Keyframe Extraction Algorithm Based on Moving Object Features (Postprint)

Authors: Lihua Tian, Zhang Mi, Li Chen

Date: 2018-09-12T00:00:00+00:00

Abstract

To address the problem that features in sports videos are difficult to extract and that the keyframe results tend to contain many missed frames, a keyframe extraction algorithm based on moving object features is proposed. This algorithm emphasizes moving object features while weakening background features, thereby preventing missed detection and redundancy caused by moving objects being too small while the background occupies the main content of the video frame. First, frames with significant color changes are selected as partial keyframes based on video frame entropy; for frames without sudden color changes, feature points of moving objects within the frame are obtained using Scale-Invariant Feature Transform (SIFT) of moving objects. Finally, video keyframes are extracted respectively based on frame entropy and the distribution of SIFT points of moving objects. Experimental results show that the keyframe result set obtained by this algorithm not only has a low miss detection rate but also can accurately represent the content of the original video.

Full Text

Preamble

Key Frame Extraction Algorithm Based on Feature of Moving Target

Tian Lihua, Zhang Mi, Li Chen†

(School of Software Engineering, Xi'an Jiaotong University, Xi'an 710086, China)

Abstract: Motion features are difficult to extract, which easily leads to missed and redundant frames in the result. To address this problem, this paper proposes a method of key frame extraction based on feature of moving target. The method reduces redundant rate and missed rate of the result set by emphasizing the features of the moving target and weakening the background features of the frame. In this method, a frame with a burst entropy change is taken as part of

the key frame firstly. Then, our method extracts the SIFT points of the moving target from the frame of which the entropy value has not suddenly changed. Finally, it extracts key frames according to the entropy and SIFT distribution respectively. Experimental results show that the miss rate of this algorithm is low. At the same time, key frame results can accurately and completely describe the main content of the video.

Key words: key frame extraction; gaussian mixed model detection; SIFT; perceptual hashing

0 Introduction

Key frame extraction is an important step in video retrieval, and the quality of its extraction results directly affects the efficiency of video retrieval. Due to frequent scene switching in motion videos, there are many missed detection frames in their key frame results. Various methods for key frame extraction in motion videos have been proposed. Literature [1,2] suggests extracting key frames by analyzing the optical flow field to understand motion fields. Ma et al. [3] propose extracting key frames when the state of motion objects changes. Li et al. [4] propose a motion-focusing key frame extraction algorithm that uses uniformly moving objects as reference to obtain motion targets in video and extract key frames. For object motion caused by camera movement in videos, Guironnet et al. [5] propose extracting key frames by detecting camera motion, flexibly adjusting the key frame extraction method according to different lens motion patterns. However, the above methods have limited capability for key frame extraction in motion videos, and when there are fast-moving objects in the video, the key frame results are prone to missed detection frames. Since local features can better preserve frame semantics, there are currently many algorithms that extract key frames based on local features. Among them, Hannane et al. [6] propose extracting key frames based on frame SIFT distribution histograms. Barbieri et al. [7] propose extracting candidate frames based on fixed-size windows, then extracting key frames based on SIFT feature distances between candidate frames. However, this method cannot accurately describe the change process of moving targets.

Combining the characteristics of motion videos, this paper proposes a key frame extraction algorithm based on moving target features. The algorithm first selects frames with sudden color changes as partial key frames based on frame entropy values. For frames without sudden entropy changes, the SIFT features of moving objects are extracted based on the Gaussian mixture model to obtain the change degree of moving objects within the frame to determine whether they are key frames. Extracting key frames based on the SIFT distribution characteristics of moving objects emphasizes the change of moving targets within the frame while weakening the influence of background changes, thus effectively capturing the change process of moving objects. Finally, the perceptual hashing algorithm is used to obtain frame fingerprints, and the Hamming distance is calculated to remove redundant frames in the results, further improving the

effective expression of key frames to the original video.

Technical Background

1.1 Gaussian Mixture Model

Gaussian mixture modeling (GMM) is suitable for modeling videos with complex backgrounds to detect moving targets [8,9]. To detect moving objects according to the Gaussian mixture model, we first calculate the probability density of pixel values in a frame over a time period, then determine whether each pixel belongs to the background based on common principles in statistical difference. When reading a new video frame, the model is updated first, then each pixel in the frame to be processed is traversed to determine whether it is a background point. After processing all pixels in the frame, the background model of that frame is obtained, and then the foreground is obtained based on the background. The Gaussian mixture model can effectively model videos with complex content.

1.2 SIFT Features

SIFT (scale invariant feature transform) feature points are stable local features. Representative points on the object surface are selected to describe the object. These points are usually called interest points [10], such as points with higher brightness in darker areas, points with lower brightness in bright areas, or special points on object edges. Interest points do not change significantly due to image scaling or rotation, and have certain robustness to light changes and noise interference.

1.3 Perceptual Hashing

Image perceptual hashing technology, also known as digital fingerprinting, is a summary of multimedia information. Perceptual hashing is a one-way mapping from multimedia representation to hash values, where each image has its corresponding fingerprint. The perceptual hashing algorithm first removes image details, ignores image size and scale differences, and only retains basic information such as structure and shadows. Then it calculates the DCT coefficient matrix of each frame' s grayscale image, keeping only the top-left 8×8 matrix. It calculates the average grayscale value of all pixels, compares each pixel in the matrix with this average value, marks it as "1" if greater than or equal to the mean and "0" otherwise, and finally generates the frame fingerprint. The Hamming distance is calculated based on frame fingerprints to determine frame similarity.

2 Key Frame Extraction Based on Moving Target Features

To emphasize changes of moving targets in videos, this paper proposes an algorithm for extracting key frames based on moving target features. The algorithm first calculates frame entropy values and selects frames with sudden color

changes as partial key frames. For frames without sudden color changes, key frames are extracted based on the SIFT distribution of moving objects within the frame. Finally, redundant frames are removed based on the perceptual hashing algorithm. The core steps of the algorithm are shown in Figure 1 [Figure 1: see original paper].

The algorithm flow is as follows: First, read the video frame sequence and calculate frame entropy values according to formulas (1) and (2), selecting frames with sudden entropy changes as partial key frames. For frames without sudden entropy changes, use the Gaussian mixture model (GMM) to detect moving objects in the frame and perform morphological erosion and dilation processing. Then extract the SIFT features of the frame' s moving content, calculate the inter-frame distance based on adjacent frames' SIFT point distribution to extract key frames. Finally, calculate the percentage of moving target pixels in the frame (Motion-Rate) to determine whether the current frame is disturbed by background motion, and use the perceptual hashing algorithm to remove redundant frames from the result set.

2.1 Gaussian Mixture Model for Extracting Moving Target Features

Read the frame sequence and calculate the ratio of entropy values between adjacent frames, selecting frames with entropy mutations as partial key frames. For frames without sudden entropy changes, further calculate the change degree of moving objects within the frame to determine whether they are key frames. The specific process is as follows:

a) For frames without sudden entropy changes, use GMM to obtain the foreground moving objects in the video frame, with results shown in Figure 2(b).

b) To make the contours of moving objects within the frame clearer, perform erosion and dilation processing on the foreground content. Figure 2(c) shows the result of erosion and dilation processing on the frame in Figure 2(b).

c) Obtain the motion content of all frames without sudden entropy changes in the video one by one, finally obtaining a set of binarized images containing only moving objects.

Extract the foreground content from video frames to obtain binarized images containing only moving objects, where white areas represent moving objects in the video and black areas represent background content. Then extract the SIFT features of the frame' s motion content and calculate the inter-frame distance based on SIFT point distribution of adjacent frames.

As shown in Figure 3(a), this is the result after erosion and dilation processing of moving objects detected by Gaussian mixture in a video frame. The SIFT points extracted and marked from Figure 3(a) are shown in Figure 3(b). From

the feature point distribution in Figure 3(b), we can see that SIFT points can effectively mark the positions of moving objects within the frame.

2.2 Motion-SIFT Distance Calculation

After calculating the SIFT feature points of moving objects in the frame (Motion-SIFT), key frames are extracted based on their distribution in each sector. To calculate the SIFT distribution distance, we first need to obtain the SIFT feature points of moving objects in the frame, then divide each frame into multiple sector regions. Next, convert the SIFT feature points to polar coordinates, count the number of SIFT points in each sector, and calculate inter-frame distances to extract key frames. The Motion-SIFT distance calculation process is as follows:

1) Frame Sector Partitioning

To count the SIFT distribution of moving objects in video, our algorithm divides video frames into multiple sectors. Since the main content of motion videos may appear in any region of the frame, the video frame is divided into multiple sectors based on different radii and angles. Let h be the height of the video frame to be partitioned and w be the width. The center of the video frame in the Cartesian coordinate system is O(xo,yo), with xo = w/2 and yo = h/2. Using the video frame center point as the origin, the frame is divided into multiple sectors based on different radii and angles, with results shown in Figure 4 [Figure 4: see original paper].

The specific steps for video frame sector partitioning are as follows:

a) First obtain the height h and width w of the video frame. Using the video frame center O as the center, draw circles with radii r1 = w/6, r2 = w/3, and r3 = w/2. The video frame is divided into 3 annular regions by these three concentric circles.

b) Partition SIFT feature points according to different angles: using 45 degrees as a unit, four lines passing through the video frame center O divide the frame into 8 angular ranges. Finally, based on different angles and distances, the video frame is divided into 24 sectors.

2) Coordinate System Transformation

After completing frame partitioning, calculate the distribution quantity of SIFT feature points in each region. Count the number of feature points in each region and convert all feature points from Cartesian coordinates to polar coordinates with the original video frame center O as the pole. (xi,yi) represents the Cartesian coordinates of the ith SIFT feature point, and the polar coordinate representation of this feature point (xi,yi) is (ri, i).

As shown in formula (3), first calculate the coordinate value (xi' ,yi') of the feature point with coordinates (xi,yi) in the Cartesian coordinate system with the frame center as the origin.

$$\begin{cases} x_i' = x_i - w/2 \\ y_i' = y_i - h/2 \end{cases}$$

Then convert each feature point to the polar coordinate system with the video frame center point O (w/2, h/2) as the pole. Calculate the corresponding polar coordinates (ri, i) of the SIFT point (xi,yi) in the polar coordinate system according to formulas (4) and (5).

$$\theta_i = \arctan\left(\frac{y_i' - y_o}{x_i' - x_o}\right)$$

$$r_i = \sqrt{(x_i' - x_o)^2 + (y_i' - y_o)^2}$$

Repeat the above steps to obtain the polar coordinates of all feature points, count the distribution quantity of feature points in each sector, and extract key frames.

3) Motion-SIFT Distribution Statistics

To calculate the SIFT distribution distance, we first need to count the number of feature points in each region and record it in Count[i][j], where i and j represent the radius r range and angle range of the SIFT feature points respectively. When r = 0, it means the pixel point is within the smallest circle (r1 = w/6). When r = 2, it means the pixel point is within or outside the largest circle (r3 = w/2). When 0 < < /4, the corresponding array j value for this feature point' s angle is 0.

After obtaining the number of SIFT points in each sector of the frame, determine which region the SIFT feature point belongs to based on its polar coordinates (r,). The sector region (i,j) that the feature point (r,) belongs to can be calculated by formulas (6) and (7), and this feature point is recorded in Count[i][j].

4) Inter-frame Distance Calculation

Extract Motion-SIFT feature points based on moving objects to calculate inter-frame distances. Let two consecutive frames in the video be fk and fk+1, with corresponding Motion-SIFT feature point distribution arrays Countk[i][j] and Countk+1[i][j]. As shown in formula (8), calculate the feature point distribution distance between the two video frames (SiftCountDiff).

$$SiftCountDiff(f_k, f_{k+1}) = \sum_{i=0}^{2}\sum_{j=0}^{7} |Count_k[i][j] - Count_{k+1}[i][j]|$$

Since motion videos may misdetect background as moving targets, causing a sudden increase in SiftCountDiff values, using the average value as a threshold

to extract key frames can easily cause missed detection of key frames where motion changes rapidly. Therefore, this algorithm uses the ratio of feature point distribution distances between adjacent frames to measure the degree of motion feature change, with the calculation process shown in formula (9).

$$MotionChange(f_k, f_{k+1}) = \frac{SiftCountDiff(k, k+1)}{SiftCountDiff(k-1, k)}$$

The ratio of Motion-SIFT distribution distances between adjacent frames in the video is recorded as MotionChange. The inter-frame feature distance is judged based on the ratio of adjacent frames to determine whether it is a key frame. When the MotionChange(fk, fk+1) value undergoes a sudden change, further determine whether the current frame fk+1 is disturbed by background motion. If not disturbed by background, the frame is selected as a key frame and added to the key frame set; otherwise, compare the feature point distribution distances of the next two adjacent frames sequentially until all frames are traversed.

Method: Extracting Key Frames Using Frame Entropy and Motion-SIFT Distribution Distance

```
Count[i][j] ← 0      // number of feature points of each sector
for n = 0 → iNum do          // number of frames of video
    extract Motion-Object of frame by GMM
    extract SIFT points of frame
    divide frame into sectors
    for each SIFT point do
        calculate polar coordinates
        Count[i][j] ← Count[i][j] + 1
    end for
     call formula(4-9) calculate SiftCountDiff
     call formula(4-10) calculate MotionChange
     if MotionChange >   then
         if not background interference then
             add frame to keyframe set
         end if
     end if
     calculate Hamming distance of keyframes
     remove redundant frames
 end for
```

3 Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, we implemented and tested it on a computer with Windows 7 operating system, Intel Core i3 processor, and 2GB memory, using Visual Studio 2010 and OpenCV 3.0 as the development platform. Multiple motion videos were selected for testing, including soc-

cer games, basketball games, sports teaching, gymnastics, skating, and fencing videos.

3.1 Evaluation Criteria

Precision (P) and Recall (R) are effective evaluation standards for result sets. Recall is used to measure the missed detection of key frames as shown in formula (12). Precision reflects the accuracy of extraction results as shown in formula (13).

$$R = \frac{N_c}{N_c + N_m} \times 100\%$$

$$P = \frac{N_c}{N_c + N_f} \times 100\%$$

Where Nc, Nm, and Nf represent the number of correctly extracted key frames, missed detection frames, and false detection frames in the results, respectively. Since it is difficult to balance R and P values in key frame extraction, this paper uses the harmonic mean F of the two to comprehensively measure the effectiveness of the result set, defined as formula (14).

$$F = \frac{2 \times P \times R}{P + R}$$

3.2 Experimental Results

3.2.1 Visual Results of Key Frame Extraction In the experiments, 15 motion video clips were selected for testing, and parameter values were determined to analyze and summarize the key frame extraction results. The results show that when the Hamming distance n is 5, the judgment of similar frames is most accurate; when the MotionChange ratio is greater than 3, it can be determined that the frame has undergone a sudden change relative to the previous frame.

To verify the effectiveness of the proposed algorithm, it was compared with the SIFT distribution histogram-based key frame extraction algorithm (SIFT-HD) [6]. The visual results are shown in Figures 5 [Figure 5: see original paper] to 7 [Figure 7: see original paper].

In motion videos, the change of moving objects within frames accounts for a small percentage of the entire frame content. Therefore, this paper uses the Gaussian mixture model to detect moving objects in frames and calculates the percentage of white pixels in the entire frame (Motion-Rate) to determine the change of motion content. When the Motion-Rate value between adjacent frames changes significantly within a local range, it is determined that a large amount of background information is involved.

To determine whether the key frame extraction process is disturbed by moving background, we first need to count the number of all white pixels in the frame, recorded as MotionCount. Then calculate the ratio of the total number of pixels of moving targets in the current frame to be judged and the previous key frame (MotionChange). The ratio of the total number of moving target pixels between adjacent key frames is shown in formula (10).

$$MotionChange(i) = \frac{MotionCount(i)}{MotionCount(cur)}$$

The percentage of moving targets in the frame is calculated as formula (11).

$$MotionRate = \frac{MotionCount}{Allpixel} \times 100\%$$

Based on the pixel percentage of moving targets in the frame and the change degree between the current frame and its previous frame, we can determine whether the current frame is disturbed by background motion. When there is background interference, continue to determine whether the next candidate frame is a key frame; when there is no background interference, add the frame to the key frame set.

Finally, use the perceptual hashing algorithm to remove redundant frames from the obtained key frame set. Calculate the Hamming distance between adjacent frames in the result set, and consider redundancy when the distance between two frames is less than n.

Figure 5(a) detected 4 key frames, while Figure 5(b) detected 8 key frames. Both Figure 5(a) and Figure 5(b) have 1 redundant frame. According to the original video, the athlete completed multiple sets of movements, including two separate movements and three joint movements. Figure 5(a) only reflects the two joint lifting movements, so Figure 5(b) expresses the video content more accurately.

As shown in Figure 6 [Figure 6: see original paper], the SIFT-HD algorithm detected 2 key frames, while our algorithm detected 8 key frames. In the original video, the player on the right attacked twice and defended once, while the player on the left attacked once and defended twice. As shown in Figure 6(b), our results can completely express the attack and defense situations of both athletes, while Figure 6(a) cannot accurately express the original video content.

As shown in Figure 7(a), the key frame result can only describe the athlete' s three-step layup process, losing the action of the athlete attempting to shoot and being blocked. Figure 7(b) can completely describe the entire process.

3.2.2 Analysis and Comparison of Experimental Results The proposed algorithm was compared with the SIFT distribution histogram-based key frame

extraction algorithm (SIFT-HD) [6] and the frame SIFT feature-based key frame extraction algorithm (KS-SIFT) [7]. The redundancy rate, missed detection rate, and their harmonic values were calculated. Since it is difficult to manually label key frames in motion videos, the results of our algorithm and comparison algorithms were used as mutual reference standards to determine missed detection frames.

The statistics of missed detection and redundant frames in key frame results between our algorithm and the SIFT-HD algorithm [6] are shown in Table 1 .

Table 1. Comparison of Missed Detection and Redundancy with SIFT-HD Algorithm

Video	Total Frames	Literature [6]	Our Algorithm	Literature [6]	Our Algorithm	Literature [6]	Our Algorithm
Gymnastics	155	-	-	-	-	-	-
Basketball 1	149	-	-	-	-	-	-
Basketball 2	115	-	-	-	-	-	-
Soccer 1	85	-	-	-	-	-	-
Soccer 2	82	-	-	-	-	-	-
Soccer 3	240	-	-	-	-	-	-
Soccer 4	102	-	-	-	-	-	-
Fencing	125	-	-	-	-	-	-

As shown in Table 1, both our algorithm and the SIFT-HD algorithm have a small amount of redundancy in their results, but our algorithm has fewer missed detection frames than literature [6]. The key frames detected by the SIFT-HD algorithm are all included in the results of our algorithm; however, some key frames in our algorithm results were not detected by the SIFT-HD algorithm. Therefore, our algorithm achieves better reproduction of the original video.

Our algorithm was further compared with the KS-SIFT algorithm [7]. The statistics of missed detection and redundant frames in the results of both algorithms are shown in Table 2 .

Table 2. Comparison of Missed Detection and Redundancy with KS-SIFT Algorithm

Video	Total Frames	Literature [7]	Our Algorithm	Literature [7]	Our Algorithm	Literature [7]	Our Algorithm
Gymnastics	155	-	-	-	-	-	-
Basketball 1	149	-	-	-	-	-	-
Basketball 2	151	-	-	-	-	-	-
Soccer 1	85	-	-	-	-	-	-
Soccer 2	82	-	-	-	-	-	-
Soccer 3	240	-	-	-	-	-	-
Soccer 4	102	-	-	-	-	-	-
Fencing	125	-	-	-	-	-	-

As shown in Table 2, both our algorithm and the KS-SIFT algorithm [7] have very few redundant frames in their results. Our algorithm results have very few missed detection frames, while the reference algorithm has more missed detection frames. Therefore, with similarly low redundancy, our algorithm has higher recall and more complete expression of video content.

The F-values of key frame results from the three algorithms were compared, and the statistical results are shown in Figure 8 [Figure 8: see original paper]. As shown in Figure 8, our algorithm has better comprehensive performance, and the F-values of our algorithm results are more stable compared to the other two algorithms. The F-values of the SIFT-HD algorithm and KS-SIFT algorithm are comparable overall, but both have individual videos with F-values below 50%. The key frame result sets extracted by the SIFT-HD algorithm [6] and KS-SIFT algorithm [7] both have many missed detection frames, so their F-values are slightly lower than our algorithm.

We further compared the running efficiency of the three algorithms. In the experiment, 8 motion video clips were tested, all with frame size 480×272. The average frame processing time of the three algorithms is shown in Table 3 .

Table 3. Comparison of Algorithm Processing Time

Algorithm	Average Time per Frame (s)
KS-SIFT	-
SIFT-HD	-
Our Algorithm	-

As shown in Table 3, the KS-SIFT algorithm takes the least time because it selects frames at fixed positions as candidate frames, then selects key frames from candidate frames, which can greatly reduce the amount of data to be processed. However, although this method is simple and fast, it is not flexible enough, and the selection of candidate frames often has deviations. The SIFT-HD algorithm extracts key frames based on frame SIFT distribution histograms, which is certainly more time-consuming than the KS-SIFT algorithm, but generally has better results. Compared with the SIFT-HD algorithm, our algorithm first obtains moving objects in the frame through Gaussian mixture modeling, then extracts SIFT feature points of moving objects. This method reduces the number of SIFT feature points to be processed, but requires some time for Gaussian mixture extraction of moving targets, so its execution time is slightly higher than the SIFT-HD algorithm. Although our algorithm takes slightly more time, the key frame combination obtained is more accurate in expressing the original video, and the time consumption is in the same order of magnitude, meeting engineering requirements.

4 Conclusion

For motion videos, this paper proposes a key frame extraction algorithm based on moving target features. The algorithm emphasizes changes of moving objects in videos and weakens changes of moving backgrounds, thereby preventing missed detection of key frames caused by excessive or overly complex background content that makes moving target features inconspicuous. Experimental results show that even when the motion background is relatively complex and moving targets occupy a small percentage of the frame, the algorithm can still effectively detect changes in moving targets. The key frame results obtained by the moving target feature-based key frame algorithm have high fidelity to the original video content and can completely and accurately express the main content of the original video.

References

[1] Cui Zhigao, Wang Hua, Li Aihua, et al. Moving object detection based on optical flow field analysis in dynamic scenes [J]. Acta Physica Sinica, 2017, 66(8): 97-104.

[2] Shao Ling, Ji Ling. Motion histogram analysis based key frame extraction for human action//activity representation [C]// Proc of Canadian Conference on Computer and Robot Vision. Washington DC: IEEE Computer Society, 2009: 88-92.

[3] Ma Yanzhuo, Chang Yilin, Yuan Hui. Key-frame extraction based on motion acceleration [J]. Optical Engineering, 2008, 47(9): 957-966.

[4] Li Congcong, Wu Yita, Yu S S, et al. Motion-focusing key frame extraction and video summarization for lane surveillance system [C]// Proc of IEEE

International Conference on Image Processing. 2010: 4329-4332.

[5] Guironnet M, Pellerin D, Guyader N, et al. Video summarization based on camera motion and a subjective evaluation method [J]. Eurasip Journal on Image & Video Processing, 2007, 2007(1): 060245.

[6] Hannane R, Elboushaki A, Afdel K, et al. An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram [J]. International Journal of Multimedia Information Retrieval, 2016, 5(2): 89-104.

[7] Barbieri T, Goularte R. KS-SIFT: a keyframe extraction method based on local features [C]// Proc of IEEE International Symposium on Multimedia. 2015: 13-17.

[8] Huang Dongjun, Yang Yinghua. Modified Gaussian mixture background model for moving object detection [J]. Application Research of Computers, 2017, 34(6): 1862-1866.

[9] Yu Zhengqiang, Pan Yun, Huan Ruohong. A moving detection method combining frame diffrence and gaussian mixture model [J]. Computer Applications and Software, 2015(4): 129-132.

[10] Li Haiyang, Wen Yongge, He Hongzhou. An improved SIFT feature point detection method [J]. Computer Applications and Software, 2013(9): 147-150.

Note: Figure translations are in progress. See original paper for figures.