KS-SIFT: a keyframe extraction method based on local features

http://www.producao.usp.br/handle/BDPI/48796

Downloaded from Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
KS-SIFT: a keyframe extraction method based on local features

Tamires Tessarolli de Souza Barbieri
Computer Science Department
Sao Paulo University - ICMC
Sao Carlos, Brazil
tamires@icmc.usp.br

Rudinei Goularte
Computer Science Department
Sao Paulo University - ICMC
Sao Carlos, Brazil
rudinei@icmc.usp.br

Abstract— In this work we propose a new keyframe extraction method based on SIFT local features. We extracted feature vectors from a carefully selected group of frames from a video shot, analyzing those vectors to eliminate near duplicate keyframes, helping to keep a compact set. Moreover, as the keyframe extraction is based on local features, it keeps frames latent semantics and, therefore, helps to keep shot representativeness. We evaluated our method in the scene segmentation context, with videos from movies domain, developing a comparative study with three state of the art approaches based on local features. The results show that our method overcomes those approaches.

Keywords— Keyframe Extraction, Scene Segmentation, Visual Features

I. INTRODUCTION

The video scene segmentation is still an open research field and presents many challenges. It is more computationally complex than the problems of frame or shot segmentation, mainly due to the subjectivity of the concept and the semantics involved [6, 19].

Among the approaches for scene segmentation, those based on shot clustering are the most common [11, 15]. It is due to availability, computational cost and performance. The first step towards a successful segmentation process is to obtain a compact shot representation, which should be used to properly characterize shots. The majority of works found on literature use one keyframe as shot representation, defined, for example, as the shot first frame or median frame. But just one frame, in most cases, is not able of representing the variety of information in a shot, usually composed by hundreds of images that can have different content. Besides, with few exceptions, these works describe the keyframe using color histograms. However, color features have limited semantic and, despite the use of histograms presenting a low computational cost, it tends to lead to low efficiency when applied in segmentation tasks [13].

Therefore, in this work, we propose a new keyframe extraction method based on local features. Local features have being applied successfully in the image retrieval domain, mainly due their capabilities of to provide robust descriptors and to retain image semantics in a latent fashion [2, 3, 7, 8]. However, in spite of their potential, local features have been poorly explored in the video keyframe extraction field.

Using SIFT, we extract visual local features, represented as feature vectors, from a carefully selected group of frames from a video shot – a Keyframe Set (KS). Then, the proposed KS-SIFT method analyzes those feature vectors eliminating near duplicate keyframes, helping to keep a compact KS. Moreover, as the keyframe extraction is based on local features, it keeps frames latent semantics and, therefore, the shot representativeness.

In order to demonstrate our method efficiency for scene detection tasks, we have developed a naive baseline scene detection algorithm in a modular fashion. This algorithm is completely decoupled of keyframe extraction methods. In this way, we could implement three state of the art keyframe extraction approaches (local features based) found in literature [1, 4, 14] and compare their efficiency against the proposed KS-SIFT method. The comparison was made measuring classic Precision and Recall values when using each of the four keyframe extraction methods at time in the scene detection algorithm to segment videos in the movie domain. It must be stressed that the aim was to evaluate the keyframe extraction methods, not the scene detection.

The experiments show that our method is a viable alternative for keyframe extraction, achieving performance close to the related works and being more effective.

II. RELATED WORK

The scene segmentation techniques usually adopt compact approaches to represent a shot and most of them, as discussed at Section I, are based on keyframes and color histograms, but present low representativeness [13]. As can be seen in Image Retrieval domain, local features can bring benefits as an alternative to represent keyframes. So, in this section, we discuss keyframe extraction methods based on local visual features found at state of the art works related to video scene segmentation.

Baber et al. [1] describe each shot by one keyframe, defined as the shot median frame. Then, SURF (Speeded Up
Robust Features) features are extracted from each keyframe, which are used in their scene detection algorithm. This approach has low computational cost, since it considers only a small fraction of the available frames. On the other hand, there is the issue of selecting an image that doesn’t represent the most relevant content of the shot. In some cases, the median frame may not be the most suitable choice.

Chergui et al. [4] adopted a similar strategy; they also select a single keyframe to represent each shot. But their keyframe extraction method is less rigid. They consider that a relevant image contains rich visual details. Thus, they defined the keyframe as the frame with the highest number of points of interest in the shot. Despite using images content, it is not possible to guarantee that the frame with the highest number of points of interest is the most representative one in all cases. Besides, one image may not be enough to describe the diverse content of some shots and important information can be lost. This method is also more computationally demanding, because the selection step involves processing all shot frames.

Tapu and Zaharia [14] developed an approach to extract a variable number of keyframes from each shot. Using a window size parameter N, the first frame is selected N frames after a detected shot transition. Next, they analyze images located at integer multipliers of the window size N. These images are compared with the existing keyframes set already extracted. If the visual dissimilarity (defined as the chi-square distance of HSV color histograms) between them is significant (above a pre-established threshold), the current image is added to the keyframes set. Then, they discard irrelevant frames, computing points of interest with SIFT descriptor. If the number of keypoints is zero, the image is removed. After that, the keyframes are described by SIFT features. This keyframe extraction method has the advantage that not all shot frames are processed. However, many parameters need to be set (window size N, dissimilarity threshold, histograms quantization), what can influence the quality of the shot representation.

The related work presented in this section show that the use of local features can be an alternative for keyframe representations. However, as discussed, the current approaches present problems of representativeness and, sometimes, computational costs leading to high processing times.

III. **KS-SIFT Keyframe Extraction**

We developed a keyframe extraction method based on SIFT descriptor, designed to deal with the problems identified in related work and discussed at Section II, i.e., representativeness and computational cost due to high processing times.

A. **Choosing Keyframes Candidates**

In order to select the best frames to be the keyframes of each shot, we, initially, group some of them into a set we call Keyframes Candidates Set (KCS). The first frame to be included in the KCS is defined as the shot first frame. This has the goal of guarantee that each shot will be represented by, at least, one keyframe.

The next frames to be included in the KCS follow a windowing rule. We defined a window of size n and the frames at positions $n+1$, $2n+1$, $3n+1$, and so on, are selected for later analysis. We found the value 25 for n is a good one, in this case, because most of movies have 25 fps as capturing/exhibition rate and, generally, within 1 second there is no significant variation on consecutive frames content.

The next step is to extract SIFT [8] features from the frames in the KCS. The result is a number of feature vectors, of 128 dimensions, representing each frame. The exact number of vectors varies according to the frames content but it is generally high. This is another reason to adopt the windowing rule (mentioned before) instead of to use all frames in the shot.

In spite of the SIFT vectors extraction computational cost to be higher than broadly used color histograms, local features (like SIFT) provide a set of Points of Interest (PoIs) that identify an image. Moreover, these PoIs are invariant against illumination, rotations and scale conditions [3, 6, 7, 8]. In this way, in terms of representativeness, they preserve more image semantics than color histograms.

B. **Building a Keyframe Set**

In this step, a new set is built – the Keyframe Set (KS). The first frame included in the KS is the first one in the KCS. Then, each frame in the KCS is analyzed according to the following criterion: it will be inserted into the KS only if its number of PoIs is ± 60% different from the number of PoIs of each frame already in the KS.

The reasoning behind this criterion is to avoid the insertion of similar frames into KS, since similar frames do not add value to representativeness. This try to ensure the KS to have a minimum, but representative, set of keyframes. The 60% threshold was defined empirically. We used our base of videos (Section V) in order to test a range of rates between 10% (a more liberal one) to 90% (a more conservative one). The rate of 60% demonstrated to have a better tradeoff. Moreover, it is important to mention that we are using a variable threshold, defined based on features of a keyframe, instead of a fixed one. This is done in that way because the number of PoIs varies according to the frame content, being hard to find a good fixed value for the whole video.

C. **Analyzing the Feature Vectors**

If the criterion explained in the last section (III-B) fails, it means that the frame being analyzed to be included in the KS, let’s say α, has a number of PoIs similar to some other frame in the KS, let’s say β. So, we need to decide if α should be included in KS or not.

Even the number of PoIs being similar, α and β may be different if there are no coincident PoIs between them. So, the feature vectors describing those PoIs and representing those frames must be analyzed through some similarity measure. In this work we use the One-to-One Symmetric
matching (OOS) measure [17] to compare the feature vectors. We describe OOS later, at Section III-D.

The analysis is made verifying if there are more than 10% of feature vectors in common between \( \alpha \) and \( \beta \). If so, it means that \( \alpha \) can be represented by \( \beta \) and, in this way, \( \alpha \) do not need to be inserted into KS. Otherwise, if there are less than 10% feature vectors in common between \( \alpha \) and \( \beta \), it means that they are different and \( \alpha \) must be inserted into KS. It is important to notice that \( \alpha \) will be compared with every frame in KS, in the same way we described its comparison with \( \beta \), before it can be inserted into KS. The 10% threshold was defined empirically, in the same way described earlier at Section III-B.

The result of the KS-SIFT method, described in this Section (III), is a set of keyframes (KS) representing a video shot. Obviously the method is applied to all shots in a video, resulting in a number of KSs, one for each shot. The number of keyframes in a KS (KS size) may vary depending on the content diversity found in the respective shot. Figure 1 presents an example of keyframes selected using the KS-SIFT method for a shot of the A Beautiful Mind movie. It is possible to notice that the frames belonging to the KS have differences in semantics, representing different shot moments.

**D. Matching Measure**

To be possible to evaluate the proposed method in scene segmentation context, a similarity measure is necessary. We used OOS (One-to-One Symmetric matching) technique [17, 18] to identify matching points of interest described by SIFT, i.e., matching feature vectors, between two data sets. This technique aims to optimize the matches and presents better performance compared to other methods of literature, like [7], since it removes a large number of matches caused by noise [18].

OOS uses a partial matching scheme, based on cosine angle, i.e., only a subset of feature vectors is matched to exclude pairs with low similarity. The threshold that we used to specify the minimum similarity between two feature vectors for a potential match was 0.95, which is considered a very restrictive value.

Therefore, after using the OOS technique, we have as similarity measure of two data sets the number of matched feature vectors between them.

**IV. SCENE SEGMENTATION**

We developed a naive automatic scene segmentation technique to evaluate keyframe extraction methods applied to that domain. The technique searches for valleys in the used similarity measure between adjacent shots, for example, in the number of matched feature vectors.

We only consider a valley when the reduction and respective increase in the similarity value are significant. A meaningful reduction/increase rate depends on the characteristics of the analyzed video. So, instead of using a fix threshold, we developed a method to determine it. We compute all reduction values in the similarity measure, considering that it can occur within 5 shots. Then, we calculate a typical reduction to the video, i.e., we delete the 10% higher and 10% lower reduction values and compute the mean to the remaining ones. The result is the meaningful reduction/increase rate for the analyzed video. This is the minimum variation necessary to valley identification.

The next step is the scene transitions identification. For this, we cover all similarity values and verify if within 5 successive shots there was a reduction equal or higher than the reduction/increase rate previously determined. In the positive case, we still verify if within the next 5 shots there was an increase equal or higher than the same rate. If it was true again, then we identify a valley, i.e., a scene transition.

It is important to highlight that the developed scene segmentation technique is decoupled of the keyframe extraction method. Moreover, it is naive - it doesn’t present any heuristic, elaborated approaches or filters to improve the results. In this way, we can change the used keyframe extraction method and similarity measure and perform the scene segmentation in the same way. This characteristic is important to make possible to compare different keyframe extraction methods.

**V. EXPERIMENTS**

We evaluated the proposed (KS-SIFT) keyframe extraction method using five videos segments from movies domain. This domain presents videos with large quantity and variety of shots and scenes.

<table>
<thead>
<tr>
<th>Movie</th>
<th>Initial Frame</th>
<th>Number of Frames</th>
<th>Number of Shots</th>
<th>Number of Scenes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Beautiful Mind</td>
<td>2152</td>
<td>69013</td>
<td>609</td>
<td>28</td>
</tr>
<tr>
<td>Ice Age</td>
<td>685</td>
<td>72019</td>
<td>905</td>
<td>43</td>
</tr>
<tr>
<td>Gone in Sixty Seconds</td>
<td>4836</td>
<td>72010</td>
<td>1047</td>
<td>72</td>
</tr>
<tr>
<td>Pirates of the Caribbean</td>
<td>396</td>
<td>72008</td>
<td>1041</td>
<td>57</td>
</tr>
<tr>
<td>Back to the Future</td>
<td>4831</td>
<td>71970</td>
<td>572</td>
<td>37</td>
</tr>
</tbody>
</table>

We used the initial 50 minutes of each movie, disregarding the opening effects. These videos were converted to frames using FFMpeg (www.ffmpeg.org). Besides, KS-SIFT method considers that the videos were previously segmented into shots, so we did this step manually aiming to not influence the quality of the results. We also manually segmented the videos into scenes in order to create a ground truth to compare the results obtained with the automatic scene segmentation technique. In our work, a scene is defined as consecutive shot sequences that occur at
the same place [10, 12]. The characteristics of the videos used in the experiments are presented in Table I.

We compared KS-SIFT method with three other keyframe extraction schemes found in literature and previously discussed at Section II: Baber et al. [1], Chergui et al. [4] and Tapu & Zaharia [14]. These works belong to the state of art and contribute with important results for the video scene segmentation area. We implemented their strategies to select keyframes and, then, extracted SIFT features from the selected ones. For the four methods we consider the same similarity measure (number of matched feature vectors) and perform the scene segmentation in the same way, i.e., using the technique presented in Section IV. So, we can fairly compare the impact of the different keyframe extraction approaches in the results.

For the keyframes extraction method of Tapu & Zaharia, was necessary to set some parameters. One of them was the window size N. Analyzing our video database, we conclude that the best value for N is 10, since some movies have shots of small size. We also had to establish a threshold to define the visual dissimilarity between frames was significant. In this case, we determined empirically that the most appropriate threshold value is 0.3.

We evaluated the scene segmentation results using precision, recall and F1 measures. It is important to highlight that we used the Hanjalic’s evaluation [5] to match the ground truth with the automatic detected scene transitions, i.e., if the detected scene boundary is within four shots from the boundary detected manually, it is counted as a correct one. This criterion is commonly adopted by works related to scene segmentation [19].

The implementations needed in these experiments were developed using MATLAB (MATrix LABoratory), version R2012a.

VI. RESULTS AND DISCUSSION

A. Effectiveness Results

Table II presents the scene segmentation results obtained from the experiments described at Section V. We are interested in evaluating the impact that the different keyframe extraction methods have on scene segmentation task. So, the values presented should be analyzed on that context – it is not expected to have high values since the main portion of the whole process, the scene segmentation method by itself, is naive.

Analyzing the results, it is possible to note that the proposed KS-SIFT method presents better performance than the other approaches. Considering all videos, our method achieves F1 measure, on average, 5.7% higher than the other ones.

Besides, using our KS-SIFT method, it is possible to achieve similar values for precision and recall, indicating that the method is well-balanced. This is a relevant behavior, because as important as to identify only correct scene transitions is to guarantee that all scenes in the database were covered. In 3/5 of the movies the KS-SIFT presented very close values for precision and recall. Moreover, the average precision and recall were, both, 51% (consequently F1 was 51% too). None of the other methods achieved that balance, except Tapu and Zaharia, but with just 46%. These results are summarized at Table III.

<table>
<thead>
<tr>
<th>Movie</th>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Beautiful Mind</td>
<td>KS-SIFT</td>
<td>57%</td>
<td>59%</td>
<td>58%</td>
</tr>
<tr>
<td></td>
<td>Baber et al.</td>
<td>53%</td>
<td>37%</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>Chergui et al.</td>
<td>39%</td>
<td>44%</td>
<td>41%</td>
</tr>
<tr>
<td></td>
<td>Tapu &amp; Zaharia</td>
<td>42%</td>
<td>41%</td>
<td>42%</td>
</tr>
<tr>
<td>Ice Age</td>
<td>KS-SIFT</td>
<td>40%</td>
<td>50%</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>Baber et al.</td>
<td>46%</td>
<td>62%</td>
<td>53%</td>
</tr>
<tr>
<td></td>
<td>Chergui et al.</td>
<td>42%</td>
<td>52%</td>
<td>47%</td>
</tr>
<tr>
<td></td>
<td>Tapu &amp; Zaharia</td>
<td>47%</td>
<td>57%</td>
<td>52%</td>
</tr>
<tr>
<td>Gone in Sixty Seconds</td>
<td>KS-SIFT</td>
<td>56%</td>
<td>45%</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td>Baber et al.</td>
<td>47%</td>
<td>37%</td>
<td>41%</td>
</tr>
<tr>
<td></td>
<td>Chergui et al.</td>
<td>58%</td>
<td>42%</td>
<td>49%</td>
</tr>
<tr>
<td></td>
<td>Tapu &amp; Zaharia</td>
<td>52%</td>
<td>39%</td>
<td>45%</td>
</tr>
<tr>
<td>Pirates of the Caribbean</td>
<td>KS-SIFT</td>
<td>46%</td>
<td>48%</td>
<td>47%</td>
</tr>
<tr>
<td></td>
<td>Baber et al.</td>
<td>38%</td>
<td>41%</td>
<td>39%</td>
</tr>
<tr>
<td></td>
<td>Chergui et al.</td>
<td>40%</td>
<td>48%</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>Tapu &amp; Zaharia</td>
<td>46%</td>
<td>54%</td>
<td>50%</td>
</tr>
<tr>
<td>Back to the Future</td>
<td>KS-SIFT</td>
<td>54%</td>
<td>53%</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>Baber et al.</td>
<td>53%</td>
<td>44%</td>
<td>48%</td>
</tr>
<tr>
<td></td>
<td>Chergui et al.</td>
<td>46%</td>
<td>44%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>Tapu &amp; Zaharia</td>
<td>41%</td>
<td>39%</td>
<td>40%</td>
</tr>
</tbody>
</table>

B. Processing Times Versus Effectiveness Results

We also compare the processing time necessary to extract the keyframes for each one of the four methods analyzed. This comparison was made calculating the processing time for 50 shots randomly obtained from the base described at Section V. Table IV shows the results. The term “Keyframe” means the total time spent to extract the keyframes and the SIFT feature vectors. The term “Match”, in turn, means the time spent to match the feature vectors (Section III-D), identifying correspondences.

<table>
<thead>
<tr>
<th>Keyframe</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>KS-SIFT</td>
<td>12.1</td>
</tr>
<tr>
<td>Baber et al.</td>
<td>2.3</td>
</tr>
<tr>
<td>Chergui et al.</td>
<td>198.8</td>
</tr>
<tr>
<td>Tapu &amp; Zaharia</td>
<td>6.9</td>
</tr>
</tbody>
</table>

Analyzing Table IV one can notice that, for all methods, the time necessary to extract keyframes is much higher than the time necessary to measure similarity, meaning that the
adopted matching strategy do not highly influence the whole process. It is also possible to notice that the processing times are very different, in spite of all methods are keyframe-based. The less time consuming method is Baber et al., however, it is also the simpler one – it uses the shot median frame as shot keyframe – and it is the less effective (F1 is 45%) one.

The approach of Tapu & Zaharia also has low processing time. This is due to the fact that they use color histograms (computationally less time consuming than local features) to select a set of candidate frames and, only then, they filter and represent some of them as SIFT-based keyframes. Despite that histogram-based filtering approach to reduce the processing time, it loses important information and also reduces representativeness – average effectiveness rate were 46% (Table III).

Chergui et al. presents the higher costs for processing time. This can be explained by the used keyframe extraction method, in which all frames are processed in order to extract points of interest before select the one that will represent the shot. Unfortunately, this computational effort is not reflected into effectiveness, since the average F1 was the worst (45% - Table III).

In turn, as the KS-SIFT processes more frames than the former two methods (Tapu & Zaharia and Baber et al.), it was expected a substantial increase in the processing time. But, the increase was, on average, 1.8 times higher than Tapu & Zaharia, 5.3 times higher than Baber et al. and 16.4 times lower than Chergui et al. It is worth to mention that, in the KS-SIFT case, this relative increase in processing time was compensated with better representativeness, leading to a better effectiveness rate (51% - Table III), as discussed at Section VI-A.

VII. Final Remarks

In this work, we presented a keyframe extraction method based on visual local features. This method uses SIFT to extract features in order to build a representative set of keyframes. We showed comparative results in scene segmentation context. The proposed approach proved to have superior effectiveness to three state of the art related segmentation context. The proposed approach proved to have superior effectiveness to three state of the art related methods.

Acknowledgment

The authors thank FAPESP (grant 2012/19025-0) and CNPq for their financial support to this work.

References