International Research Journal of Mathematics, Engineering and IT
Volume 4, Issue 1, January 2017 **Impact Factor- 5.489**
ISSN: (2348-9766)
Website- www.aarf.asia, **Email:** editor@aarf.asia , editoraarf@gmail.com

NEAR-DUPLICATE SEGMENTS DETECTION IN THE VIDEOS USING A DENSITY-BASED CLUSTERING METHOD

Abbas Ghavam

Assistant Professor, Department of Environmental Science, Institute of Sciences and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran

Nasim Afhami

Master of Computer Engineering, Institute of Sciences and High Technology and Environmental Sciences, Graduate University of Advanced Technology, Kerman, Iran

Abstract

With the increasing volume of online videos, near duplicate videos have also risen sharply. Now, the problem of detection and removal of the near duplicate videos has become an essential requirement for the video storage and indexing. Also, due to the multiple video editing, some segments of a video are sometimes repeated in other videos. The redundancy of videos with near duplicate content leads to the loss of storage space and can degrade the performance of the web video search engine. Thus, the detection of near duplicate videos is one of the essential requirements for the manufacturers of video and video search engine. In this paper, we try to identify the segments that are available in the different videos and have near duplicate content. For this purpose, we have used a two-step approach. Firstly, by extracting the fingerprints and using a hash function, we select some of the candidate videos. In the second stage, among the selected videos, the near duplicate segments are extracted by the feature extraction and investigation of these features. For this purpose, the SURF feature and density-based clustering method have been used. This method has been used on a number of video and the experimental results show that this method can identify the near duplicate segments.

Key Words: Image processing, detection of near duplicate videos, hash, density-based clustering.

1. Introduction

Since the massive amounts of video content are now available in the media applications and the digital video contents can be easily edited, there are a very large volume of videos copied and near duplicate videos. With the increasing volume of online multimedia data, we need the ways to search the various search engines. In addition, some methods are needed to identify the near duplicate videos for great video archives. For this purpose, the near duplicate video retrieval (NDVR) is used. NDVR helps the video search and is used in many new applications, such as detection of illegal copies of video, video tagging, monitoring the use of online video, etc.

In this paper, we consider two videos with nearly identical visual content as near duplicate videos. Of course, two videos may be different in terms of format, compression quality, as well as some spatio temporal edits such as adding a logo, light changes, frame removal, etc.

In this paper, it is tried to detect the near duplicate video segments. Much research has been done to identify the near duplicate videos, some of which are described below:

In paper (1), a three-step algorithm is presented: conceptual sign production, video matching (using semantic concepts) and at a later stage, the near duplicate video detection. For this purpose, first some objects or concepts are considered, then they are checked in shape. An array is created with length of concepts number and decisions are made with the help of this array.

In the paper [2], a hierarchical method is used to detect the near duplicate videos and the bipartite graph is used to match. The maximum matching method (MM) is used on the graph.

In the paper [3], a two-step method is used. Firstly, a quick comparison is done in the space of compact feature and some candidate segments are selected. In the second stage, a pixel selection method based on entropy is used and the feature vectors are constructed using it. In the paper (4), the hash approach is used to match the near duplicate videos using local and global features. In the paper [5], the parallel processing is used to solve the problem due to the large volume of data, and GPU and MapReduce framework has been used. In [6], matching pair of elements in the video is used and data are grouped using clustering based on density. In [7], clustering is used to identify the near duplicate video groups.

The contents of this paper are set in a way that first we will present an overview of the proposed algorithm in the next section and then in the section 3, the candidate videos selection algorithm will be expressed. In the section 4, extracting the near duplicate segments will be discussed. Finally, we examine the proposed method using some videos and the results of its implementation are examined.

2. General description of the proposed method

We need to have the fast methods and features with low complexity to compare the large databases. In this section, a method is expressed for selecting a candidate set (videos that may contain the near duplicate segments) in order to increase the speed. For this purpose, we first define a series of temporal informative representative images (TIRI). Out of every 4 frames, a TIRI is extracted. Then, DCT is extracted from them. The extracted features are passed to the hash function and the near duplicate videos are found through the hash function. Next, the near duplicate video segments must be found for the set of input candidate videos. To this end, the feature extraction is done. The SURF feature is applied and the similarity is checked by comparing them. However, the density-based clustering is used to decide on the best series of frames in terms of similarity and frame continuity.

The algorithm used is as follows:

1. Extracting the fingerprints from the video frames.

2. Applying a hash function to reduce the search time.

3. Selecting a set of videos that can be included the near duplicate segments.

4. SURF feature extraction

5. Applying the density-based clustering

3. Candidate video selection algorithm

To select the candidate videos, spatio–temporal feature is derived from them. Then, they (TIRI) are divided, and DCT coefficients are derived from them. Using these coefficients, the feature vectors are made and then with the help of the feature vector and hash function, the near duplicate videos are diagnosed and at the later stage, the near duplicate video segments are selected among them. In the following, the TIRI extraction method is explained in the section (1.3), the feature extraction from them by DCT is described in the section (2.3) and in (3.3), applying the features and hash function is discussed.

1.3 Spatio–temporal feature (image)

The spatio–temporal feature or TIRI is used to evaluate the temporal characteristics as well as spatial changes. The TIRI is calculated by the short video segments and it can be considered the frame features of that segment using it.

This feature that represents the temporal information has been introduced in [8] and in the paper [9], it is used to create the fingerprint of video segments and the segment features are shown. To this end, the video is divided into the segments and each segment is shown with a photo. How to create these photos will be described later. The photos represent the changes of video snippets over time as well as spatial variations occurred and show both temporal and spatial concepts. The way that [2009] has introduced to display the temporary information (TIRIs) of a series of video has also a good performance for displaying the spatial information. In the paper [2010], it is used the average weighted frames of a segment to create an image that contains the spatio–temporal information of a video segment (TIRI). The result is normally a fading image, including the possible movements of that video segment. The TIRI is calculated according to the following formula:

(1)

$$L`_{m,n} = \sum_{k=1}^{J} L_{m,n,k} \, w_k$$

(2)

$$w_k = 0.65 \, ^\wedge \, k$$

This formula is calculated for each segment that includes J-frame. K represents the number of frames. Total weighted features from one to the total number of frames of that segment are calculated. In this formula, the variables m and n specify the number of pixels and w_k represents the weight of each frame. As shown in the formula (2), it is used the exponential weighting for weighting.

In this work to calculate TIRI, the video is divided into the segments of 4 frames and features of four frames are displayed in a photo with the help of TIRI. And it is used TIRI instead of using the frames to identify and extract a set of near duplicate videos. This will increase the search speed in the video database. In order to extract the TIRI, first a pre-processing is done on the video and the dimensions of the frames are considered to be equal. All frames are converted to the size of W × H. Then, TIRI is extracted from the segments of 4 frames which are available in one shot and we use it to make decisions. To create TIRI, we must gather the successive frames in one segment with the help of a weighting method. Three methods for weighting are displayed in Figures 2, 3 and 4. As stated in equation (2), we use the exponential weighting.

Figure (1). The four photos used to calculate their TIRI

Figure (2). TIRI Photo with exponential weighting

Figure (3). TIRI Photo with the fixed weight 1

Figure (4). Linear weighting

2.3 Discrete cosine transform

Discrete cosine transform (DCT) displays a limited series of the data points as a sum of cosine functions oscillating at different frequencies. Using the DCT, the image can be divided into the sections with different importance according to the contents of the video.

3.3 Candidate videos selection

At this stage, the obtained fingerprint is used for searching and matching. We have used the Hamming distance to calculate the similarity between them. As shown in the following algorithm, after the feature extraction (fingerprint) from the frames, the near duplicate videos are selected using the hash function. When there is a large database of images, it is very difficult and time-consuming to search in it. The hash function is used to reduce search time. The hash function gives the possibility to consider the similarity of photos via Hash Code dedicated to them and quickly returns the photos that are adjacent to the desired photo in the hash table.

Out of every 4 frames, a TIRI is extracted, which represents all the frames. The DCT feature is derived from TIRI and the first horizontal and vertical coefficients are used. These two coefficients are joining together and then the middle of the resulting vector is calculated. The vector elements are replaced with one and zero according to the median. This vector is used as a feature vector of the TIRI and using the hash function, the TIRI resemblance to the existing elements in the database is specified.

Hash function can record the input data with a desired size instead of the fixed-size data. In this work, we have used the SHA1 hash function. This algorithm is very useful.

Algorithm (1): Algorithm used to find the candidate videos [2010]

1. TIRI production for each 4 frames by weight (γ = 0/65) $W_K = \gamma^K$

2. Each TIRI is divided into a series of overlapping blocks of size 2W × 2W. $B^{i,j}=\{l'_{x,y}|x\epsilon\ iw\pm w, y\epsilon jw\pm w\}$

3. Two DCT coefficients is derived for each block. The first horizontal and vertical coefficients near the DC coefficients are used.

The first vertical coefficient is as follows:

$\alpha_{i,j}=V^T B^{i,j} 1$

And also the first horizontal coefficient:

$B^{i,}=1^T B^{i,j}$ V

4. All coefficients are joined to form the vector f.

5. m is placed against the middle of f.

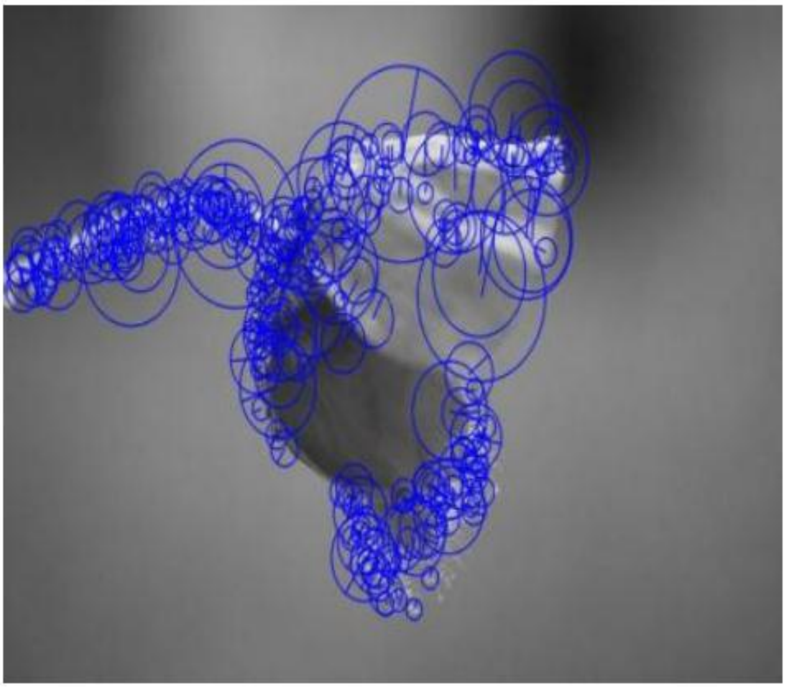
Then, the binary hash is made. Below median values are considered zero and also more than median, 1.

4. Extracting the near duplicate video segments

To extract the near duplicate sub-segments, first it must be found the videos that can include the near duplicate segments and then the segments are identified using the SURF feature extraction method. The frames that are similar to the clustering based on density are presented so that it can be extracted an appropriate group from them. In the following, the SURF feature extraction is described. Moreover, the density-based clustering method is stated in the section (3.4).

1.4. SURF feature

The SURF (Speeded-up Robust Features) algorithm is used to extract the features and descriptors. The feature is resistant against the rotation and changing the scale, and also has good speed and accuracy. The algorithm uses the Hessian blob detector for the detection of desired points. After identifying the key points, their direction is determined in a circle around the points. The feature extraction is done using the Haar wavelet. [15] This is shown in the following figure:



SURF feature extraction from the photo

2.4. Density-based clustering

The feature is extracted to find the near duplicate segments among the candidate videos, and the frames that are most similar are examined to find the sub-videos. Clustering is done based on the density and the related sub-videos are extracted.

The density-based clustering algorithm [16] is described in the following:

Algorithm (2). The DBSCAN algorithm

```
For each o Є D do
  If o is not yet classified then
    If o is a core-object then
      Collect all object density-reachable from o
         And assign them to a new cluster
   Else
         Assign o to NOISE
```

In this algorithm, each element (o) is selected from a set of input (D) and then based on location, are clustered. The time complexity of this algorithm is of the order of o (the cost of finding the neighbors *n).

5. Experimental results

In this section, the proposed method implementation on a series of videos is explained. For the program input dataset, 20 videos of near-duplicate web video dataset (CC-WEB-VIDEIO) are used. The video dataset[1] is designed to detect the near duplicate videos and also the video shots are identified in them. In addition, it has the information such as the key video frames, etc. The results of this algorithm on the input dataset are as follows:

Table (1). Output of the segments detection algorithm

Average Precision	83.32
Average Recall	97.03

In the above table, the algorithm shown by recall and precision is able to detect the near duplicate segments and has a good performance.

6. Conclusion

In this paper, a method for extracting the near duplicate segments was expressed. First they are extracted from the video frames of TIRI and the DCT feature can be obtained for them, then a series of the near duplicate videos are selected by applying the hash function and in the next step, the similarity between frames is investigated by extracting the SURF feature. And finally, the near duplicate segments are detected using the density-based clustering. This method is able to identify and extract the segments accurately.

For future works, it can be examined the methods for extracting the near duplicate video segments using the semantic clustering of various scenes.

[1] http://vireo.cs.cityu.edu.hk/webvideo/Download.htm

References

[1]. M. Hyun-seok, Ch. JaeYoung, D.N. Wesley, and M.R.Yong, Near-Duplicate Video Detection Using Temporal Patterns of Semantic Concepts, 11th IEEE International Symposium on Multimedia, pp. 65-71, 2009.

[2]. X. Hui, L. Lu, S. Li-Feng, Y. Shi-Qiang, Fast Robust Detection of Near-Duplicates In Web Video Data Base, IEEE International Conference on Multimedia and Expo, pp. 293-296, 2008.

[3]. L. Lchiro, K. Noda, T. Takahashi, H. Murase, Genre-Adaptive Near-Duplicate Video Segment Detection, IEEE International Conference on Multimedia and Expo, pp. 484-487, 2007.

[4]. J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, Multiple feature hashing for real-time large scale near-duplicate video retrieval, in Proceedings of the International Conference on Multimedia – MM,11. New York, NY, USA: ACM Press, pp. 423–432, 2011.

[5]. Y. Jiang, G. Li, J. Feng, and W. Li, String Similarity Joins: An Experimental Evaluation, in Proceedings of the VLDB Endowment, pp. 625–636, 2014.

[6]. Y. Kim, K. Shim, M.-S. Kim, and J. Sup Lee, DBCURE-MR: An efficient density-based clustering algorithm for large data using MapReduce, Information Systems, vol. 42, pp. 15–35, 2014.

[7]. H.B. Dasilva, Z.K.G. do Patrocinio,G. Gravier , L. Amsaleg, Near-Duplicate Video Detection Based on An Approximate Similarity Self-Join strategy, International WorkShop on Content- Based Multimedia Indexing (CBMI), pp. 1-6, 2016.

[8]. M. Malekesmaeili, M. Fatourechi, R. K. Ward, Video copy detection using temporally informative representative images, in Proc. Int. Conf. Machine Learning and Applications, pp. 69-74, Dec. 2009.

[9]. M. Malekesmaeili, R.K. Ward, Robust video hashing based on temporally informative representative images, in Proc. IEEE Int. Conf. Consumer Electronics, pp. 179-180, Jan. 2010.

[10]. M.G.Chung, J.Lee, H.kim, S.M.-H.Song and W.M.Kim, Automatic video segmentation based on spatio-temporal features. Korea Telecom Journal, 4(1), pp. 4-14, 1999

[11]. S.J.F.Guimaraes, M.Couprie, N.J.Leite, A.A.Araujo, A method for cut detection based on sypnposium on computer graphics and image processing –SIBGRAPI, IEEE Computer society press, ISBN 0769513301,pp. 297-304, 2001

[12]. A. Jinda-Apiraksa, V. Vonikakis, S. Winkler, California-ND: An annotated dataset for near-duplicate detection in personal photo collections. In Proc. 5th International Workshop on Quality of Multimedia Experience, July 3-5, Klagenfurt, Austria, 2013.

[13]. E. Apostollidis, V. Mezaris, FAST SHOT SEGMENTATION COMBINING GLOBAL AND LOCAL VISUAL DESCRIPTORS, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), pp. 6583- 6586, 2014

[14]. H. Min, J. Y.Choi, W.D.Neve, Y. M. Ro, Near-Duplicate Video Detection Using Temporal Patterns of Semantic Concepts, 11th IEEE International Symposium on Multimedia, pp. 65-71, 2009

[15]. H. Bay, A. Ess, T. Tuytelears, L. V. Gool, Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding, vol. 110, pp. 346-359, 2008

[16]. M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on

19th ACM international conference on Multimedia. pp. 1173-1176, 2011 Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231, 1996

[17]. T. Chen, Sh. Jiang, L. Chu, Q. Huang ,Detection and location of near-duplicate video sub-clips by finding dense subgraphs. Proceeding of the