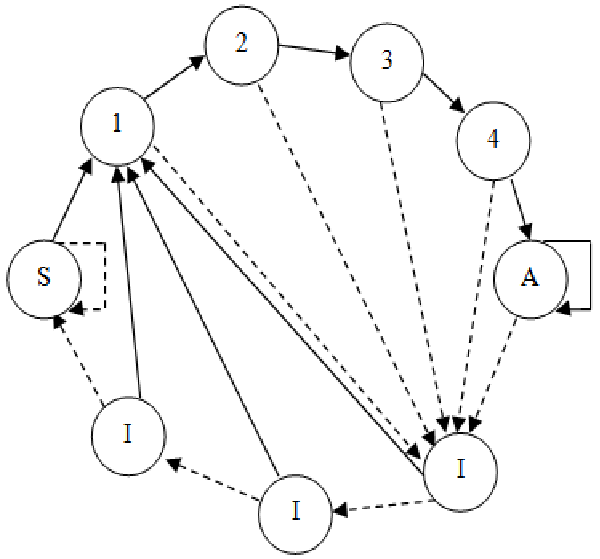

Fig. 5 Finite State Machine for Action sequence detection

VI. Conclusion and future work

In this paper, we examined shot change detection technique and support vector machine for classification of audio clip of shot whether it is speech, silence or music. Zero crossing rate Average energy, Pitched ratio and Silence ratio are powerful features to detect whether scene contain speech, music, silence and other audio. For this task SVM is excellent classifier. Shot detection technique based on colorhistogram is 100% accurate too. Some of the shots are shown in figure 6.

In future we are going to develop FSM for action sequence detection from video as shown in figure 5.

References

[1] M. M. Yeung and B. L. Yeo, Time-constrained clustering for segmentation of video into story units, in proceedings of 13th International Conference on Pattern Recognition, 1996, pp. 375-380.

[2] Yun Zhai, Z. Rasheed, M. Shah, “A Framework for Semantic Classification of Scenes using Finite State Machines.”

[3] Liang-Hua Chen, Chih-Wen Su, Chi-Feng Weng and Hong-Yuan Mark Liao, “Action Scene Detection with Support Vector Machines” Journal of Multimedia, Vol. 4, No. 4, August 2009.

[4] L.Chen and M. T. Ozsu, “Rule-based scene extraction from video,” in Proceedings of IEEE International Conference on Image Processing, pp. 737–740, September 2002.

[5] L. Chen, S. J. Rizvi and M. T. Ozsu, “Incorporating Audio Cues into Dialog and Action Scene Extraction” In Proceedings of SPIE Conference on Storage and Retrieval for Media Database, pp.252 - 264, San Jose, CA, 2003.

[6] Priyadarshinee Adhikari, Neeta Gargote, Jyothi Digge, and B.G. Hogade “Abrput Scene Change Detection ” World Academy of Science, Engineering and Technology 42 2008.

[7] S.-Z. Li and G. Guo, “Content-based audio classification and retrieval by support vector machines,” in PRCM (invited talk), 2000.

[8] Lei Chen, Sule Gunduz and M. Tamer Ozsu“Mixed type audio classification with support vector machine” in Proc. ICME 2006.

[9] C. J. C. Burges., “A tutorial on support vector machines pattern recognition,” Data Mining and Knowledge Discovery,1998.

VII. RESULT FOR SHOT CHANGE DETECTION

Fig 6. Shown the first frame of shot. The colour histogram difference of last frame of previous shot and first frame of coming shot is very large compared to previous difference as shown in fig 7.

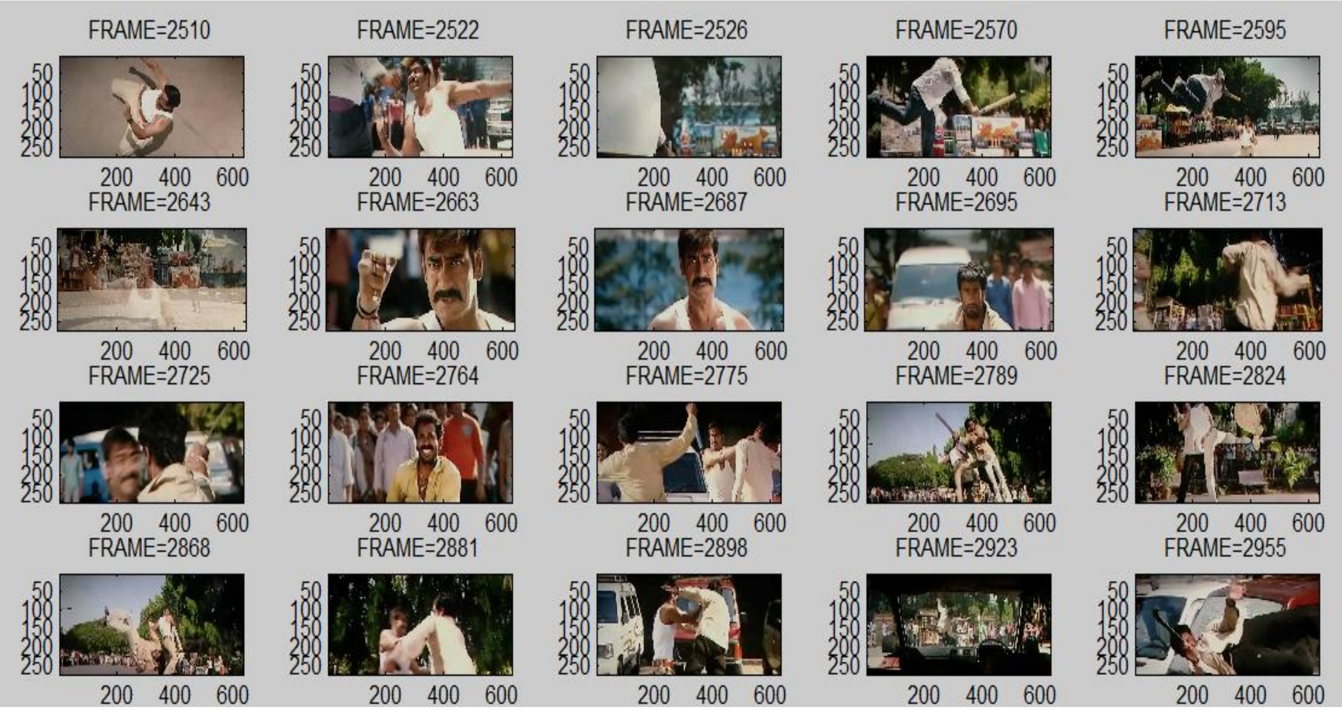

Fig. 6 First frame of shot

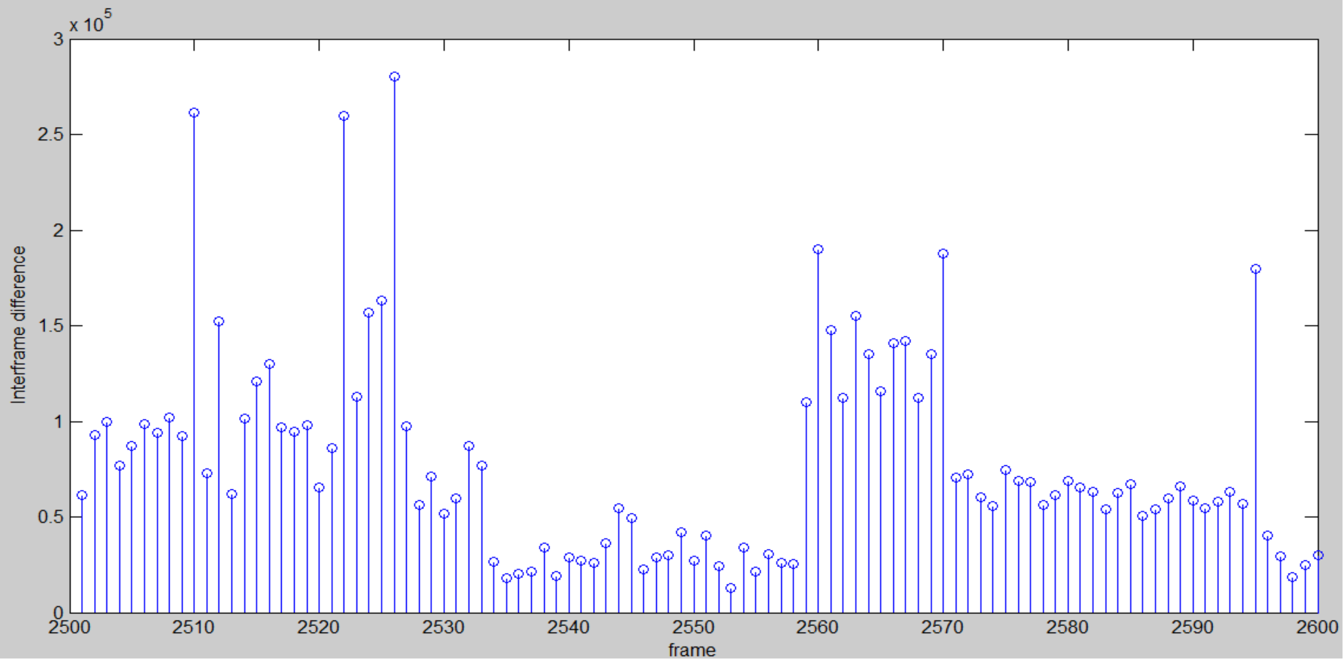

Fig. 7 Color histogram