PERGAMON

Pattern Recognition 35 (2002) 1847–1867

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Spatial–temporal joint probability images for video segmentation

Ze-Nian Li *, Xiang Zhong, Mark S. Drew

School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, BC, Canada V5A 1S6

Received 1 December 2000; received in revised form 22 June 2001; accepted 22 June 2001

Abstract

Effective annotation and content-based search for videos in a digital library require a preprocessing step of detecting, locating and classifying scene transitions, i.e., temporal video segmentation. This paper proposes a novel approach—spatial–temporal joint probability image (ST-JPI) analysis for temporal video segmentation. A joint probability image (JPI) is derived from the joint probabilities of intensity values of corresponding points in two images. The ST-JPT, which is a series of JPIs derived from consecutive video frames, presents the evolution of the intensity joint probabilities in a video. The evolution in a ST-JPI during various transitions falls into one of several well-defined linear patterns. Based on the patterns in a ST-JPI, our algorithm detects and classifies video transitions effectively.

Our study shows that temporal video segmentation based on ST-JPIs is distinguished from previous methods in the following way: (1) It is effective and relatively robust not only for video cuts but also for gradual transitions; (2) It classifies transitions on the basis of predefined evolution patterns of ST-JPIs during transitions; (3) It is efficient, scalable and suitable for real-time video segmentation. Theoretical analysis and experimental results of our method are presented to illustrate its efficacy and efficiency.

Keywords: Temporal video segmentation; Spatial–temporal joint probability images

1. Introduction

Digital libraries are important for their ability to manage huge amounts of heterogeneous data, such as text, sound, images and even digital videos. Among all the data forms that are stored and managed by digital libraries, digital videos are the most informative because of their high information throughput.

Conventionally, a digital video is often considered as merely a large collection of digital images in chronological order, with images completely independent at one another. This challenges attempts of storing and manipulating videos in digital libraries since a typical one and a half hour movie consists of 162,000 frames, with 0.9 Mbyte per frame (true color and 640×480 resolution). With the growing number of videos in digital libraries, storage and manipulation become prohibitive tasks.

Fortunately, videos usually have a high-level structure that facilitates storage and manipulation tasks. A video is composed of a series of video scenes. Every video scene consists of a varying number of frames, which visually relate to one another by sharing objects, background, luminance, etc. Differences among frames belonging to same video scene are caused by video content movement, camera motion and zooms, but are often insignificant. The number of frames comprising a video scene differs greatly from scene to scene, but is often large.

A video index that is based on video scenes should be very compact, and at the same time keep the narrative

* Corresponding author. Tel.: +1-604-291-3761; fax: +1-604-291-3045.

E-mail address: li@cs.sfu.ca (Z.-N. Li).

0031-3203/02/$22.00
PII: S0031-3203(01)00134-0

evolution of a video [1]. With the help of such an index, video manipulation tasks, such as fast browse and content-based search, can be performed on a very compact image space. Thus temporal video segmentation, which divides a video into a series of video scenes, is a critical preprocessing step for effective video manipulation [2].

Temporal video segmentation is often accomplished by detecting video transitions, which are procedures changing one scene to another. One scene can change into another scene by following various patterns, and transitions produce various visual effects in videos. Frequently used video transitions include cuts, fade in/fade out, cross dissolve, dither dissolve, etc.

Three tasks must be accomplished by a temporal video segmentation algorithm: reporting the appearances of video transitions; locating the transition's starting and ending position; and classifying the transitions according to predefined transition categories. These are non-trivial tasks, not only because of the ambiguous distinction between transitions and other video effects but also because of the variety of characteristics that appear in various types of transitions.

A large number of color histogram-based algorithms [3–8] have been proposed to segment color video streams. In these algorithms, histograms of consecutive video frames are generated for distance comparison. A pair of consecutive video frames with large frame distance is classified as a video transition.

Swain and Ballard [9] estimated frame distance using histogram intersection, defined as

$$Intersection(\boldsymbol{h}_1, \boldsymbol{h}_2) = \frac{\sum_{i=1}^{N} \min(h_1[i], h_2[i])}{N}, \tag{1}$$

where $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the histograms derived from two image frames.

Chi-square test histogram distance, which is defined as

$$\chi^2 = \begin{cases} \dfrac{1}{N^2} \displaystyle\sum_{i=1}^{N} \dfrac{(h_1[i] - h_2[i])^2}{h_2[i]} & \text{if } h_2[i] \neq 0, \\ \dfrac{1}{N^2} \displaystyle\sum_{i=1}^{N} \dfrac{(h_1[i] - h_2[i])^2}{h_1[i]} & \text{if } h_2[i] = 0, \end{cases} \tag{2}$$

is used by Nagasaka and Tanaka [6] to normalize histogram bin distances. Frame distances are sharpened by the chi-test.

Proposed by Niblack et al. [7], histogram similarity is interesting for its ability to reflect similarity between colors according to human perception. The similarity between histogram $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ is defined by

$$Similarity(\boldsymbol{h}_1, \boldsymbol{h}_2) = (\boldsymbol{h}_1 - \boldsymbol{h}_2)^{\mathrm{t}} \boldsymbol{A} (\boldsymbol{h}_1 - \boldsymbol{h}_2),$$

where $\boldsymbol{A} = [a_{ij}]$ is a similarity matrix and weights a_{ij} reflect the similarity between colors for bin i and j of a histogram. The weights can be determined by human perceptual similarity or by L_1 distance, $a_{ij} = 1 - d_{ij}/d_{\max}$, where d_{ij} denotes the L_1 distance between bin i and j of the histogram, and $d_{\max}$ denotes the distance between black and white color.

Drew et al. [4] develop a normalized chromaticity based algorithm to address the issue of global illumination variations inside one clip. Two-dimensional histograms for chromaticity images are used, where chromaticity is defined by

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}.$$

The idea behind this is that changes in the brightness of the frame may not be correlated with the content change. As well, chromaticity removes shading, in a Lambertian model.

Bouthemy et al. [10] analyze the temporal evolution of the size of the support associated with the estimated dominant motion. Transitions are detected by a downward jump of corresponding supports, followed by an upward jump of corresponding supports. If the two jumps are detected to be successive, a cut is reported. If they are separated by one or several frames, a gradual transition is reported. The dominant image motion is represented by a two-dimensional affine model.

An integrated method was presented by Ford et al. [11,12] for detecting and classifying transitions in uncompressed video sequences. Multiple estimation strategies for frame distances are used: color histogram distance, statistics-based distance, and pixel difference distance. Frame distance estimates are integrated by a fuzzy logic system to generate the final results. This method works for cuts, fade-in/fade-out, wipes and dissolves.

Zabih et al. [13] described an intensity edge feature-based algorithm where scene transitions are detected based on the emergence or disappearance of edges. When one scene changes into another, the intensity edges of the first scene gradually disappear and the edges of second scene emerge. A global motion computation is used to compensate camera or object motion, and the ways in which edges emerge and disappear relate to the type of transition—good detection and classfication results are obtained for a variety of scene transitions. The disadvantage of this method is that it only works for few types of gradual transitions.

Ngo et al. [14] presented a spatial–temporal method for detecting gradual scene changes. Spatially, a strip (or just a row or column) from each video frame is extracted. Over time, the strips form a spatial–temporal image, i.e., a *video slice*. Good results were obtained by using a Markov energy model to locate the trajectories of wipes in the spatial–temporal image. For dissolves, the variance in the slice will change and form a concave upward parabola.

Most of the above algorithms work well for cuts [1], but their reliability for gradual transition detection is

very low. Distance estimates for consecutive frames derive from statistics for various visual cues or the statistics of distances between corresponding macro-blocks. However, transition pattern information is lost when computing statistics for low-level or intermediate-level visual cues and as well these methods have difficulty distinguishing transitions from other video effects, such as camera movements. Further classification results are often not available, and hard-coded thresholds are often required by these algorithms.

Li and Wei [15] introduced a spatial–temporal joint probability analysis for temporal video segmentation: the time behavior of joint probability images derived from gradual transitions is analyzed. Here we provide further development and implementation of the theory and analysis in the paradigm.

2. Video transitions

Temporal video segmentation is usually accomplished by detecting, locating and classifying video transitions, i.e., boundaries between consecutive video shots. A video shot is a sequence of frames that were continuously captured by a camera without interruption. During a video shot, video contents can be affected by object movements and camera pans, tilts or zooms—these are not viewed as video transitions: a video transition is an artificial effect that connects two consecutive video shots. Some other artificial effects, such as captions, also have impact on the video contents. Transitions differ from these in the following way: a transition is a boundary between two visually independent video shots with a narrative change, while an artificial effect other than a transition occurs in a single video shot by means of some added visual effect, without a narrative change.

Various kinds of transitions exist, and each transition type is distinctively associated with a certain transition pattern. A transition can be characterized as one of two kinds of transitions: instantaneous cuts and gradual transitions. An instantaneous cut changes one video shot into another one abruptly without any intermediate transition procedure.

2.1. Cross transition and dither transition

One video shot can gradually change into another in one of the following two ways:

- *Cross transition*: Every pixel value gradually changes from one video shot to another. Fig. 1 illustrates the visual effect of a cross transition.
- *Dither transition*: A small portion of pixels abruptly change from pixels values from the first shot to those of the second shot every moment. With time, more and more pixels change until all of the pixels change into the second video shot. Fig. 2 illustrates four frames in a dither transition.

The above classification is based on the way in which pixels change from one shot to another. Gradual transitions can also be classified according to their visual effects. Table 1 lists some frequently used transitions. Mathematical definitions for these two types of transition are given below.

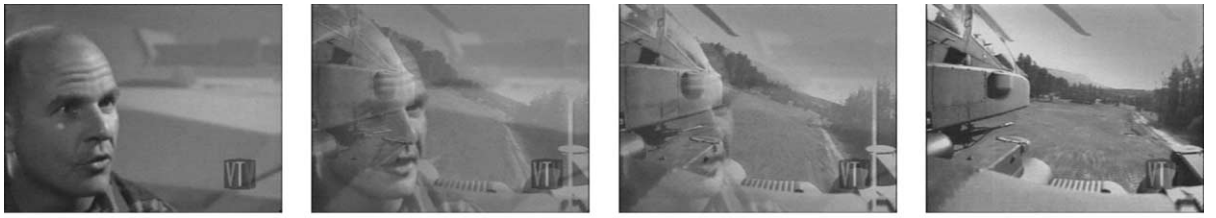


Fig. 1. A cross transition: the duration is from Frame 0 to Frame T.

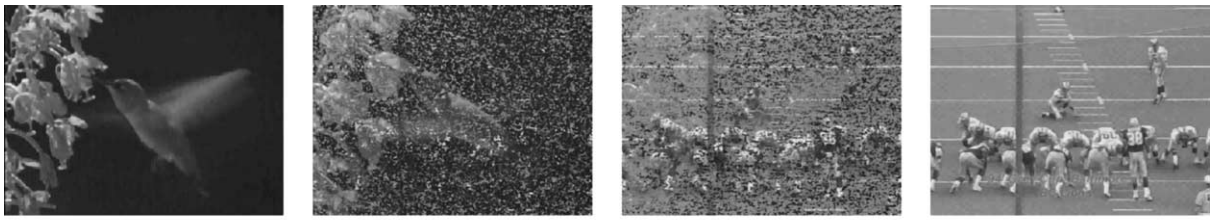


Fig. 2. A dither transition: the duration is from Frame 0 to Frame T.

Table 1
Transitions in digital video

Abrupt transitions	Gradual transitions	
	Cross transitions	Dither transitions
Cut	Cross dissolve	Dither dissolve
	Fade-in/fade-out	Various types of wipes
	Additive dissolve	Non-additive dissolve

Definition 1 (*Cross transition*). Given two video shots $\boldsymbol{A}$ and $\boldsymbol{B}$, a cross transition obeys

$$D_t(x,y)=\alpha(t)A_t(x,y)+\beta(t)B_t(x,y),\quad t\in[0,T],\qquad(3)$$

where $D_t(x,y)$, $A_t(x,y)$ and $B_t(x,y)$ are the pixel values of frame t at position (x,y) for transition shot $\boldsymbol{D}$, shot $\boldsymbol{A}$ and shot $\boldsymbol{B}$, respectively. $\alpha(t)$ and $\beta(t)$ are transition functions and are often defined as linear functions

$$\beta(t)=\frac{t}{T}\quad\text{and}\quad\alpha(t)=1-\frac{t}{T}.$$

The so-called "fade-in" and "fade-out" are transitions generated from one video shot. They can be viewed as special cases of cross transitions with $A_t(x,y)=0$ and $B_t(x,y)=0$, respectively.

It is worth mentioning that the transition functions can be non-linear functions. Furthermore, the sum of $\alpha(t)$ and $\beta(t)$ may not equal 1 for a few types of transitions. For example, the transition functions for an additive dissolve, illustrated in Fig. 3, are defined by

$$\alpha(t)=\begin{cases}1 & t\in[0,\frac{T}{2}),\\ 2(1-\frac{t}{T}) & t\in[\frac{T}{2},T],\end{cases}\qquad(4)$$

$$\beta(t)=\begin{cases}\frac{2t}{T} & t\in[0,\frac{T}{2}],\\ 1 & t\in(\frac{T}{2},T].\end{cases}\qquad(5)$$

Because the sum of $\alpha(t)$ and $\beta(t)$ is not bounded by 1, the transition video may saturate, as illustrated in Fig. 4.

Definition 2 (*Dither transition*). Given two video shots $\boldsymbol{A}$ and $\boldsymbol{B}$, a dither transition obeys

$$D_t(x,y)=(1-\alpha(x,y,t))A_t(x,y)+\alpha(x,y,t)B_t(x,y),\quad t\in[0,T],\qquad(6)$$

where $D_t(x,y)$, $A_t(x,y)$ and $B_t(x,y)$ are the pixel values of frame t at position (x,y) for transition shot $\boldsymbol{D}$, shot $\boldsymbol{A}$ and shot $\boldsymbol{B}$, respectively. The transition function $\alpha(x,y,t)$

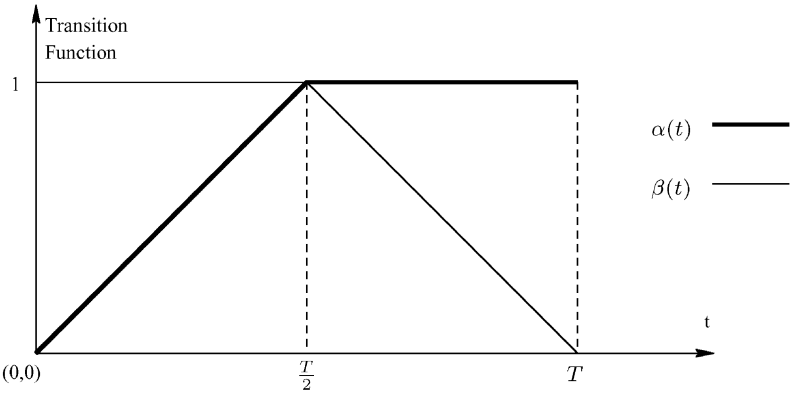


Fig. 3. Transition functions for additive dissolve.

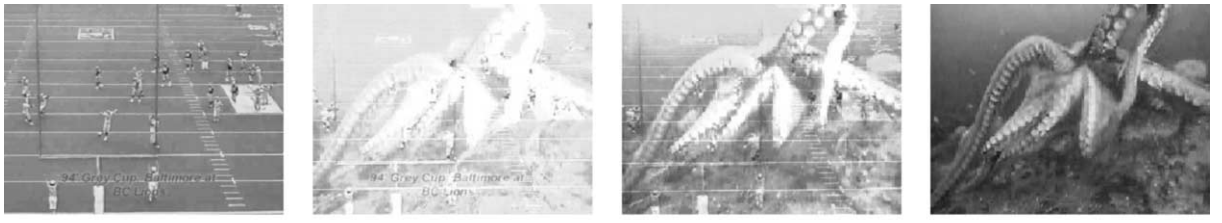


Fig. 4. An additive dissolve.

Fig. 5. A non-additive dissolve.

is defined by

$$\alpha(x,y,t)=\begin{cases}1 & (x,y)\in S_t,\\ 0 & (x,y)\notin S_t,\end{cases} \tag{7}$$

where $\boldsymbol{S}_t$ is a subset of $\boldsymbol{S}$, the pixel coordinate space. $\boldsymbol{S}$ is related to $\boldsymbol{S}_t$ by

$$\boldsymbol{S}=\bigcup_{t=0}^{T}\boldsymbol{S}_t$$

and $\forall t_1,t_2\in[1,T]$, if $t_1\neq t_2$, then

$$(\boldsymbol{S}_{t_1}-\boldsymbol{S}_{t_1-1})\cap(\boldsymbol{S}_{t_2}-\boldsymbol{S}_{t_2-1})=\phi.$$

Like cross transitions, most dither transitions use linear transition functions, i.e., an equal number of pixels change every moment; this implies

$$|\boldsymbol{S}_{t_1}-\boldsymbol{S}_{t_1-1}|=|\boldsymbol{S}_{t_2}-\boldsymbol{S}_{t_2-1}|.$$

Note that the transition functions only determine the number of pixels that are changed each moment. They do not give any spatial information for which pixels change. The positions of such pixels can be random, e.g., in the dither dissolves, shown in Fig. 2. However, if the transition functions follow one of several predefined spatial patterns, then they describe a *wipe*, which is a frequently used dither transition.

A non-additive dissolve is a very special dither transition, illustrated in Fig. 5. The condition for a pixel to change from video shot $\boldsymbol{A}$ to shot $\boldsymbol{B}$ is illustrated in Fig. 6 by the shaded area. A point $(B(x,y),A(x,y))$ in the rectangle (Fig. 6) corresponds to those pixels with luminance $B(x,y)$ and $A(x,y)$ in $\boldsymbol{B}$ and $\boldsymbol{A}$, respectively. A front-line l rotates counterclockwise across the rectangle from $\theta=0$ to $\pi/2$, and leaves the shadow area behind. The value of θ linearly increases with the progress of time. If the intensity of a pixels falls into the shadow area at a certain moment, its value in the transition $\boldsymbol{D}$ equals the value at corresponding position in video shot $\boldsymbol{B}$. Otherwise, its value equals the value of video shot $\boldsymbol{A}$. The transition function $\boldsymbol{S}_t$ is defined by

$$\boldsymbol{S}_t=\left\{(x,y)\,|\,\arctan\left(\frac{A_t(x,y)}{B_t(x,y)}\right)\in\left(0,\frac{t\pi}{2T}\right)\right\}.$$

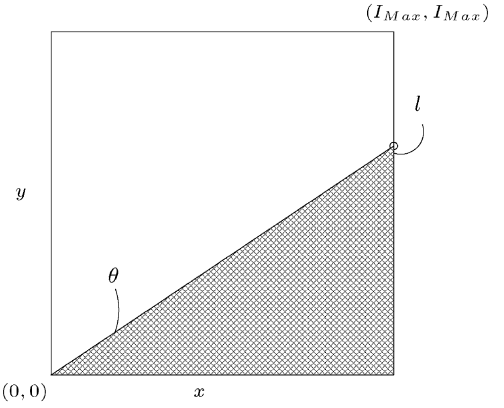


Fig. 6. Conditions for a pixel to change in a non-additive dissolve.

3. ST-JPI for video segmentation

3.1. Joint probability image

As a measure of intensity co-occurrence between two images, joint probabilities of two images have been used together with independent component analysis (ICA) to effectively separate reflections from paintings [16]. In this paper, joint probabilities are viewed as a similarity estimate between two images. For simplicity, luminance images [17] are used in discussions:

$$\boldsymbol{Y}=0.299\boldsymbol{R}+0.587\boldsymbol{G}+0.114\boldsymbol{B}.$$

The transition detection method developed here is based on luminance images. It can readily be extended to RGB color space images.

Joint probability images were proposed by Li and Wei [15] for use in video segmentation:

Definition 3 (*Joint probability image*). Suppose $\boldsymbol{A}$ and $\boldsymbol{B}$ are two images of the same size. Let $A(x,y)$ and $B(x,y)$ be the luminance of image $\boldsymbol{A}$ and $\boldsymbol{B}$ at the position (x,y), respectively. A joint probability image ($\boldsymbol{JPI}$) is a matrix

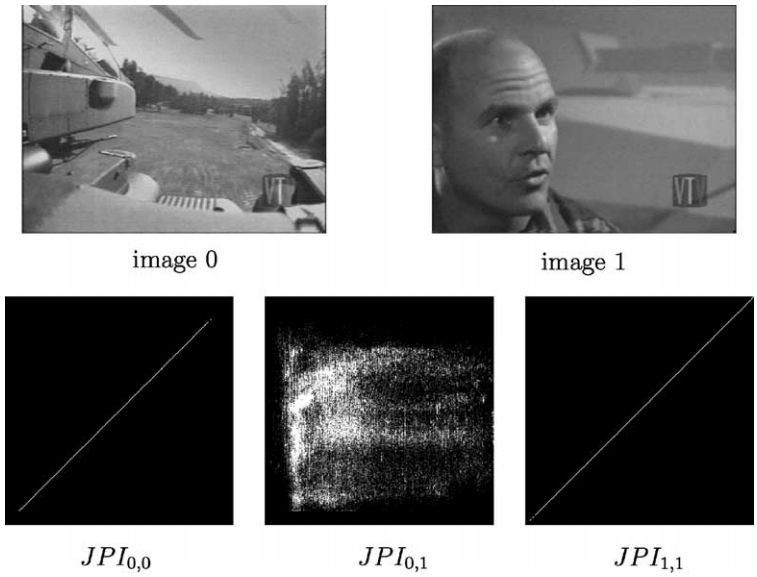


Fig. 7. Illustration of the behavior of JPIs.

with element value:

$$JPI_{A,B}(i_1,i_2) = P(A(x_1,y_1) = i_1, B(x_2,y_2) = i_2, (x_1,y_1) = (x_2,y_2)) \quad (8)$$

where $P(A(x_1,y_1) = i_1, B(x_2,y_2) = i_2, (x_1,y_1) = (x_2,y_2))$ is the probability that luminance i_1 and i_2 appear at the same position in image $\boldsymbol{A}$ and image $\boldsymbol{B}$, respectively.

Each pixel of a joint probability image corresponds to an intensity pair in two images. Its value can be calculated by counting its co-occurrence as a pair at any position of the two images. Spatial information for co-occurrence intensity pairs is not preserved. The sum of all the components in a JPI equals 1. The distribution of the values in a JPI maps the correlation between two images.

Fig. 7 depicts two extreme cases: a JPI for two identical images and a JPI for two independent images. In the first case, the JPI shows a diagonal line, and in the second the JPI consists of a uniform distribution.

Because of the continuity of image contents, video frames within a single video shot highly relate to one another. A JPI derived from two frames belonging to the same video shot usually has a narrow distribution along the diagonal line. However narrative and visual content changes occur between two consecutive video shots, and then a uniform distribution is expected in the JPI. This strong impact of a transition on JPI behavior is the basis of our transition detection method.

3.2. Spatial–temporal joint probability image

A single JPI, which is derived from a pair of video frames, illustrates the correlation of two tiny intervals in a video. A spatial–temporal joint probability image (***ST-JPI***), which consists of a series of JPIs in chronological order, reflects the temporal evolution of video contents. An ***ST-JPI*** for a video $\boldsymbol{D}$ can be expressed as

$$\boldsymbol{ST\text{-}JPI}_{T_0,T_1} = \{\boldsymbol{JPI}_{T_0,t} \mid \boldsymbol{JPI}_{T_0,t} = \boldsymbol{JPI}_{\boldsymbol{D}_{T_0},\boldsymbol{D}_t}, t \in [T_0,T_1]\} \quad (9)$$

The image $\boldsymbol{D}_{T_0}$ that is shared by all the JPIs is the base image of the ***ST-JPI***.

The evolution pattern of the JPIs directly relates to the effects in the video. The frames within a same video shot highly relate to one another because of the continuity of visual contents. Across two video shots, images are visually independent of one another. If a video cut happens at a certain frame between frame 0 and T, where an ST-JPI is derived, the JPIs before the cut have very limited dispersion from the diagonal line. For video frames after the cut, uniform JPIs are usually obtained. The shift from narrow dispersion JPIs to uniform JPIs happens instantaneously at the position where a cut appears. By estimating the uniformity of JPIs, cuts can be detected and reported.

3.3. Spatial–temporal pattern of gradual transitions

The above scenario works well for cuts, but for gradual transitions it is too coarse to produce good results. The dispersion of JPIs gradually extends to a uniform one

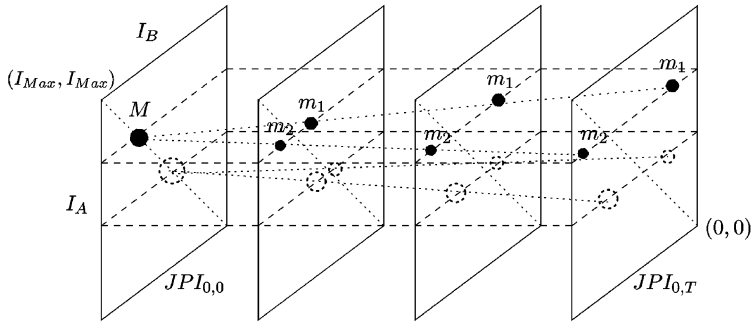


Fig. 8. Spatial–temporal JPIs in a cross transition.

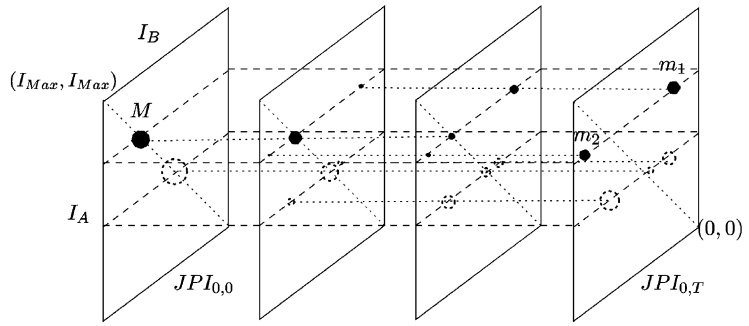


Fig. 9. Spatial–temporal JPIs in a dither transition.

when one video shot changes into another video shot. The process can take a long time determined by the length of the transition.

For convenience of discussion, some initial assumptions are made. The situations when the assumptions are not satisfied will be discussed later. Suppose a gradual transition $\boldsymbol{D}$ is generated from two video shots $\boldsymbol{A}$ and $\boldsymbol{B}$ between frame 0 and frame T, where $\boldsymbol{ST\text{-}JPI}_{0,T}$ is generated, let us assume:

- The transition $\boldsymbol{D}$ has a linear transition function.
- The transition is generated from two static video shots. i.e., no movement exists in either original video shot during the period of the transition.
- For the dither transitions, the choices of pixels changed are entirely independent of the pixel's visual characteristics, such as luminance.

Consider a pixel set, which is formed by the pixels with intensity i_0 in video shot $\boldsymbol{A}$; its corresponding point in $\boldsymbol{JPI}_{0,0}$ falls on the diagonal line, and is denoted by M in Figs. 8 and 9.

In $\boldsymbol{JPI}_{0,T}$, which is the last JPI of the ST-JPI, the point M splits into a number of small points. Two of those points are denoted by m_1 and m_2 in Figs. 8 and 9, which correspond to intensity i_1 and i_2 in video shot $\boldsymbol{B}$. Between the first and last JPIs, the JPIs gradually evolve from the first JPI to the last one with the progress of time. The evolution pattern of an ST-JPI is determined by the type of the transition.

For a cross transition, the intensities of pixels corresponding to intensity i_0 in video shot $\boldsymbol{A}$ and intensity i_1 in the video shot $\boldsymbol{B}$ linearly increase from i_0 to i_1. Their corresponding point m_1 in the ST-JPI forms a straight line as illustrated by Fig. 8. And the mass (pixel count) of m_1 does not change with the progress of time. Because of the rich color existing in a video, a large number of straight lines with different slopes exist in an ST-JPI.

For a dither transition, a portion of pixels abruptly changes from video shot $\boldsymbol{A}$ to $\boldsymbol{B}$ at each moment. The point M, which represents pixels with intensity i_0 in both $\boldsymbol{A}$ and $\boldsymbol{D}$, gradually loses the pixels that have different intensities in $\boldsymbol{A}$ and $\boldsymbol{B}$. With a linear transition function, its mass linearly decreases. Statistically, the intensities into which the pixels change are likely to distribute according to the intensity distribution of the video shot $\boldsymbol{B}$. The point m_1, which corresponds to intensity i_0 and i_1 in $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively, gradually accumulates the pixels. With a linear transition function, its mass linearly increases as illustrated by Fig. 9. With time, the point m_1 and M form straight lines without slope. Because of the rich color in a video, a large number of straight lines exist in an ST-JPI.

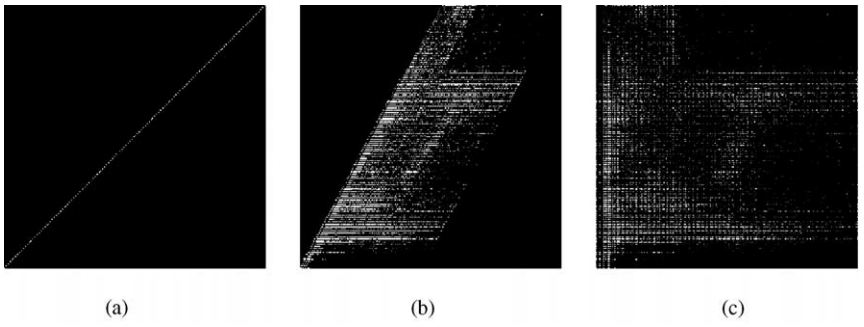


Fig. 10. Sample JPIs of an ST-JPI for a cross transition. (a) $JPI_{30,30}$. (b) $JPI_{30,60}$. (c) $JPI_{30,90}$.

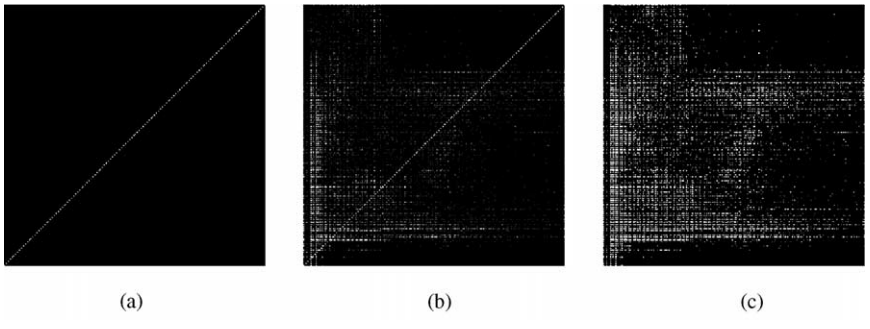


Fig. 11. Sample JPIs of an ST-JPI for a dither transition. (a) $JPI_{30,30}$. (b) $JPI_{30,60}$. (c) $JPI_{30,90}$.

Figs. 10 and 11 illustrate several sample JPIs from ST-JPIs for a cross transition and a dither transition, respectively. The evolution pattern for a cross transition is reflected by expansion of the breadth of the JPI distribution. However, the evolution pattern for a dither transition is reflected by the linear increase of the masses of the pixels that do not fall on the diagonal line and the linear decrease of masses of the pixels that fall on the diagonal line.

Explicitly, the spatial–temporal pattern of an ST-JPI for a cross transition is expressed by

Theorem 3.1 (Spatial–temporal pattern of cross transition). *Given an* $\boldsymbol{ST\text{-}JPI}_{0,T}$ *derived from a video* $\boldsymbol{D}$ *with a cross transition between frame* 0 *and* T, $\forall t \in [0,T]$ *and* $\forall i_0 \in [0, I_{\max}]$,

$$\text{if } i \in [(1-t/T)i_0, (1-t/T)i_0 + (t/T)I_{\max}],$$

$$JPI_{0,t}(i_0, i) = JPI_{0,T}\left(i_0, i_0 + \frac{T}{t}(i - i_0)\right) \tag{10}$$

$$\text{if } i \notin [(1-t/T)i_0, (1-t/T)i_0 + (t/T)I_{\max}],$$

$$JPI_{0,t}(i_0, i) = 0. \tag{11}$$

The cutting lines between Eqs. (10) and (11), which are $i = (1-t/T)i_0$ and $i = (1-t/T)i_0 + (t/T)I_{\max}$, are the left and right boundaries of the parallelograms illustrated in Fig. 10. Their width ($(t/T)I_{\max}$) increases linearly with the progress of time t.

Explicitly, the spatial–temporal pattern of an ST-JPI for a dither transition is expressed by

Theorem 3.2 (Spatial–temporal pattern of cross transition). *Given an* $\boldsymbol{ST\text{-}JPI}_{0,T}$ *derived from a video* $\boldsymbol{D}$ *with a dither transition between frame* 0 *and* T, $\forall t \in [0,T]$ *and* $\forall i_0, i \in [0, I_{\max}]$, *we have*

$$JPI_{0,t}(i_0, i) = \begin{cases} \frac{t}{T} JPI_{0,T}(i_0, i) & i \neq i_0, \\ JPI_{0,0}(i_0, i_0) + \frac{t}{T}(JPI_{0,T}(i_0, i_0) - JPI_{0,0}(i_0, i_0)) & i = i_0. \end{cases} \tag{12}$$

This theorem predicts the behavior of the ST-JPI generated from a dither transition. The masses of the points on the diagonal line linearly decrease and the masses of the points that are not on the diagonal line linearly increase.

Examination of the above models might suggest a dissolve–detection algorithm based on line-detection in the ST-JPI. Considering the required detection of possibly a large number of lines, this approach is less than desirable. Further analysis in the rest of this section discloses the evolution pattern of an ST-JPI for a gradual

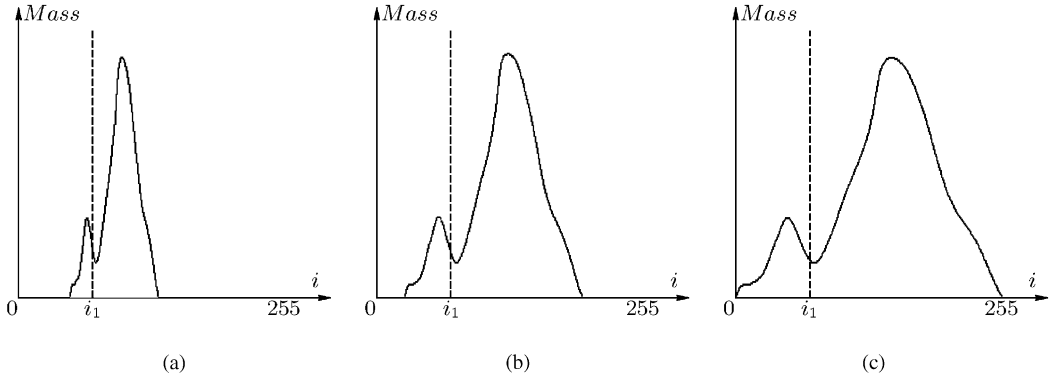


Fig. 12. Pattern evolution of cross transition. (a) Strip i_1 of $JPI_{30,50}$. (b)Strip i_1 of $JPI_{30,70}$. (c) Strip i_1 of $JPI_{30,90}$.

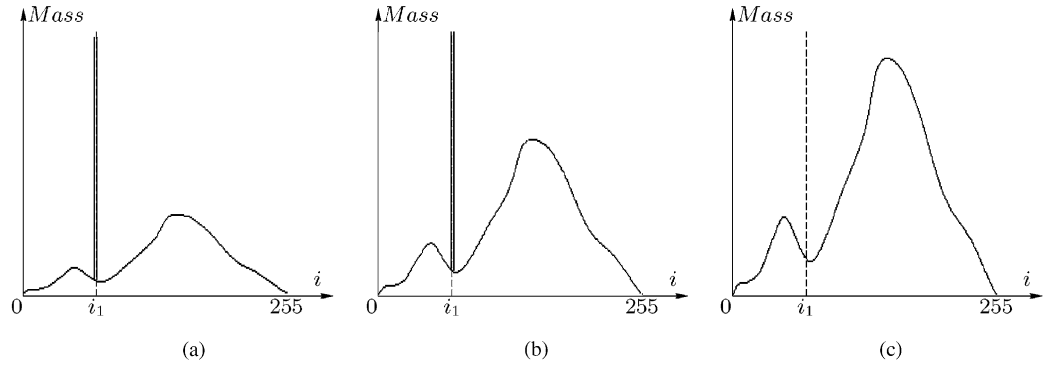


Fig. 13. Pattern evolution of dither transition. (a) Strip i_1 of $JPI_{30,50}$. (b)Strip i_1 of $JPI_{30,70}$. (c) Strip i_1 of $JPI_{30,90}$.

transition and leads to an approach based on a pattern match. Figs. 12 and 13 illustrate the pattern evolution of cross and dither transitions.

The theorems show that an ST-JPI that is derived from a cross or dither transition is entirely determined by $JPI_{0,T}$, the last JPI of the ST-JPI. An ST-JPI can be generated for a potential gradual transition. A cross transition model ST-JPI can be predicted by the last JPI, with Eqs. (10) and (11). A dither transition model ST-JPI can be predicted by the last JPI, using Eq. (12), and the predicted model is compared with the ST-JPI. If the ST-JPI matches the cross transition model, a cross transition is reported. If the ST-JPI matches the dither transition model, a dither transition is reported. If the ST-JPI matches neither the dither nor the cross transition model, the pattern match fails and no transition is reported.

3.4. Joint probability uniform transform

As we have seen, every JPI of an ST-JPI generated from a gradual transition shares the same pattern, but their widths or altitudes change with the progress of time. Before pattern matching is performed, they should be unified into patterns with an identical width and altitude. For the cross transition, which is illustrated in Fig. 12, the widths of (a) and (b) should be expanded to the width of (c). For the dither transition, which is illustrated by Fig. 13, the altitudes of (a) and (b) should be expanded to the altitude of (c).

Definition 4 (*Cross uniform transform*). Given an $\boldsymbol{ST\text{-}JPI}_{0,T}$ derived from a video $\boldsymbol{D}$, a cross uniform transform U_C is defined as

$$JPI_{0,t}^{U_C}(i_0, j) = JPI_{0,t}\left(i_0, i_0 + (j - i_0)\frac{T}{t}\right). \tag{13}$$

Definition 5 (*Dither uniform transform*). Given an $\boldsymbol{ST\text{-}JPI}_{0,T}$ derived from a video $\boldsymbol{D}$, a dither transform U_D is defined as

$$JPI_{0,t}^{U_D}(i_0, j) = \begin{cases} \dfrac{T}{t} JPI_{0,t}(i_0, j) & \text{if } j \neq i_0, \\ 0 & \text{if } j = i_0. \end{cases} \tag{14}$$

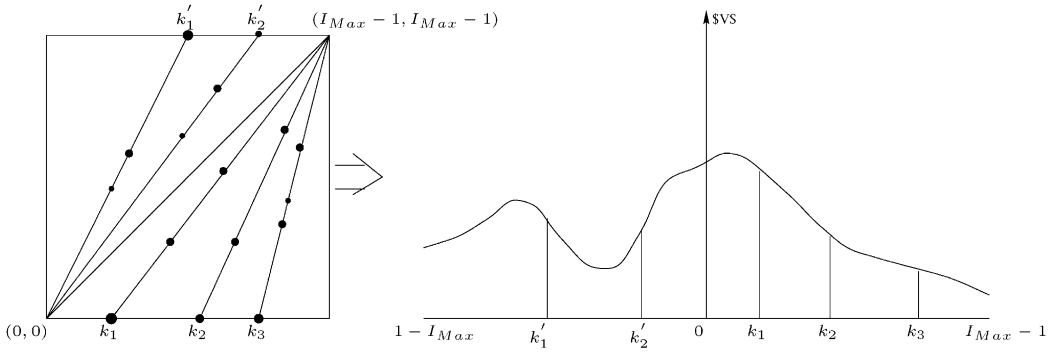


Fig. 14. Joint probability projection vector of a joint probability image.

Obviously an ST-JPI from a cross transition after a cross uniform transform becomes a sequence of identical JPIs. And an ST-JPI from a dither transition after a dither uniform transition also becomes a sequence of identical JPIs. The dither uniform transform throws away the pixels on the diagonal line. This operation does not lose any useful pattern information, because the pixels on the diagonal lines always behave in an exactly opposite way from each other.

3.5. Joint probability projection vector

Illustrated in Figs. 12 and 13 by i_1, the center of the pattern is determined by the corresponding luminance in the base image of the strip. For the convenience of the pattern matching, the center of the pattern should be normalized to a position that is independent of the luminance.

The pattern matching operation can be performed on every strip of an ST-JPI. The number of the strips in an ST-JPI equals the number of the luminance levels of the video. The efficiency of this method can be significantly increased by cutting the ST-JPI into a few slices and performing pattern matching on each slice. Each slice of an ST-JPI consists of a number of strips with different corresponding intensities in the base image.

The pattern of a slice can be obtained by computing a joint probability projection vector (***JPPV***), which is defined by

Definition 6 (*Joint probability projection vector*). Given a joint probability image $\boldsymbol{JPI}_{0,t}$, and luminance $I_1, I_2 \in [0, I_{\max}]$, its joint probability projection vector is

$$V_{I_0,I_1,t}(k) = \begin{cases} \sum_{i=I_0}^{I_1} JPI_{0,t}\ (i, (1-\frac{k}{I_{\max}}i)) & k \in [0, I_{\max}-1], \\ \sum_{i=I_0}^{I_1} JPI_{0,t}\ (i, \frac{I_{\max}-i}{I_{\max}}k+i) & k \in [1-I_{\max}, 0). \end{cases} \quad (15)$$

Fig. 14 illustrates the procedure of deriving a joint probability projection vector from a JPI: projecting all JPI pixels to the upper or lower boundary of the JPI. If a pixel is on the upper-left side of JPI, i.e., $i > j$, the original point of the projection is $(0,0)$, and the pixel is projected to the upper boundary of the JPI. Otherwise, the projection will be made from the point $(I_{\max}, I_{\max})$ to the lower boundary of JPI.

It is obvious that the temporal evolution pattern is preserved after this projection. The center of the pattern, which represents identical luminance, is now at $V(0)$, regardless of what the corresponding intensities are.

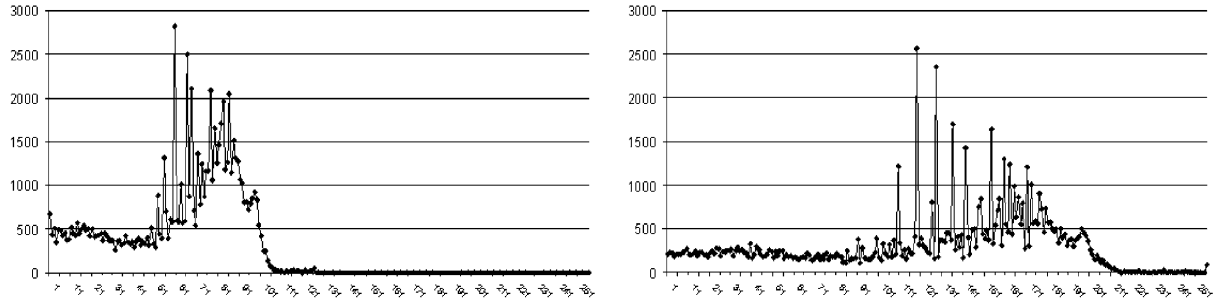
Theorem 3.3. *Given a cross transition* ***D****, an ST-JPI*$_{0,T}$ *generated from* ***D*** *and any intensities* $I_0, I_1 \in [0, I_{\max}]$ *its joint probability projection vectors satisfy*

$$V_{I_0,I_1,t}(k) = V_{I_0,I_1,T}\left(k\frac{T}{t}\right).$$

Theorem 3.4. *Given a dither transition D, an ST-JPI*$_{0,T}$ *generated from* ***D*** *and any intensities* $I_0, I_1 \in [0, I_{\max}]$, *its joint probability vectors* $V_{I_0,I_1,t}$ *satisfy*

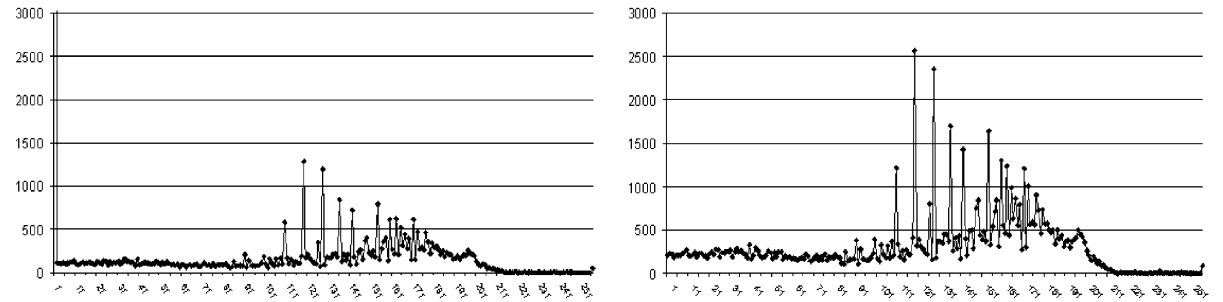
$$V_{I_0,I_1,t}(k) = \begin{cases} \frac{t}{T}V_{I_0,I_1,T}(k) & \text{if } k \neq 0, \\ V_{I_0,I_1,0}(0) + \frac{t}{T}(V_{T_0,I_1,T}(0) - V_{I_0,I_1,0}(0)) & \text{if } k = 0. \end{cases} \quad (16)$$

Observing that overall luminance differences often exist between the two original video shots that generate the transition, either the upper-left or lower-right part of a JPI often is sparse. Figs. 15 and 16 illustrate a few JPPVs derived from cross and dither transitions. For cross transitions, the width of the distribution pattern is different. For dither transitions, the amplitude of the distribution pattern is different. The patterns are slightly influenced by motions in the video.



(a) (b)

Fig. 15. Joint probability projection vectors for a cross transition. (a) The JPPV of $JPI_{0,15}$. (b) The JPPV of $JPI_{0,30}$.



(a) (b)

Fig. 16. Joint probability projection vectors for a dither transition. (a) The JPPV of $JPI_{0,15}$. (b) The JPPV of $JPI_{0,30}$.

3.6. Pattern matching

The last step of transition detection is evaluating whether the uniform joint probability projection vectors in an ST-JPI are identical to a transition model prediction. The JPPVs of the last JPI in the ST-JPI are usually used as the model prediction.

Definition 7 (*Similarity between a JPPV and its model prediction*). Given a JPPV, $\boldsymbol{V}_{i,t_1}$, its similarity with transition model prediction $V_{i,T}$ is

$$\boldsymbol{V}_{i,t_1} \ominus \boldsymbol{V}_{i,T} = (\boldsymbol{V}_{i,t_1} - \boldsymbol{V}_{i,T})^{\mathrm{t}} \boldsymbol{A} (\boldsymbol{V}_{i,t_1} - \boldsymbol{V}_{i,T}), \tag{17}$$

where $A = [a_{ij}]$ is a distance matrix [18]. The weight a_{ij} denotes the distance between component i and j of the projection vector. A Gaussian function, $a_{ij} = 1 - \mathrm{e}^{-(i-j)^2/2\sigma^2}$, can be used to determine this similarity matrix. In our implementation, $\sigma = 10$. Making use of the similarity matrix determined by the Gaussian function increases the robustness of the measure to error caused by quantization and motions in the video, and still keeps enough resolution to distinguish transitions from other video effects.

The value of this similarity is always between 0 and 1. Value 0 represents a perfect match, and indicates that all the projection vectors compared are identical. If the distance is smaller than a preset threshold, a transition is successfully detected. Its type is determined by the uniform transform used.

We use this distance measure because it tolerates noise. There also exist some other measures that can be used to measure distances, such as intersection, which is defined by

$$\boldsymbol{V}_{i,t} \otimes \boldsymbol{V}_{i,T} = \frac{\sum_{i=0}^{I_{\max}} \min(V_{i,t}(j), V_{i,T}(j))}{\sum_{i=0}^{I_{\max}} V_{i,t}(j)}. \tag{18}$$

The intersection is sensitive to video movements and quantization errors. With a deliberate Gaussian smoothing process, its performance may be significantly improved.

The similarity between an ST-JPI and the transition model is defined as the average of the similarities obtained from all the JPPVs:

$$S(\boldsymbol{ST\text{-}JPI}_{0,T}) = \frac{\sum_{t=0}^{T} \sum_{i=0}^{I_{\max}} \boldsymbol{V}_{i,t} \ominus \boldsymbol{V}_{i,T}}{TI_{\max}}. \tag{19}$$

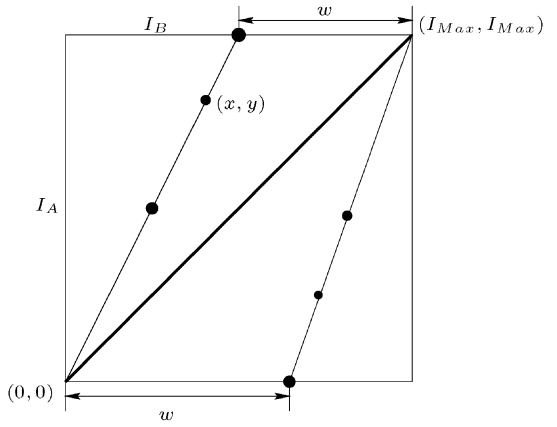


Fig. 17. JPPC of a joint probability image.

4. Fast video segmentation algorithm

The ST-JPI pattern matching only succeeds when the entire ST-JPI is derived from the frames with in the range of a transition. Even a few frames that are not part of transition can cause a mismatch. Usually a video consists of a huge number of frames. For efficiency reasons it is undesirable to generate ST-JPIs and perform pattern matching everywhere, so a preprocessing step that highlights potential transitions is required before activating the ST-JPI pattern match.

The *joint probability projection centroid* (JPPC) is proposed as a distance estimate between two images. Since it is a single number that can be obtained with quick computation, JPPCs are suitable for preprocessing (Fig. 17).

Definition 8 (*Joint probability projection centroid*). label8Given a joint probability image ***JPI***, its joint probability projection centroid is

$$JPPC_{JPI} = \sum_{i=0}^{I_{\max}} \sum_{j=0}^{I_{\max}} (JPI(i,j) \cdot w) \tag{20}$$

where w is the distance between $(I_{\max}, I_{\max})/(0,0)$ and the projection of (x, y), and can be computed by

$$w = \begin{cases} \dfrac{j-i}{i} & j \geqslant i, \\ \dfrac{i-j}{I_{\max}-j} & j < i. \end{cases} \tag{21}$$

The value of a JPPC is between 0 and 1. It is a measure of the width of a JPI's dispersion. Value 0 means the JPI is derived from identical images. A large JPPC suggests the JPI is derived from independent images. When JPPCs are computed for JPIs in an ST-JPI, an abrupt change from a small JPPC to a large one suggests a cut.

Abrupt changes of JPPCs are usually absent for a gradual transition. For a cross transition, the pattern keeps the same altitude; however its width increases linearly. As a result, the JPPCs increase linearly. Fig. 18(a) illustrates a series of JPPCs computed from a cross transition. For a dither transition, the width of the pattern does not change. However the altitude of the pattern increases linearly. As a result, the JPPCs increase linearly. Fig. 18(b) illustrates a series of JPPCs computed from a dither transition. The motions in the videos have a minor influence on the linear pattern.

Theorem 4.1. *Given a video,* ***D****, with a linear cross or dither transition between frame* 0 *and* T. $JPPC_{JPI_{0,t}}$ *and* $JPPC_{JPI_{0,T}}$ *represent JPPCs for* $JPI_{0,t}$ *and* $JPI_{0,T}$, *respectively*. $\forall t \in [0, T]$, *we have*,

$$JPPC_{JPI_{0,t}} = \frac{t}{T} JPPC_{JPI_{0,T}}. \tag{22}$$

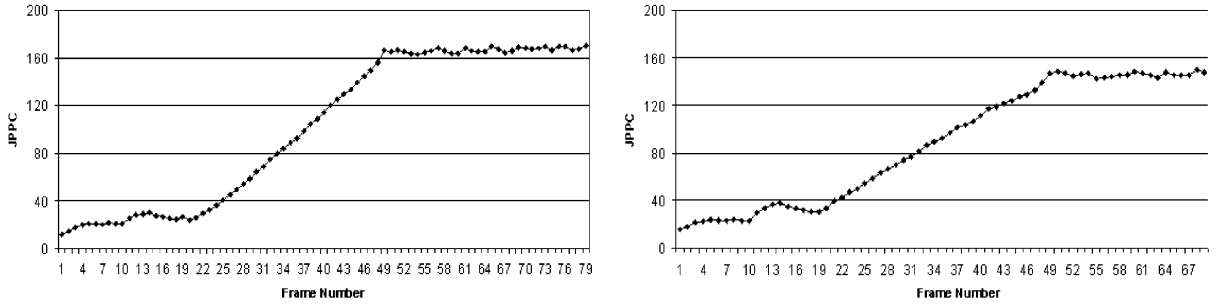
The linear patterns for cross and dither transitions are exactly the same. Although the above does not distinguish cross and dither transition and is only a necessary condition for a transition, and it works well as a filter. With a line-detection algorithm, this pattern can be detected without many difficulties.

Some assumptions about characteristics of video are made to facilitate the discussion of a video segmentation algorithm:

- A gradual transition should last at least 16 frames, which is equivalent to about half a second of video.
- The interval between two transitions should be longer than 32 frames, which is equivalent to about 1 s of video.

Our algorithm can be divided into six major steps:

1. Quickly browsing the video by calculating the JPPCs of pairs of frames at 32 frame distances: D_t and D_{t+32}. If the JPPC is smaller than a preset threshold, then the frames belong to a same video shot. No transitions exist within these 32 frames.
2. If a JPPC is found larger than the threshold, the algorithm activates a binary search. For each step, the binary search attempts to divide the range that is investigated, and continue the binary search in the half with the larger JPPC, if the criteria of a cut are satisfied. The cut criteria are that there is a significant difference between the JPPCs of the two half parts and the larger one is not less than 75% of the JPPC of JPI_{D_t,D_t+32}. If these criteria are always satisfied and binary search succeeds to detect a large JPPC of two consecutive frames, a cut is reported. If the criteria are not satisfied at some stage with a reasonable duration, say eight frames, there is a possibility that a gradual transition happens here.
3. An ST-JPI is generated within this duration. The JPPCs are calculated one by one. A progressive mean



(a) (b)

Fig. 18. JPPCs for a gradual transition (duration: frame 20–50). (a) Cross transition. (b) Dither transition.

square error (MSE) line detection [19] is performed. If JPPCs fit into a straight line (i.e., MSE less than a threshold), a potential transition is reported.

4. The duration usually is only a portion of a transition. The full duration of the transition is discovered by progressively expanding the ST-JPI in both forward and backward directions.
5. Based on the ST-JPI, the JPPVs are generated and normalized by uniform transitions. The similarity is calculated. If it is less than the preset threshold, a transition is detected. Its category corresponds to the uniform transition used.

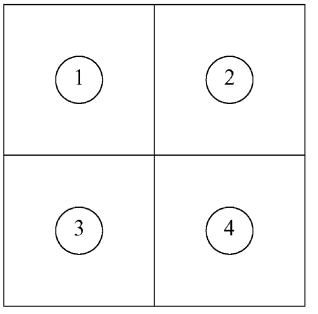


Fig. 19. Screen division for wipe detection.

4.1. Discussion

Transitions can be classified into more specific categories based on transitions' visual effects. A cross transition can be a cross dissolve, fade in or fade out. If the first JPI of an ST-JPI shows a horizontal line, a fade-in is detected. If the last JPI of an ST-JPI shows a vertical line, a fade-out is detected.

A dither transition randomly changes a certain number of pixels at each moment, and the ways to choose the changing pixels vary. As one of the dither transitions, a wipe changes pixels following a pre-specified spatial pattern. The ST-JPI does not retain such spatial information; thus it lacks the ability to distinguish wipe patterns. A solution for this problem is to divide the screen into four parts, as illustrated in Fig. 19. An ST-JPI can be derived from each portion of the screen. Pattern matching can be performed on each portion of the screen. A dither transition will be detected at each portion of the screen. For a dither dissolve, four dither transitions occur at exactly the same time. For a wipe, four dither transitions often fall into a regular chronological order determined by the wipe direction. The relation between the wipe direction and the chronological order is shown in Fig. 20. A spatial–temporal chromatic histogram image based algorithm, which works for general wipes, was proposed in Ref. [20].

Not all transitions have linear transition functions. For a transition with predefined non-linear transition functions, the uniform transform can always be adjusted to unify the frames. Even for a transition without a predefined transition function, or with an unknown transition function, the JPPC ratio between $JPI_{0,T}$ and $JPI_{0,t}$ can be used to unify $JPI_{0,t}$. This strategy eliminates the requirement of previous knowledge of transition functions. Its drawback is that the errors of the JPPCs are introduced into the uniform transform and increase the pattern matching error.

Additive dissolve is another type of challenging transition, because it may saturate. When saturation occurs, the pattern information is lost. Fortunately, if the luminance of two video scenes is not very high, we still have enough information to perform pattern matching. Fig. 21 illustrates two strips of an ST-JPI for an additive transition. The left strip corresponds to the middle luminance in the base image. Enough space exists for the pattern to show up in the ST-JPI. The right strip corresponds to a frame close to the high end of the luminance. The transition quickly saturates in this strip.

It was assumed that the choice of changing pixels at each moment is independent of visual characteristics, e.g., the pixels' intensities. This assumption is not true for a few dither transitions. For example, a non-additive

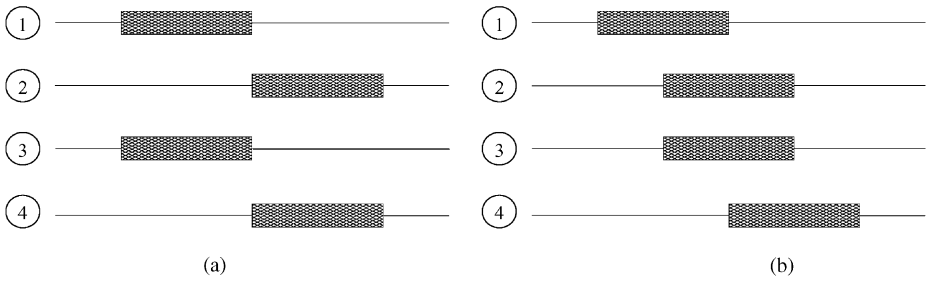


Fig. 20. Temporal patterns for wipes with various directions: the shadow areas represent a dissolve at corresponding block. (a) Left-to-right wipe. (b) Upper-left to bottom-right wipe.

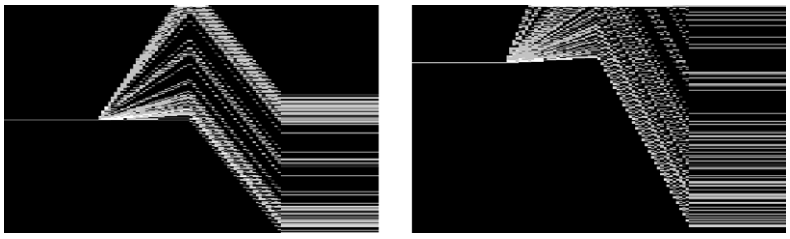


Fig. 21. Illustration of the behavior of JPIs: two slices of the ST-JPI of an additive transition.

Fig. 22. Illustration of the behavior of JPIs: two slices of the ST-JPI of a non-additive transition.

dissolve (shown in Fig. 5) changes pixels with certain intensity co-occurrences at each moment. Its spatial–temporal pattern is illustrated by Fig. 22.

Finally, motion can significantly affect the transition patterns. Besides transitions, object and camera movements also cause video content changes and affect the behavior of an ST-JPI. Fortunately, object and camera movements tend to be moderate and have modest influences on the behaviors of ST-JPIs. The JPIs for dither transitions tend to be more sensitive to motions. Fig. 23(a) illustrates a JPPV that is influenced by movement. Ideally the $JPI_{0,15}$ should keep the pattern information in two parts: the vector components with non-zero index; component $\boldsymbol{V}[\mathbf{0}]$ holds pixels that are not changed. However, various numbers of pixels that are not changed do not fall into $\boldsymbol{V}[\mathbf{0}]$, as expected, because of motions. They are very likely to fall into components close to zero (Fig. 23). Based on this scenario, the components close to zero are often considered unreliable and are discarded.

5. Experimental results

5.1. Experimental results

Two parameters, recall and precision, are often used to evaluate the efficacy of a video segmentation algorithm. They are defined by

$$Recall = \frac{D}{D+U}, \quad Precision = \frac{D}{D+F},$$

where D denotes the number of transitions that are detected, U denotes the number of transitions that are not detected, and F denotes the number of false alarms made by the algorithm. Recall and precision reflect the completeness and correctness of a transition report. A good

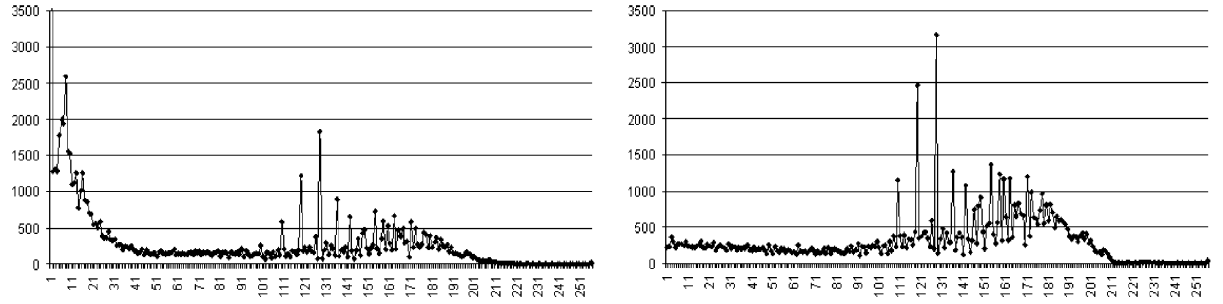


Fig. 23. JPPVs of a dither transition. (a) The JPPV for $JPI_{0,15}$. (b) The JPPV for $JPI_{0,30}$.

Table 2
Transition detection experiment results

	D	*U*	*F*	Recall (%)	Precision (%)
Cut	43	1	2	97	96
Cross transition	14	3	1	82	93
Dither transition	9	3	2	75	81

transition–detection algorithm should balance precision and recall and obtain high values for both of them.

Experiments have been performed on several digital videos to evaluate the efficacy of the algorithm. One digital video, which is recorded from television, is used as the training set to optimize the thresholds. The test videos consist of a variety of contents, such as sports and battlefields. Fast object and camera movements frequently appear in the videos. Furthermore, artificial effects, such as captions, often show up in the videos. The algorithm accurately detects, locates and classifies almost all cuts and most of the gradual transitions. Table 2 illustrates statistics of the experimental results. It shows that our algorithm detected almost all the cuts and most of the gradual transitions. It performs well for both recall and precision. Fig. 24 illustrates a segmentation result for a specific digital video. The text shows the detected transitions, reporting the transition types and durations. Key frames between each transition is shown as a thumbnail.

For these results, thresholds are determined by the training set, for the different types of transitions and different extent of motions. Then these same thresholds are used for all experiments on test videos.

Timing results for the non-optimized code we used were dominated by disk I/O for reading and writing video. For the examples shown, it took about 2–2.5 min to process a video clip. However, in theory the algorithm is very fast because it only needs to buffer 8 frames. The algorithm can work on any length video for the same reason.

Tests were performed on several artificial video transitions with different degrees of motion. For each test, a transition that lasts for 30 frames were generated from two video shots. To estimate the motions in the two original video shots, JPPCs of the consecutive frames of the two video shots were derived (Fig. 25(a) and (b)). The averages of those JPPCs were used as the quantitative estimation of motion. The JPPCs between the first frame in the transition video and every following frame (Fig. 25(c)) were generated. The potential transition was detected by line-detection. An ST-JPI was generated by sampling one frame in every three frames during the potential transition. The ST-JPI was divided into eight slices and a JPPVs was derived for each slice. The uniform transforms were used to unify all the JPPVs. Each slice was compared with the corresponding slice of the last JPI and a similarity was derived. For each sample frame, its similarity with the last frame in ST-JPI was computed by the weighted average of similarities derived with Eq. (17), where the weights are the percentages of the pixels that fall in the corresponding slices. The average was used as the similarity of this pattern matching.

5.2. Performance under different motion conditions

Figs. 25 and 26 illustrate the tests performed on the artificial video Bird and the video Tennis, respectively. For the video Bird, in which modest motions appear, the pattern matching result is very good. Because of the strong motions in the video Tennis, the pattern matching result deteriorates.

Table 3 illustrates several tests performed on video shots with different degrees of motions: Static, Bird, Sport and Tennis. The results show that the pattern matching results gradually deteriorate with the increase of the motions in videos. The differences in the timing results are so small that they do not suggest any difference in cost for different videos and different kinds of pattern match.

Fig. 24. An example of video segmentation.

The pattern match time is only a small portion of total processing time. The major computer costs is incurred by generating JPIs. Further improvements in efficiency should be obtained through efficient methods for generating JPIs.

5.3. Comparative results: histogram intersection vs. similarity

Both histogram intersection and similarity can be used as a measure of the similarity between an ST-JPI and

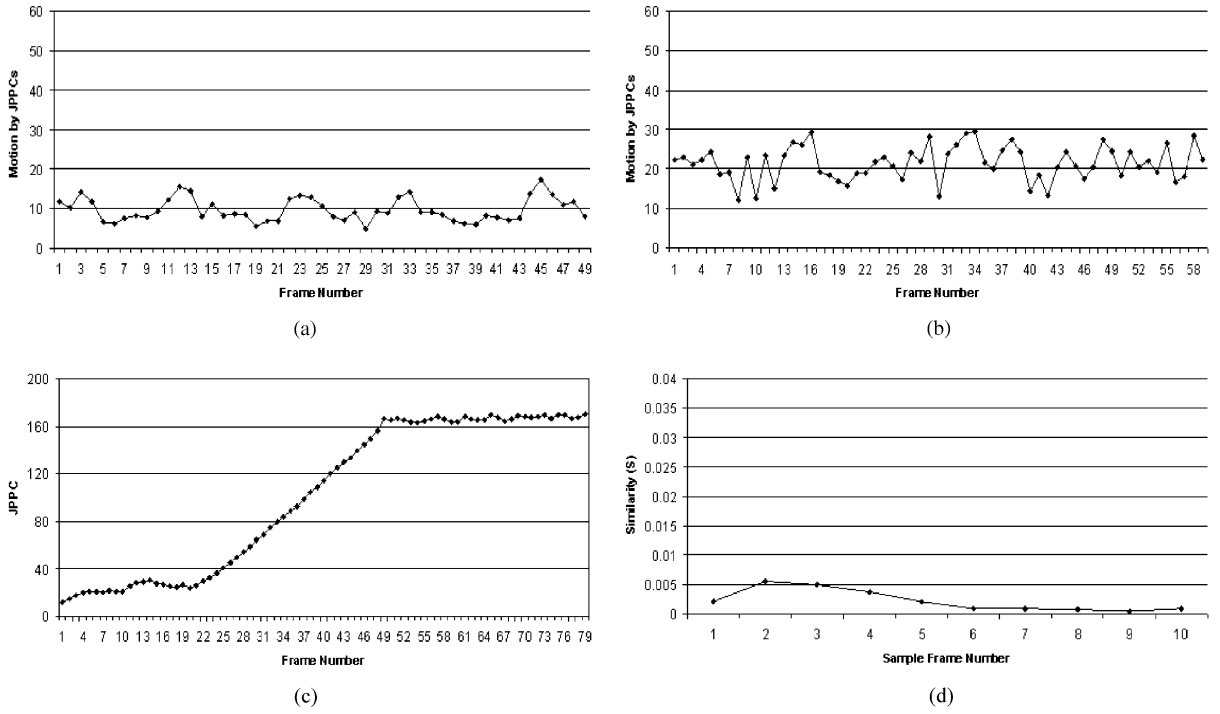


Fig. 25. Relation between algorithm performance and motion (Bird, cross transition). (a) JPPCs of consecutive frames in video I. (b) JPPCs of consecutive frames in video II. (c) JPPCs compared with the first frame. (d) Similarities (sample frames: the last frame).

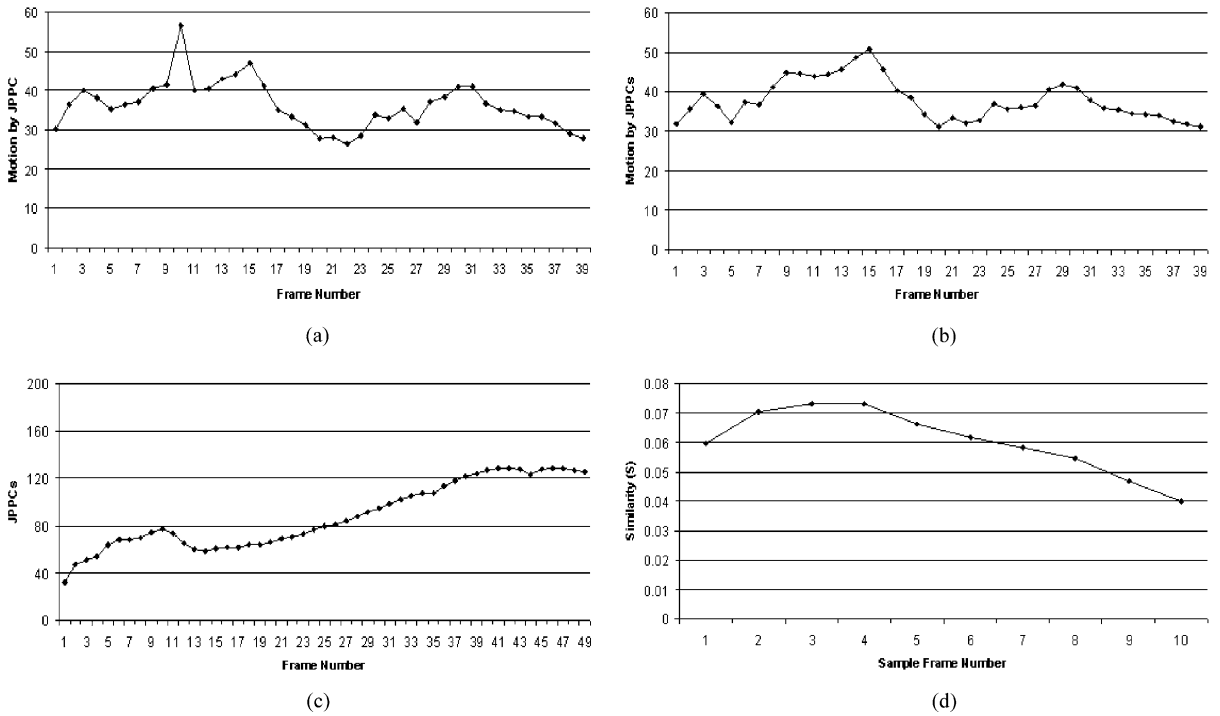


Fig. 26. Relation between algorithm performance and motion (Tennis, cross transition). (a) JPPCs of consecutive frames in video I. (b) JPPCs of consecutive frames in video II. (c) JPPCs compared with the first frame. (d) Similarities (sample frames: the last frame).

Table 3
Relations between algorithm performance and motion in video. All results reported are for a Pentium-III machine with speed 500 MHz

		Motion by *JPPC*	Similarity (S) ($\sigma = 10$)	Total processing time (ms)	Pattern matching time (ms)
Cross transition	Static	0	0.0002	440	120
	Bird	13.70	0.0022	391	110
	Sports	26.05	0.0165	430	100
	Tennis	35.12	0.0604	360	90
Dither transition	Static	0	0.0004	350	120
	Bird	13.70	0.0059	481	101
	Sports	26.05	0.0273	480	150
	Tennis	35.12	0.0719	420	150

Table 4
Comparative results for gradual transitions: similarity vs. intersection

		Motion by *JPPC*	Similarity (S) ($\sigma = 10$)	Intersection (I)
Cross transition	Static	0	0.0002	0.7558
	Bird	13.70	0.0022	0.6686
	Sports	26.05	0.0165	0.7550
	Tennis	35.12	0.0604	0.6788
Dither transition	Static	0	0.0003	0.7348
	Bird	13.70	0.0059	0.5276
	Sports	26.05	0.0237	0.6496
	Tennis	35.12	0.0719	0.6376
Total processing time (ms)			441	341
Pattern matching time (ms)			81	1

the one under the transition model. Table 4 illustrates a performance comparison of them. For similarity, 0 is a perfect match; for intersection, 1 is a perfect match. The performance of intersection appears to be poor. Even for perfect transitions derived from static video, its performance is still not satisfactory.

5.4. Comparative results: Y vs. RGB

The above tests are performed on luminance images. This can readily be extended to RGB color space images. Table 5 illustrate the comparison of performance on luminance images and RGB color images. Motion of the videos in RGB space are estimated by

$$JPPC^{RGB} = 0.299JPPC^{R} + 0.587JPPC^{G} + 0.114JPPC^{B}.$$

The similarities between the ST-JPI and transition model S^{RGB} are calculated by

$$S^{RGB} = 0.299S^{R} + 0.587S^{G} + 0.114S^{B}.$$

Apparently, these two results are very comparable, with RGB taking nearly three times as much time, as expected.

6. Conclusion and future work

6.1. Conclusion

Video segmentation is an important preprocessing step for digital video processing. Although a variety of research work has been conducted for this problem, an efficient method for detecting, locating and classifying gradual transitions was still not available. This paper addresses this problem through spatial–temporal joint probability analysis. Spatial–temporal joint probabilities accurately characterize the temporal evolution of the visual characteristics of digital videos. Spatial–temporal joint probabilities derived from a video transition show a certain pattern of behavior determined by the transition type. The patterns are so regular that they can be reliably distinguished from video effects other than transitions. Patterns for various transitions differ with one another, and thus are very distinguishable.

In this paper, we build our transition method on the basis of spatial–temporal joint probability images, which are visual sketches of joint probabilities. Through performing a pattern match on an ST-JPI with predefined pattern models, transitions are detected and classified. To speed up the video segmentation process, a fast algorithm is designed and presented.

The main contributions of this paper are:

1. Mathematical models are defined for two types of gradual transitions, i.e., cross transition and dither transition. All gradual transitions can be characterized as one of the two types of transitions.
2. The spatial–temporal joint probability image is proposed for modelling the evolution patterns of gradual transitions. The ST-JPI patterns for both types of grad-

Table 5
Comparative results for gradual transitions: Y vs. RGB

		Motion in Y by $JPPC^Y$	Similarity (S^Y)	Motion in RGB by $JPPC^{RGB}$	Similarity (S^{RGB})
Cross transition	Static	0	0.0002	0	0.0004
	Bird	13.70	0.0022	15.45	0.0036
	Sports	26.05	0.0165	28.68	0.0216
	Tennis	35.12	0.0604	38.72	0.0617
Dither transition	Static	0	0.0004	0	0.0006
	Bird	13.70	0.0059	15.45	0.0085
	Sports	26.05	0.0237	28.68	0.0257
	Tennis	35.12	0.0719	38.72	0.0586
Total processing time (ms)			421		1171
Pattern matching time (ms)			81		360

ual transitions are analyzed. An ST-JPI based pattern matching method is proposed for detecting and classifying transitions. A fast algorithm that can detect most transitions is designed.

3. Tests are performed on the videos that are generated from video shots with various degrees of camera and/or object movements. Both experimental results and error analysis demonstrate the relationship between the quality of the proposed video segmentation technique and movements in the video.

Good experimental results demonstrate that ST-JPI based transition detection is a promising method for temporal video segmentation.

6.2. Future work

Although the algorithm based on spatial–temporal joint probability images works relatively well, compared to previous methods, it does have some limitations. Future research work should be conducted on the following aspects.

- Perhaps the most serious limitation of the method is that it can be influenced by object and camera movements. Movements in video have some temporal continuity. By estimating movements in video before and after potential transitions, prediction of movements in the two individual video shots can be made. With the help of this motion information, motion compensation can greatly improve the quality of ST-JPI.
- Our fast algorithm is built on the basis of line-detection, and thus only works for transitions with linear transition functions. If knowledge of transition functions is available for these transitions, the line-detection algorithm can be adjusted to detect all those predefined functions. If a transition function is not known, an estimation of the transition function can be made by computing the ratios between projection centroids, as reported above. But the estimation of the transition functions can be influenced by the motions in the video, and thus the reliability of a later pattern match is affected.
- A few types of dither transition schemes determine pixels to be changed on the basis of pixel value. Their transition pattern is strongly related to video content. Our algorithm is likely to fail for those dither transitions, although regular patterns still appear in those ST-JPIs. Individual ST-JPI patterns can be defined for each special dither transition, and our pattern match should include such patterns in order to detect all types of dither transitions.

Appendix

Algorithm (ST-JPI-based video segmentation)

Input: A video stream O
Output: A report of the transitions in the video.
Every transition is specified by three parameters: Type, Location and Duration
Method: (τ, ε, and δ represent predefined thresholds.)

$t_0 = 0$
While ($t_0 <$ length of video O){
  if $JPPC_{JPI_{O_{t_0},O_{t_0+32}}} > \tau$ { /* possible shot change */
    set time $t = t_0$ and range $r = 16$
    While ($r > 0$){
      if max ($JPPC_{JPI_{O_t,O_{t+r}}}$, $JPPC_{JPI_{O_{t+r},O_{t+2r}}}$)
        $> 0.75 \cdot JPPC_{JPI_{O_{t_0},O_{t_0+32}}}$
          if ($r = 1$) report a transition:
            Type = cut; Location = r;
            Duration = 1
          Break;
          else if ($JPPC_{JPI_{O_t,O_{t+r}}} \geqslant$

```
                    JPPC_{JPI_{O_{t+r},O_{t+2r}}})
                        then r = r/2
                else t = t + r, r = r/2
        }
        if (r > 2){ /* it is not a cut. */
            Derive JPPC_{JPI_{O_t,O_{t+i}}} from image O_t and
              O_{t+i}, i ∈ [1, r];
            Calculate the best fit straight line, and
              the mean square error (MSE)
            if MSE > ε then no transition found
                        t_0 = t_0 + 32
                        Continue.
            Do {
                r = r + 1
                Derive JPPC_{JPI_{O_t,O_{t+r}}} from pairs of
                  images O_t and O_{t+r};
            } while (distance of JPPC_{JPI_{O_t,O_{t+r}}} from
              line 1 < ε)
            Initialize i = 0
            Do {
                i = i − 1
                Derive JPPC_{JPI_{O_t,O_{t−i}}} from image O_t
                  and O_{t−i},;
            } while (distance of JPPC_{JPI_{O_t,O_{t−i}}} from
              line l < ε)
            Generate JPPVs from the ST-JPI
            Do cross uniform transition
            Calculate Similarity
            If Similarity < δ then Report a transition:
                        Type = cross Transition;
                          Location = t − i;
                          Duration = r + i
                    t = t + r + 1
                    Continue
                    Do dither uniform transition and
                      calculate Similarity
            if Similarity < δ then Report a transition:
                        Type = Dither transition;
                          Location = t − i;
                          Duration = r + i
                    t = t + r + 1
        }
    }
}
```

References

[1] U. Gargi, R. Kasturi, S.H. Strayer, Performance characterization of video-shot-change detection methods, IEEE Trans. Circuits Systems Video Technol. 10 (1) (2000) 1–13.

[2] P. Salembier, F. Marques, Region-based representations of image and video: segmentation tools for multimedia services, IEEE Trans. Circuits Systems Video Technol. 9 (8) (1999) 1147–1169.

[3] C.M. Lee, M.C. Ip, A robust approach for camera break detection in color video sequence, Proceedings of the IAPR Workshop Machine Vision Applications, 1994, pp. 505–505.

[4] M.S. Drew, J. Wei, Z.N. Li, Illumination-invariant image retrieval and video segmentation, Pattern Recognition 32 (1999) 1369–1388.

[5] A. Hampapur, R. Jain, T. Weymouth, Production model based digital video segmentation, J. Multimedia Tools Appl. 1 (1) (1995) 9–46.

[6] A. Nagasaka, Y. Tanaka, Automatic video indexing and full-video search for object appearances, Proceedings of the IFIP Second Working Conference on Visual Database Systems, 1992, pp. 113–117.

[7] W. Niblack, R. Berber, M. Flickner W. Equitz, E. Glasman, D. Petkovic, P. Yanker, The QBIC project: query images by content using color, texture and shape. Proceedings of SPIE, 1993, pp. 173–181.

[8] B.M. Mehtre, M.S. Kankanhalli, A.D. Narasimhalu, G.C. Man, Color matching for image retrieval, Pattern Recognition Lett. 16 (1994) 325–331.

[9] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vision 7 (1) (1991) 11–32.

[10] P. Bouthemy, M. Gelgon, F. Ganansia, A unified approach to shot change detection and camera motion characterization, IEEE Trans. Circuits System Video Technol. 9 (7) (1999) 1030–1044.

[11] R.M. Ford, C. Robson, D. Temple, M. Gerlach, Metrics for shot boundary detection in digital video sequences, ACM Multimedia Systems 8 (1) (2000) 37–46.

[12] R.M. Ford, A fuzzy logic approach to digital video segmentation, SPIE Conference on Storage and Retrieval for Image and Video Databases VI, February 1998.

[13] R. Zabih, J. Miller, K. Mai, A feature-based algorithm for detecting and classifying scene breaks, ACM Multimedia Systems 7 (2) (1999) 119–128.

[14] C.W. Ngo, T.C. Pong, R.T. Chin, Detection of gradual transitions through temporal slice analysis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '99), 1999, pp. 36–41.

[15] Z.N. Li, J. Wei, Spatio–temporal joint probability images for video segmentation, Proceedings of the IEEE International Conference on Image Processing (ICIP 2000), Vol. II, 2000, pp. 295–298.

[16] H. Farid, E.H. Adelson, Separating reflections and lighting using independent component analysis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '99), 1999, pp. 262–267.

[17] A.M. Tekalp, Digital Video Processing, Prentice-Hall, PTR, Englewood Cliffs, NJ, 1995.

[18] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, W. Niblack, Efficient color histogram indexing for quadratic form distance functions, IEEE Trans. Pattern Anal. Mach. Intell. 17 (7) (1995) 729–736.

[19] J.L. Devore, Probability and Statistics for Engineering and the Sciences, Wadsworth Publishing Inc., 1995.

[20] M.S. Drew, Z.N. Li, X. Zhong, Video dissolve and wipe detection via spatio–temporal images of chromatic histogram differences, Proceedings of the IEEE International Conference on Image Processing (ICIP 2000), Vol. III, 2000, pp. 929–932.

About the Author—ZE-NJAN LI received the B.S. degree in Electrical Engineering from the University of Science and Technology of China in 1970, and the M.S. and Ph.D. degrees in Computer Sciences from the University of Wisconsin-Madison in 1981 and 1986, respectively. From 1970 to 1979 he was an electronic engineer in charge of digital and analogical system design. He was an Assistant Professor at the University of Wisconsin-Milwaukee from 1986 to 1987. In 1988 he joined the School of Computing Science at Simon Fraser University, where he is currently a Professor and Director of Vision and Media Lab. His current research interests include multimedia, computer vision and pattern recognition.

About the Author—XIANG ZHONG received the B.Sc. degree in Computer Science and the M.Sc. in Electronic Engineering from Huazhong University of Science and Technology, China. Subsequently, he received the M.Sc. in Computing Science from Simon Fraser University, Canada, in 2000. He is currently working at Lumic Electronics Inc. His interests include video compression, video segmentation, and multimedia databases.

About the Author—MARK S. DREW is an associate Professor in the School of Computing Science at Simon Fraser University. He received the B.A.Sc. in Engineering Science at the University of Toronto in 1970, the M.Sc. in the Foundations of Quantum Mechanics in 1971 in the Mathematics Department at the same University, and the Ph.D. in Theoretical Physics from the University of British Columbia in 1976. Combining work on Energy Systems and Computer Applications, he held an Industrial Postdoctoral Fellowship and subsequently an Industrial Research Fellowship in industry until he joined SFU in 1982. His interests lie in the fields of multimedia, computer vision especially focusing on color, photorealistic computer graphics, and computer algorithms for color reproduction. Dr. Drew is the holder of a US patent in digital color processing and a Director of Lightseer Ltd., a UK color research firm.