TRƯỜNG ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH
TẠP CHÍ KHOA HỌC
KHOA HỌC TỰ NHIÊN VÀ CÔNG NGHỆ
ISSN: 1859-3100 Tập 14, Số 9 (2017): 24-33

HO CHI MINH CITY UNIVERSITY OF EDUCATION
JOURNAL OF SCIENCE
NATURAL SCIENCES AND TECHNOLOGY
Vol. 14, No. 9 (2017): 24-33

Email: tapchikhoahoc@hcmue.edu.vn; Website: http://tckh.hcmue.edu.vn

CONTENT BASED VIDEO RETRIEVAL SYSTEM USING PRINCIPAL OBJECT ANALYSIS

Bui Van Thinh[1], Tran Anh Tuan[1], Ngo Quoc Viet[2*], Pham The Bao[1]

[1]University of Science Ho Chi Minh City

[2] Ho Chi Minh City University of Education

Received: 25/7/2017; Revised: 04/9/2017; Accepted: 23/9/2017

ABSTRACT

Video retrieval is a searching problem on videos or clips based on the content of video clips which relates to the input image or video. Some recent approaches have been in challenging problem due to the diversity of video types, frame transitions and camera positions. Besides, that an appropriate measures is selected for the problem is a question. We propose a content based video retrieval system in some main steps resulting in a good performance. From a main video, we process extracting keyframes and principal objects using Segmentation of Aggregating Superpixels (SAS) algorithm. After that, Speeded Up Robust Features (SURF) are selected from those principal objects. Then, the model "Bag-of-words" in accompanied by SVM classification are applied to obtain the retrieval result. Our system is evaluated on over 300 videos in diversity from music, history, movie, sports, and natural scene to TV program show.

Keywords: Video retrieval, principal objects, keyframe, Segmentation of Aggregating Superpixels, SURF, Bag-of-words, SVM.

TÓM TẮT

Hệ thống truy vấn video
dựa trên nội dung sử dụng phân tích thành phần chính

Truy vấn video nhằm tìm kiếm nội dung trong video hoặc clip gần giống với với ảnh hoặc video đầu vào. Một số thách thức khi thực hiện bài toán này bao gồm sự đa dạng của kiểu video, chuyển khung ảnh và vị trí camera. Ngoài ra, việc lựa chọn độ đo tương đồng cũng là vấn đề quan trọng cần giải quyết. Trong bài viết này, chúng tôi đề nghị hệ thống truy vấn video dựa trên nội dung trong một số bước chính nhằm đạt được hiệu suất cao. Với mỗi video, các khung ảnh quan trọng và các đối tượng chủ chốt được trích dựa trên giải thuật Segmentation of Aggregating Superpixels (SAS). Sau đó, mỗi đối tượng chủ chốt sẽ được tạo đặc trưng SURF. Sau cùng, sử dụng mô hình "Bag-of-words" kết hợp với bộ phân loại SVM để xác định kết quả truy vấn. Chúng tôi đã thực nghiệm trên 300 video thuộc các chủ đề khác nhau như âm nhạc, lịch sử, phim ảnh, thể thao, tự nhiên, và các chương trình truyền hình.

Từ khóa: Video retrieval, các đối tượng chính, khung chính, phân đoạn superpixel, SURF, đặc trưng túi từ, SVM.

[*] Email: *vietnq@hcmup.edu.vn*

1. Introduction

Internet development helps everyone to access a huge of online data easily. For example of video data, based on the Youtube web statistics, the number of people watching video monthly increases 50% than the previous year. There are 300 hours of video which are uploaded every minute. Therefore, data has been accumulated every day and every hour and it has become a huge database. A challenge is emerged: how we could search our interest or desired video from such huge database quickly and effectively? We need to set up a retrieval system that is able to process a content-based video search [1].

Video retrieval is a complicated process. The process generally is divided into many steps. Each step has its own target and the previous result will affect directly the next result. The preprocessing step target is: partitioning video into shots which have the same content frames. The retrieving step target is: extracting features from shots, clustering these features and classifying.

There are two main approaches in video retrieval problem: context-based video retrieval and content-based video retrieval. Context-based video retrieval is an approach using information such as text or audio. Advantages of such information are to search video based on the content from spoken words in the conversations. However, the performance in this kind will totally depends on the spoken word recognition process. Content-based video retrieval mainly focuses on visual features such as: color, texture, shape, motion, etc… The advantages of visual features are that there are a lot of information in video but the classification is more difficult than context-based classification.

Hybrid video retrieval is the combination of content and context based approaches with the desire of more accurate result. Some optimistic results in such approach is the sports video retrieval system SportsVBR of China [2].

Although we follow all of above approaches, there are still many obstacles in video retrieval. The demand of searching video quickly and effectively is a question because of a huge database and the diversity of video types, frame transitions, and camera angles. For the purpose of overcoming all difficulties robustly and flexibly, we propose a system including steps:

Step 1: Selecting keyframes and principal objects using Segmentation of Aggregating Superpixels (SAS) algorithm.

Step 2: Extracting SURF features from principal objects.

Step 3: Classifying video using SVM based on "Bag-of-words" model.

In the organization of this paper, we present the algorithm to find all shots from video in Section 2. Section 3 is about SURF feature extraction algorithm from each shot. And then, SVM is applied to classify video in Section 4. Some experiments and performance result are discussed in Section 5.

2. Shot detection

A shot is defined as the consecutive frames which are subtracted from video and have the minimum difference in content. In order to detect shots from a video, we choose the combination of measures [4]. The first measure is entropy of two frames and the second measure is subtraction of two frames. This combination give us a guarantee of an accurate shot boundary. Boundary of a shot must ensures that frame within a shot has a low difference in content and the transition of two shot is high difference. Figure 1 shows us an array of shots after being taken from a video.

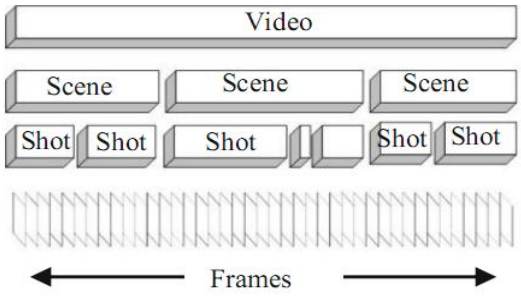

Figure 1. *An Array of shots extracted from video*

A shot is defined as the consecutive frames which are subtracted from video and have the minimum difference in content. In order to detect shots from a video, we choose the combination of measures [4]. The first measure is entropy of two frames and the second measure is subtraction of two frames. This combination give us a guarantee of an accurate shot boundary. Boundary of a shot must ensures that frame within a shot has a low difference in content and the transition of two shot is high difference. Figure 1 shows us an array of shots after being taken from a video.

Depending on the mentioned approach, we process three entropy and frame differences for calculations as:

- Difference between frame f(i) and the first frame of shot f(i_0) and their entropy difference.
- Difference between frame f(i+1) and the first frame of shot f(i_0) and their entropy difference.
- Difference between frame f(i+1) and the first frame of shot f(i) and their entropy difference.

Where f(i) and f(i+1) are the frame (i)th and (i+1)th, f(i_0) is the first frame of a shot. Figure 2 depicts us these symbols. These calculations are processed in iteration.

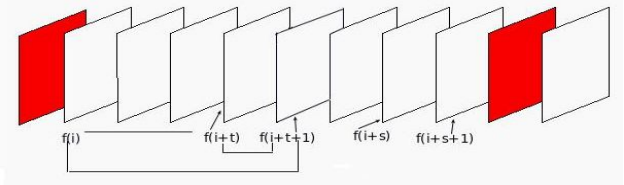

Figure 2. *Frames within a shot*

Using entropy and frame differences detect a shot is explained in formulas (1), (2) and (3).

$$bp2 = \sqrt{(preEnt - entFrm2)^2 + (preDiffEnt - diffCntEnt)^2} \quad (1)$$

$$bp3 = \sqrt{(bp2)^2 + (preRate - nmRate)^2} \quad (2)$$

$$bp = \sqrt{(preEnt - entFrm2)^2 + (preRate - nmRate)^2} \quad (3)$$

Where

- *entFrm2* is the entropy of frame f(i+1).
- *preEnt* is entFrm2 when go to the next iteration (i+2).
- *diffCntEnt* is the subtraction | entFrm2 - preEnt |.
- *preDiffEnt* is diffCntEnt when go the the next iteration (i+2).
- *nmRate* is the subtraction of f(i) from the first frame f(i_0).
- *preRate* is assigned by nmRate when go the the next iteration (i+1).

If *bp3* value is higher than a threshold, we can segment a video to a new shot. The result shows us a high accurate shot detection. It will be demonstrated in the Section 5. After shot detection, we define a vector which is represented a frame v as below, it has 09 dimensions and will be used for the next step to perform feature extraction from a shot.

$$\upsilon = (i0, i, entFrm2, nmRate, |preEnt - entFrm2|, |preDiffEnt - diffCntEnt|, |preRate - mnRate|, bp2, bp3).$$

3. Surf feature extraction

3.1. Principal Object Detection

Principal object is the main object which is focused by a camera. The principal object always have a highest color, sharpness and area information among the surrounding objects. A principal object belongs to the foreground of an image [3].

In order to detect the principal object in a image, we have a procedure in two steps: object segmentation and principal object detection.

3.1.1. Object segmentation

Assume that there are k objects in an image which are denoted by {O_1,O_2, …, O_k}. The algorithm of Segmentation of SAS aims to group all pixels in the same properties. These pixels are called superpixels. The below algorithm is SAS algorithm in detail [5]. Figure 3 depicts the result of SAS algorithm processing on an input image with k = 9.

***Algorithm**: Segmentation of Aggregating Superpixels [6]*

***Preprocessing**: Calculate value k (number of groups) by using histogram optimization.*

***Input**: Image I and the value k*

***Output**: k segmented objects*

a. Collect all superpixel S of I

b. Construct bipartite graph G

c. Cluster k groups from G

d. Evaluate pixels belongs to groups.

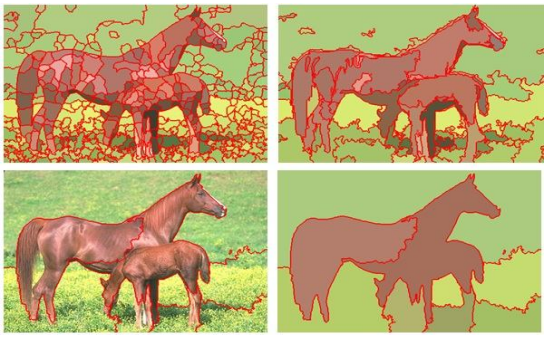

Figure 3. *The result of SAS on an input image with k = 9*

3.1.2. Principal Object Detection

From a set of objects $\{O_1, O_2, \ldots, O_k\}$, assume each O_i has the center (x_i, y_i) and size sz_i. We check two distances from center to border of image and size of O_i with a threshold. If the distances greater than d_1 and d_2 and the size greater than a threshold, O_i is the principal object. The algorithm of principal object detection is described below. Figure 4 is an illustration of value d_1, d_2 and the object O_i. The figure 5 is an example of algorithm output.

Algorithm*: Principal Object Detection*
Input*: Input image I, the value thresholdSize, d_1, d_2*
Output*: A set of principal objects*

For i=1: k
If ((size of O_i sz_i ≥ thresholdSize) and
(center O_i: distance from (x_i, y_i) to border of image is
greater than d_1, d_2))
O_i is determined as principal object
Else
continue
End
End

3.2. SURF Feature Extraction

SURF are scale and rotation-invariant interest point detector and descriptor [7-8]. It uses a Hessian matrix-based measure for detector and a distribution-based descriptor. A set of principal object will be the input to the feature extraction algorithm to provide features for each object. Figure 6 is the procedure of feature extraction on all objects. The algorithm is described in detail belows.

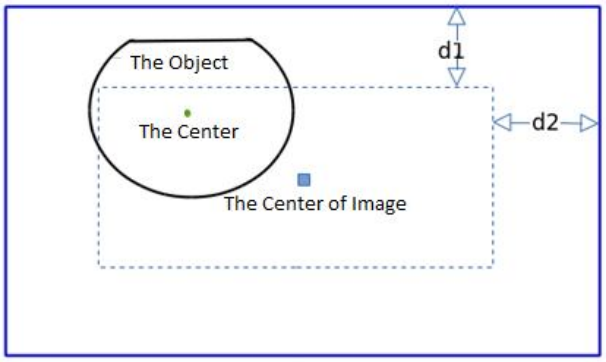

Figure 4. *Illustration of algorithm to detect principal object*

Figure 5. *The result of principal object detection*

Algorithm*: Feature Extraction from Principal Objects*
Input*: Image I, a codebook with size k*
Output*: a set of vector {v_1, v_2, …,v_m} for all k of principal objects.*
For i = 1: m
SURF feature calculation Fea_i = (f_1, f_2, …, f_n) of O_i
- *Calculate the frequency of feature Fea_i through codebook, we obtain $vecObj_i$ (frequency and codebook is described in Section 4 in BoW model)*
- *Save a frequency vector $vecObj_i$ = (v_1, v_2,…,v_k).*

End

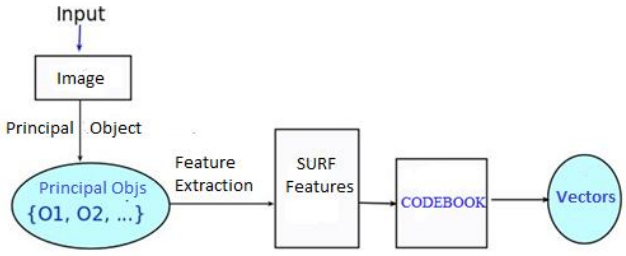

Figure 6. *The algorithm to extract SURF features from principal objects*

4. Video retrieval

4.1. "Bag-of-words" model

The bag-of-words (BoW) model is commonly used in methods of document

classification where the frequency of each word is used as a feature for training a classifier [9-11]. In similarity, we are able to apply this model in the other problem classification by the way of constructing some discriminative features in replacement of words, figure 7. As mentioned above, our features are SURF features which represent principal objects in one shot. The model is generally worked through three steps:

- Feature Extraction: we apply some method to extract discriminative features
- Codebook Construction: Codebook is a number of group after clustering all the features
- BoW Feature Representation: with each feature, we assign a codeword to codebook. And then, we construct a bin representation in which the value of bin is a frequency (or occurrence) of each feature. Figure 8 is an example of BoW representation.

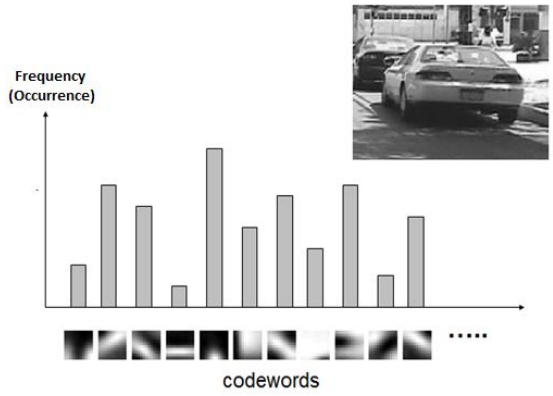

Figure 7. *BoW representation*

4.2. Video retrieval

Our content based retrieval system is constructed by SURF features of principal objects and then applied by SVM for classification based on BoW model, figure 8. There are totally 6 steps in the process of system.

- Shot Detection and choose Key Frames
- Principal Object Selection
- SURF feature extraction from Principal Objects
- Training Set Construction
- SVM Training [12]
- Video retrieval

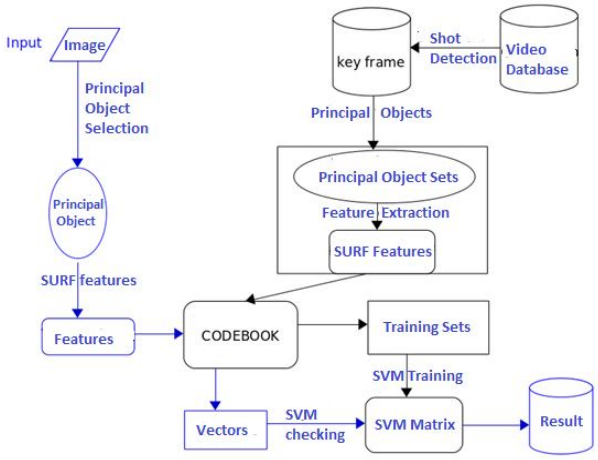

Figure 8. *The total model description of our proposed system*

The following figure is total model description of all process step by step for implementation. This model is a highest flexible and robust to any input video. That is our proposal to ensure and get a high performance of video retrieval.

5. Experimental results

5.1. Database and Environment

We construct database using 300 videos in which the content range spreads from music, history, comic, movies. interview, sports, natural scenes to TV shows. There are about 200 GB from TRECVID 2010. Out environment implementation is on Matlab R2012a and processed in desktop CPU Core i3 550 @ 3.20GHz, RAM 4GB.

5.2. Experimental Results

Video retrieval is a challenging problem for many researches. The accuracy is rather lower than expectation. However, by using our system, we can increase the accuracy to near about 70% for most of video types. Here are some results in related to each steps.

Table 1. *Shot detection result*

Consuming Time	Considered Value	Recall	Precision
67037 seconds	bp2, bp3	54.3%	0.7%
66482 seconds	bp	60.8%	41.7%

Table 2. *Consuming time for each steps*

Steps	Consuming Time
Feature Extraction	1670 seconds
“Bag-of-Words”	594 seconds
Training	3765 seconds
SVM	4435 seconds

Table 3. *Recall and Precision of our proposed model*

Codebook size	Recall	Precision
30	53.37%	64.74%
40	51.81%	56.34%
45	**63.67%**	**65.85%**
50	46.77%	49.44%
60	29.08%	38.41%
70	30.74%	39.55%
80	29.48%	39.76%

In the Figure 9, we can see that if we increase the size of codebook from 30 to 45, the accuracy of video retrieval is increase to nearly 70%. However, when we conduct some more experiements to increase codebook to 125, the accuracy descrease much. From this observation, we conclude that we should choose the codebook about 45 to get the optimal result in our model. The accuracy of 70% is an optimising result in comparison to the other approaches. Every year, there are some competition about video retrieval hold in the world in the purpose of increasing video searching to 80% but the algorithm is so complicated and time-consuming.

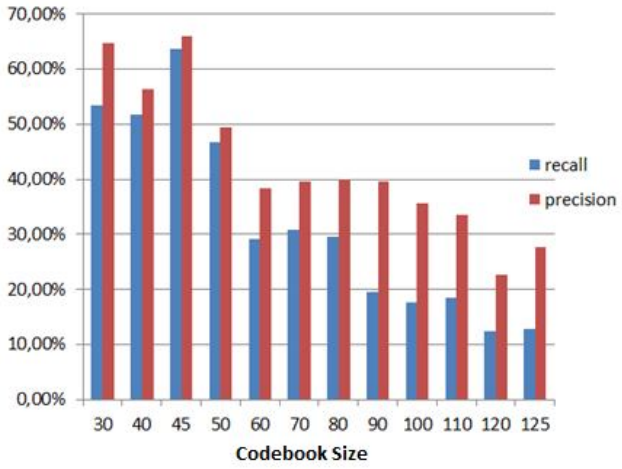

Figure 9. *Codebook size is 45 gives us the optimal result*

REFERENCES

[1] Mr. Ganesh.I.Rathod, Mrs. Dipali.A.Nikam, "Review on Event Retrieval in Soccer Video," *International Journal of Computer Science and Information Technologies*, vol. 5(4), 2014.

[2] Liu Huayong, Zhang Hui, "SportsVBR: A Content-Based TV Sports Video Browsing and Retrieval System," *ICEC'05 Proceedings of the 4th international conference on Entertainment Computing*, pp. 106-113, 2005.

[3] Bui Ngoc Nam and Pham The Bao, "Principal Objects Detection Using Graph-Based Segmentation and Normalized Histogram," *IJCSI International Journal of Computer Science Issues*, vol. 9, Issue 1, No 1, pp. 47-49, 2012.

[4] Gautam Pal, Dwijen Rudrapaul, Suvojit Acharjee, Ruben Ray, Sayan Chakraborty, and Nilanjan Dey, "Video Shot Boundary Detection: A Review," *Proceedings of the 49th Annual Convention of the Computer Society of India CSI* , vol. 2, pp. 119-127, 2015.

[5] Yuri Boykov, and Paria Mehrani Olga Veksler, "Superpixels and Supervoxels in an Energy Optimization Framework," *ECCV'10 Proceedings of the 11th European conference on Computer vision*, 2010.

[6] Zhenguo Li, Xiao-Ming Wu, Shih-Fu Chang, "Segmentation Using Superpixels: A Bipartite Graph Partitioning Approach," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool (2008), "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.

[8] Herbert Bay, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features," *9th European Conference on Computer Vision*, pp. 404-417, 2009.

[9] Gerard M. Salton, and Michael J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill: New York, 1986.

[10] L. Fei-Fei, "Recognizing and Learning Object Categories (slides)," *Stanford Vision Lab, Princeton University*.

[11] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision*, ECCV, pp. 1-22, 2004.

[12] Lipo Wang, *Support Vector Machines: Theory and Applications, Springer-Verlag*, New York, 2005.