Multimedia Tools and Applications, 15, 227–245, 2001

Visual Rhythm and Shot Verification

HYEOKMAN KIM hmkim@cs.kookmin.ac.kr
Department of Computer Science, Kookmin University, Songbuk-gu Chungnung-dong 861-1, Seoul 136-702, Korea

JINHO LEE papajino@skylife.co.kr
IT Business Center, Korea Digital Satellite Broadcasting, Jongno-gu Gongpyeong-dong 100, Seoul 110–702, Korea

JAE-HEON YANG jhyang@cs.kaist.ac.kr
Department of Computer Science, KAIST, Yusong-gu Kusong-dong 373-1, Taejon 305-701, Korea

SANGHOON SULL sull@korea.ac.kr
WOONKYUNG M. KIM mwkim@korea.ac.kr
School of Electrical Engineering, Korea University, Songbuk-gu Anam-dong 5 Ga 1, Seoul 136-701, Korea

S. MOON-HO SONG smsong@snu.ac.kr
School of Mechanical and Aerospace Eng., Seoul National University, Kwanak-gu Shinrim-dong 56-1, Seoul 151-742, Korea

Abstract. Typical result of an automatic shot change detection algorithm expectedly includes a certain number of undetected shots as well as falsely detected shots. Even though automatic shot change detection algorithms are continuing to improve, the ultimate goal of automatically detecting all shot changes without false alarms may never be achieved. Thus, allowing a human operator to intervene—to review and verify the result of a shot change detection algorithm, to delete falsely detected shots as well as to find undetected shots—may be the most viable approach currently available for increasing the accuracy of the overall shot detection process. For this exact purpose, we propose a *shot verifier* based on the *visual rhythm*.

The visual rhythm, an abstraction of the video, is a single image, a sub-sampled version of a full video in which the sampling is performed in a pre-determined and in a systematic fashion. It is a representation of the video, which includes the overall content of the video. But most importantly, the visual rhythm contains patterns or visual features that allow the viewer/operator to distinguish and classify many different types of video effects (edits and otherwise): cuts, wipes, dissolves, fades, camera motions, object motions, flashlights, zooms, etc. The different video effects manifest themselves as different patterns on the visual rhythm. Using the visual rhythm, it becomes possible, without sequentially playing the entire video, to find false positive shots as well as undetected shots. Thus, inclusion of the visual rhythm in the shot verification process will aid the operator to verify detected shots as well as to find undetected shots fast and efficiently.

Our newly developed shot verifier based on the visual rhythm has been designed for operator efficiency. The design of our shot verifier presented and the usefulness of the visual rhythm during the shot verification process will be demonstrated.

Keywords: shot verification, visual rhythm, video edit effects

1. Introduction

With the advancement of digital video applications, *video indexing* is now becoming an integral part of multimedia services. The services involve a wide variety of areas such as video editing, digital library, video database, video contents production, streaming video on the web, etc. Since video is a flat sequence of frames with no other indications, it has only limited interactivity, typically found in modern VCRs: fast-forward, rewind, pause and play. For advanced video applications, an interpretation level above the raw video information is needed, for instance to parallel those found in text applications, such as sentences, paragraphs, chapter titles and table of contents. Video indexing is a mechanism that provides such syntax and structure supplying users with entry points in video. The user is then able to quickly access a particular part of the video. In addition, the indexing mechanism can be exploited to provide a summary of the entire video so that the user may become quickly acquainted with the video content. The first step in this important video indexing procedure is to detect shot and scene changes in the video. The video content indexing becomes possible only after accurate shots have been detected to represent the entire video with certain syntax for efficient user access.

Before proceeding with details, we note that the terminology used in the video indexing literature vary somewhat. Thus, we precisely define the following terms, which will be consistently used throughout the paper. A *shot* is an unbroken sequence of frames, which present a continuous action; it usually results from a single operation of the camera. In other words, it is a sequence of frames that is generated by the camera from the time it begins recording until the time it stops [3]. A *scene* is defined as a collection of one or more adjoining shots that focus on an object or objects of interest [1]. In general, shots have a duration of 1–10 seconds, but the duration of scenes vary greatly depending on the genre of videos from TV commercials to soap operas.

Manual detection of shot and scene changes is extremely tedious and naturally time-consuming. In addition, such manual detection will be prone to errors, due to operator fatigue. Therefore, it is essential to develop and utilize automatic methods for the detection and the multimedia research community is actively pursuing methods for better automatic detection of shots and scenes [3–6, 8–11].

In general, the shot change detection step (or segmentation) precedes the scene change detection step (or clustering). In addition, during shot change detection, one or more key frames are usually obtained automatically for each shot. Therefore, the accuracy of the shot change detection algorithm will have a great influence on the performance of the scene change detection algorithm. If the detected shot changes are inaccurate, then the resulting scene changes will also be inaccurate. As important as accurate shot change detection maybe, as Bayesian decision theory points out, there is always a trade-off between the detection probability and the false alarm rate. Thus, it is only natural to assume that it is impossible to automatically detect all shots of a video accurately. In addition, certain video editing effects produce frames so that even a visual determination of scene/shot changes becomes difficult.

Most shot change detection algorithms result in many *false positives* and *missing shots*. For accurate detection of all shots in the video, it is essential to manually remove false

positives and find missing shots (as detected by an automatic algorithm), the process we call *shot verification*. A robust shot change detection system must include a shot verification step to further enhance the overall system performance. In this paper, we introduce the concept of the *visual rhythm*, an abstraction of a video. The visual rhythm is a single image, a sub-sampled version of a full video in which the sampling is performed in a pre-determined and systematic way. It is basically a representation of the video, which includes the overall content of the video. But most importantly, a visual rhythm includes visual features that allow the operator to distinguish and classify many different types of video effects (edits and otherwise): cuts, wipes, dissolves, fades, camera motions, object motions, flashlights, as well as zooms. These different video effects manifest themselves as different patterns on visual rhythms. Using the visual rhythm, it becomes possible, without sequentially playing the entire video, to find false positive shots as well as undetected shots. Thus, inclusion of the visual rhythm in the shot verification process will aid the operator to verify the shots fast and efficiently. For this purpose we have developed a new tool, a shot verifier including the visual rhythm. The usefulness of the visual rhythm in the shot verification process will be presented.

The paper is organized as follows. Section 2 reviews the current state-of-the-art shot detection methods, and leads to the importance of shot verification. Section 3 formally introduces the concept of visual rhythm and shows its features and characterizations for different video effects. Our design of a shot verifier is presented in Section 4 with some of its major functions and user interfaces. Lastly, Section 5 concludes the paper with certain merits and limitations of the proposed shot verifier.

2. Edit effects and shot verification

The process of making a video involves production and editing of individual shots. The video editor assembles different shots together in a continuous stream, which becomes the final video. During this editing process, a number of different effects are introduced at the boundary of the shots. These effects are typically referred to as *edit effects*. Thus, the problem of shot change detection can be posed as edit effects detection [3]. The most frequently used edit effects are *cuts*, *dissolves* and *wipes*. A cut is simply a concatenation of two shots. In other words, no special effects are introduced. A dissolve is a simultaneous application of a fade-out (of an outgoing shot) and a fade-in (of an incoming shot). A wipe is another edit effect where an incoming shot appears at one or more parts of an outgoing shot, and then grows gradually until it covers the entire frame. Cuts are typically classified as an abrupt change where dissolves and wipes are classified as gradual changes, because during dissolves and wipes, shot changes occur gradually over many contiguous video frames. Modern digital video editors provide, not only two-dimensional effects mentioned above, but also three-dimensional gradual effects such as rolling, flying, doppler effects, etc.

Abrupt changes have a clear boundary since the last several frames of an outgoing shot and the beginning ones of an incoming shot usually have different contents. Thus, abrupt changes can be detected fairly easily by applying simple methods such as pixel differences, statistical differences, histograms, motion vectors and so on. However, a sequence of frames with gradual effects has a visually indistinguishable boundary since each frame in the boundary has visual characteristics of both outgoing and incoming shots. Therefore, it is

much more difficult to detect gradual changes. Perhaps, for this reason much work has been reported for the detection of abrupt changes [5, 6, 8–11], but only a few for gradual changes [3, 4]. We note in passing that even abrupt changes may be missed in certain video contexts, for instance, shots with diverse applications of camera motions such as zooming and panning, shots with fast moving objects, quick changes in luminance, etc. This illustrates the importance of a shot verifier.

Boreczky et al. [1] analyzed various genres of test videos and summarized the frequency of different edit effects. According to their analysis, "more than 30% of those scene changes were marked by gradual changes, whereas just over 10% of non-scene shot changes were marked by gradual changes." Thus, without shot verification, if scene change detection (clustering) step is executed without the correct shot changes (including gradual changes), more than 30% of the scenes may not be found correctly. This percentage is large enough for the detection of gradual changes to be included as an important part of any shot and scene change detection algorithm. Although most automatic shot change detection methods report about 10–20% false positive and/or missing shots [1], the reported detection rate would have been much worse had the test videos included a diverse range of gradual changes, such as those typically found in commercials. This further illustrates the need for an efficient shot verifier.

For a straight forward manual shot verification, one has to compare each shot detected by an automatic algorithm using the raw video stored on disk or tape. During this verification process, the raw video must be played back and carefully viewed, using VCR functions, such as slow motion, pause, fast-forward, etc. Note that the operator will be required to playback the same video frames repeatedly, over where the shot changes occur. The amount of time required for such verification process may be just as much as a manual shot change detection. Therefore, the shot verification process must be done quickly, with a smartly designed graphical tool. Our newly developed tool, based on the proposed visual rhythm, serves this need.

3. Visual rhythm

3.1. Visual rhythm and pixel sampling

We have established that viewing the entire video for shot verification is too time consuming. If one chooses to verify shots in such a way, one might as well detect the shots by playing the entire video in the first place. Therefore, the shot verifier, although manual, must be designed for a quick and accurate operation.

For the shot verifier to be quick, we resort to viewing a portion of the original video. This portion of the video must retain most, if not all, video edit effects. We claim that our visual rhythm, defined below, satisfies the requirement. The user, upon viewing the visual rhythm, must be able to distinguish different video edit effects and verify the detected shots and find new missing shots quickly and efficiently. We note that the terminology visual rhythm was first used in the context of video icons [12], where it was defined to be the vertical or horizontal sides of the video icon. This idea is generalized to sample any collection of pixels from a frame (e.g., pixels along the diagonal) across time.

Let $f_V\ (x, y, t)$ be the pixel value at location (x, y) and time t of an arbitrary video V. Then, the video V may be represented as:

$$V = \{f_V(x, y, t)\}, \quad x, y, t \in \{0, 1, 2, \ldots\}.$$

Let $f_{Thumbnail}\ (x, y, t)$ be the representation of a reduced frame of a *spatially reduced video* $V_{Thumbnail}$ of the original video V. Each reduced frame, or thumbnail, is a (horizontally and vertically) reduced image of its corresponding frame in the video V by a factor r. Thus, the spatially reduced video or sequence of thumbnails may be expressed as:

$$V_{Thumbnail} = \{f_{Thumbnail}(x, y, t)\}, \quad x, y, t \in \{0, 1, 2, \ldots\}.$$

The relationship between the video V and its spatially reduced video $V_{Thumbnail}$ can be represented using their pixel correspondences as follows

$$f_{Thumbnail}(x, y, t) = f_V(rx + k_x, ry + k_y, t), \quad x, y, t \in \{0, 1, 2, \ldots\},$$
$$0 \leq k_x, k_y \leq r - 1$$

where k_x and k_y are offsets in pixel units. Using the spatially reduced video, we define the *visual rhythm*, *VR*, of the video V as follows:

$$VR = \{f_{VR}(z, t)\} = \{f_{Thumbnail}(x(z), y(z), t)\}$$

where $x(z)$ and $y(z)$ are one-dimensional functions of the independent variable z. Thus, the visual rhythm is a two-dimensional image consisting of pixels sampled from a three-dimensional data (video). That is, the visual rhythm is constructed by sampling a certain group of pixels in each thumbnail and by temporally accumulating the samples along time. Thus, the visual rhythm is a two-dimensional abstraction of the entire three-dimensional video content.

Depending on the mappings $x(z)$ and $y(z)$, various types of visual rhythms can be obtained. For an arbitrary plane, for instance, $y = y_0$ or $x = x_0$, horizontal or vertical pixel sampling can be done by applying $(x(z), y(z)) = (z, y_0)$ or $(x(z), y(z)) = (x_0, z)$, respectively. For a constant α, diagonal pixel sampling is achieved by the application of $(x(z), y(z)) = (z, \alpha z)$. Figure 1 shows several different sampling strategies, a partial list of all realizable mappings.

The sampling strategies, $x(z)$ and $y(z)$, must be carefully chosen for the visual rhythm to retain the edit effects that characterize shot changes. Figure 2 shows three visual rhythms constructed from an identical video clip with different sampling strategies. The first four abrupt changes as well as the final wipe are readily visible on all visual rhythms. However, the visual features for shot changes are most pronounced in the diagonal sampling as shown in figure 2(a). The visual rhythm obtained by the cross sampling, shown in figure 2(b), is marked by a single horizontal line through the center, along which the visual rhythm has been effectively divided into upper and lower halves. These independent halves may sometimes be too narrow for a correct reading. Lastly, the area sampling shown in figure 2(c) preserves the overall characteristics of the video. However, it consists of many undesirable

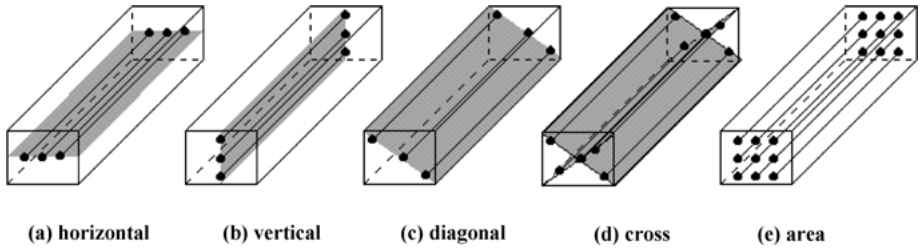


Figure 1. Pixel sampling.

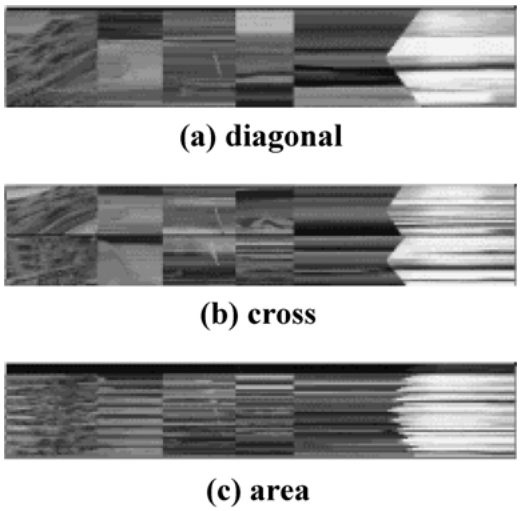


Figure 2. Visual rhythms by different sampling schemes.

horizontal lines. Our experience on reading these visual rhythms, indicate that diagonal sampling of figure 2(a) provides the best visual features for distinguishing various video edit effects. We will discuss the details of these visual features in Section 3.3. All visual rhythms presented hereafter are generated using the diagonal sampling.

3.2. Fast generation of visual rhythms

If the digitized video is in its raw form, the construction of its corresponding visual rhythm will be trivial. However, the digital video is usually compressed due to its huge volume. But, because many compression schemes use the discrete cosine transform (DCT) for intra-frame encoding, fast generation of the visual rhythm becomes possible. To be more specific, Motion JPEG and I-frames of MPEG use the DCT for intra-frame encoding, where the frames partitioned into 8×8 blocks are processed. Each block is transformed to the frequency domain representation resulting in one DC and 63 AC coefficients. Thus, the extraction of the DC coefficients can be performed fast, without fully decoding the

compressed video. Since the DC coefficient is actually the average pixel value of the block (more precisely, eight times the average, since the transforms are defined as unitary), the collection of these DC coefficients over the frame can serve as our thumbnail images. Noting this fact, Yeo and Liu [9] introduce the *DC image* which consists of the DC coefficients of the original frame, and a sequence of DC images called the *DC sequence*.

By accepting the DC image as a thumbnail and the DC sequence as a spatially reduced video, the thumbnail and the original video are related as shown below:

$$f_{Thumbnail}(x, y, t) = \frac{1}{8} \sum_{k_x=0}^{7} \sum_{k_y=0}^{7} f_V(8x + k_x, 8y + k_y, t), \quad x, y, t \in \{0, 1, 2, \ldots\}.$$

Thus, the construction of the visual rhythm is possible without the inverse DCT. We simply extract the DC coefficients by sampling (the diagonal pixels of) the DC sequence. As for the P- and B-frames of MPEG, algorithms for determining the DC images from inter-frame compressed P- and B-frames of MPEG1 [9] and MPEG2 [7] have already been developed. Therefore, it is possible to generate visual rhythms fast, at least for the DCT-based compression schemes, such as Motion JPEG and MPEG videos.

3.3. Characteristics of visual rhythm and shot verification

The visual rhythm, an abstraction of a video, is itself a two-dimensional image. It is a depiction of the entire video, which retains visual features of shot changes. In a visual rhythm, pixels along a vertical line are those pixels uniformly sampled along the diagonal of a frame. The vertical lines of a visual rhythm will have similar visual features if the lines are from the same shot. The lines will have different visual features if they belong to different shots. Thus, if there is a shot change, shot boundaries will become apparent, as visual features of the vertical lines will change. Note that the terminology *visual rhythm* is quite appropriate as it allows "a user to easily see variations in video content without actually playing the video sequentially" [12], corresponding to *rhythm* of the music.

We will now discuss how different shot changes result in different visual features on visual rhythms. The discussion will illustrate that shot changes as well as different camera and object motions can be clearly identified visually. Based on these characteristics of the visual rhythm, it then becomes possible to verify the detected shot changes from an automatic detection algorithm without playing the entire video repeatedly and sequentially.

Figure 3 shows some typical visual rhythm patterns arising from different types of edit effects. Figure 3(a) shows the expected pattern on the visual rhythm for an abrupt change or a cut. Note that a cut will appear as a *vertical line* on the visual rhythm. If two adjacent vertical lines of a visual rhythm belong to different shots, their visual difference leads to a vertical line right at the shot boundary as indicated in figure 3(a).

Figure 3(b) shows a left-to-right horizontal wipe. Note that this particular wipe appears as an *oblique line*. Top-to-bottom vertical wipe will also have the same pattern. Both right-to-left horizontal and bottom-to-top vertical wipes will also appear as an oblique line, but with opposite slope. However, center-to-outskirts expanding wipe will appear as a *curved line*, as shown in figure 3(c). If an incoming shot is uniformly expanded in time, the wipe will appear

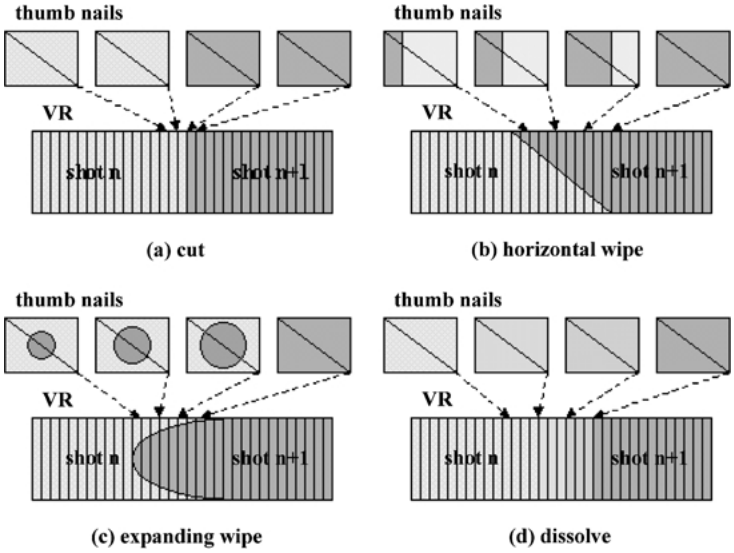


Figure 3. Illustration of edit effects on visual rhythms.

as two oblique lines which have opposite slopes and meet at the center. Outskirts-to-center absorbing wipe will similarly appear as a curved line, except in the opposite direction. The important point is that all wipes will generate lines from top to bottom. And if this line is over a range of frames, instead of a vertical line over a single frame, then the shot change is probably due to a wipe. Obviously, a straight vertical line over a single frame implies an abrupt change.

The dissolves, on the other hand, do not generate any distinguishable lines on the visual rhythm, since during dissolve, every pixel will have characteristics of both outgoing and incoming shots. However, the dissolving frames do have a visual feature and will show linearly increasing or decreasing luminance and chrominance values as indicated in figure 3(d).

Figure 4 shows three visual rhythms generated from real video clips. As explained earlier, and indicated on the figure, edit effects such as cuts, wipes and dissolves (including fades) can easily be identified from these visual rhythms. Furthermore, camera motions such as zoom-ins and zoom-outs, flashlights, object motions can also be identified visually. The camera and object motions are major causes for false detection and therefore such visual rhythms will serve the purpose of eliminating false positives during the shot verification process.

As mentioned earlier, we prefer diagonal sampling over other sampling approaches. This is partially, if not entirely, due to wipes as we have observed that diagonal sampling results in visually distinctive patterns on visual rhythms for most commonly used wipes. The following experiment will indicate that diagonal sampling will result in a least number of misreadings.

There are hundreds if not thousands of wipes that modern digital video editors can produce, and a user of the editor may still create an entirely different wipe. However,

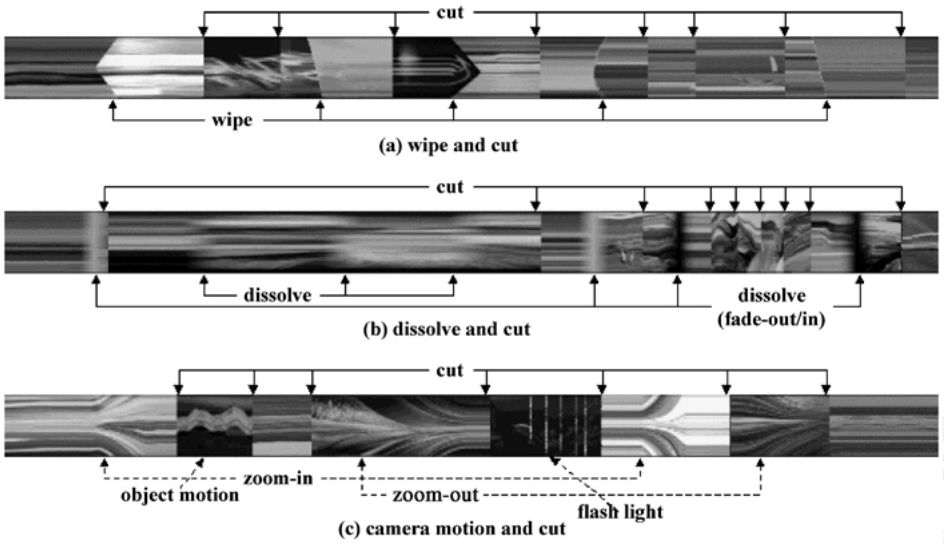


Figure 4. Edits and other effects on visual rhythm.

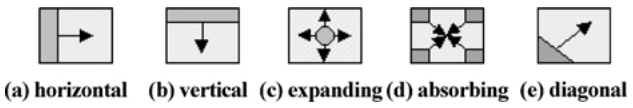


Figure 5. Types of wipes.

most wipes can be categorized into horizontal, vertical, expanding, absorbing, diagonal and miscellaneous types. Figure 5 shows these five types of wipes most frequently used during normal video editing.

As previously described, detection of wipes essentially requires the operator to detect lines that run from the top to the bottom of the visual rhythm. Table 1 summarizes different manifestations of the five wipes on visual rhythms for different sampling strategies: horizontal, vertical and diagonal samplings. Each entry in the table describes the different visual characteristics in the visual rhythm, due to various combinations of wipes and sampling strategies. For instance, for horizontal sampling, vertical and absorbing wipes will show vertical lines on the visual rhythm, which then be misread as a cut. Table 1 shows that the diagonal sampling leads to the least amount of such misreadings. Note that even though Table 1 shows that diagonal sampling will result in misreading a diagonal wipe as a cut, if this wipe had been a diagonal wipe along the other diagonal, it will then have generated the visual rhythm with an oblique line. Then, it would have resulted in a correct reading. In any case, from the viewpoint of detecting shots, misreadings are much better than missing shots.

Figure 6(a) through (f) shows three different real wipe and animation sequences respectively. The corresponding visual rhythm is shown in figure 6(g). Note the incidental cuts, or abrupt changes, on the visual rhythm. As expected, abrupt changes appear as a straight

Table 1. Pixel samplings and visual rhythm.

	Horizontal wipe	Vertical wipe	Expanding wipe	Absorbing wipe	Diagonal wipe
Horizontal sampling	Oblique line	Vertical line (cut)	Curved line	Vertical line (cut)	Oblique line
Vertical sampling	Vertical line (cut)	Oblique line	Curved line	Vertical line (cut)	Oblique line
Diagonal sampling	Oblique line	Oblique line	Curved line	Curved line	Vertical line (cut)

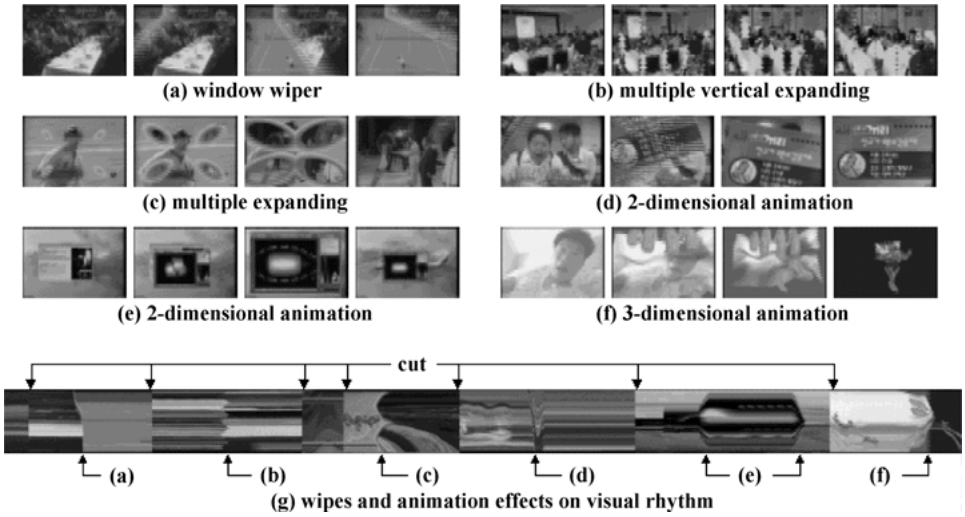


Figure 6. Miscellaneous wipes and animations.

vertical line on the visual rhythm. The "window wiper," shown in figure 6(a), results in a visually distinguishable line on the visual rhythm. The line begins slanted which then appears to become vertical towards the bottom. As the line is not a straight vertical line, we may infer that it is due to a wipe. Both expanding and absorbing wipes, shown in figure 6(b) and (c), also result in visually distinguishable lines (or curves) that run from the top to the bottom of the visual rhythm. Note that both 2-D and 3-D effects, shown in figure 6(d) through (f), also result in similar lines that run from the top to the bottom. What is common here is that all wipes and wipe-like effects including 2-D and 3-D animation (as well as cuts) manifest themselves as visually distinguishable lines (or curves) that run from the top to the bottom. Therefore, the operator, upon observation of any such pattern, may zoom-in towards the frames corresponding to that pattern. The operator may then view and/or play a certain number of frames surrounding those frames for further assessment. Our proposed shot verifier, with smartly designed graphical user interface allows the operator for such operation. The details of the proposed shot verifier will be presented in the next section.

4. Shot verifier

4.1. Functions of shot verifier

The design of our shot verifier incorporates the visual rhythm to aid the operator to quickly verify detected shots as well as to find new ones. The shot verifier is expected to run after the automatic shot detector returns with a list of detected shots. However, it may also be used by itself to find shots manually. In either case, incorporation of the visual rhythm in the shot verifier will speed to process. Our shot verifier has following attributes:

- Visual rhythm which express various video effects and shot boundaries
- Frame-accurate display controls
- VCR functions
- Addition and deletion of shot boundaries

We have implemented the shot verifier for Motion JPEG compressed video with the above mentioned features. The incorporation of the visual rhythm into the shot verifier greatly enhances the operator efficiency. One quick glance at the visual rhythm allows visual determination of shot boundaries, as different edit effects will appear as various lines on visual rhythms. In addition, The visual rhythm can be displayed with time scaling. In other words, the visual rhythm can be subsampled along time. Figure 7 shows that a visual rhythm of an eight minutes video clip can be displayed in a single window with most of its visual characteristics in tact.

4.2. Verification processes

Figure 8 shows the graphical user interface of our shot verifier. With this shot verifier, by viewing the visual rhythm and using frame-accurate controls, the operator determines the frames that appear to be candidates for false positives and/or undetected shot boundaries.

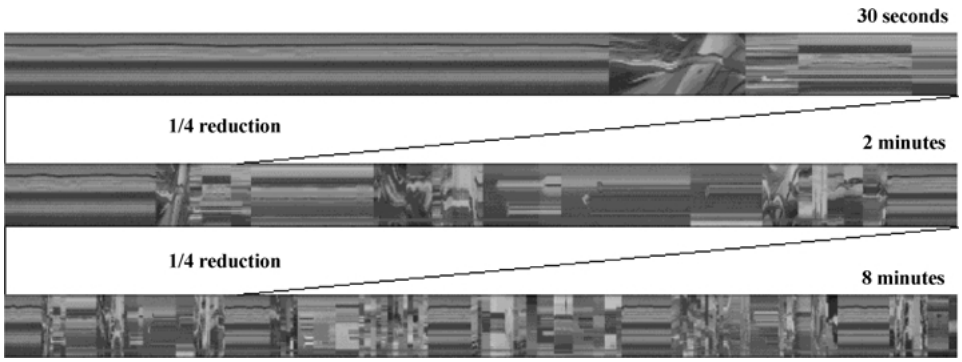


Figure 7. A visual rhythm at various time resolutions.

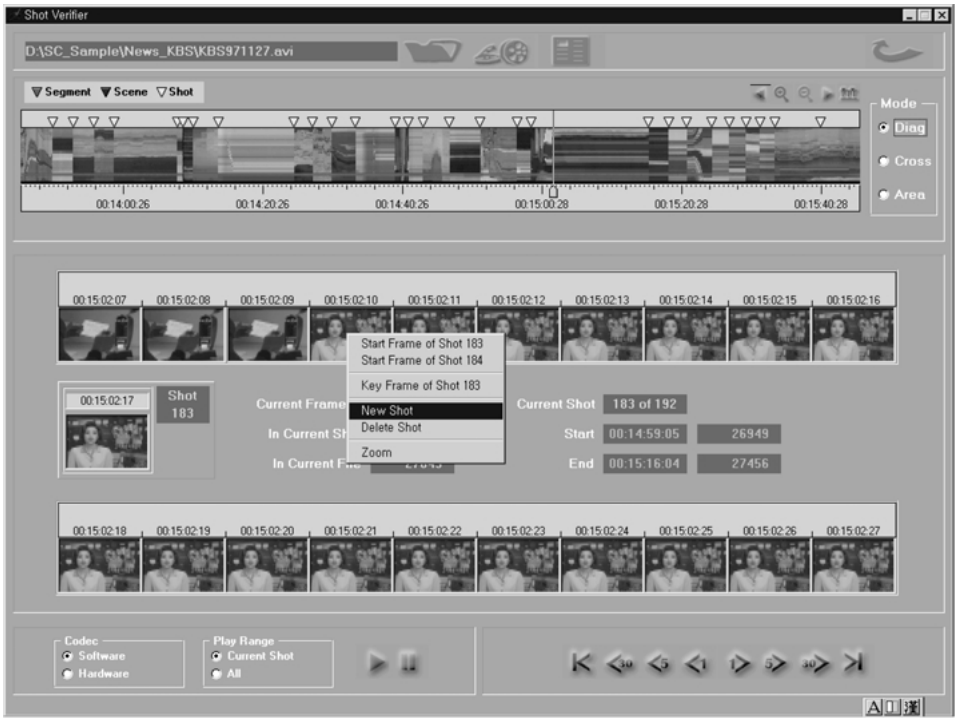


Figure 8. User interface of shot verifier.

The shot verifier provides a sequence of 90×60 thumbnails to aid the operator for such operations. These thumbnails are sixty-four times smaller than the original 720×480 frames. The shot verifier can also provide the original frames at the user request. In addition, it provides VCR functions for shot-by-shot playing and frame-accurate random positioning. With these frame-accurate controls and VCR functions, the operator can manually delete false positives and add newly found shots. The operator navigates the three major parts of the interface: the visual rhythm along with certain markers, the list of thumbnails surrounding the current frame, and the frame accurate play controls. We describe these individual parts in detail.

At the top of the interface, the visual rhythm is displayed with automatically detected shot boundaries marked at their corresponding locations with small inverse triangles or *shot boundary markers*. On the upper right corner of the visual rhythm, small buttons for zooming (enlarging and reducing) and scrolling the visual rhythm is provided. On top of a visual rhythm, a cursor is provided to indicate the current position (time code or frame number). By dragging and dropping the cursor at any position of the visual rhythm, the user can change the location of the current frame.

At the middle of the interface, the thumbnail list of twenty-one frames surrounding the current frame is displayed. At the center of the list, showing "Shot 183," is the current

frame, as indicated by the cursor on the visual rhythm. Other thumbnails, ten above and ten below, are the frames immediately preceding and following the current frame, respectively.

At the bottom of the interface, various options/buttons for frame accurate controls are provided. For instance the operator is able to move the current frame by 1, 5, 30 frames forward or backward. Recall that the current frame can also be updated by moving the cursor on the visual rhythm. In any case, such current frame update results in redisplay of the thumbnail list. When the play button is selected, a new window appears to play from the current frame. The "Play Range" option indicates whether to play the entire video or the current shot. Currently, no fast forward nor rewind buttons are provided as we have provided the random positioning function.

Following the automatic shot detector, the shot verification process may proceed as follows. If a certain frame appears suspicious, upon viewing the visual rhythm, the cursor is moved near that frame. For instance, figure 8 shows the visual rhythm where there appears to be a shot boundary without the boundary marker (inverse triangle). Therefore, the cursor has been moved near that frame by the drag and drop operation. Then, twenty-one thumbnails are redisplayed according to this new current frame. Quick glance over the thumbnail display indicates that the fourth thumbnail is actually the first frame of a new shot, missed by the automatic detector. False positive shots are detected similarly. In any case, when detected, deletion and/or addition of shot boundaries is processed with a left mouse button click, at which a small menu appears as shown in figure 8. The menu is self-explanatory; it basically provides the mechanics for deleting and adding shot boundaries. Such updates to shot boundaries result in redisplay of the markers (inverse triangles) on top of the visual rhythm at the top of the interface. For instance, the user may select "New Shot" to insert a missing shot or "Delete Shot" to discard a false positive. The user may also select the "Key Frame" for the current shot (183). Other interface is also available for clustering various shots in segments and scenes.

To summarize, our shot verifier fully utilizes the visual characteristics of the visual rhythm. The visual rhythm contains many visually distinguishable characteristics for various types of video effects, edits and otherwise. Therefore, the user of the shot verifier can quickly view the visual rhythm along with detected shots. When in doubt the user displays nearby frames using frame accurate controls. Since this process is relatively quick and in practice, the result of an automatic shot change detection for an hour long video can be verified within minutes. This tool is implemented with C and Delphi under Windows NT.

5. Summary

Manual shot verification is the process of correcting the shot boundaries detected by an automatic detection algorithm. In other words it strives to remove false positive shots and to find missing shots manually. If automatic shot detection algorithms can find all shot boundaries without any false positives and undetected shots, there would be no need for such a verification process. However, it is impossible for such perfect detection and one is always faced with a trade-off between the detection and false-alarm rates. Thus, the process of verifying detected shots must be included for accurate video indexing. In this paper, the

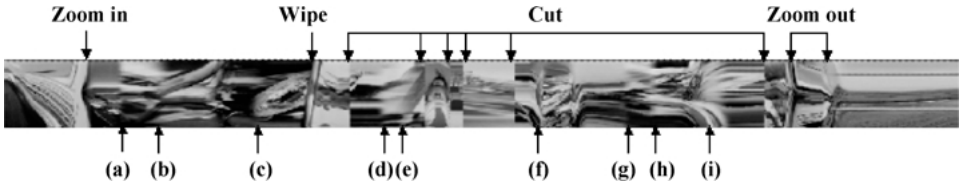


Figure 9. Visual rhythm of a commercial clip. (a) through (i) are dissolves.

concept of visual rhythm is introduced which has led to a fast and easy to operate shot verifier. The operation of our shot verifier is discussed with a typical scenario for deleting and adding shot boundaries. The operator efficiency is greatly increased by the incorporation of the visual rhythm into our shot verifier.

However, not all shot boundaries are clearly visible on visual rhythms. Figure 9 shows an example visual rhythm with a wipe, several cuts and nine dissolves. The clip also has several camera zooms. Note that camera zooms maybe mistakenly determined as wipes. In addition, the dissolves—marked (a) through (i)—are difficult to visually distinguish. Although (a), (b), (d), (g), (h) and (i) are somewhat recognizable, others are extremely difficult to recognize as dissolves. As a reference, figure 5 shows thumbnails of the corresponding dissolving frames. Even though there are cases where shot boundaries are not clearly visible on the visual rhythm, the user, when in doubt, can always review the doubtful frames of the video via the random access capability using the visual rhythm itself. Because, the visual rhythm, in addition to providing visually distinguishable patterns for the frames where shot boundaries occur, also provides random entry points into the video.

If a user wishes to view a particular frame with certain color characteristics, the user, upon viewing the visual rhythm, can simply move the cursor on the visual rhythm. Once the cursor is moved, 21 thumbnails about the selected frame will be displayed which can then be used to quickly determine the nature of the shot. This function of the visual rhythm may also be applied to application areas such as video editing, browsing, indexing and modeling. We are actively implementing a complete video indexer, which includes capabilities such as automatic shot detection, shot verification, hierarchical modeling and browsing. All these capabilities will be based on the visual rhythm. Currently, our shot verifier only handles Motion JPEG compressed video, but we are currently extending the verifier to handle MPEG1 as well as MPEG2.

We have also developed a new automatic shot detection algorithm using our visual rhythm [4]. The accuracy of our automatic algorithm is similar to other published algorithm for abrupt changes. However, for gradual changes, our algorithm is able to detect almost all conventional wipes and most dissolves with small amounts of motion. In any case, there are no known algorithms to find every shot correctly without false positives. Thus, the need for a shot verifier cannot be overemphasized and this precisely has been the reason for our development and presentation of a shot verifier, a tool to systematically verify the results of an automatic shot change detection. Our current plan is to integrate this automatic shot detection algorithm with the proposed shot verifier, at which time a quantitative evaluation of the integrated system will be performed.

Figure 10. Thumbnails of corresponding dissolves in figure 9.

Acknowledgments

We would like to thank the reviewers for their comments and suggestions. The work of H. Kim has been supported in part by the University Research Program supported by Ministry of Information and Communication in South Korea.

References

1. J.S. Boreczky and L.A. Rowe, "Comparison of video boundary detection techniques," in Proc. of SPIE Storage and Retrieval for Image and Video Database IV, SPIE, 1996, Vol. 2670, pp. 170–179.
2. G. Davenport, T.A. Smith, and N. Pincever, "Cinematic primitives for multimedia," IEEE Computer Graphics and Applications, pp. 67–74, July 1991.
3. A. Hampapur, R. Jain, and T. Weymouth, "Digital video segmentation," in Proc. of ACM Multimedia '94, 1994, pp. 357–564.
4. H. Kim, S.-J. Park, J. Lee, W.M. Kim, and S.M. Song, "Processing of partial video data for detection of wipe," in Proc. of SPIE Storage and Retrieval for Image and Video Databases VII, SPIE, 1999, Vol. 3656, pp. 280–289.
5. J. Meng, Y. Juan, and S. Chang, "Scene change detection in a MPEG compressed video sequence," in Proc. of SPIE Digital Video Compression: Algorithms and Technologies, SPIE, 1995, Vol. 2419, pp. 14–25.
6. Y. Nakajima, K. Ujihara, and A. Yoneyama, "Universal scene change detection on MPEG-coded data domain," in Proc. of SPIE Visual Communication and Image Processing, SPIE, 1997, Vol. 3024, pp. 992–1003.
7. J. Song and B.-L. Yeo, "Spatially reduced image extraction from MPEG-2 video: fast algorithms and application," in Proc. of SPIE Storage and Retrieval for Image and Video Database VI , SPIE, 1998, Vol. 3312, pp. 93–107.
8. S.M. Song, T.-H. Kwon, W.M. Kim, H. Kim, and B.-D. Rhee, "On detection of gradual scene changes for parsing of video data," in Proc. of SPIE Storage and Retrieval for Image and Video Database VI, SPIE, 1998, Vol. 3312, pp. 404–413.
9. 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, pp. 533–544, 1995.
10. M.M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in Proc. of SPIE Multimedia Computing and Networking, SPIE, 1995, Vol. 2417, pp. 399–413.
11. H. Zhang, K. Kankanhalli, and S. Smoliar, "Automatic partitioning of full-motion video," ACM Multimedia Systems, Vol. 1, No. 1, pp. 10–28, 1993.
12. H. Zhang and S.W. Smoliar, "Developing power tools for video indexing and retrieval," in Proc. of SPIE Storage and Retrieval for Image and Video Database II, SPIE, 1994, Vol. 2185, pp. 140–149.

Hyeokman Kim received his B.S., M.S., and Ph.D. degrees in computer Engineering from the Department of Computer Engineering, Seoul National University, Seoul, Korea in 1985, 1987 and 1996 respectively. From 1996 to 1999, he was engaged in research on digital video library for the Multimedia Technology Research Laboratory,

Korea Telecom where he was a senior member of technical staff. Currently, he is an assistant professor of the department of Computer Science at the Kookmin University, Seoul, Korea. His research interests include digital library, video database and modeling, xml database, digital videos and their applications on the web.

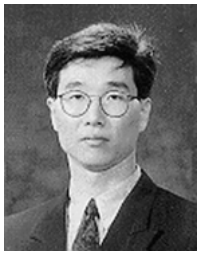
Jihno Lee received B.S. and M.S. degrees in computer Engineering from the Department of Computer Science, Soongsil University, Seoul, Korea in 1990 and 1993 respectively. From 1996 to 1999, he was engaged in research on digital video library for Korea Telecom where he was a member of technical staff. Currently, he is working for Korea Digital Satellite Broadcasting where he is a director of data business team. His research interests include digital library, video database and modeling, Internet broadcasting and their applications on the web.

Jae-Heon Yang received the B.S. and M.S. degrees in Computer Engineering from Seoul National University in 1985 and 1987, respectively, and the Ph.D. degree in Computer Science from the University of Maryland at College Park in 1994. Since July 1996, he has been an Assistant Professor of Computer Science at KAIST in Taejon, Korea. Prior to joining KAIST, he was an Assistant Professor of Computer Science at Mills College in Oakland, California, U.S.A.. His research interests include distributed multimedia, network protocols, and operating systems.

Sanghoon Sull received the B.S. degree with honors in electronics engineering from the Seoul National University, Korea, in 1981, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Korea, in 1983, and the Ph.D. degree in electrical and computer engineering from the University of

Illinois, Urbana-Champaign, in 1993. In 1983–1986, he was with the Korea Broadcasting Systems, working on the development of the teletext system. In 1994–1996, he conducted a research on motion analysis at the NASA, Ames Research Center. In 1996–1997, he conducted a research on video indexing/browsing and was involved in the development of the IBM DB2 Video Extender at the IBM, Almaden Research Center. He joined the School of Electrical Engineering at the Korea University as an Assistant Professor in 1997 and is now an Associate Professor. His current research interests include multimedia data management including search/browsing. MPEG-7, image processing, and Internet applications.

Woonkyung Michael Kim received two B.S. degrees in Electrical Engineering and Mathematics from Massachusetts Institute of Technology in 1985, one B.S. degree in Computer Science from Boston University in 1986, one M.S. degree in Engineering Sciences from Harvard University in 1993, and one Ph.D. degree in Engineering Sciences from Harvard University in 1994. His work towards an M.S. degree in Computer Science from Boston University was halted in 1994 when he joined the Artificial Intelligence Lab (Department of Cognitive Sciences) at Massachusetts Institute of Technology as a Postdoctoral Fellow. He has briefly interned at Electronic Telecommunications Research Institute (ETRI) in 1997. He now holds an Associate Professorship at the School of Electrical Engineering at Korea University where his current theoretical interests include Communications (CDMA and PN Codes), Multimedia (MPEG-X), Morphological Signal Processing, Virtual Reality and Intelligent Systems. He heads the three laboratories–Communications Signal Processing Lab, Multimedia Information Communication Lab, and Intelligent System Lab– on campus at Korea University which specialize in finding further applications of theory in Internet, GIS, ITS and IMT-2000. He also serves as a director at Multimedia Wiz, an Internet Video Infrastructure company in Irvine, California, that he helped found. In addition to a couple of patents already pending through the current work, he is in the process of filing a number of patents on Multimedia Delivery and Processing Systems. He is a member of IEEE and IS&T.



S. Moon-Ho Song received the B.S. degree from the Massachusetts Institute of Technology in 1982, the M.S. degree from the University of California, Los Angeles, in 1985, and the Ph.D. degree from the University of Southern California in 1991, all in electrical engineering. He was with Litton Guidance and Control Systems from 1982 to 1983, where he worked on developing various inertial navigation systems. From 1983 to 1991, he was a member of the technical staff at Hughes Aircraft Company, where he developed radar signal processing algorithms and software. From 1992 to 1993, he was a research associate at the Richard M. Lucas Magnetic Resonance Imaging and Spectroscopy, Stanford University, where he developed new image processing techniques

for MR and CT applications. From 1994 to 1995, he served on the faculty of the Department of Radiology at the University of California, San Francisco. From 1995 to 2001, he was on the faculty of the School of Electrical Engineering at Korea University, Seoul, Korea. In 2001, he joined the faculty of the School of Mechanical and Aerospace Engineering at Seoul National University, Seoul, Korea, where he is currently an associate professor. His current research interests include vision and visualization techniques for multimedia and medical applications, numerical algorithms, signal processing, and non-destructive testing using tomography. Dr. Song holds three US patents and has several other patents pending. He is a member of IEEE, IS&T, IEEK, KICS and Eta Kappa Nu.