International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.11 (2015), pp.385-396
http://dx.doi.org/10.14257/ijmue.2015.10.11.37

A Key Frame Extraction Algorithm Based on Clustering and Compressive Sensing

Lei Pan, Xin Shu and Ming Zhang

School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003, China
just_panlei@just.edu.cn

Abstract

Key frame extraction is considered as one of the most critical issues in content-based video retrieval technology (CBVR). In this paper, an efficient key frame extraction algorithm based on unsupervised clustering and compressive sensing is proposed. Firstly, three types of filters with various scales are employed to generate high dimensional feature of each frame in one shot, which will be projected to low dimensional feature by a very sparse random projection matrix that satisfies the condition of Restricted Isometry Property (RIP), and then sub-shot segmentation is conducted by an unsupervised clustering method in order to divide one shot into sub-shot collections, in which each class of clustering represents one sub-shot. Finally, the Bhattacharyya coefficient is used to measure the similarity between frame and class center, the frame with the maximum similarity value is selected as the key frame in each sub-shot. The experimental results demonstrate that the proposed method could extract key frames efficiently and properly.

Keywords: *Key frame extraction, unsupervised clustering, compressive sensing*

1. Introduction

Compared with text, audio and image, video is human's favorite multimedia data type due to its abundant amount of information and intuitive experience. With rapid progress of computer and network technologies, the amount of video data increases fast, massive storage and frequent retrieval inevitably lead to huge spatio-temporal cost [1], how to manage, classify and retrieve the massive video data efficiently becomes a challenging issue in the field of multimedia management, and then content-based video retrieval technology (CBVR) appears. Figure 1 shows a typical four-layer video structure from bottom to top, a corresponding CBVR system includes several technologies, such as shot boundary detection, video summary, video abstract, scene analysis, and so on, among which key frame extraction plays a critical role in video indexing, query and browse, and it is the connection between shot detection and advanced semantic information acquisition of video. Because of its importance, more and more attentions have been paid to key frame extraction in recent years.

Key frames are a subset of all still frames extracted from different video shots [2], and can be theoretically defined as the most informative and representative frames that reflect most of the visual contents in one video [3-4]. According to the definition, the purpose of key frame extraction algorithm is to extract correct and proper key frames from each video shot, which can perfectly represent the whole visual and semantic contents of the shot. Through key frame extraction, the amount of video storage can be compressed and the video indexing complexity can be decreased, which would speed up and simplify the work of quick browse and query. In general, the traditional key frame extraction methods can be categorized as shot boundary-based methods, motion analysis-based methods, clustering-based methods, visual content-based methods, compressed domain-based methods, *etc.*

ISSN: 1975-0080 IJMUE

The rest of this paper is organized as follows. The related work, including key frame extraction methods, compressive sensing theory, the motivation of the proposed algorithm, is briefly introduced in Section 2. The proposed algorithm is discussed in detail in Section 3. The experimental results are demonstrated in Section 4. Finally the conclusion and future work are described in Section 5. Note that because the emphasis of this paper is mainly on the key frame extraction, all shot boundaries are supposed to be detected correctly in advance.

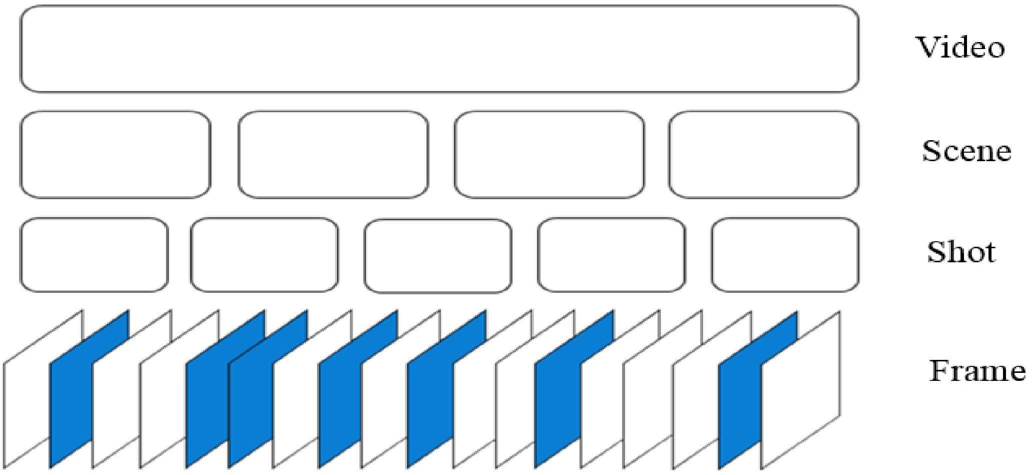


Figure 1. A Typical Video Structure, a Blue Rectangle Means One Key Frame

2. Related Work

In this section, the traditional key frame extraction methods, as well as some new methods appeared recently are briefly reviewed. After that, the compressive sensing concept and the motivation of the proposed method are introduced.

2.1. Existing Key Frame Extraction Methods

Shot boundary-based methods suppose that visual and semantic contents in one shot are mainly stable and change slightly, only the first, the last and the middle frames in each shot are selected as the key frames, which is obviously not robust to most videos. Motion analysis-based methods select key frames in local minimum through the computation of optical-flow, however, the computational cost is huge and the result is not always accurate. Clustering-based methods and visual content-based methods use the difference between adjacent frames to select key frames, and these two methods can be easily affected by noise and motion. Compressed domain-based methods use the compressed feature of frames to determine key frames, the efficiency relies on the compressed algorithm and few features could be used, which leads to the instability.

Some novel and improved key frame extraction methods are presented recently. X. Liu, *et al* [1] proposed a method based on Maximum a Posteriori (MAP) to estimate the positions of key frames. N. Ejaz, *et al*. [2] proposed an aggregation mechanism to combine the visual features extracted from the correlation of RGB color channels, the color histogram and the moments of inertia to extract key frames. Q. Xu, *et al* [3, 5] developed a Jensen–Shannon divergence, Jensen–Rényi divergence and Jensen–Tsallis divergence-based approach to measure the difference between neighboring frames and extract key frames. J.L. Lai, *et al* [6] used a saliency-based visual attention model and selected the frames with maximum saliency value as key frames. M. Kumar, *et al* [7] analyzed the spatio-temporal information of the video by sparse representation and used a normalized clustering method to generate clusters, the middle frame in each temporal order-sorted cluster was selected as a key frame.

Despite that numerous methods have been presented in the literature, key frame extraction remains a challenging and difficult problem due to the complexity and diversity of video data. An excellent key frame extraction algorithm requires not only fast processing speed, but also accurate results, however, the methods discussed previously either use single feature of frame which lead to the inaccuracy or use multi-features of frame which lead to the increase of computation complexity. In this paper, a simple yet efficient algorithm based on unsupervised clustering and compressive sensing is proposed to address the problem mentioned above.

2.2. Compressive Sensing

Dimensionality reduction and signal reconstruction are basic research areas in the field of pattern recognition. Compressive sensing (CS) [8-14] is a novel and effective sampling theory proposed in last decades, which can reconstruct original signal with the frequency that is much smaller than Nyquist frequency. It is well-known that high dimensional signal sampling can be formulated as follow:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \tag{1}$$

Where ***x*** means the *K*-sparse original signal and ***y*** means the sampled one, ***Φ*** is a *m*×*n* measurement matrix. As compressive sensing theory shows that when *n*>>*m*, ***y*** can reconstruct ***x*** with high probability if ***Φ*** satisfies the condition of Restricted Isometry Property (RIP), which is shown below as equation (2):

$$(1-\varepsilon)\left\|\boldsymbol{u}_i-\boldsymbol{u}_j\right\|^2 \le \left\|\boldsymbol{v}_i-\boldsymbol{v}_j\right\|^2 \le (1+\varepsilon)\left\|\boldsymbol{u}_i-\boldsymbol{u}_j\right\|^2 \tag{2}$$

Where $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ are two arbitrary high dimensional signals with the same *K*-basis, $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the corresponding samplings computed by (1), ε is a small positive number between 0 and 1. It is known that some special types of matrices [13], such as random Gaussian matrix, Fourier matrix and symmetric Bernoulli matrix have the characteristics of RIP. In other words, if matrix ***Φ*** satisfies RIP condition, it is easy to reconstruct original signal from its sampling.

It has been mentioned that several types of matrices satisfy the condition of Restricted Isometry Property, however, these matrices are dense and have a number of nonzero elements, which can cause huge computational cost during the processing. In general, a practical matrix with fewer nonzero elements is defined as follow:

$$\boldsymbol{\Phi} = (\varphi_{ij}) = \sqrt{\rho} \times \begin{cases} 1 & P(1) = \dfrac{1}{2\rho} \\ 0 & P(0) = 1 - \dfrac{1}{\rho} \\ -1 & P(-1) = \dfrac{1}{2\rho} \end{cases} \tag{3}$$

Where *P* means the probability and ρ is a parameter that controls the number of nonzero elements. It has been proved that if $\rho = 1$ or $\rho = 3$, ***Φ*** satisfies RIP condition, when $\rho = 1$, ***Φ*** is a uniform nonzero matrix that has the same number of 1 and -1, when $\rho = 3$, the whole computational cost would be twofold down because only one-third of the elements in ***Φ*** are nonzero and need to be computed, furthermore, because $\boldsymbol{\Phi}$ matrix is very sparse, it does not need to be stored in practice, only the positions and values of those nonzero elements should be remembered and saved, which makes the memory cost obviously slight.

Apparently, the number of nonzero elements in $\boldsymbol{\Phi}$ is determined by ρ , besides $\rho = 1$ and $\rho = 3$, another two values of ρ are often used in practice, as shown in equation (4):

$$\rho = \sqrt{n} \quad \text{or} \quad \rho = \frac{n}{\log n} \tag{4}$$

P. Li, *et al* [8] suggest that the former equation in (4) can achieve significant speedup with little loss in accuracy. However, even the dimensionality of a low-resolution frame could reach 10^4, the one of a high-resolution frame would be even much higher, which directly leads to a lot of nonzero elements in $\boldsymbol{\Phi}$. In this paper, the latter equation in (4) is alternatively employed because it can exponentially decrease the number of nonzero elements in $\boldsymbol{\Phi}$, and only a few elements need to be computed, which significantly speeds up the computation.

2.3. Motivation of the Proposed Algorithm

Video data is composed of numerous frames and needs more amount of storage than any other types of multimedia documents. For example, a common 90 minutes' film with 720P resolution (1280×720) contains about 160,000 frames with 920,000 pixels each, needs about 3GB storage space. It is no doubt that direct use of all features in one frame would produce huge amount of computation as well as expensive cost of memory, however, the accuracy would decrease if only few features in one frame are used, because the preserved information about the frame is not adequate.

As mentioned in 2.2, compressive sensing theory shows that even a low dimensional signal could contain adequate information to reconstruct the original high dimensional signal, in other words, the low dimensional signal can not only efficiently speed up the computation, but also nearly be as accurate as the high dimensional one. Motivated strongly by the CS theory, a sparse matrix similar to (3) is adopted in this paper, which acts as a transition between high and low dimensional signals, after that, key frame extraction will use the low dimensional features of frames instead of those high dimensional ones, and the processing speed would be very fast even though the original dimensionality of frames is very high. Meanwhile, as proved by CS theory, these low dimensional features preserve almost the entire information of the corresponding high dimensional ones, the accuracy would be hardly affected at all.

3. Proposed Algorithm

The proposed algorithm is presented in detail in this section. The whole process can be mainly divided into four steps, including high dimensional feature construction, low dimensional feature generation, sub-shot segmentation and key frame extraction, as shown in Figure 2.

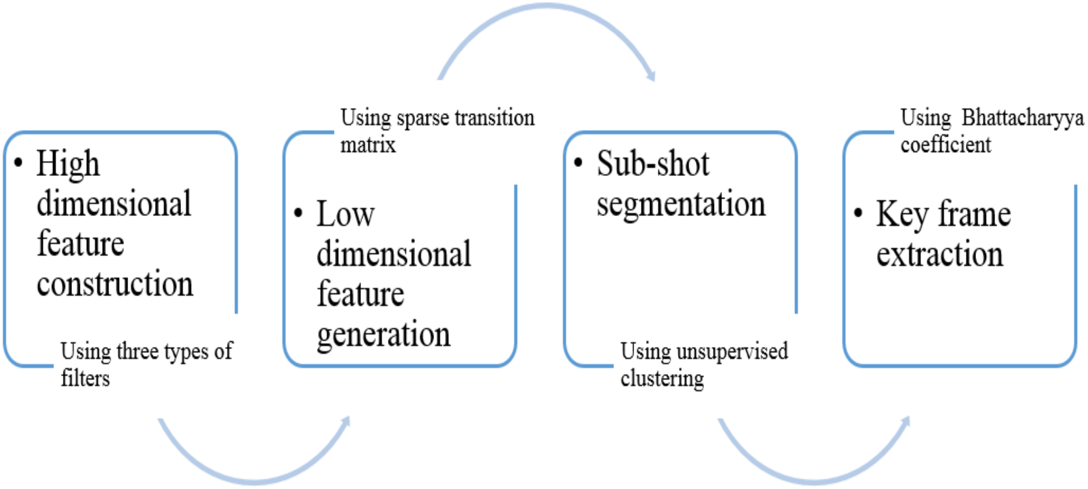

• High dimensional feature construction
Using three types of filters
Using sparse transition matrix
• Low dimensional feature generation
• Sub-shot segmentation
Using unsupervised clustering
Using Bhattacharyya coefficient
• Key frame extraction

Figure 2. Four Steps of the Proposed Algorithm

3.1. High Dimensional Feature Construction

In this step, a very high dimensional feature of each frame would be constructed by three types of filters with various scales. In general, a high dimensional representation of an image is often processed by convolving the image with several filters or templates, as shown in equation (5):

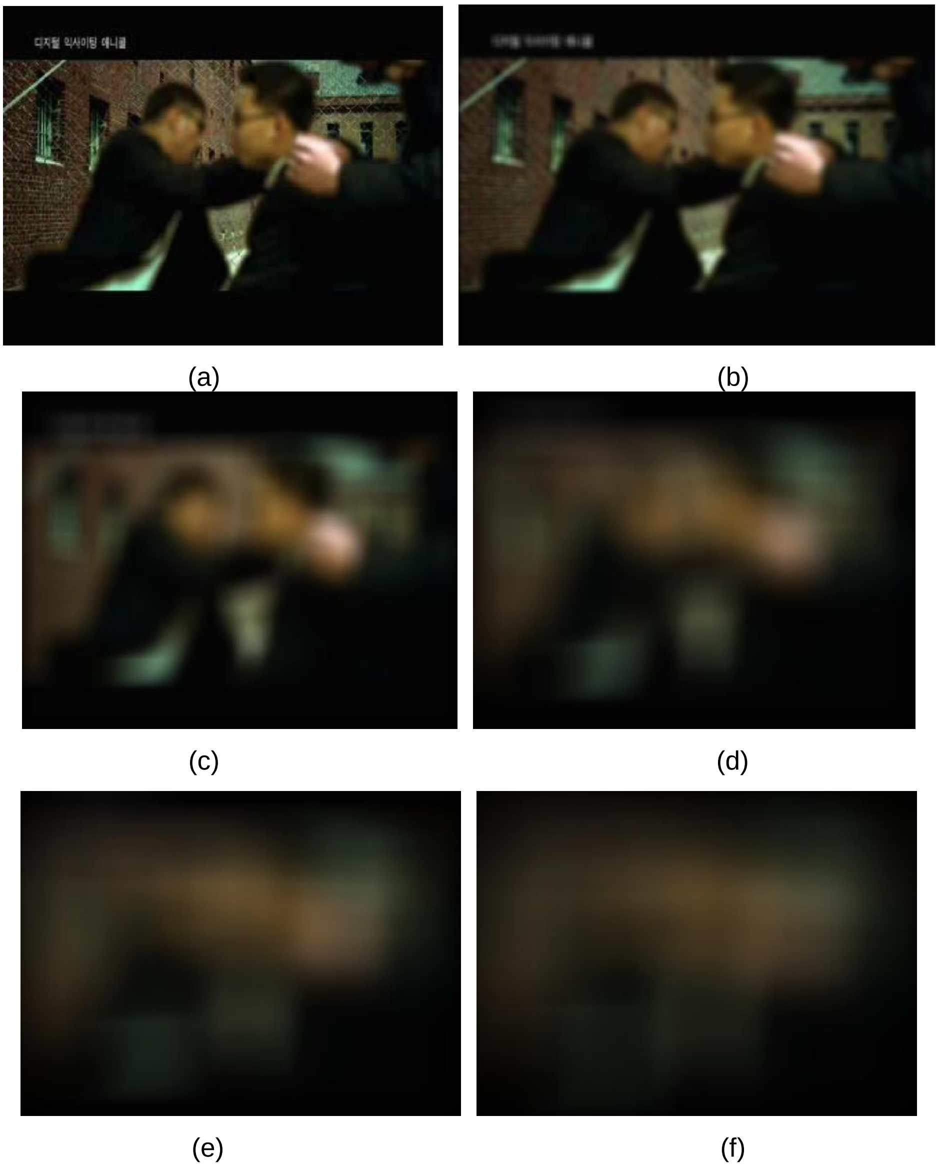
$$\boldsymbol{f}^{h} = \boldsymbol{f} * \boldsymbol{Z} \qquad (5)$$

Where $\boldsymbol{f}$ is an image, $\boldsymbol{f}^{h}$ is the high dimensional representation of $\boldsymbol{f}$, $\boldsymbol{Z}$ is a filter or template, symbol "*" means convolution.

Considering the diversity of video frames, it is not reasonable to use only one filter with various scales to generate high dimensional representation of each frame, sometimes pixels near the center of filter should be given larger weights and pixels far away from the center of filter should be given smaller weights, on the contrary, sometimes all pixels should be given the same weights, the strategy used to assign weights is up to the types of frames, different strategies should be used even in the same video. In this paper, three types of filters with various scales including anisotropic average filter, median filter and Gaussian filter, are employed to address the problem mentioned previously, as shown in equation (6):

$$\boldsymbol{f}^{h} = \bigcup_{c=1}^{3} \bigcup_{i=1}^{0.4w} \bigcup_{j=1}^{0.4h} [\boldsymbol{F}_{c}(i, j) * \boldsymbol{f}] \qquad (6)$$

Where $\boldsymbol{f}$ is a frame, $\boldsymbol{f}^{h}$ is the high dimensional representation of $\boldsymbol{f}$, $\boldsymbol{F}$ is one of the specified filters, symbol "*" means convolution and symbol "$\bigcup$ " means concatenation, w and h are width and height of the frame, respectively. In our experiments, when scale of filter exceeds a certain range, the image produced by convolution between frame and filter losses most of the information so that it is useless in subsequent process, as shown in Figure 3. According to this consideration, the maximum scale of each filter is less aggressively limited to 40 percent of original frame width and height in order to contain more useful and positive information in high dimensional representation of frame. With (6), the high dimensional representation of each frame would be produced and used as the high dimensional feature of each frame.



(a) (b)

(c) (d)

(e) (f)

Figure 3. Results of Filtering with different Scales, (a) is the Original Frame, (b-f) are Filtered Image with Scale 5×5, 21×21, 41×41, 61×61, 81×81, Respectively

3.2. Low Dimensional Feature Generation

Once high dimensional feature of each frame is constructed, the next step is to project it to low dimensional space and generate corresponding low dimensional feature. As mentioned above, compressive sensing is employed in this step, firstly a very sparse random projection matrix $\boldsymbol{\Phi}$ that satisfies condition of RIP is constructed, and here equation (3) and the latter equation in (4) are employed because fewer nonzero elements would be produced, which results in faster computation speed and lower memory requirement. In a word, the process of this step can be formulated as follow:

$$\boldsymbol{f}^{l} = \boldsymbol{\Phi}\boldsymbol{f}^{h} \qquad (7)$$

Where $\boldsymbol{f}^{l}$ is the low dimensional feature of $\boldsymbol{f}^{h}$.

Suppose that the magnitude of high dimensional feature n is 10^p, it is easy to know from (4) that the probability of nonzero elements in $\boldsymbol{\Phi}$ is about p/n, which means the number of nonzero elements in $\boldsymbol{\Phi}$ is mp, m is the row count of $\boldsymbol{\Phi}$ and usually small, that is, the average amount of nonzero elements in each row of $\boldsymbol{\Phi}$ is p, which is usually less than 10. Obviously, only the entries of these nonzero elements need to be stored, and the dimensionality of generated low dimensional feature is equal to m. It is not difficult to conclude that the computation speed of low dimensional feature generation is fast owing to the very sparse projection matrix $\boldsymbol{\Phi}$.

3.3. Sub-Shot Segmentation

A shot is a set of frames captured by one camera in a continuous time period. The early literatures often assumed that the visual contents change slightly in one shot. However, it is not true in most video sequences which produces the necessity of sub-shot segmentation in one shot. In this section, a simple yet efficient unsupervised clustering-based approach is proposed to accomplish sub-shot segmentation.

Suppose that $\boldsymbol{S}$ is a correctly detected shot, firstly the low dimensional feature of each frame in $\boldsymbol{S}$ can be computed using (6) and (7), the corresponding feature column vector group of $\boldsymbol{S}$ would be obtained, which is described as follow:

$$\boldsymbol{S}_l = \left\{ \boldsymbol{f}_l^{\,1}, \boldsymbol{f}_l^{\,2}, \cdots, \boldsymbol{f}_l^{\,N} \right\} \tag{8}$$

Where $\boldsymbol{f}_l^{\,i}$ means the low dimensional feature of the i-th frame. Secondly, $\boldsymbol{f}_l^{\,j}$, which is the first frame that does not belong to any class is initialized as $\boldsymbol{C}_k$, which represents the center of the new class k, then adjusted cosine similarity (acs) is employed as the similarity measurement between $\boldsymbol{C}_k$ and $\boldsymbol{f}_l^{\,j+1}$:

$$S(\boldsymbol{C}_k, \boldsymbol{f}_l^{\,j+1}) = \frac{\left[\boldsymbol{C}_k - \overline{\boldsymbol{C}}_k, \boldsymbol{f}_l^{\,j+1} - \overline{\boldsymbol{f}}_l^{\,j+1} \right]}{\left\| \boldsymbol{C}_k - \overline{\boldsymbol{C}}_k \right\| \cdot \left\| \boldsymbol{f}_l^{\,j+1} - \overline{\boldsymbol{f}}_l^{\,j+1} \right\|} \tag{9}$$

Where the symbol "$\overline{\ }$" is used to generate a vector in which all elements are average value of the specified vector, if $S(\boldsymbol{C}_k, \boldsymbol{f}_l^{j+1})$ is larger than a threshold α (α=0.7 in this paper), $\boldsymbol{f}_l^{\,j+1}$ is regarded as a part of k-class, then $\boldsymbol{C}_k$ and subscript j are modified as follow:

$$\boldsymbol{C}_k = \mu \boldsymbol{C}_k + (1-\mu)\, \boldsymbol{f}_l^{\,j+1} \qquad j = j+1 \tag{10}$$

Where $\mu \in (0,1)$ is a weight value which is used to balance the importance of $\boldsymbol{C}_k$ and $\boldsymbol{f}_l^{\,j+1}$, we consider that the visual contents always vary with time, so that $\mu = 0.49$ is adopted, which means the new center of each class is slightly more dependent on $\boldsymbol{f}_l^{\,j+1}$. If $S(\boldsymbol{C}_k, \boldsymbol{f}_l^{j+1})$ is less than α, the current clustering is ended and a new clustering would start in the same way.

The traditional cosine similarity is more sensitive to the angle between two vectors than to the value difference, so adjusted cosine similarity is adopted to avoid this problem. The total flowchart of sub-shot segmentation is shown in Figure 4.

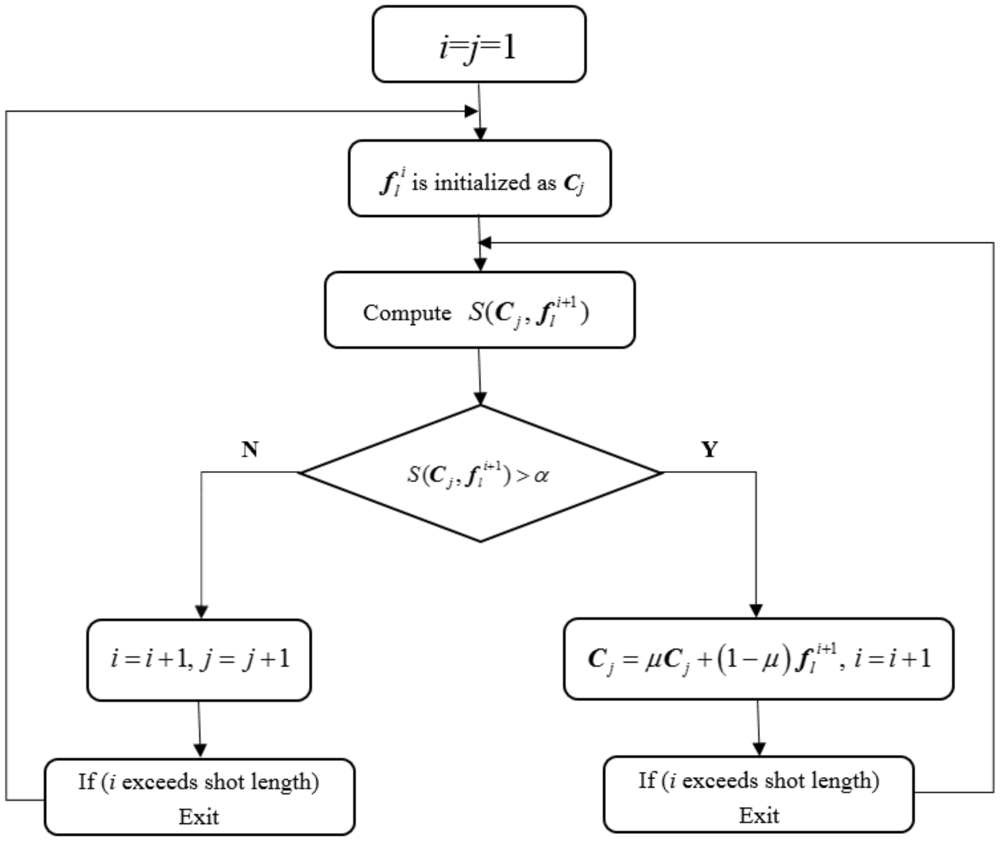

$i=j=1$
f_l^i is initialized as C_j
Compute $S(C_j, f_l^{i+1})$
$S(C_j, f_l^{i+1}) > \alpha$
N
Y
$i=i+1, j=j+1$
$C_j = \mu C_j + (1-\mu) f_l^{i+1}, i=i+1$
If (i exceeds shot length)
Exit
If (i exceeds shot length)
Exit

Figure 4. Flowchart of Unsupervised Clustering-Based Sub-Shot Segmentation

3.4. Key Frame Extraction

Shot $\boldsymbol{S}$ is segmented into a set of sub-shots after sub-shot segmentation presented in 3.3. The visual contents keep stable in each sub-shot so that only one key frame needs to be extracted to represent the sub-shot. Assume that there are totally k sub-shots in $\boldsymbol{S}$, as illustrated in Figure 5. The final step of our algorithm is how to extract the most proper frame from every sub-shot as the key frame and construct a set of key frames to describe shot $\boldsymbol{S}$.

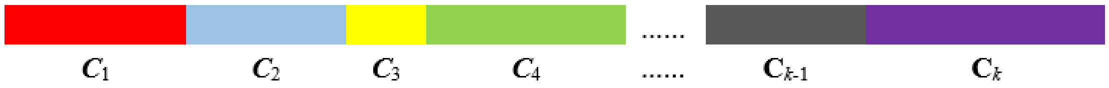


Figure 5. Totally *k* Sub-Shots in Shot *S*, Each Color Represents a Sub-Shot, C_i is the Class Center of *i*-th Sub-Shot Represented by its Low Dimensional Feature

The class center is the core of all elements in the same class. A naive and natural strategy to extract key frame from each sub-shot is to extract the frame nearest to the class center, which is formulated as follow:

$$P = \arg\max_{j} \left\{ B(\boldsymbol{C}_i, \boldsymbol{f}_l^{\,j}) \right\} \tag{11}$$

Where P is the position of the key frame in i-th sub-shot, $\boldsymbol{C}_i$ is the class center of i-th sub-shot, and $\boldsymbol{f}_i^{\,j}$ is each frame in i-th sub-shot. $B(\bullet)$ is the Bhattacharyya coefficient which measures the similarity between two distributions as follow:

$$B(\boldsymbol{x},\boldsymbol{y}) = \sum_{u=1}^{N} \sqrt{p_u(\boldsymbol{x})q_u(\boldsymbol{y})} \tag{12}$$

Where $\boldsymbol{x}$, $\boldsymbol{y}$ are two samples and $\boldsymbol{p}_u$, $\boldsymbol{q}_u$ are their respective distributions.

4. Experiments

Five different video clips are used in this work to test and evaluate the performance of the proposed method, the first one is a basketball match containing 140 frames and 5 shots, the second one is a news report containing 399 frames and 5 shots, the third one is an advertisement containing 917 frames and 24 shots, the fourth and fifth ones are MTVs, the former contains 2769 frames and 135 shots and the latter contains 5406 frames and 171 shots. All the key frames are marked manually as the ground truth.

Qualitative and quantitative tests are both conducted in the experiments. For most of the key frame extraction methods, the recall rate would reach 100% if the threshold is set properly. The usual strategy is that redundant key frames are permitted, but missed key frames are not accepted, so the criterion used in quantitative test and comparison is the maximum precision rate under 100% recall rate, as shown in (13):

$$Precision = \frac{k}{k+f} \tag{13}$$

Where k is the number of key frames extracted correctly, and f is the number of key frames extracted falsely. Methods in [5] and [6] are compared with our proposed algorithm in order to demonstrate the efficiency and accuracy of our work. Table 1 and Figure 6 show the quantitative test results in 5 video clips mentioned above, Figure 7 to Figure 11 partially illustrate the qualitative results in chronological order.

Table 1. Quantitative Results of the Proposed Algorithm

Video sequence	Number of frames	Number of real key frames	Number of detected key frames	Precision rate
basketball match	140	7	7	1
news report	399	12	12	1
advertisement	917	96	105	0.91
MTV-1	2769	413	450	0.92
MTV-2	5406	668	747	0.89

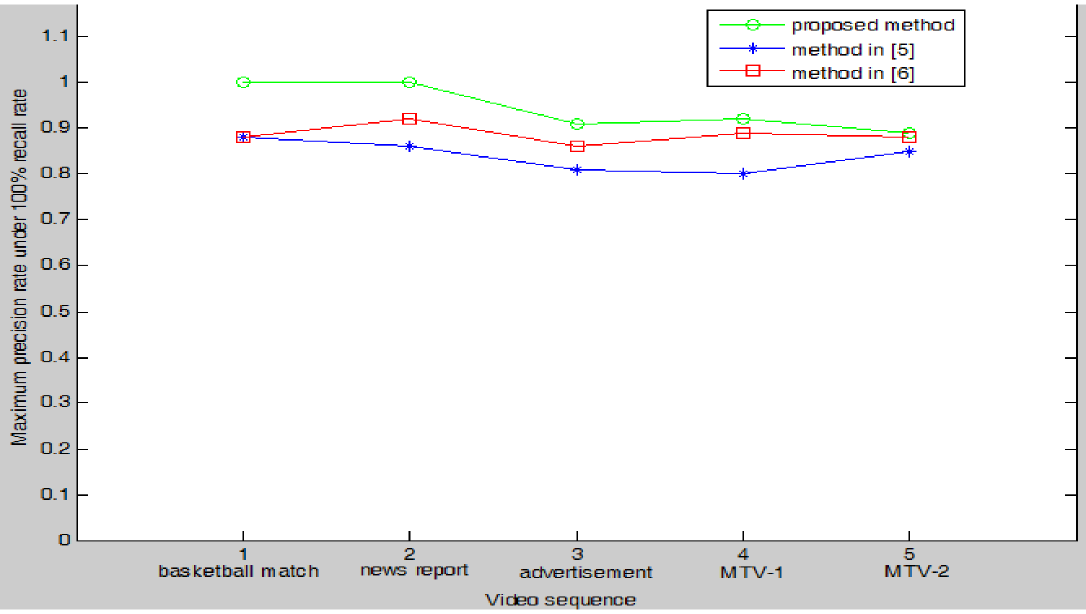


Figure 6. Comparison Results of the Proposed Algorithm

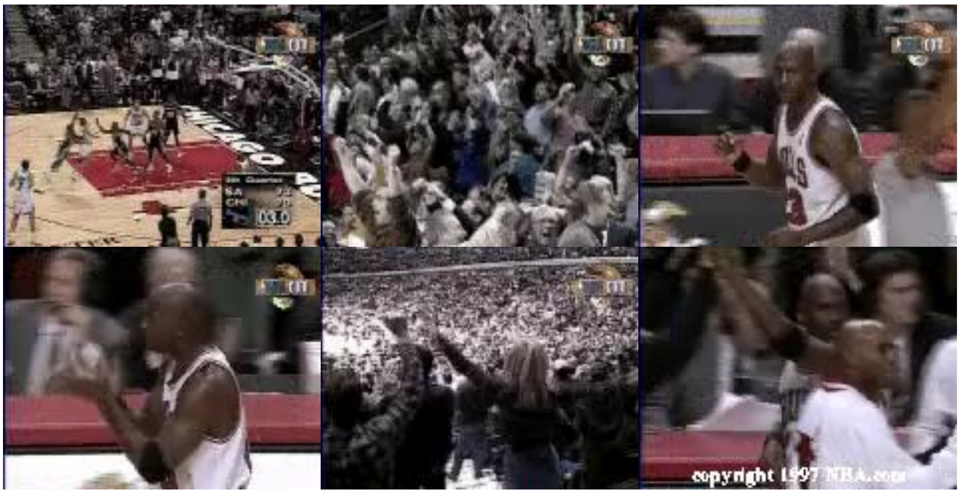


Figure 7. Some Key Frames Extracted from Basketball Match Video

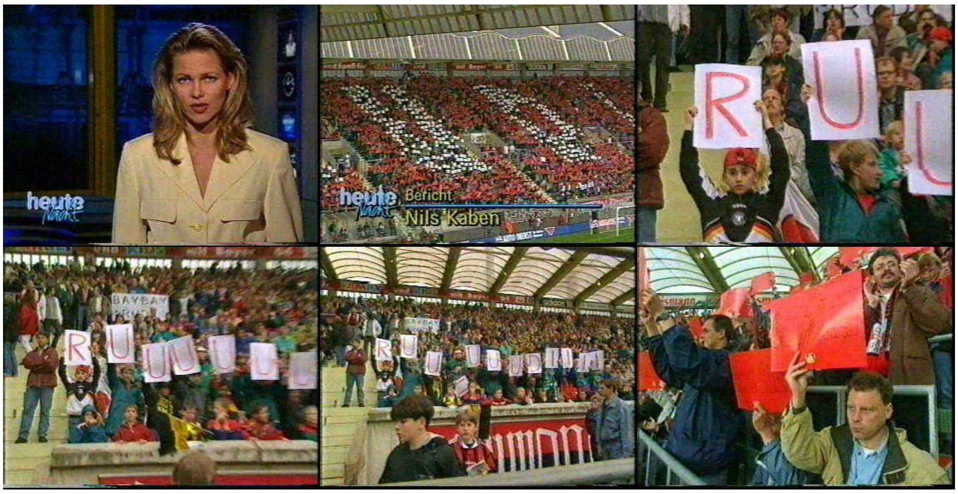


Figure 8. Some Key Frames Extracted from News Report Video

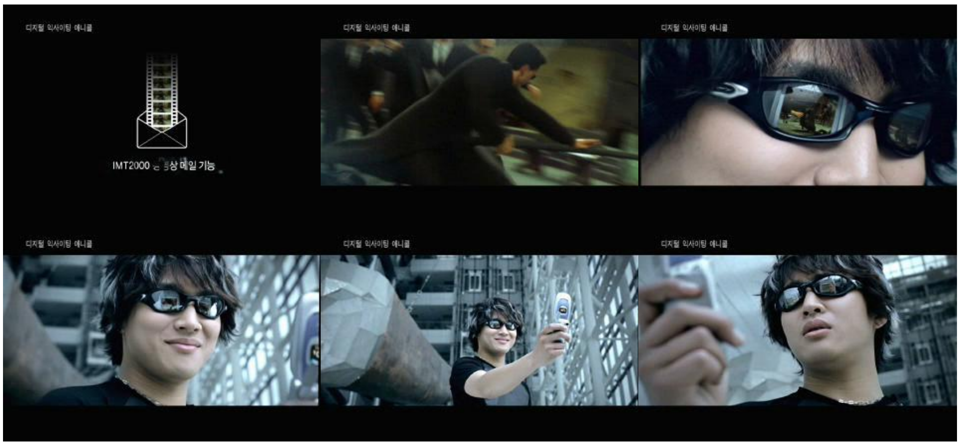


Figure 9. Some Key Frames Extracted from Advertisement Video

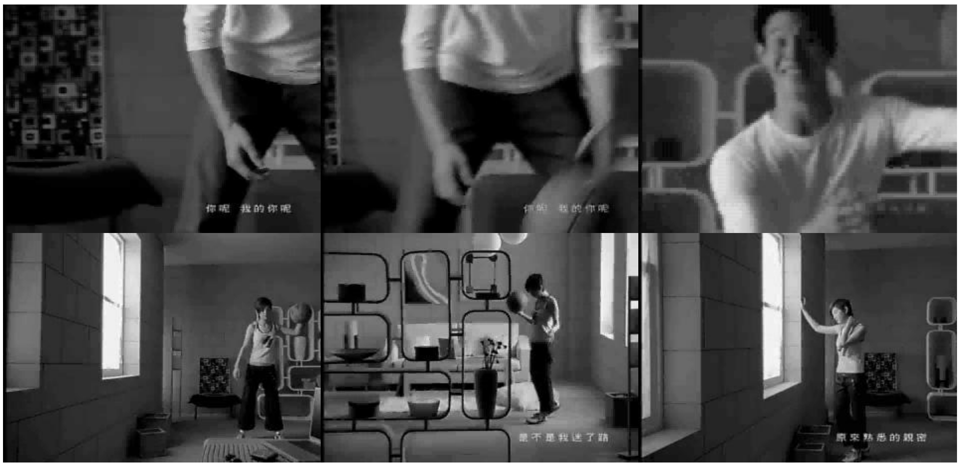


Figure 10. Some Key Frames Extracted from MTV-1 Video

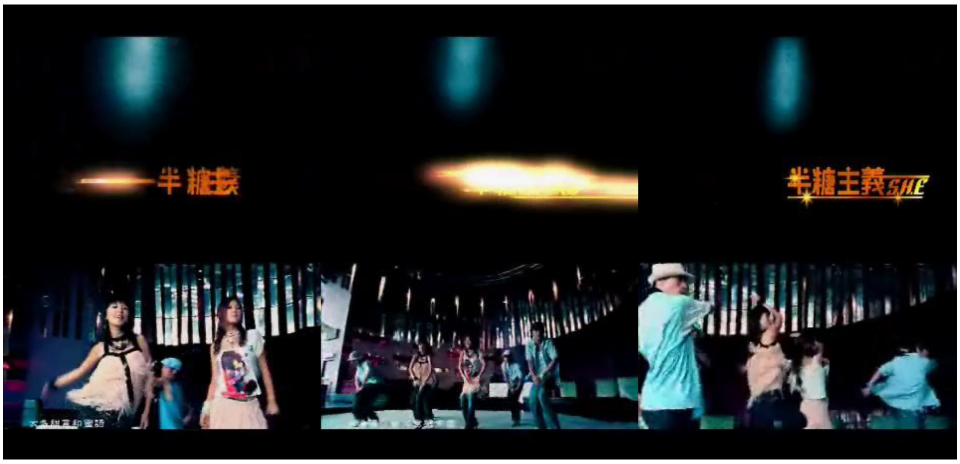


Figure 11. Some Key Frames Extracted from MTV-2 Video

The experimental results show that the proposed method could extract key frames from video more efficiently and properly. Because missed key frames would reduce the experience and lead to incomplete understanding and description, the strategy of key frame extraction should be emphasized again, that is, redundant key frames are permitted, but missed key frames are not accepted. The redundant key frames are mainly produced by falsely sub-shot segmentation, especially in those videos with complex motions and special effects.

5. Conclusion

In this paper, an efficient key frame extraction algorithm that exploits compressive sensing and unsupervised clustering is proposed. Because of the excellent property of the sparse transition matrix, the process that produces the low dimensional feature is fast and could conduct offline with few demands for memory and computation. The sub-shot segmentation is accomplished in order to extract key frames more accurately. Compared with other two mainstream algorithms, our proposed method works more efficiently and properly in the five test videos. The false sub-shot segmentation and redundant key frames are unavoidable because only low-level features of frames are used, which are not enough to precisely represent frames. Future work would mainly be focused on the application of semantic feature in key frame extraction.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China under Grant Nos. 61103128, 61471182, 61170120, 61305058, Natural Science Foundation of Jiangsu Province under Grant Nos. BK20130473, BK20130471, BK20140419.

References

[1] X. Liu, M. L Song, L. M. Zhang and S. L. Wang, “Joint Shot Boundary Detection and Key Frame Extraction”, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), **(2012)**, pp. 2565-2568.

[2] N. Ejaz, T. B. Tariq and S. W. Baik, “Adaptive key frame extraction for video summarization using an aggregation mechanism”, Journal of Vision Communication and Image Representation, vol. 23, **(2012)**, pp. 1031-1040.

[3] Q. Xu, Y. Liu, X. Li, Z. Yang, J. Wang, M. Sbert and R. Scopigno, “Browsing and exploration of video sequences: A new scheme for key frame extraction and 3D visualization using entropy based Jensen divergence”, Information Sciences, vol. 278, **(2014)**, pp. 736-756.

[4] L. Pan, X. J. Wu and X. Shu, “Key Frame Extraction Based on Sub-shot Segmentation and Entropy Computing”, Proceedings of Chinese Conference on Pattern Recognition (CCPR), **(2009)**, pp. 1-5.

[5] Q. Xu, Y. Liu, X. Li, Z. Yang, J. Wang, M. Sbert and J. F. Li, “Key frame selection based on Jensen-Rényi divergence”, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), **(2012)**, pp. 1892-1895.

[6] J. L. Lai and Y. Yi, “Key frame extraction based on visual attention model”, Journal of Vision Communication and Image Representation, vol. 23, **(2012)**, pp. 114-125.

[7] M. Kumar and A. C. Loui, “Key Frame Extraction from Consumer Videos Using Sparse Representation”, Proceedings of the 18th IEEE International Conference on Image Processing (ICIP2011), **(2011)**, pp. 2437-2440.

[8] P. Li, T. J. Hastie and K. W. Church, “Very Sparse Random Projections”, Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD2006), **(2006)**, pp. 287-296.

[9] E. J. Candes and T. Tao, “Decoding by linear programming”, IEEE Transactions on Information Theory, vol. 51, no. 12, **(2005)**, pp. 4203–4215.

[10] E. J. Candes and T. Tao, “Near-Optimal Signal Recovery from Random Projections: Universal Encoding Strategies”, IEEE Transactions on Information Theory, vol. 52, no. 12, **(2006)**, pp. 5406-5425.

[11] D. L. Donoho, “Compressed sensing”, IEEE Transactions on Information Theory, vol. 52, no. 4, **(2006)**, pp. 1289–1306.

[12] R. Baraniuk, M. Davenport, R. DeVore and M. Wakin, “A Simple Proof of the Restricted Isometry Property for Random Matrices”, Constructive Approximation, vol. 28, no. 3, **(2008)**, pp. 253-263.

[13] K. H. Zhang, L. Zhang and M. H. Yang, “Fast Compressive Tracking”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 10, **(2014)**, pp. 2002-2015.

[14] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen and S. Y. Lee, “Compressive sensing: From theory to applications, a survey”, Journal of Communications and Networks, vol. 15, no. 5, **(2013)**, pp. 443-456.