A Motion-Insensitive Dissolve Detection Method with SURF

Yaqiong Wang Yang Yang Tongwei Ren Gangshan Wu
State Key Laboratory for Novel Software Technology at Nanjing University
Meng Mingwei Building, NO.22 Hankou Road, Nanjing
{wyq,yangy,rtw}@graphics.nju.edu.cn, gswu@nju.edu.cn

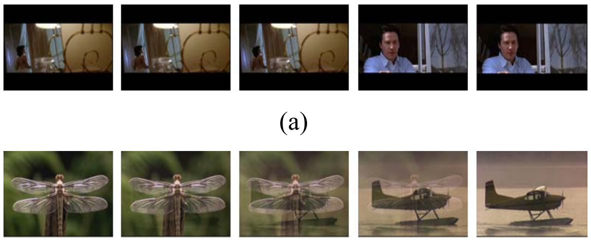
Abstract

As dissolve is the most common gradual shot transition, dissolve detection plays an important role in video segmentation which is the fundamental step for efficient video indexing and retrieval. However, the existing detection methods easily confuse dissolve with camera motion or object motion when using global features. Besides, when using local features' change tendency, they can't get accurate trajectories to reflect this. In this paper, we propose a SURF feature based dissolve detection algorithm which can well differentiate dissolve from motion appearances. We get the trajectories of SURF key points by matching between two successive frames. Then a candidate set of dissolves is obtained according to the distribution of the starting points and ending points on the trajectory, filtering part of motions. Dissolves are further located by analyzing the curve of the proportion of sub-trajectories with monotonous variation through each frame. Experiments demonstrate the effectiveness and efficiency of the proposed method.

1. Introduction

The rapid development of multimedia technology leads to the wide application of video data, such as distance learning, surveillance, and TV on demand [1]. To manage the exponentially growing video data, many techniques are proposed for efficient video organization. As one of the core video organization techniques, shot boundary detection is used to detect the video shot transitions and partition the original video into more manageable parts [2]. It is the foundation of the further video indexing, retrieval and other analysis tasks.

Video shot transitions is usually classified into two categories: abrupt cut and gradual transition. Abrupt cuts take place between two consecutive frames due to camera switch (Figure 1(a)) [1]. It can be efficiently detected from the obvious change between the two frames. Compared with cut, gradual transition has more complex representations and can further be classified into dissolve, fade-in/out, wipe and other types. Among these video gradual transitions, dissolve (Figure 1(b)) is the most common one, which takes a very large part in gradual transition. Moreover, fade-in/out can also be treated as the special case of dissolve. Therefore, dissolve detection is crucial in gradual transition detection and draws much attention of the researchers [2]-[14].



(a)

(b)

Figure 1. Examples of cut and dissolve

Many existing shot boundary detection methods try to detect dissolve with other types of shot transition using the same mechanism [2]-[9]. These methods can not deal with the complicated nature of dissolve well and usually have a low precision and recall in dissolve detection. To address this problem, several methods are specially proposed for dissolve detection [10]-[13] or dealing with motion problem in a special way from cut [14]. However, the existing dissolve detection methods cannot efficiently differentiate dissolve to the appearances of camera motion and object motion, because the motion and dissolve have very similar appearance in selected feature [14]-[13] but they can't deal with this situation well or they didn't take this problem considered [10]-[12].

In this article we focus on this problem and propose a novel and robust method. We deeply analyze the difference between dissolve and motion. And with accurate trajectories of SURF feature points, we can make a model which reflects the difference well. This model can make a promising result.

The rest of this paper is organized as follows. We give a brief overview of the previous work in section 2. In section 3, we describe the dissolve detection method with SURF feature in detail. In section 4, we examine the proposed method and compare it with the current typical methods. The paper is closed with conclusion and future work.

2. Related work

Recently, shot boundary detection, especially gradual transition detection, has attracted much attention. According to the detection strategy, the existing methods can be roughly categorized into two types, feature-based method and model-based method [6]. Feature-based methods devote to analyze the shot transitions from the extracted features, such as color histogram in decompressed domain or motion vectors in compressed domain. Zhang *et al.* [2] utilize the color histograms as the feature and propose a twin-comparison method with different thresholds for abrupt cuts and gradual transitions detection. Zabih *et al.* [4] propose a method based on the local maximum fraction of edge changes, which is the larger one of the proportion of the exiting edge pixels in the former frame and the proportion of the entering edge pixels in the latter frame. Model-based methods formulate shot boundary detection with the predefined mathematical or statistical model(s). Zhang *et al.* [7] use HMM model to detect each type of shot transition with statistical corner change ratio and HSV color histogram difference. Jones *et al.* [8] formulate the problem with CRFs model and detect shot transitions with principled parameter learning and inference. Boccignone *et al.* [9] calculate the perception of the visual details changes in each scene and generate a consistency measure of the foveation sequences, and then detect the shot transitions based on Bayesian inference. The above methods, both feature-based and model-based, devote to detect all types of shot transitions with a uniform mechanism. They do not deal with the special complicated nature of dissolve well, so they usually have low precision and recall in dissolve detection.

As an important and difficult part of shot boundary detection, dissolve detection has drawn attention of many researchers. Nam *et al.* [12] address this problem by fitting the gray-level intensity changes with B-spline interpolation method and locating the dissolves in the positions with low fitting error and high variances of gray-level intensity changes. However, similar to the previous two methods, this method easily confuses dissolve with camera motion and object motion in many situations. To solve this problem, Su *et al.* [13] calculate the proportion of pixels, whose gray-level intensities monotonically change within a window, and locate the dissolves in the positions where the proportion is larger than a pre-defined threshold. Lawrence *et al.* [14] identify the parts with obvious motion with first-order partial derivative and detect the dissolve in the local maximum locations based on temporal partial derivative without the motion area. However, the above methods may also confuse dissolve with motion sequences when motion sequences have the similar behavior using intensity change because they can't get the accurate tracks to reflect the real intensity change tendency of points. This is the main reason which causes false and misdetections in these algorithms. The further discussions will be shown in section 4.

3. SURF feature based dissolve detection

To efficiently differentiate dissolve from motion sequence, we propose a detection method based on SURF feature. We first detect the feature points with SURF algorithm in each video frame and track the feature points by matching the feature points in successive frames. Then, we calculate the weighted sum of the appearing and disappearing feature points in each frame and treat the video parts with high weighted sums as the dissolve candidates. Third, each trajectory of a dissolve candidate is checked, locating the monotonous part. Finally, locations of dissolves are calculated by the coherence rate of monotonous part.

3.1. SURF feature extraction

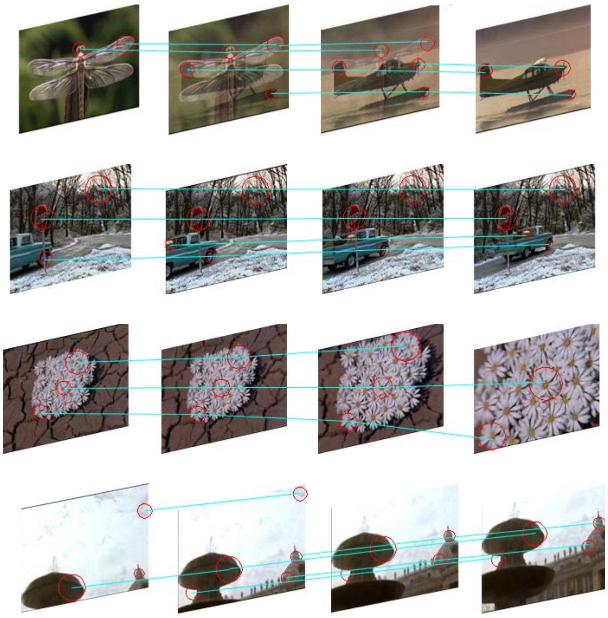
SURF feature is an extension of SIFT, which has better robustness and efficiency than SIFT [15]. It is invariant to luminance changing, contrast adjusting, motion and geometric distortion. Moreover, it is good at differentiating two video shots with different content and widely used in copy detection. Compared with the luminance change and other feature in dissolve detection, SURF feature can represent better stability on the unchanged video content and more obvious variation when the video content changes. Hence, we utilize SURF feature as the feature in the proposed method.

We first detect the feature points in video frames. To reduce the time cost, we constrain the number of feature points in each frame should no more than a certain number. And generally speaking, 200 points will be enough for our situation.

After detecting the feature points, we extract the feature for each feature point, and match the feature points in the successive frames based on the similarities between the feature point features. To make the matching more reliable, we introduce a Hough space filter to estimate an affine transformation between two frames which can make us obtain a near 100% precision matching rate.

With the matching between successive frames, we connect the matched feature points and obtain the tracks of feature points. To avoid the influences of noise points, we remove the tracks whose length less than length of 2 frames and ignore the feature points in these tracks. In this way, we can define the status of each feature point p_k according to its position in the track. If p_k is in the beginning of a track, its status is "appearing"; if p_k is in the end of a track, its status is "disappearing"; otherwise, its status is "stable". To simplify the expression, we collectively refer the disappearing feature point and the appearing feature point to "unstable feature point".

Figure 2(a) shows an example of some trajectories of a dissolve sequence. The feature points are marked with red circles in each frame and the tracks are represented with the green fold lines through the frames. Similar examples are shown in Figure 2(b), Figure 2(c) and Figure 2(d) which belong to object motion, camera zooming, camera panning respectively



3.2. Dissolve candidate extraction

Dissolves can be defined as the combination of the last frames of disappearing shot and the first frames of the appearing shot in the duration from t_1 to t_2 [13]:

$$f(t)=\alpha(t)f_{dis}(t)+\beta(t)f_{app}(t),\quad t\in[t_1,t_2],\qquad (1)$$

where $f(t)$ is the frame of dissolve in time t ; $f_{dis}(t)$ and $f_{app}(t)$ are the frames of the disappearing shot and the appearing shot at time t respectively; $\alpha(t)$ and $\beta(t)$ are the monotonic increasing function and the monotonic decreasing function in the range of [0,1] respectively.

The descriptors of SURF cannot keep invariant during the dissolve due to combination of two sequences. Trajectory made by feature points may be broken at some position during the dissolve. Meanwhile, new trajectory starts due to the appearing of new feature points.

Besides dissolve, some other reasons may also cause the unstable feature points. For example, abrupt cuts may cause a lot of disappearing feature points and appearing feature points in the last frame of the disappearing shot and the first frame of the appearing shot respectively. In camera motion, pan and title may cause the unstable feature points near the frame boundaries, and zoom in/out may cause some unstable feature points in the whole frame. Strong object motion may cause the unstable feature points within the region(s) of object(s). And noise cause a little of unstable feature points sometimes. However, the feature points are much more stable within a shot which don't have strong object motions and camera motions.

Considering the difference in the unstable feature point positions caused by different reasons, we can get a dissolve candidate set without the normal video frames within a stable shot, part of pan/title motion and small object motion sequence by analyzing the tendency of a simple weight sum of the unstable feature points.

We define the weight of feature point according to its position in the video frame. Assume the coordinate of feature point p_k is (x_k, y_k), we define its weight w_k as follows:

$$w_k = 1-2\times\max\left\{\frac{|x_k-x_0|}{W},\frac{|y_k-y_0|}{H}\right\},\qquad (2)$$

where (x_0, y_0) is the coordinate of the frame center, W and H are the weight and height of the video frame respectively. It means the feature points near frame center have larger weight than the ones near frame boundaries.

Based on the feature point weight definition, we calculate the weight sum sum_{weg} of the feature points in each frame f_t:

$$sum_{weg}(f_t) = \frac{\sum_{p_k \in P_{ust}} w_k}{N_{ust} + N_{stb}}, \quad (3)$$

where P_{ust} is the set of unstable feature points in frame f_t; N_{ust} and N_{stb} are the numbers of the unstable feature points and the stable feature points, respectively. To avoid missing the frames with low weight sum in dissolve, we further instead the weight sum of each frame with the mean value of the weight sums of its neighboring frames:

$$\overline{sum_{weg}}(f_t) = \frac{1}{2T_{sum}+1} \sum_{k=t-T_{sum}}^{t+T_{sum}} sum_{weg}(f_k), \quad (4)$$

where T_{sum} determines the number of neighboring frames of f_t. In our experiment, we set T_{sum} 2 because the shortest dissolves last six frames in a video of 30fps.

If some continuous frames whose weight sum value are larger than a threshold and the range of these continuous frames is over six frames, we treat these frames as a dissolve candidate.

3.3. Dissolve Location

The generated dissolve candidates include many other types of video sequence, such as strong object motion, camera zoom in/out and intense camera pan/title with many noises. To filter these motion sequences, we calculate the feature variation monotonicities of the feature points within a window. Considering the length of the shortest dissolve in our experiment, we also set the length of the window six frames. According to the definition of dissolve in Equation (1), the feature values of the feature points in the same trajectory transform from the feature of appearing point to the feature of the disappearing point monotonically. Compared with dissolve, the feature values of the feature points in a trajectory within motion sequence usually have little changes, and the feature values of the feature points caused by noise will change randomly. Figure 3 shows how the trajectories change through motion and dissolve. The white parts in a trajectory show this part change in a monotonous way. The area in the blue rectangle is the change tendency through motion and the green area is the change tendency through dissolve. We can see clearly the difference between the trajectories part through dissolve and motion.

In order to reflect this character, we should set a model to reflect the change tendency of part trajectories.

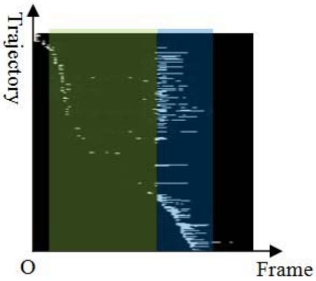


Figure 3. Monotonous points on trajectories

We define the feature value variation vat_k of feature point p_k as the distance between its feature and the feature of the starting point of a window in the same track:

$$vat_k = dis(p_k, p_0), \\ k \in \{1,2,3,4,5\} \quad (5)$$

where p_0 is the starting feature point in the same window of p_k, $dis(p_k, p_0)$ is the distance between feature values of p_k and p_0. When this trajectory is through a motion, the five feature value variation will in a stable state. However, when it is through a dissolve sequence, these five values will behavior in a monotonous way. Considering the effect of noise, there can also be randomly fluctuating sometimes during the dissolve but the monotonous tendency always exists while this situation rarely happens during a motion sequence. In order to weaken the influence of noise, we only calculate the variation in a window from the fourth frame.

When most of trajectories whose monotonous part are through a continuous frames we can take this sequence a “monotonous sequence” and think this sequence is a dissolve. In order to locate this sequence, we calculate the monotonicity of the feature point track pt_j within a window with the start frame f_t as follows:

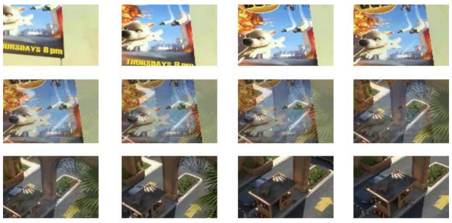
$$mot(pt_j, f_t) = \prod_{i=4}^{6} sign(vak_i) \quad (6)$$

where $sign(x)$ equals to 1 when $x > 0$, otherwise $sign(x)$ equals to 0.

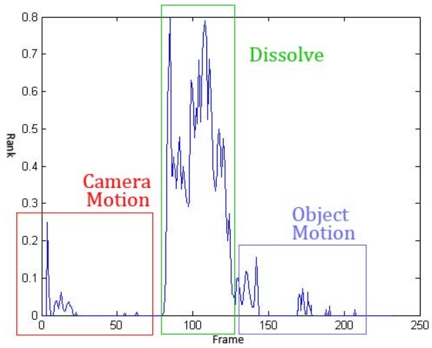
If $mot(pt_i)$ equals 1, then we can assume this part of the trajectory is a monotonous part and may be through a dissolve sequence. So the start frame f_t many be in a dissolve sequence. However, due to the existence of noise, when a certain number of this kind of trajectories through this frame we can make this conclusion. A model as follows is given to reflect this nature, we mark each frame with a rank and this rank reflects the proportion of monotonous sub-tracks.

$$rank(f_t) = \frac{\sum_{j=1}^{num} mot(pt_j, f_t)}{num} \quad (7)$$

where num is the number of tracks through the frame f_t. If a frame have a very high rank and its neighbors within window all have continuous much higher values of $rank(f_t)$, we can take this sequence a dissolve.



(a)



(b)

Figure 4. Curve of ranks for a sample video

4. Experiments

We evaluate our algorithm on ten videos, which include three television serials, two documentaries, four movies and one music video. All the data is extracted from original video without any manual editing. The dissolves in these videos are manually marked as background. Before we start taking our algorithm we use [16] to detect the cuts in the videos, and our algorithm deals with the sequences between the cuts. The information about the dissolves in videos is listed in table 1. The videos are all in MPEG1 format and the frame size of all the video is 720 × 480. Music videos do not contain any dissolves but they contain many sequences with large motion. The two documentaries are taken from the BBC documentaries. They contain many complicate dissolves and motions which have very similar behavior. The TV series and Movies have very complex scenes with rapid camera motion and object motion.

We use traditional method with the recall and the precision to evaluate the performance of our algorithm.

We take the first step on sequences without cuts to get the trajectories and a candidate set of dissolve. In our experiment, we set the threshold of weight sum value to be 0.02. After this, we get a curve of rank value for every sequence. Figure 4(a) shows a sequence with camera motion, dissolve, and object motion. Figure 5(b) shows the curve of $rank(f_t)$, we can see that our algorithm can differentiate them well. When the rank of frame over 0.3 and the duration of continuous frames of this kind last over six frames, we take this sequence a dissolve.

Table 1. Test video data information

video	#frames	#Dissolves
television serials	276045	138
Documentaries	89460	82
Movie	605884	125
Music video	11470	9

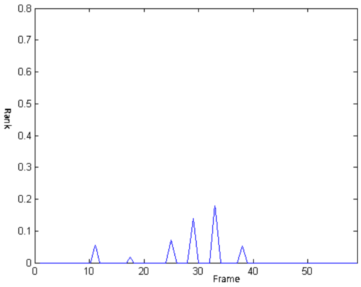


Figure 5. Ranks of a zooming video

Here we compare our work with some representative methods on the same dataset. The methods in [2]-[4] are classic method using twin-comparison method and ECR, and in [11]-[12] are algorithms dealing with dissolve specially. The algorithms in [13]-[14] are motion-insensitive methods which both use the intensity information as their feature. When motion have similar behavior with dissolve or large camera zoom in/out and object motion happens, they use MVs to track the points but this kind of information can’t help much .As a result, these algorithms can’t get very promising results in these situations.

SURF feature is robust to geometric distortion. That enables us to get much more accurate trajectories, based on which, we analysis change tendency. Figure 5 shows the curve of Figure 2(c), which is a camera zooming with obvious intensity change. The result is very promising. The comparison result is showed in table 2 as follows, and we can see that our algorithm performs much better.

Table 2. Experiment results

Method	recall	precision
twin-comparison method	0.626	0.685
ECR	0.712	0.423
Statistical Methods	0.803	0.391
B-Spline Interpolation	0.823	0.742
Monotonous Intensity Change	0.805	0.732
Partial derivatives	0.813	0.796
Proposed algo.	0.913	0.859

5. Conclusion

We propose a novel dissolve detection method by analyzing the change tendency of SURF feature points' trajectories to tell difference between dissolve and motion. Because of the robustness of SURF, we can obtain a more accurate trajectory through camera zooming or object motion than using MVs. Ranking of monotonous sub-trajectories locates dissolve well as demonstrated in experiment. SURF descriptor is not stable under extreme shape distortions, so we cannot trace feature points with very rapid or intensive motions. Our feature work will add some extended features to get more excellent performance.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China under the Grant 60533080.

7. References

[1] V. Chasanis, A. Likas, and N. Galatsanos, "Simultaneous detection of abrupt cuts and dissolves in videos using support vector machines", *Pattern Recognition Letters,* Elsevier Science Bv, Amsterdam, Netherlands, 1 Jan. 2009, vol. 30(1), pp. 55-65.

[2] C. Cotsaces, N. Nikolaidis, I. Pitas, "Video shot detection and condensed representation", *IEEE Signal Processing Magazine,* IEEE, NJ, USA, MAR. 2006, vol. 23(2), pp. 28-37.

[3] H.J. Zhang, A. Kankanhalli, and S.W. Smoliar, "Automatic partitioning of full-motion video", *Multimedia Systems*, Springer, NJ, USA, 1993, vol. 1(1), pp. 10-28.

[4] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects", *Multimedia Systems*, Springer Verlag, NY, USA, Mar. 1999, vol. 7(2), pp. 119-128.

[5] H. -W. Yoo, H. -J. Ryoo, and D. -S. Jang, "Gradual shot boundary detection using localized edge blocks", *Multimedia Tools and Applications*, Springer, Dordrecht, Netherlands, 2006, vol. 28(3), pp. 283-300.

[6] M. A. Fouad, F.M. Bayoumi, "Real-time shot transition detection in compressed MPEG video streams", *Electronic Imaging*, I S & T, VA, USA, 2008, vol. 17(2).

[7] W.G. Zhang, J.Q. Lin, and X.P. Chen, "Video Shot Detection Using Hidden Markov Models with Complementary Features", *the First International Conference on Innovative Computing, Information and Control*, IEEE Computer Society, CA, USA, 2006, vol. 3, pp. 593-596

[8] J.H. Yuan, J.M. Li, and B. Zhang, "Gradual Transition Detection with Conditional Random Fields", *Proceedings of the 15th international conference on Multimedia*, ACM, Augsburg, Germany, Sep. 2007, pp. 277-280.

[9] G. Boccignone, A. Chianese, V. Moscato and A. Picariello, "Foveated Shot Detection for Video Segmentation", *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, NJ, USA, MAR. 2005, vol. 15(3), pp. 365-377.

[10] G. -S. Lin, M. -K. Chang, and S. -T. Chiu, "Dissolve Detection Scheme With Transition Duration Refinement", *3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IEEE Computer Society, Kaohsiung TAIWAN, vol. 1, 26-28 Nov. 2007, pp. 155-158.

[11] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Fade and dissolve detection in uncompressed and compressed video sequences", *International Conference on Image Processing*, IEEE Computer Society, Kobe, Japan, 1999, vol. 3, pp. 299–303.

[12] J. Nam and A. H. Tewfik, "Detection of Gradual Transitions in Video Sequences Using B-Spline Interpolation", *IEEE Transactions on Multimedia*, IEEE Computer Society, NJ, USA, Aug. 2005, vol. 7(4), pp. 667-679.

[13] C. -W. Su, H. -Y. M. Liao, H. -R. Tyan, K. -C. Fan and L. -H .Chen, "A Motion-Tolerant Dissolve Detection Algorithm", *IEEE Transactions on Multimedia*, IEEE, NJ, USA, Dec. 2005, vol. 7(6), pp. 1106-1113

[14] S. Lawrence, D. Ziou, M.F. Auclair-Fortier, S. Wang, "Motion Insensitive Detection of Cuts and Gradual Transitions in Digital Videos", *Pattern Recognition and Image Analysis*, Springer, 2002.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features", *European Conference on Computer Vision*, Springer, Berlin, Germany, 2006, vol. 3951, pp. 404-417.

[16] C.W. Ngo, T.C. Pong and R.T. Chin "Video Partitioning by Temporal Slice Coherency", *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, NY, USA, Aug. 2001, vol. 11(8), pp. 941-953.