TEMPORAL MULTI-RESOLUTION ANALYSIS FOR VIDEO SEGMENTATION

Yi Lin & Mohan S Kankanhalli & Tat-Seng Chua

School of Computing
National University of Singapore
Singapore 119260
Email: {linyi/mohan/chuats}@comp.nus.edu.sg

ABSTRACT

Video segmentation is an important step in many of the video applications. We observe that the video shot boundary is a multi-resolution edge phenomenon in the feature space. Based on this observation, we have developed a novel temporal multi-resolution analysis (MRA) based algorithm using Canny wavelets to perform temporal video segmentation. Information across multiple resolutions is used to help detect as well as locate abrupt and gradual transitions. We present the theoretical basis of the algorithm followed by the implementation as well as the results. In this paper the MRA technique has been implemented using color histogram as the feature space. Experimental results shows that this method can detect as well as characterize both the abrupt and gradual shot boundaries. The technique also shows good noise tolerance characteristics.

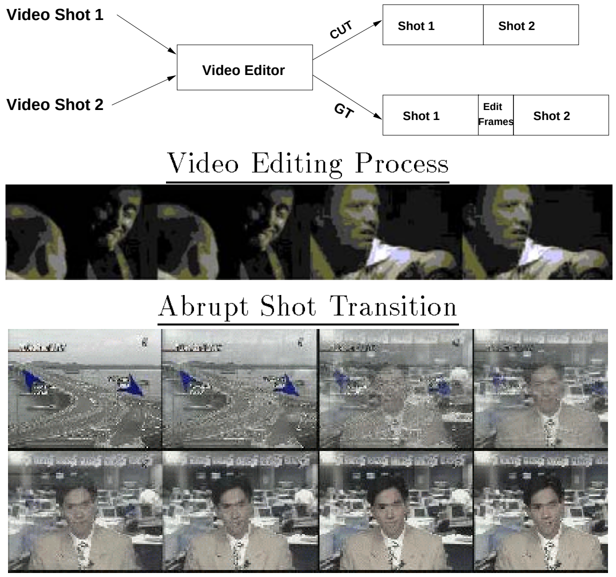
1. INTRODUCTION

The rapid proliferation of digital video information has created a need for many video applications which in turn has necessitated the development of many video data processing techniques. Many of these applications require the ability to represent, index, store and retrieve videos efficiently. Since video is a continuous time-based medium, it is very important to break the video streams into basic temporal units, known as shots [31]. These elemental units can then be utilized for many kinds of operations.

This kind of temporal segmentation is very useful in most video applications. For example, in video modeling, when the video stream is segmented, the low-level structure of video can be extracted. The temporal segmentation is the first basic step in the structured modeling of video [1, 3, 22]. When attempts to let computers understand videos are made, say like reconstructing the story from the video data stream, these shots serve as the basic bricks to construct the whole story. The understanding of video usually requires the understanding of the relationship between shots. While in an application concerning video storage & retrieval, indexing the shots seems to be an inevitable step [1]. Therefore a good algorithm for the temporal segmentation of digital video can be helpful in all of these applications. Consequently, there have been many research efforts in this direction.

The temporal partitioning of video is generally called *video segmentation* or *shot boundary detection* or *scene change detection* [9, 10]. To fulfill the task of partitioning the video, any video segmentation technique needs to detect the joining of two shots in the video stream and locate the exact position of these joins. These joins are made by the video editing process, and they can be of two different types based on the technique involved in the editing process [8]. If the video editor does nothing but directly concatenates the two shots together, the join is termed to be an *abrupt transition*, which we denote as CUT. On the other hand, if the video editor uses some special technique such as fade-in/fade-out, wipe, dissolve or morphing to make the joint appear visually smooth, the join will be a *gradual transition*, which is denoted by GT. Figure 1 gives an example of a CUT and a

GT.



Video Editing Process

Abrupt Shot Transition

Gradual Shot Transition

Figure 1: Video Shot Transitions

Due to the existence of the above two types of the transitions and the fact that GTs can be of varying temporal durations, the transition of video shots is not a single resolution phenomenon. By resolution, we mean the temporal resolution of a video stream which could be either high (i.e. the original video stream) or low (i.e. due to some form of temporal sub-sampling obtained by grouping of frames). For example, although longer GTs cannot be observed at a high temporal resolution, it is easily apparent at a low temporal resolution of the same video stream. So we claim that the transition of video shots is a *multi-resolution* phenomenon. In this paper, we have developed a novel multi-resolution temporal analysis based approach to solve the problem of scene change detection. In this approach, information across multiple resolutions will be utilized to help detect as well as locate both the CUT and GT transition points. Since wavelets are known to be suitable for studying multi-resolution phenomenon [14, 23], we use *Canny wavelets* for this multi-resolution analysis.

The rest of this paper is organized as follows. The past work has been reviewed in Section 2. Section 3 describes our understanding of the problem and introduces the theoretical basis. Section 4 describes the algorithm in detail. Section 5 discusses the experimental results, and Section 6 provides the conclusion and also outlines many areas for future work.

2. BACKGROUND

A lot of work has been done on detecting the CUT and many algorithms have been developed to handle the GT detection [17, 6, 25, 32, 19, 24, 27, 20, 29, 30]. By assuming that different shots have different content-features, the video segmentation algorithms for CUT detection usually work by sequentially measuring successive inter-frame differences and studying their variances. Zhang et al. measured the difference of color histogram between successive frames and used a global threshold to detect CUT. When the difference is above the global threshold, a CUT is declared. When dealing with the GT, they used two thresholds. Accumulated differences of successive frames were computed when the inter-frame difference is above a lower threshold. When this accumulated difference exceeds the high threshold, a GT is declared. Template matching method compares the pixels or regions of two images across the same location [31]. DCT comparison works directly on the compressed domain comparing the DCT coefficients [19, 27]. Most of these methods need careful threshold selection and are affected by noise [9]. The threshold for CUT detection can be chosen by modeling the noise as a Gaussian distribution [31] – it can be easily set as a range and then used for considering whether the inter-frame difference is significant enough. However, it appears that the threshold for GT can only be chosen by experience. The threshold that works for one video may not work for some other video. Also, when using the color feature, object motion and camera flash may cause false detection, therefore they need special treatment [27].

There are some other approaches also. Hampapur et al. [8] proposed a model based method by studying the video production techniques which describes different models for different editing effects. Some other researchers have adopted a statistical approach, studying the distribution of the difference-histogram and characterized different type of transitions with different types of distributions [16]. A feature-based algorithm has also been proposed for detecting the GT by observing the pattern of the increasing/decreasing edges in the frames in a GT [30]. In the Hong et al. method [29], wavelets are used to spatially decompose every frame into a low-resolution component and a high-resolution component. They then detect the possible gradual transitions by extracting the edge spectrum average feature in the high-resolution component to detect fade and apply double chromatic difference on the low resolution component to identify the dissolve transitions.

Our proposed approach differs from these previous works in the following ways. In our work, we measure the difference between sets of frames instead of just between every two successive frames. In this way we can vary the size of each set which corresponds to different temporal resolutions. Thus ours is a multi-resolution approach as opposed to the single resolution approach adopted in the past work. Hong et al. have multi-resolution approach but this is purely in the spatial domain. So temporally, their approach is still a single resolution approach. Thus our multi-resolution analysis is in the temporal domain in contrast to those using the multi-resolution method which are in the spatial domain. Lastly because of the simultaneous use of multiple resolutions, the result of the detection is not very sensitive to the threshold selection. Moreover, our algorithm has good performance for both abrupt and gradual transitions with low sensitivity to noise.

3. MULTI-RESOLUTION VIDEO ANALYSIS USING WAVELETS

In this section, we will introduce the necessary background for the development of our algorithm. In particular, we will explain the basis for the video representation in the feature space and the use of wavelets for the multi-resolution analysis.

3.1. Video Representation

We mathematically model the video using the content of the frames in the video data stream like in [21]. The content is usually some low-level features of the video frame and it could be of any of the following types: color, shape, texture or motion. Let us assume that we have chosen the color feature. For visualizing this representation, we first map the video stream into the 3D feature (color) space. Consider a very simple mapping – for every frame f in the video stream, we compute the average RGB color for this frame. Note that we use this mapping only for the sake of illustration and has not been used in in our algorithm. Using $I_{xy} = (R_{xy}, G_{xy}, B_{xy})$ to represent the RGB value of every pixel (x, y) of the frame, we can compute $f = I_{average} = \frac{\Sigma I_{xy}}{w \times h}$ where w and h are the width and height of the frame. In other words, we only use one average color to represent a frame. In this way, every frame is mapped into a point in the RGB color space. If we connect these points in their temporal order of occurrence in the video stream, the *temporal sequence* of frames will map into a *spatial trajectory* of points in the feature space. Figure 2 shows the mapping of a video sequence into the RGB space.

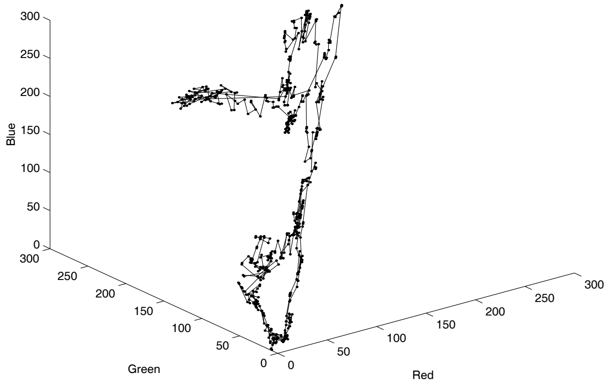


Figure 2: Mapping a Video into RGB-space using Average Color

A different content feature will lead to a different feature space in which the video sequence will trace a corresponding trajectory. The dimension of the space depends on the dimensionality of the chosen feature. For example, if a 64-bin color histogram is used to represent each frame, we will have obtain a $64-D$ feature space. If we carefully observe the mapped video in the feature space, we can notice that the similarity between any two frames is well reflected by the distance between the two corresponding points in the space. Similar frames will be closer in the feature space and dissimilar frames will be far apart. Of course, the exact distance will depend on how well can this feature space approximate the perceptual content. Since the frames belonging to one particular shot usually have a high similarity in content, the corresponding points in the feature space tend to be clustered.

In this way, we can easily map any video sequence into a corresponding trajectory of points in the multi-dimensional feature space. This trajectory of points can be considered as a sequence of samples of a multi-dimensional signal. Since the color-histogram representation has been found to be useful for the video segmentation problem [31], we also use the n-color histogram to model the content of each frame of a video. We have used $n = 64$. Every frame of video stream can be expressed as $f = (x_1, x_2, ..., x_n)$ where x_i is the i^{th} bin in histogram representation. We can then use $v = f(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ to represent the temporal video stream v. Basically, the video is modeled in an n-dimensional feature space with every bin of the histogram corresponding to an axis of the space. Besides this feature space representation, we also use derivatives of the video signal trajectory to detect the shot transitions.

By empirically observing GTs in many video streams, we have found that different types of GTs exist like fade-in/fade-out, dissolve, wipe, morphs etc. Moreover, the length of the transition can vary widely. Although GTs lasting for thousands of frames exist, most of the GTs span lengths of 5 to 100 frames. Most of the existing algorithms assume that in general the content between shots changes much more than the intra-shot change. The veracity of this assumption has been borne out by observing several video trajectories in the feature space. We observe that different types of shot transitions are observable at different resolutions in the feature space. We have also noticed that whatever be the type or the length of the transition, there will always be a big enough change that is observable at some resolution. Figure 3 shows that the GT, which is not observable at a high resolution, can be easily detected at a lower resolution. But these two phenomena can be observed at different resolutions. For CUT, we can observe the change both at a high resolution, and at a coarse resolution. On the other hand, GTs show the change only at coarse temporal resolutions. So the transitions must be defined with respect to different resolutions. By viewing the video at multiple resolutions (scales) simultaneously, the detection of both CUTs and GTs could be unified. The only difference is that GTs correspond to local maxima (which are also edges or boundaries) of the multi-dimensional signal (feature-space video trajectory) in the low resolution while CUTs correspond to boundaries in *all* the resolutions. The longer the duration of a GT means the lower is the resolution at which the boundaries are observable. By making this fundamental observation that a video shot boundary is a multi-resolution phenomenon, we can characterize the transitions with the following features:

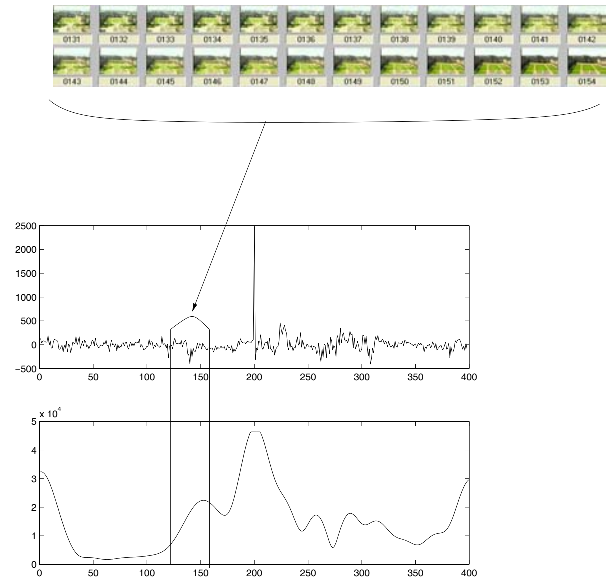


Figure 3: Multi-resolution Nature of a Gradual Transition

- The resolution of the transition (denoted by parameter a explained in section 3.2)
- The strength of the transition (which is computed by $|W_{2^j}f(x)|$ explained in section 3.2)
- The regularity of the transition point (which is characterized by the Lipschitz α explained in section 3.2)

Using these ideas, we have developed a unique multi-resolution analysis technique to detect and characterize both CUT and GT shot boundaries. In this paper, we mainly use the color feature for video representation but our technique is not limited to this feature.

3.2. Wavelet Analysis

We first introduce the basic theory of the analysis in the function space $L^2(\mathbb{R})$. We then apply it to digital video by considering the temporal sequence of video frames as samples from a continuous signal. The convolution of a 1D signal $f(x)$ with a Gaussian function θ with variance a, $a \in \mathbb{R}$ can be considered as a coarse representation of the signal. Since the convolution in the signal domain is equivalent to multiplication in the frequency domain, the convolution of f and θ acts like a low-pass filter. That means the details (high frequency components) in the time/spatial domain are removed in the new transformed signal obtained after the convolution. This new signal function obtained after convolution can also be considered as a lower resolution representation of the original function. Denoting the Gaussian kernel by

$$\theta(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{1}$$

and the convolution at point b for resolution a as

$$f * \theta(\frac{x-b}{a}) \;=\; \frac{1}{a}\int_{-\infty}^{\infty} \theta(\frac{x-b}{a}) f(x)dx \tag{2}$$

For temporal video segmentation, it is commonly assumed that the shot transition cause a rapid change in the video content. Therefore, these transition points can be located by finding the the points where the content change occurs most rapidly. These points of rapid variations are precisely those for which the modulus of the first order derivative has a local maximum. Thus the detection of the maximum of the first order derivative after transforming the signal can help locate the transition points in some resolution. The abrupt transition will be observed at the high resolution while a gradual transition will be apparent only at a coarser resolution. So, if we could somehow determine these maxima in all resolutions of the signal (each resolution obtained through convolution which is essentially smoothing), we would be able to detect the transition points in all the resolutions. This provides the *basis* for detecting the GT and CUT in one (multi-resolution) framework. The resolution (defined with respect to a reference resolution) will characterize the nature of the transitions. The concept of multi-resolution analysis is related to the mathematical theory of wavelets. Wavelets are useful since they are well localized in both time and the frequency domain.

We briefly present here the basics of wavelet theory related to this paper [2, 13]. A function h is called a wavelet if it satisfies the admissibility condition,

$$C_h = 2\pi \int \frac{|\hat{h}(\omega)|^2}{|\omega|} d\omega < +\infty \tag{3}$$

By the process of dilations and translations, the mother wavelet h builds a family of analyzing functions. They are normalized as:

$$h_{a,b}(x) = a^{-1/2} h(\frac{x-b}{a}), a \in \mathbb{R}, b \in \mathbb{R} \tag{4}$$

Then the wavelet transform is defined as:

$$W_{a,b}f(x) = f(x) * h_{a,b} = \int a^{-1/2} f(x) h(\frac{x-b}{a})dx \tag{5}$$

The n derivatives of the Gaussian θ:

$$\theta^{(n)} = \frac{\partial^n \theta(x)}{\partial x^n} \tag{6}$$

can be shown to be wavelets for $n > 0$ by the following lemma: [5]
Definition: Let $n \in \mathbb{N}$. A wavelet ψ has n vanishing moments(is of order n), if for all integers $k < n$:

$$\int_{-\infty}^{\infty} \psi(x) x^k dx = 0 \tag{7}$$

and

$$\int_{-\infty}^{\infty} \psi(x)x^n dx \neq 0 \qquad (8)$$

Lemma: Let ϕ be a n-times differentiable function and ϕ,$\phi^{(n)} \in L^2(\mathbb{R})$,$\psi^{(n)} \neq 0$. Then it follows that $\psi = \phi^{(k)}$ is a wavelet.

Let us denote $\theta_a = \frac{1}{a}\theta(\frac{x}{a})$ as the Gaussian at resolution a, then we obtain:

$$W_a f(x) = f{*}\psi_a(x) = f{*}(a\frac{d\theta_a}{dx})(x) = a\frac{d}{dx}(f{*}\theta_a)(x) \qquad (9)$$

This also shows that the first derivative of smoothed signal f at some resolution a can also be obtained by smoothing f with the first derivative of the Gaussian θ_a. And all these could be done using the wavelet transform by using this particular mother wavelet based on the Gaussian. The first order derivative of the Gaussian $\frac{\partial\theta(x)}{\partial x}$ forms the **Canny wavelet**. Figure 4 shows the Canny wavelet at different resolutions.

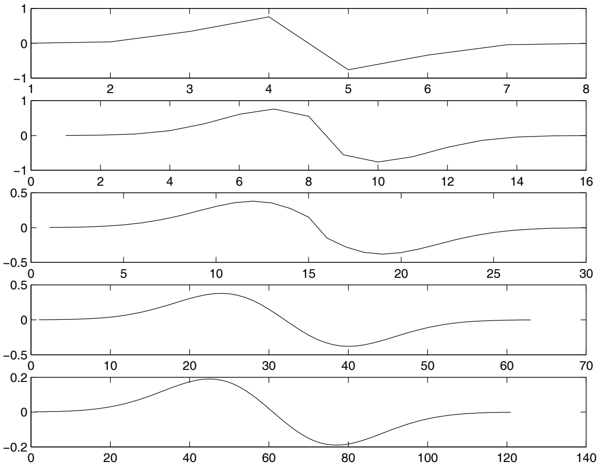


Figure 4: Canny Wavelet at Resolutions 2^0, 2^1, 2^2, 2^3 & 2^4

Using the Canny wavelets, we could transform the first derivative of any f in $L^2(\mathbb{R})$ to a representation in the scale space [13]. In other words, the derivative of function f has representations at different resolutions of the scale space. The higher the resolution, the more precise is the representation. This fact is used in the transition location algorithm to precisely locate the end-points of a transition. From the earlier discussion, we can easily see that the video shot transitions can be detected by the absolute maxima points in the scale space generated by the Canny wavelets. Over and above detecting the transitions, wavelets also provide the ability to characterize the transitions using these maxima points by utilizing the Lipschitz exponent. The regularity of a function at a point x_0, can be characterized with a Lipschitz exponent α. A function $f(x)$ is said to be Lipschitz α regular in x_0 $(0 \leq \alpha \leq 1)$, if and only if for all x in a neighborhood of x_0, we have $|f(x) - f(x_0)| = O(|x - x_0|^\alpha)$. The following theorem relates the Lipschitz regularity of a function to the amplitude of its wavelet transform [2]:

Theorem: For a function $f(x) \in L^2(\mathbb{R})$, $f(x)$ is Lipschitz α regular in all points of an open interval if and only if for all x in this interval [14]

$$|W_{2^j} f(x)| = O(2^{\alpha j}) \qquad (10)$$

We can utilize the above theorem in the following way: we can construct the scale space by the dyadic scale 2^j, and study the evolution of the amplitudes of the absolute maxima across the scales. We can obtain the degree of smoothness and the corresponding scale where the transition exist. Using these information, we can also distinguish the shot transitions from noise (such as object motion or flash) as well as CUTs from GTs. The details of how exactly this can be done is presented in the algorithm description. Conceptually, we consider the sequence of video frames to be the result of sampling of a continuous signal. In our case, since the digital video feature space is multi-dimensional, the video is a sequence of multi-dimensional samples. Since this feature space is multi-dimensional, we use the separability property to compute the one-dimensional Canny wavelet transform of these samples along each axis separately. The absolute maxima in the multi-dimensional space are then obtained by combining them using a Euclidean measure.

4. THE ALGORITHM

The theoretical basis of the algorithm has been explained in the previous section. In this section, we present a multi-resolution algorithm for shot boundary detection. The input is a video stream and the output consists of (a) the frame numbers where CUTs occur and (b) pairs of frame numbers corresponding to the start & end of GTs. The overall algorithm proceeds in this manner:

1. Compute the 64-bin histogram color feature for each frame of the video sequence [31].

2. Compute the Canny wavelet transform on the trajectory corresponding to the video sequence in the 64-D feature space.

3. Select the potential transition points at all resolutions. The potential transition points are basically the points which correspond to the local maxima. The details of this step are presented in Section 4.1.

4. Reject the potential transition points which correspond to noise. This procedure is explained in Section 4.2. We are then left with the genuine transition points which can be utilized to locate the shot transitions.

5. Distinguish the transition points corresponding to CUTs from those corresponding to GTs. The algorithm for this step is given in section 4.4.

6. Locate the exact transition points in the highest resolution (and thus their corresponding frame numbers) and output them. This step is elaborated in Section 4.3.

Note that compared to the existing shot-detection techniques which rely on threshold selection, resolution selection is relatively easier and is more adaptive. Although we could try to find a way to get the "perfect resolution" for each video stream, we do not really need to do so. This is because most of the information in multi-resolution analysis is contained across all resolutions, and thus the sensitivity to resolution selection is not much. In the actual implementation, we perform analysis on resolutions from 0 to 5. The wavelet transform can be construed to be a filter convolution operation which means that the lengths of the wavelet filter ranging from 2 to 100 will cover most of the lengths of GT that can occur in real videos.

4.1. Selecting Potential Transitions

We have earlier seen that shot transitions correspond to local maxima in the multi-resolution scale space. This is nothing but the multi-resolution edge [14]. For all resolutions, we compute the points corresponding to local maxima in strength using $|W_{2^j}f(x)|$. Notice that by this process we will be selecting spurious transition points which actually correspond to noise such as object movement or a camera flash. We there need to separate the points which really correspond to shot transitions from the ones that correspond to noise.

For a given potential transition point at the highest resolution, we find the corresponding point at all the resolutions. We can thus link the potential transition points across resolutions. First select a potential transition point in the highest resolution. In the next lower find the nearest transition point. Link this point with the initial one and iterate for rest of the resolutions. We call a set of linked potential transition a *maxima path.* If there are i potential transition points in the highest resolution, we will obtain i maxima paths.

4.2. Noise Removal

The basic idea used to remove potential transition points corresponding to noise points is that the noise will get blurred at coarser resolutions and hence the strength of the such maxima points will get severely attenuated at coarser resolutions. On the other, the strength of transition points corresponding to CUTs and GTs will not be affected much. We could pick any resolution to do this, but in practice, we first pick all the maxima points and compare their strengths across all the resolutions using the maxima paths. If the strength reduces significantly at lower resolutions (scales 3 to 5), then such points are rejected. After this phase is completed, the left-over transition points are the ones which really correspond to transitions.

4.3. Locating the Transitions

The goal of video segmentation is not only to detect the existence of transitions, but also to locate precisely the positions of the CUTs and GTs in the video sequence. For a GT, both the start and the end positions need to be detected. Our method for doing this is somewhat similar to the techniques presented in [15, 5, 12].

There is a shifting of the maxima points (corresponding to the same transition) at different resolutions. When the resolution becomes lower, the signal is blurred, and this causes the position of each maxima point to shift. If we connect the corresponding maxima points at all resolutions, a maxima path is formed which traces the shifts. For the GT, both the location of the start point and the end point of the transition have a maxima in the highest resolution. These two points delimit the gradual transition region of the video. If we observe the behavior of these two points across the different resolutions, we will notice that as the resolution decreases, there will be a shift and during this shift, the two points will move towards each other to fuse into one single point at some low resolution. Basically at this resolution, the GT has become very significant. In other words, the GT maxima point in that low resolution will correspond to the transition region (delimited by the two end-points) in the high resolution. Thus, we could perform a do-while loop to find out the region and thus locate the start & end of the gradual transition.

We now provide the detailed description of the algorithm to locate the GT region. First choose points that can correspond to a GT transition. These are termed as the significant points. Then a low to high resolution search is performed. In the search, for every resolution and every significant point, we find the two nearest significant points at the next higher resolution. The one in the left is marked as SL and the one on the right is marked SR. We recursively trace up the SL from the coarsest resolution to the highest resolution in order to find the final SL. This SL corresponds to the left boundary point of the maxima region at the finest resolution. A similar thing is done for SR and thus the (SL,SR) denotes the maxima region at the highest resolution (which corresponds to a GT).

For the CUT transition, a similar tracing is done from the highest resolution to the coarsest resolution. In the case of CUT, the significant points are located at the highest resolution. Then a corresponding point is found at the next lower resolution and the process is iteratively continued till the coarsest resolution is reached. Note that there will be only one corresponding point in case of the CUT. This procedure pre-supposes that we can distinguish the transition points corresponding to CUTs from those corresponding to GTs. This can be done by the procedure explained in the following sub-section.

4.4. Distinguishing CUTs from GTs

For the previous procedure to work, we need to label whether a transition point corresponds to a CUT or to a GT. This is achieved by using the regularity of the wavelet transform using the Lipschitz α. We first compute:

$$\beta \;=\; \log |W_{2^j} f(x)| \tag{11}$$

Based on the discussion of the measure of regularity, Lipschitz α, we can obtain from eqn. (11):

$$\beta \;=\; O(\alpha j) \tag{12}$$

So we can compute the β value for a potential transition point at different resolutions along its maxima path to obtain:

$$\beta_{\text{average}} \;=\; \frac{\sum \beta}{\sum j} \tag{13}$$

A high value of β_{average} corresponds to a faster transition. By empirically observing the range of β_{average} values for CUTs as well as GTs, we can easily threshold them into CUTs and GTs using the value of β_{average} equal to 4. Therefore, if $\beta_{\text{average}} \geq 4$, then the points correspond to a CUT and if $\beta_{\text{average}} < 4$, then they correspond to a GT.

5. EXPERIMENTAL RESULTS

5.1. Sample Video Data Set

To evaluate the performance of our algorithm for shot boundary detection, we carried out an experimental study using 10 different videos of varying characteristics. This video data set consists of a documentary video (a university campus description), a commercial, a music video (Michael Jackson's "Black or White"), a cartoon video, four movie videos and two news videos. This data includes many different types of shot transitions. The lengths of these transitions are also highly variable. Figure 5 shows the result of applying our algorithm on the documentary video. The figure shows the original video sequence in the topmost graph (the 1-D representation is derived by computing the magnitude of the 64-D histogram vector). The Canny wavelet transform at 5 resolutions are shown in the next 5 graphs. The bottommost graph superimposes all the resolutions onto one graph. The vertical lines show the positions of the GT and CUT at the finest resolution. Notice that both the gradual transition as well as the abrupt transition are correctly detected. We will now illustrate the ability of our algorithm handle various types of GTs. Figure 6 shows the correct detection of the *dissolve* GT in a video of a commercial. The other detected transitions are not indicated in this as well as in the following figures. A *wipe* GT transition is accurately detected in a news video which is shown in figure 7. Finally, figure 8 shows the correct detection of a *morph* GT in the music video. Many existing algorithms fail to detect morphs but our algorithm can handle this case.

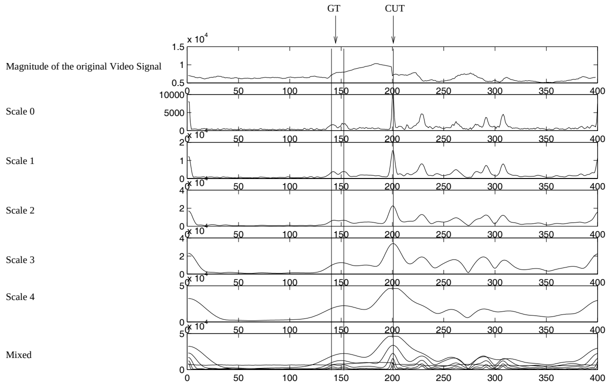


Figure 5: Result for a Documentary Video

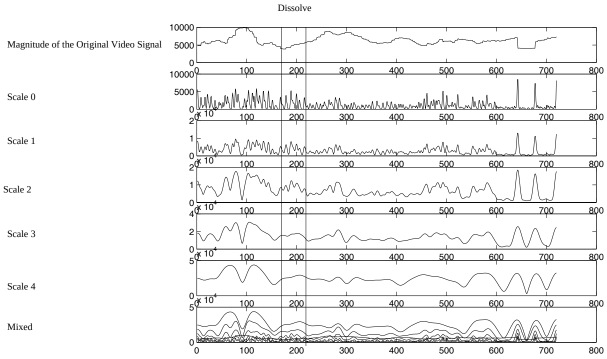


Figure 6: Result for a Commercial Video – highlighting only one *dissolve GT*

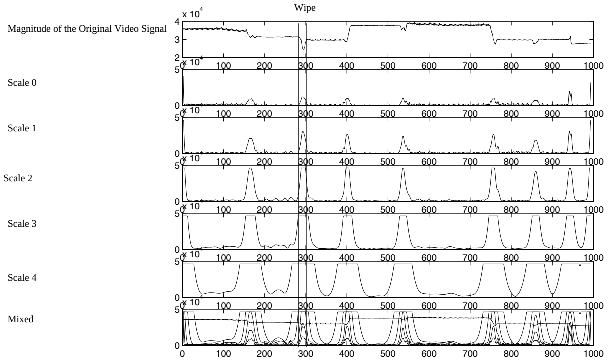


Figure 7: Result for a News Video – highlighting only one *wipe GT*

We now summarize the results of our testing. We use N_t for the total number of shot transitions, N_{CUT} for the number of CUTs and N_{GT} for the GTs. N_f denotes the number of false detects, N_m denotes the missed transitions and N_d for the number of transitions detected correctly. The results of our algorithm's performance has been summarized in table 1. Note that in the last column of the table, the type of the gradual transition has been specified – dissolve (D), wipe (W), morph (M) and fade-in/fade-out (F). One condition which causes a missed transition is the following – if a GT is followed very closely by another GT (say within a second or 30 frames), then these two GTs will be considered as one GT by the algorithm. However,

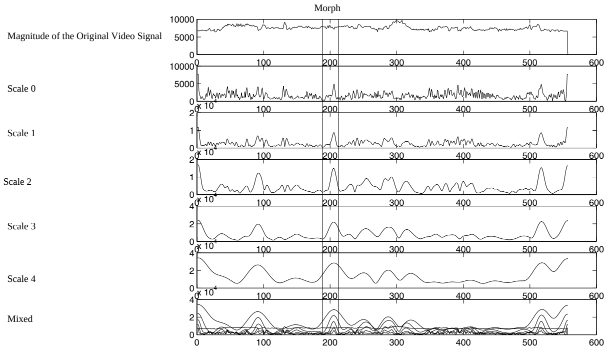


Figure 8: Result for a Music Video – highlighting only one *morph GT*

Type	N_d	N_m	N_f	N_{CUT}	N_{GT}	N_t
Doc	2	0	0	1	1 D	2
News	10	0	3	0	6 W, 1 D	10
Music	11	1	1	0	11 M	11
Comm	11	1	1	2	9 D	11
Movie	16	0	0	16	0	16
Movie	11	0	0	11	0	11
News	42	1	2	31	2 D, 6 W, 2 F	41
Cartoon	30	0	0	29	1 D	30
Movie	12	0	0	12	0	12
Movie	30	0	2	28	0	30

Table 1: Result Table

this rarely occurs in real videos. From our testing we have found that there are two basic reasons for the false detections:

- CUT: A false abrupt transition detection can occur when a large object enters within a shot. This is falsely recognized as a CUT.

- GT: A false gradual transition is detected when there is a significant camera panning movement. This could be overcome by separating camera motion like it was done in the twin-comparison method [31]. But we have not yet incorporated this into our implementation.

5.2. Noise Tolerance

Here we discuss two types of the noise that occur in most video streams:

- *Limited object movement*: We consider the situation where the camera has little movement and the object may move which can cause false detection. For this case, we have studied the strength of the wavelet coefficients in the multi-resolution space. We notice that at a coarse resolution, the noise gets severely attenuated by the Gaussian filtering. Hence we have found in our tests that limited object movement, where the movement is not very fast, does not result in false detects.

- *Camera Flash*: A flash is a special effect that causes a big change in a very short temporal duration [27]. From the point of multi-resolution analysis, it will appear to be very significant at a high resolution and drops drastically at the low resolutions. Figure 9 illustrates this from a real video data sequence. The flash occurs at the third frame. When the resolution goes down (scale goes up in the figure), the strength of the transition point drops significantly. Thus this property can be exploited to easily remove it during the tracing phase of the algorithm while detecting the potential transition points.

Note that we have not considered fast camera motion and fast object movement. We are still investigating on how to minimize their effects. The camera motion could be detected using optical flow methods [31].

6. CONCLUSIONS

In this work, we have shown how to visualize a temporal video sequence as a trajectory of points in the multi-dimensional feature space. We have made a fundamental observation regarding the multi-resolution nature of the shot boundary phenomenon in videos. A temporal multi-resolution approach using Canny wavelets has been developed to solve the problem of video shot boundary detection. Experimental results show that this method is successful in detecting abrupt transitions as well as gradual transitions and it has good tolerance to common types of noise occurring in videos.

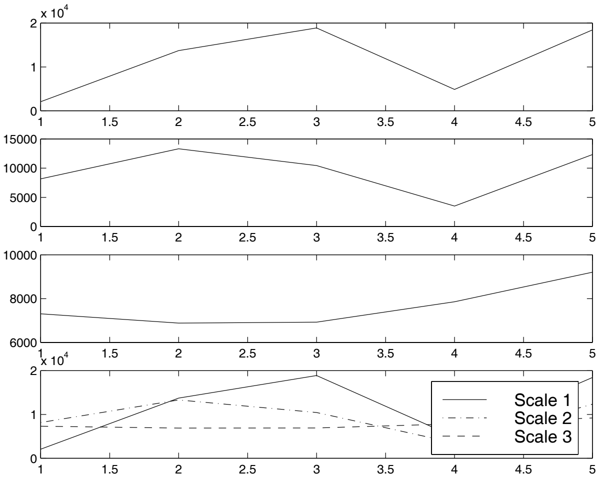


Figure 9: Illustration of a Camera Flash

We now present some issues for future research. It is very important to choose a good feature space for analyzing video data. The feature space for the video space used in this paper did not consider motion, which is the distinguishing characteristic of videos. We believe that a well constructed feature space which incorporates motion information should be helpful in further video data analysis. Although we have used wavelets to detect the gradual transitions, but we still cannot recognize the type of gradual transition. We feel that by choosing different types of mother wavelets in an appropriate feature space can help detect as well as recognize the type of gradual transitions. Our implementation has been done in the raw domain – however, it can be done directly in the compressed domain by using the DCT coefficients to construct the feature space. This can result in computationally efficient algorithm. Another issue related to the computational efficiency is the choice of the wavelet transform algorithm. Since we use the Canny wavelet transform, it is slow for computing at the coarser resolutions. It appears that the B-Spline wavelets are much more efficient from the point of computational complexity and are quite similar to the Canny wavelet [23]. So, the B-spline wavelets may be the appropriate transform for temporal analysis of video data. In general, we feel that temporal multi-resolution analysis offers a novel approach to flexibly probe the structure and content of digital videos.

7. ACKNOWLEDGEMENTS

We wish to sincerely thank Dr. Yuping Wang of the Cardiovascular Image Analysis Lab at the Washington University School of Medicine for his help with the theoretical aspects of the paper.

8. REFERENCES

[1] T S Chua and L Q Ruan. A Video Retrieval and Sequencing System, *ACM Transactions of Information Systems*, Vol. 13, No. 4, pp. 373-407, 1995.

[2] A Cohen and R D Ryan. *Wavelets and Multiscale Signal Processing*, Chapman and Hall Publishers, 1995.

[3] J M Corridoni and A Del Bimbo. Structured Representation and Automatic Indexing of Movie Information Content, *Pattern Recognition*, Vol. 31, pp. 2027-2045, 1998.

[4] G Davenport, T Aguierre-Smith and N Pincever. Cinematic Primitives for Multimedia, *IEEE Computer Graphics and Applications*, pp. 67-74, July 1991.

[5] C J G Evertsz, K Berkner and W Berghorn. A Local Multiscale Characterization of Edges applying the Wavelet Transform, *Proc. Nato A.S.I, Fractal Image Encoding and Analysis*, Trondheim, July 1995.

[6] A M Ferman and A M Tekalp. Editing Cues for Content-Based Analysis and Summarization of Motion Pictures, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, pp. 71-80, 1998.

[7] A Finkelstein, C E Jacobs and D H Salesin. Multiresolution Video, *Proc. ACM SIGGRAPH 1996*, pp. 281-290, 1996.

[8] A Hampapur, R Jain and T E Weymouth. Production Model Based Digital Video Segmentation, *Multimedia Tools and Applications*, Vol. 1, No. 1, pp. 9-46, 1995.

[9] F Idris and S Panchanathan. Review of Image and Video Indexing Techniques, *Journal of Visual Communication and Image Representation*, Vol. 8, No. 2, pp. 146-166, 1997.

[10] H Jiang, A Helal, A K Elmagarimid and A Joshi. Scene Change Detection Techniques for Video Database Systems, *ACM Multimedia Systems*, Vol. 6, pp 186-195, 1998.

[11] J Lee and B W Dickinson. Multiresolution Video Indexing for Subband Coded Video Databases, *SPIE Conference on Storage and Retrieval for Image and Video Database*, Vol. 2185, pp. 162-173, 1994.

[12] L M Lifshitz and S M Pizer. A Multiresolution Hierarchical Approach to Image Segmentation Based on Intensity Extrema, *IEEE*, 1990.

[13] S Mallat. Multiresolution Approximations and Wavelet Orthonormal Bases of $L^2(\mathbb{R})$, *Trans. of the American Mathematical Society*, Vol. 315, pp. 69-87, 1989.

[14] S Mallat and S Zhong. Signal Characterization from Multiscale Edges, *IEEE Trans. on Pattern Analysis & Machine Intelligence*, Vol. 14, No. 7, pp. 674-693, 1992.

[15] J C Olivo. Automatic Threshold Selection Using the Wavelet Transform, *CVGIP: Graphical Models and Image Processing*, Vol. 56, No. 3, pp. 205-218, 1994.

[16] I K Sethi and N Patel. A Statistical Approach to Scene Change Detection, *SPIE Conference on Storage and Retrieval for Image and Video Database III*, Vol. 2420, pp. 329-338, 1995.

[17] B Shahraray. Scene Change Detection and Content-based Sampling of Video Sequences, *SPIE Conference on Digital Video Compression: Algorithms and Technologies*, Vol.2419, pp. 2-13, 1995.

[18] M K Shan and S Y Lee. Content-based Video Retrieval based on Similarity of Frame Sequence, *IEEE*, 1998.

[19] K Shen and E J Delp. A Fast Algorithm for Video Parsing Using MPEG Compressed Sequences, *Proceedings of the IEEE International Conference on Image Processing*, pp. 252-255, October 1995.

[20] M H Song, T H Kwon, W M Kim, H M Kim and B D Rhee. On Detection of Gradual Scene Changes for Parsing of Video Data, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, pp. 404-413, 1998.

[21] X D Sun, M S Kankanhalli, Y W Zhu and J K Wu. Content-Based Representative Frame Extraction For Digital Video, *Proc. IEEE International Conference on Multimedia Computing and Systems (ICMCS'98)*, pp. 190-193, July 1998.

[22] Y Tonomura, A Akutsu, Y Taniguchi and G Suzuki. Structured Video Computing, *IEEE Multimedia*, Vol. 1, No. 3, pp. 34-43, 1994.

[23] Y P Wang and S L Lee. Scale-Space Derived From B-Spline, *IEEE Trans. on Pattern Analysis & Machine Intelligence*, Vol. 20, No. 10, October 1998.

[24] J Wei, M S Drew and Z N Li. Illumination Invariant Video Segmentation by Hierarchical Robust Thresholding, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, pp. 188-201, 1998.

[25] C S Won, D K Park and S J Yoo. Extracting Image Features from MPEG-2 Compressed Stream, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, pp. 426-435, 1998.

[26] W Xiong and J C M Lee. Automatic Dominant Camera Motion Annotation for Video, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, 1998.

[27] B L Yeo and B Liu. Rapid Scene Analysis on Compressed Video, *IEEE Transactions on Circuits and Systems For Video Technology*, Vol. 5, No. 6, December 1995.

[28] M M Yeung, B L Yeo, W Wolf and B Liu, Video Browsing using Clustering and Scene Transitions on Compressed Sequences, *Proc. Multimedia Computing and Networking*, San Jose, February 1995.

[29] H H Yu and W Wolf. Multi-resolution Video Segmentation using Wavelet Transformation, *SPIE Conference on Storage and Retrieval for Image and Video Database IV*, Vol. 3312, pp. 176-187, 1998.

[30] R Zabih, J Miller and K Mai. A Feature-Based Algorithm for Detecting and Classifying Scene Breaks, *Fourth ACM Multimedia Conference*, pp. 189-200, 1995.

[31] H J Zhang, A Kankanhalli and S W Smoliar. Automatic Partitioning of Full-motion Video, *ACM Multimedia Systems*, Vol. 1, No. 1, pp. 10-28, July 1993.

[32] H J Zhang, C Y Low, Y Gong and S W Smoliar. Video Parsing and Browsing Using Compressed Data, *Multimedia Tools and Applications*, Vol. 1, No. 1, pp. 91-111, 1995.