Static Video Summarization With Threshold Method

Igor Chagas Marques
DECOM
CEFET-MG
Belo Horizonte, Brazil
igorchagasm@live.com

Flavio Luis Cardeal Padua
DECOM
CEFET-MG
Belo Horizonte, Brazil
cardeal@decom.cefetmg.br

***Abstract*—The huge amount of videos available on the web makes to know about a specific topic a time-consuming task. Watch all the videos is not a reasonable option. To address this problem we can build Video summaries improving the process of gathering, archiving, cataloging, indexing, and as well, increase the usability of stored videos. The present work proposes a thresholding method for building static video summaries using two distinct sets of features. The sets were compared according to the CUS metric. As a result, we offered a reasonable technique. Besides, we can also highlight that color features had turned out better metrics against saliency features.**

***Keywords*-video, summaries, thresholding, histogram, saliency**

I. Introduction

The recording and accumulation of large volumes of videos were made easy by the advances in storage and digital media technology. Every minute, a huge quantity of videos is uploaded on YouTube and other video-sharing websites. Searching by a specific topic leads us to hundreds of suggestions and it's time-consuming to browse among them. Quickly retrieve this huge data is a challenge nowadays [1].

A possible solution to this problem is called Video Summarization. It is a concise and meaningful representation of a video and can be achieved with many approaches. A video can be summarized by removing redundant data or by selecting salient content. There are two main types of summaries, the static ones, and the dynamic ones. The first one is a set of static frames extracted from video, and the second one is a brief sequence of video shoots (like a film trailer) [2].

Fig. 1. Static Video Summary produced by [3]

The existing Video Summarization techniques can be classified on the basis of its used features and on the basis of the used technique. An image can be represented by a matrix of intensities, commonly mapped with the 3 RGB channels. However, this representation could have high dimensionality. To overcome this, we often extract features from the frames. Some of the features that can be used are color histograms, textures, motion histograms, saliencies, faces and so on. Among the uesed techniques we can cite clustering algorithms, thresholding algorithms and SIFT (Scale Invariant Feature Transform) [2].

Besides that, it is important to know how good a technique is. In this field of study, there are three common existing evaluating methods: result description, objective metrics, and user studies. Result description is the most popular and simple form and does not involve any comparison with other techniques. In objective metrics, the metric is often the fidelity function computed from the extracted keyframe set and the original frame sequence. The user studies involve independent users judging the quality of generated video summaries and are probably the most useful and realistic form of evaluation [3].

Contributions: This paper offers a Static Video Summarization approach based on thresholding techniques. It presents a comparison among two feature choices for a variant method inspired by [4] statistical technique. Moreover, it evaluates the techniques with the widely used CUS metric proposed by [3].

A. Related work

The literature covers several works on Video Summarization. A static clustering approach using the 16 bins hue component from HSV histogram feature was proposed by [3]. Furthermore, [3] have proposed the CUS evaluating method adopted in this paper.

[5] aiming to reduce redundancy and generate a succinct representation of the video data had presented a technique for key-frame extraction based on RGB correlation, color histograms and moments of inertia.

[1] have offered an method combining representativeness, uniformity, static attention, temporal attention and quality which includes colorfulness, brightness, contrast, hue count, edge distribution for selecting keyframes.

A clustering work regarding key-frame extraction using weighted multi-view convex mixture models and spectral clustering was offered by [6].

Using a statistical approach [4] developed a thresholding method to generate static video summaries based on histograms. This work was performed with the KTH action database [7] and its evaluation was focused on compression ratio and fidelity value.

The present paper is also a statistical thresholding method, however, regardless of [4] we propose other features besides

gray scale histograms, added two extra steps and evaluate our method focusing on CUS [3].

B. Technique overview

Our method to produce a static summary from a video can be divided into the following sequential steps:

1) Pre-sampling.
2) Feature Extraction.
3) Distance Between Frames Calculation.
4) Threshold Calculation.
5) First Frame Filtering.
6) Second Frame Filtering.

The details of each step are deep approched in the next section.

II. Technical background

In this section, we detail our proposed technique.

A. Pre-sampling

By using a sampling rate, the number of video frames to be analyzed is reduced. On this work, we used a fixed frame rate of one frame per second, textit i. e., if a video contains 900 frames and its duration is 30 seconds (30fps), after sampling, we will have a set with 30 frames.

B. Feature Extraction

At this step we perform feature extraction for each pre-sampled frame.

An important aspect of this method is to choose an appropriet set of features to represent a frame. This papers shows a comparison among two set of features:

1) Color Histogram: Histograms are used for representing value frequencies of a scalar image, or of one channel or band of a vector-valued image [8]. According to [9] color is perhaps the most expressive of all the visual features. The VSUMM project cite avila chose the 16 bins hue component histogram of HSV color space since it is a popular choice for manipulating color whereas it was developed to provide an intuitive representation of color and to be close to the way in which humans perceive and manipulate color. Our first choice of features is also the 16 bins hue component histogram of HSV color space, however regardless [3], we worked with relative histograms.

The final output is a 16-dimensional array of real numbers with ranges from 0 to 1 meaning the color frequencies.

2) Mean Saliency Screen Position: Saliency is an image region where our eye-brain system quickly focuses [10][11]. At this step we performed a spectral residual static saliency detection algorithm (implemented and well documented by OpenCV) and then with the result saliency map we compute a binary map using Otsu threshold. The result is a binary map image of the frame saliency. Thus we compute the percentual position of the saliency centers on the map for X and Y coordinate. For calculating Xm (*i.e.*, x mean) we sum up all X coordinates of saliency pixels and then divide by the number of saliency pixels multiplied by the frame width. It gives us a position focus varying from 0 to 1, where 0.5 stands for the middle of the screen.The Ym calculus is analog to Xm, but using frame height.

$$Xm = \frac{\sum_s s_x}{s_{size} * w} \tag{1}$$

$$Ym = \frac{\sum_s s_y}{s_{size} * h} \tag{2}$$

Where s stands for the set of saliency pixels, s_x and s_y stands for pixel coordinates, s_{size} is the number of elements in s, w and h are the frame width and heigth dimensions.

The final output is a two-dimensional array of real numbers with ranges from 0 to 1 meaning the ”mean saliency screen position”.

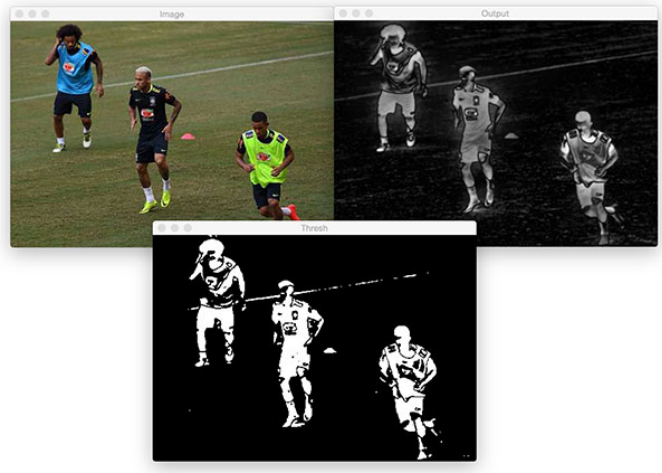

Fig. 2. Saliency Map produced by [10]

C. Distance Between Frames Calculation

As proposed by [4] to measure the distance of frames we compute the absolute distance between their features. In other words the distance is the sum of the absolute difference of each feature dimension. So we compute the distance of each frame with its next one.

D. Threshold Calculation

With the set of frame distances we compute its mean and standard deviation. The t threshold is the sum of them [4].

$$t = \mu + \sigma \tag{3}$$

The figures 3 and 4 show the distances between frames (the blue curve) and the threshold computed (the red line). As we can see, the two different feature sets gave us different choices for key-frames set.

E. First Frame Filtering

For each pre-sampled frame, we check if its distance from the next frame is greater than the threshold computed. If so it is selected. Else it is discarded. Doing this we are picking frames that show rasonable differences.

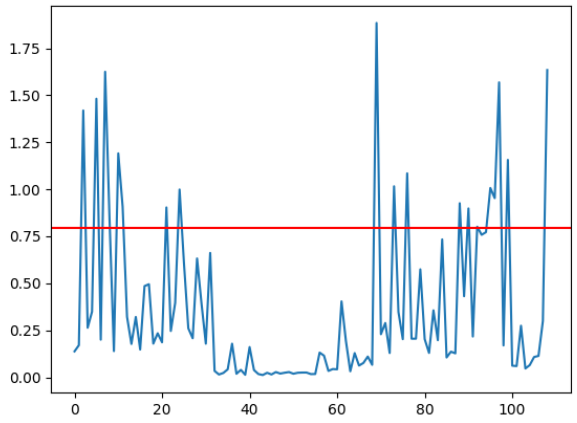

Fig. 3. Distances and Threshold computed using Color features on video The Great Web of Water from Open Video Data Base

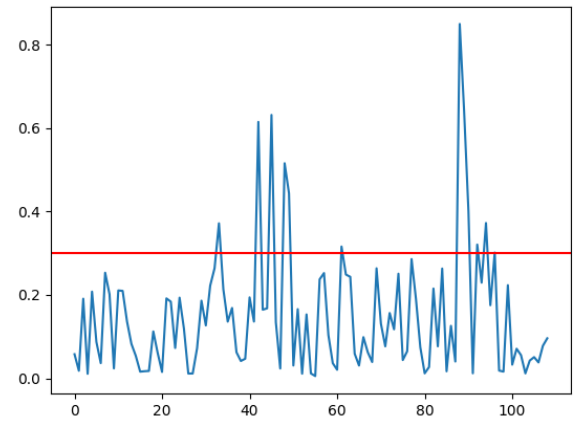

Fig. 4. Distances and Threshold computed using Saliency features on video The Great Web of Water from Open Video Data Base

F. Second Frame Filtering

After filtering the frames the first time we perform a second one for removing similar frames that were not direct neighbors on the pre-sampled set. Then, finally, as a result, we have our static summary.

III. Implementation

The code implementation of this proposed method was made with the following computer and enviroment specifications:

- Laptop Asus K47VM
- i7-3610QM CPU @ 2.30GHz
- 8GB RAM
- Nvidia Geforce GT630M 2GB
- Elementary OS 0.4.1 Loki 64-bit (Built on Ubuntu 16.04.5 LTS)
- Python 3.5.4
- OpenCV 3.4.1

Moreover the code itself can be found online on the Github [12].

IV. Quality Metrics

We validate the quality of summaries built by our technique with the *Comparison of User Summaries* (CUS) method. It was proposed by [3] and takes as reference summaries built by human users using sampled frames. With the two summaries built, two metrics are computed to express the summarie quality. CUS compares each automatic summary frame with the user summary frame. For this purpose it computes the hue component with 16 bins of the frame HSV histogram. With this feature it calculates the Manhattan distance between frames. If the distance is smaller than a threshold, we assume that the two compared frames are equal. The threshold value, estabilshed by tests is 0.5.

At the first step, a prefixed number of users are asked to watch the videos and then build up their summaries with sampled frames of these videos. They are oriented to select a set of frames that, in their opinion, is able to summarize the original video content and they can select any number of frames they think it is necessary. So, user summaries are compared with automatic summaries according to two metrics:

$$CUS_A = \frac{n_{mAS}}{n_{US}} \tag{4}$$

$$CUS_E = \frac{n_{\overline{m}AS}}{n_{US}} \tag{5}$$

Where n_{mAS} is the number of frames in the automatic summary that matches with the user summary. $n_{\overline{m}AS}$ is the number of non-matching frames in an automatic summary. And n_{US} is the number of user summary frames. CUS_A varies from 0 to 1. If 1 it means that all the frames in the user summary are present in the automatic summary. It is important to warn that if $n_{mAS} > n_{US}$, CUS_A can be equals 1 and not all frames from the automatic summary will be present in user summary. For CUS_E the values ranges from 0 (the best case, when all the keyframes from automatic summary matches with the frames from user summary) to $\frac{n_{AS}}{n_{US}}$ (when none frame from automatic summary matches with user summary). Therefore, these metrics are complementary and best quality possible is $CUS_A = 1$ and $CUS_E = 0$.

V. Results

To perform a interesting and fair comparison we adopted the same dataset as [3]. It consists in 50 videos extracted from The Open Video Data Base covering several genres (documentary, educational, ephemeral, historical, lecture) and with duration varying from 1 to 4 min. All the videos are in MPEG format (30 fps, 352 240 pixels).

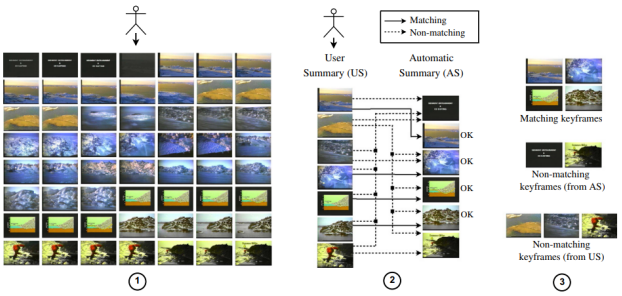

Fig. 5. CUS example, extracted from [3]

Feature Set	CUS_A	CUS_E
Color	0.65	0.60
Saliency	0.52	0.60

TABLE I
RESULT METRICS FOR FEATURE SETS

For each video we have 5 different user summaries, totalizing 250 user summaries.

We computed static summaries using our three proposed feature sets and then evaluate CUS_A and CUS_E for all of them. These metrics can be observed in I.

VI. DISCUSSION

The two chosen feature sets achieved the same metric CUS_E, however the color feature achieved reasonably better accuracy results in CUS_A. Perhaps this fact can indicate that colors are more attention captives for humans than the saliency position on the screen.

VII. CONCLUSION AND FUTURE WORK

In this paper, we introduced e presented a technique with two feature set approaches and showed a comparison between them. The results achieved are interesting and the color feature turned out better than saliency.

However, unfortunately, our results are not so good as [3]. Therefore as future work we propose to check other feature sets and also try to combine Color information and Saliency information in a weighted way.

REFERENCES

[1] M. Srinivas, M. M M, and R. M. Pai, "An improved algorithm for video summarization a rank based approach," *Procedia Computer Science*, vol. 89, pp. 812–819, 12 2016.

[2] P. Kaur, "Analysis of video summarization techniques," *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, pp. 1157–1162, 01 2018.

[3] S. Avila, A. Paula Brando Lopes, A. da Luz, and A. Arajo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, pp. 56–68, 01 2011.

[4] S. C V and N. Narayanan, "Key-frame extraction by analysis of histograms of video frames using statistical methods," *Procedia Computer Science*, vol. 70, pp. 36–40, 12 2015.

[5] N. Ejaz, T. Tariq, and S. Baik, "Adaptive key frame extraction for video summarization using an aggregation mechanism," *Journal of Visual Communication and Image Representation*, vol. 23, p. 10311040, 10 2012.

[6] *Key-Frame Extraction Using Weighted Multi-view Convex Mixture Models and Spectral Clustering*, Aug 2014.

[7] B. C. Ivan Laptev, "Recognition of human actions," 2005, access: 06/10/2019. [Online]. Available: http://www.nada.kth.se/cvap/actions/

[8] F. Cardeal, "Images in the spatial domain," 2019, access: 06/10/2019. [Online]. Available: http://cardeal.piim-lab.cefetmg.br/Teaching/cvision/Slides-02.pdf

[9] A. Tremau, S. Tominaga, and K. Plataniotis, "Color in image and video processing: Most recent trends and future research directions," *EURASIP J. Image and Video Processing*, vol. 2008, 05 2008.

[10] A. Rosebrock, "Opencv saliency detection," 2018, access: 06/10/2019. [Online]. Available: https://www.pyimagesearch.com/2018/07/16/opencv-saliency-detection/

[11] B. Davida, "Opencv static saliency detection in a nutshell," 2019, access: 06/10/2019. [Online]. Available: https://towardsdatascience.com/opencv-static-saliency-detection-in-a-nutshell-404d4c58fee4

[12] I. C. Marques, "Static video summary," 2019, access: 16/10/2019. [Online]. Available: https://github.com/Daegonny/static-video-summary