

# CycleStack: Inferring Periodic Behavior via Temporal Sequence Visualization in Ultrasound Video

Teng-Yok Lee\*

The Ohio State University

Abon Chaudhuri†

The Ohio State University

Fatih Porikli‡

Mitsubishi Electric Research Laboratories

Han-Wei Shen§

The Ohio State University

## ABSTRACT

A range of well-known treatment methods for destroying tumor and similar harmful growth in human body utilizes the coherence between the inherently periodic movement of the affected body part and periodic respiratory signal of the patient, with the objective of minimizing damage to surrounding normal tissues. Such methods require constant monitoring by an operator who observes the 3D body motion via its 2D projection onto an ultrasound imaging plane and studies the synchronism of this motion with the respiratory signal. Keeping an attentive eye on the respiratory signal as well as the ultrasound video for the entire treatment period is often inconvenient and burdensome. In this paper, we propose a video visualization technique called *CycleStack Plot* which reduces this cognitive overhead by blending the video and the signal together in a stack-like layout. This visualization reveals the inherent synchronism between the target's movement and the respiratory signal, visually highlights significant phase shifts of either of the two cyclic phenomena, with the hope of arresting the operator's attention. Our proposed visualization also provides a visual overview for the post-treatment analysis which enables educated users to quickly and effectively skim through the excessively long process. This paper demonstrates the utility of CycleStack Plot with a case study using real ultrasound videos. In addition, a user study has been performed to evaluate the merits and limitations of the proposed method with respect to the conventional way of watching a video and a signal side-by-side. Even though the motivation of the proposed visualization is improvement of medical applications that use ultrasound, the core techniques discussed here have potential to be extended to other application domains requiring analysis of cyclic patterns from videos.

**Index Terms:** I.3.8 [Computing Methodologies]: COMPUTER GRAPHICS—Applications

## 1 INTRODUCTION

As humans, we can easily perceive the spatial appearance changes from a sequence of images, yet, it becomes quite a challenge when we want to apprehend whether the temporal changes contain certain cyclic behaviors or quantify their frequency and phase. This is a natural result of the fact that we have sensors to see in 3D but not in time, without being too philosophical.

Here, we dissect an ultrasound video application, namely respiration estimation, to better appreciate the challenges laid before us. Ultrasound video is used for many purposes, one of which is the observation of tumors. Based on these observations, certain diagnostic and treatment systems are developed. They all require a high level of accuracy in identifying the location of the tumor, for

instance, to reduce the damage to the surrounding normal tissues in treatment procedures. The location of the tumor and the tissues is influenced by several factors, especially the periodic movement due to the patient's respiration. As described in Saw *et al.*'s review article [9], the respiratory movement of thoracic, pancreatic, renal or liver tumors can be more than several centimeters, which adds a considerable degree of uncertainty in the measured location of the target. While using a larger treatment area can damage healthy tissues, using a smaller area can underdose or miss the tumor, reducing the efficiency and lengthening the time of the therapy.

The clinical techniques based on the respiratory movement utilize the periodicity of the patient's respiratory signal to control the time and duration of the exposure. Several factors, however, can influence the quality of the estimation. First, the respiratory signal of the patient is measured or sampled by sensing external or internal markers, which can introduce noise and error. Second, the target's periodic movement can be quasi-dynamic, which implies that the target may not always move according to the expectation of the treatment plan. For instance, the patient may unconsciously shift his/her body during the therapy causing a change in the location and orientation of the targeting tumor, or the patient's breathing rate may vary during the treatment. Third, the treatment heavily relies on the operator who has to carefully monitor to make sure the treatment is correctly following the planned strategy and using the correct parameters for the particular patient. Not only that, he also needs to react accordingly if any abnormal condition is detected.

To monitor the course of the therapy, the operator needs to consider two different sources of inputs. One source is a video of the targeted tumor's movement which can be obtained via onboard imaging systems. Another source is the patient's respiratory signal obtained in real time. The operator's task is to examine the video as well as the respiratory signal to decide whether the location and shape of the target tumor are nearly matching the values estimated during the treatment planning stage. As long as the periodic movement of the target in the video is seen to be maintaining an expected phase difference to the respiratory signal, the treatment can be assumed to be proceeding normally. Otherwise, it implies that the therapy course is possibly not following the therapy plan, and thus requires intervention by the operator to either reset or manually drive it in order to restore correctness and quality.

However, in course of doing so, a human observer may inexplicably fail to precisely detect any abnormality in the target's movement, as the speed or frequency of the movement can be dynamic. Besides, due to the limited memory of human mind, it is hard to recollect a frame of the video or a pattern in the respiratory signal, which is basically a time series, once it has moved out of the visible window. As a result, comparing the video and the signal becomes non-intuitive and imposes cognitive burden to the observer.

Comparison between the video and the signal is also required for post-therapy analysis to assess the success of the therapy. But as the time required to watch a video is proportional to the number of frames, watching it over and over again is impractical. Even though several methods have been proposed for video surveillance applications in order to visualize a video in constant time irrespective of its original length [1] [2] [3], none of them has been designed to highlight the periodic pattern present in the video which is a re-

\*e-mail:leeten@cse.ohio-state.edu

†e-mail:chaudhua@cse.ohio-state.edu

‡e-mail:fatih@merl.com

§e-mail:hwshen@cse.ohio-state.edu

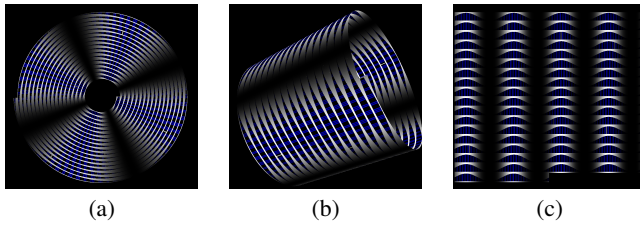


Figure 1: Existing visualization techniques to reveal periodical pattern in time series. (a): 2D spiral layout, which introduces distortion and inconsistent orientation of the patterns. (b): 3D spiral layout, which causes occlusion among the patterns. (c): Linear wrapping, which can cut through the patterns at the boundaries of the display window.

quirement in our case.

To address these shortcomings, we introduce a novel visualization algorithm called *CycleStack plot*. Given a video and a time-varying respiratory signal as the two inputs, the CycleStack algorithm simultaneously plots them in a stacked layout in such a way that the viewer can detect abnormally long or short cycles and significant phase shifts of either input leading to asynchronism between the two. CycleStack also creates an effective overview of the video which allows the user to efficiently compare the video against the signal without having to watch through all the frames. These benefits make CycleStack a useful tool for the human observers to monitor the course of dynamic changes in periodic video data.

This paper is organized as follows. Section 2 overviews the related techniques about respiration-guided therapy, visualization algorithms for videos and time series. Section 3 describes the procedure to create the CycleStack plot. Section 4 presents a case study followed by a user study on real ultrasound videos, which is followed by a short discussion on the limitations and potential directions of future work.

## 2 RELATED WORK

### 2.1 Respiration-gated and Image-guided Therapy

Details about respiration-gated therapy can be found in the following reviews [5] [9] [12]. Some alternative techniques attempt to restrict the respiratory motion of the patient during the course of the therapy, either by instructing the patient to hold his/her breath, or by fitting a physical plate around the abdominal region to reduce the moving range of the tumor. These types of treatments can be impractical for certain types of diseases, and may cause discomfort to the patient [9]. Meanwhile, a group of methods uses the respiratory signal to automatically guide the dose of medication ([11], [15]).

Image-guided therapy incorporates onboard imaging techniques to locate and track the tumor during the therapy [10] [16]. A popular method to sense the tumor’s 3D location and shape is 4-Dimensional Computational Tomography (4DCT), which contains the 3D shape of the tumor in each respiration phase [8].

### 2.2 Video Visualization and Time Series Visualization

Given a 2D video and a 1D signal or time series, our ultimate goal is intuitively highlighting the periodicity of the underlying motion in the video, and efficiently revealing its correlation with the 1D signal. Watching a video to reveal information, however, is often unacceptably time-consuming. To overcome this limitation, several work such as [1] [2] [3] [4] [13] have been published in the domain of video surveillance. By treating the video as a static spatiotemporal volume, low level image features such as gradient or motion

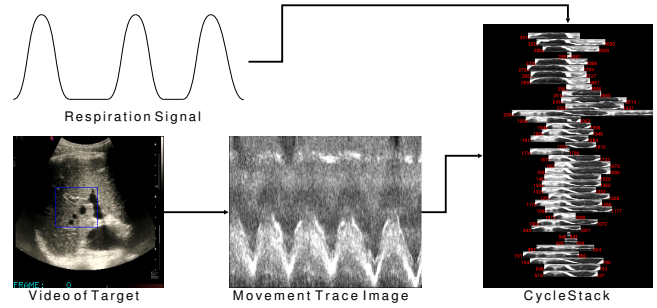


Figure 2: Overview of the system.

flow are extracted from the volume at first. Once this is accomplished, conventional volume rendering techniques can be applied to enhance those features in order to visualize the underlying events. For instance, direct volume rendering with transfer functions that assign high transparencies to static region can hide the background in the scene [3], and flow visualization techniques such as glyphs or streamlines integral can be applied to visualize the extracted motion flow [2]. In addition to direct volume rendering, a video volume can be rendered by slicing a cross section that is coplanar with the temporal axis. This can create an effect similar to slit-scan photography to visualize the trace of moving objects over time [4] [13].

While the common goal behind the above mentioned video visualization techniques is to visualize the *motion* in the video, our focus is to emphasize the *periodic patterns over time* in the video as well as in time series. Because the time series for real data can be long, efficient utilization of the screen space is required. The detail of the periodic pattern will be compressed otherwise. Weber *et al.* utilizes spiral layouts as the time axis in order to render a long time series in 2D or 3D space [14]. Figure 1 (a) and (b) are examples of using 2D and 3D spiral layouts for a synthesized sin function. One issue with the spiral layouts is that the periodic pattern gets distorted as the orientation varies, making it difficult to compare the neighboring patterns. As shown in Figure 1 (c), an alternative of the spiral layout is to wrap the time series from a 1D long array into a 2D matrix, that can be displayed as an image. But then the patterns along the image boundary may become obscure since they are likely to be placed on separate rows. Another technique is called VizTree, which clusters the segments of quantized time series into a tree structure [6]. While these techniques can be used to detect periods of constant duration in a time series; the frequency and the phase of respiratory signals, the time series dealt with in this paper, can vary with time.

## 3 CYCLESTACK

Given a video and a respiratory signal as the inputs, the proposed CycleStack plot can simultaneously display the periodic movement, if any, of an object of interest inside the video and the periodic cycle in the signal. The primary purpose of CycleStack plot is to provide the viewer with an easy way of comparing the periodic movement of the object against that of the signal.

An overview of the procedure to create CycleStack plot is presented in Figure 2. In our method, the movement of an object in the video over a span of time is represented by an image called *Movement Trace Image*. Independently, the respiratory signal is also segmented across temporal axis so that each segment represents one respiratory cycle. Each of these segments is then overlaid on the corresponding sub-image taken out from the movement trace image. The method, in essence, superimposes two different visualizations for each time period of the signal and then, stacks each such superimposed plot in a bottom-up layout to form what we call *CycleStack plot*. Each step of the procedure is described in detail in

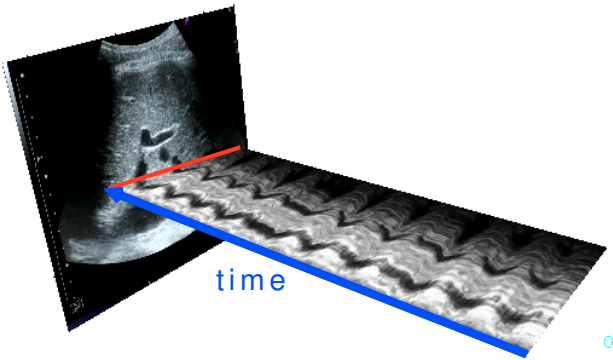


Figure 3: Movement trace. By collecting the pixel intensity along the red line on the video frame over time, an image with a apparent periodic pattern over time can be seen.

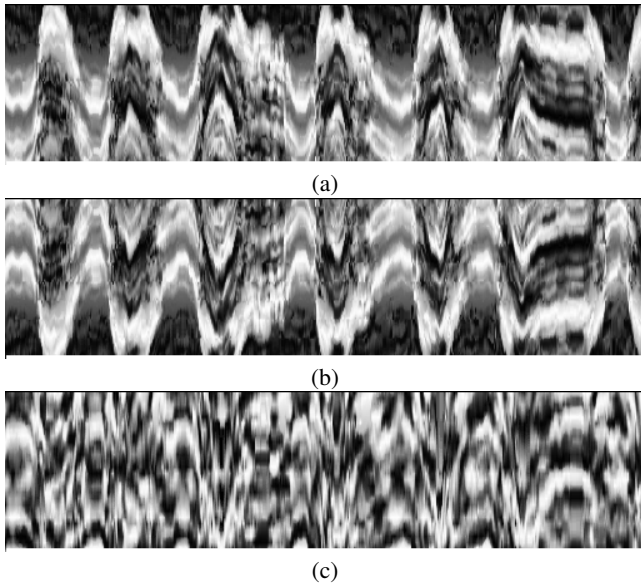


Figure 4: Sampling of the periodical movement in a video. (a): Sampling along a line with apparent periodical change. (b): Sampling along the same line in (a) with opposite direction. (c): Sampling along a line with less periodical change.

the following subsections.

### 3.1 Movement Trace Image

The purpose of this step is to sample the periodic movement of the object of interest, a tumor in our application, in the video. Intuitively, if the object of interest moves primarily along a line segment on the screen, the pixel intensities along that line segment should change with object movement. By recording the pixel intensities along that line segment over time, the *trace* of the object along that segment can be obtained. Figure 3 shows the construction of trace from a sample ultrasound video. By sampling the pixel intensities along the red line that intersects two black blobs (potential objects of interest) in the video, the trace of the blobs over time can be generated. The blue arrow indicates the direction in which time advances. This idea is similar to the Tear Slit [13] which can be used to sample the time-varying pattern in a video.

If the pixel intensity along the specified line segment is sampled as a vector, and the vectors from all time steps are vertically placed next to each other in order of time, the resulting image (see Figure

4) whose number of vertical columns is equal to the number of time steps is called a *movement trace image*. Formally, given a video of  $W \times H$  pixels with  $T$  frames and the line segment between the two points  $L_0 = (x_0, y_0)$  and  $L_1 = (x_1, y_1)$  on the screen, the pixel intensity along the line segment in the  $t$ -th frame  $I_t$  is denoted as  $I_t(L(s))$  where  $t = 1 \dots T$ ,  $s \in [0, 1]$ , and  $L(s) = L_0 + s \times L_1$ . The movement trace image  $M(s, t)$  is the cascade of the samples  $I_t(L(s))$  from all time steps  $t$ ; namely,  $M(s, t) = I_t(L(s))$ . Figure 4 (a) and (b) are two examples of movement trace images, where periodic nature of the movement is apparent, with the horizontal axis used for time (increasing from left to right).

Since the shape and the motion of the object of interest can differ widely from patient to patient, our current system requires the line segment to be specified by the operator. Before beginning the treatment, the operator can watch the video for a while to observe the motion of the tumor, and then he can specify two points on screen to define a line segment along which the periodicity will be well reflected. Some intuitions should be applied while specifying the line segment. First, if the line does not cut through any region which shows periodic movement of the object, the periodic pattern can be obscured or totally absent in the result, as the example shown in Figure 4 (c). Second, the order in which the points are being specified can be critical as well, because the order determines the direction of sampling. Figures 4 (a) and (b) which clearly display patterns opposite to each other are the movement trace images sampled from the same line segment between bounding points  $L_0$  and  $L_1$ , but along reverse directions. These two rules can help the operator to adjust the orientation and sampling direction of the line segment until he starts getting a satisfactory trace.

Two practical issues should be addressed for real object of interest. First, the real object of interest may not always move as simply as along a line segment. The periodic movement can include deformation, rotation, and scaling. However, we have found that a linear segment can represent the periodic movement reasonably well even for more complex movements. Second, the ultrasound sampling along the line segment should exhibit certain degree of heterogeneity in the scale in order to produce apparent patterns. For real ultrasound video, since the grayscale of the foreground and background can be closed, our implementation applied histogram equalization to the sampled line segment in each frame hence the contrast among the sample can be enhanced.

### 3.2 Periodic Cycle Segmentation

In this step, the temporal axis of the respiratory signal which is periodic in normal course is segmented into non-overlapping spans, each of which represents a cycle. Since the signal is similar to a sine or cosine function, it can be segmented across the time axis using a single threshold. The segmentation results in two phases. First, the consecutive time steps with signal value smaller than the threshold form a part, called the *inhale phase*, of the cycle. Similarly, the consecutive time steps having values greater than or equal to the threshold constitute the rest, named as the *exhale phase*, of the cycle. The names of the phases clearly indicate that each pair of phases, inhale followed by the exhale phase, completes a respiratory cycle. Figure 5 (a) is an example of single-threshold segmentation.

But the single threshold method is not sufficient for our application because the respiratory signal collected in real time through sensors is often noisy and the range of the signal values can be dynamic as well. As shown in Figure 5 (b), use of a single threshold to segment a noisy signal can generate short segments which do not correspond to the actual respiratory phases. In other words, local variation of the signal value around the threshold due to noise can be misinterpreted as transition of phase. To deal with this issue, a 2-threshold strategy as described below is applied. The signal value at each time step is compared to both the thresholds, say  $\delta_e$  and  $\delta_i$ .

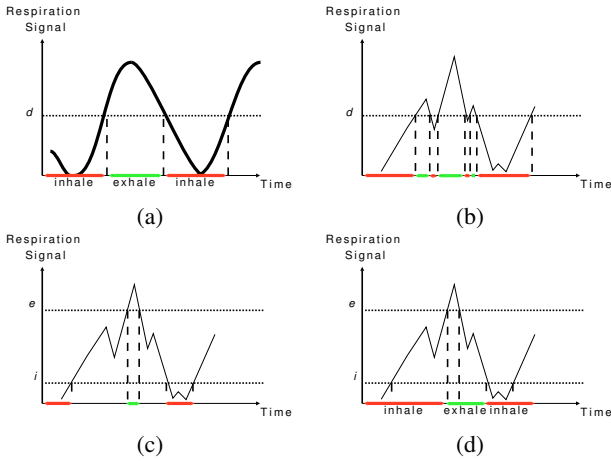


Figure 5: Periodic cycle segmentation. (a): Segmentation of a signal by using one threshold. The exhale and inhale phases are colored in red and green, respectively. (b): Segmentation of a noisy signal by using one threshold, which can create multiple short and false phases. (c) and (d): Segmentation of a noisy signal by using two thresholds. In the first step, as shown in (c), the time steps are larger than the threshold  $e$  or smaller than the threshold  $i$  are marked as red and green, respectively. In the second step, as shown in (d), each phase is extended.

If the value is higher than the threshold  $\delta_e$ , the time step is considered to be in the *exhale phase*. Otherwise, if the signal is lower than the threshold  $\delta_i$ , the time step is marked as one in the *inhale phase*. If a time step's signal value falls between the two thresholds, it is assigned the phase of its immediate predecessor, phase of which must have already been decided. This step ensures that a number contiguous time steps with values between the thresholds together gets assigned to a single phase, avoiding multiple false transitions within. Like the single threshold method, each inhale phase followed by an exhale phase builds a complete cycle. Result of the 2-threshold segmentation of the noisy signal is presented in Figures 5 (c) and (d). Absence of undesired short segments is noticeable. In our system, we have normalized the range of the respiration signal to  $[0, 1]$ , and the two thresholds  $\delta_i$  and  $\delta_e$  have been specified as 0.2 and 0.8 respectively.

### 3.3 Cycle Stack Rendering and Interpretation

Once the processes of segmenting the signal into periodic cycles and the generation of movement trace image are on, the CycleStack plot can be generated using these two inputs. The idea is to draw each periodic cycle of the signal and the movement trace for the corresponding time steps into a small individual plot and to organize each of these plots into an aligned layout so that the viewer can visually compare the signal in a particular time step with the corresponding movement trace and also can analyze plots from different time steps against one another. The detailed procedure of plot and layout generation is described below.

Each individual plot of the CycleStack occupies a 2D rectangular region called *cycle brick* on the screen. The name suggests that each plot corresponds to a cycle in the signal. Each cycle brick is segmented into two horizontally adjacent smaller rectangles called *phase bricks* (see Figure 6). Given a cycle brick, its left and right phase bricks respectively represent the inhale and exhale phases in the corresponding cycle. While the heights of all phase(or cycle) bricks are equal, the width of the a phase(or cycle) brick is linearly proportional to the duration of the corresponding phase(or cycle).

The cycle bricks are then vertically placed as a stack where the order of the bricks from bottom to up follows the corresponding

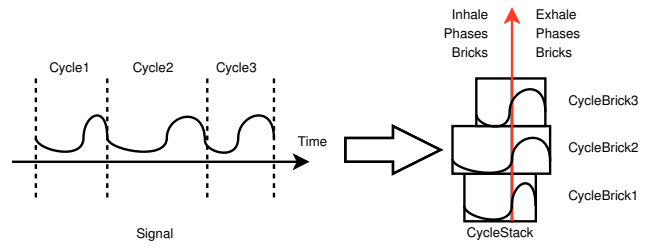


Figure 6: Layout of CycleStacks. Given a signal with three periodic cycles as an example, three cycles bricks are stacked. The left side of the exhale phase bricks of all cycle bricks are aligned to the red vertical line.

cycle's order in time, i.e., a brick is placed above another iff the cycle corresponding to the lower brick immediately precedes the cycle corresponding to the upper brick. The cycle bricks are aligned in such a way that the left (right) boundary of all exhale (inhale) phase bricks coincides on a vertical line through the screen. Figure 6 provides an illustration of the layout of a sample signal.

For each cycle ranging from time steps  $[t_0 \dots t_1]$ , the signal and the movement trace image for that range are simultaneously plotted on the corresponding cycle brick. The relevant sub-image taken out from the entire movement trace image is rendered as the background of the cycle brick. The signal is plotted on the foreground as a function over time, with time being represented along the horizontal axis (from left to right). The superimposition of the sub-image and the signal facilitates visual comparison of the two by the viewer with reduced cognitive load. To speak in a formal way, the sub-image, represented as a rectangle  $(s, t) \in [0, T] \times [0, 1]$  in the texture space, is texture mapped onto the cycle brick which is denoted as a rectangle  $(x, y) \in [l, l + w] \times [b, b + h]$ , where  $w$  and  $h$  are width and height respectively on the screen space and  $(l, b)$  denotes the bottomleft corner. The  $x(y)$  co-ordinates within the range  $[l, l + w]$  ( $[b, b + h]$ ) of the cycle brick is linearly mapped to the  $s(t)$  component of the movement trace image, as shown in Figure 7. Texture mapping is followed by rendering of the signal. The signal for the current cycle is plotted atop the background of the cycle brick as a curve composed of a sequence of 2D data points  $(x_i, y_i), i = t_0 \dots t_1$ . For each  $i$ , the horizontal coordinate  $x_i$  of the curve is determined by linearly mapping the index  $i$  within time steps  $[t_0, t_1]$  to the screen coordinate  $[l, l + w]$  and the vertical coordinate  $y_i$  is the value of the signal at  $i$ -th time step, re-scaled and translated from function space  $[0, 1]$  to screen space  $[b, b + h]$ .

The utility of the CycleStack plot thus generated is demonstrated in detail in Section 4.1 with real ultrasound videos. In brief, the CycleStack plot can convey the following information to the users:

1. Since the horizontal length of the cycle(phase) brick is proportional to the duration of the corresponding cycle(phase) of the signal, the user can easily compare the duration of all the visible cycles at any given moment to narrow down on unusually long or short ones. Similarly, due to the vertical alignment of the bricks at the point of phase transition (inhale to exhale), comparing durations of any one phase from different cycles is not difficult as well.
2. By plotting the signal atop the movement trace, we allow the user to understand how the phase of the signal is related to the movement trace, which represents the periodicity of the object of interest in general. The user should also be able to easily identify any deviation of phase difference from that general pattern.



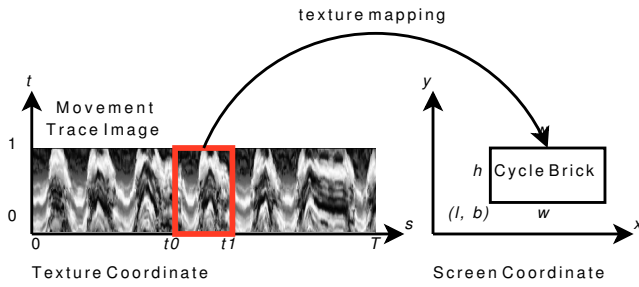


Figure 7: Texture mapping from a movement trace image to CycleStacks. A periodic cycle in time steps  $t_0 \dots t_1$  is linearly mapped from the texture coordinate to a cycle brick with  $w \times h$  pixels where the screen coordinate of the lower-left corner is  $(l, b)$ .

### 3.4 Online CycleStack

In the previous subsections, the construction of CycleStack plot has been discussed with respect to a given video and a signal with a static length. Unlike that, both the video and the signal expand with time during the treatment. As a result, the number of cycle bricks in the CycleStack plot keeps on increasing which eventually can exceed the height of the display window. Not only that, the cycle bricks corresponding to sufficiently long cycles can also exceed the width of the display window. Such an event requires the operator to manually adjust the position and scale of the plots which can be quite an overhead during the treatment.

This issue leverages introduction of some screen space management techniques. In our proposed method, the CycleStack plot automatically re-scales itself to fit a given display window. When the plot for a new time step arrives, the bounding box of the plot is updated and the CycleStack plot is re-scaled to fit the new bounding box into the display window (see the accompanying video for this).

With the automatic re-scaling, less interaction with the visualization is required of the user on events like generation of a new cycle or expansion of length of the inhale or exhale phases. When a new cycle appears, the heights of the existing bricks shrink. When a cycle with a phase longer than all the ones stacked so far appears, the width of the existing bricks corresponding to previous cycles are reduced. But as seen in the accompanying video, both the changes are gradual and subtle enough to be followed by the user.

## 4 EXPERIMENT

The effectiveness of CycleStack plots, which is designed with the primary goal of reducing overhead of the viewer, should be evaluated by allowing the users to interact with it. Hence we have conducted a user study with real and synthesized videos on a moderate sized population. The description and analysis of the user study in Section 4.2 has been placed after a thorough case study in Section 4.1 with two ultrasound videos. The primary objective of the case study is to show the readers how to interpret the CycleStack Plot so that they can appreciate the response of the user study participants in a better way.

Formally speaking, the purpose of this user study is to verify the following hypothesis:

**Hypothesis** *Compared to simply watching a video and a signal at the same time to detect any phase shift leading to a change of correlation between the two, the CycleStack plot provides the user a better means to achieve the same goal.*

In our case study and user study, we are mainly interested to see how CycleStack helps users detect two types of events, namely Asynchronous and Opposite. Both the events occur when the phase

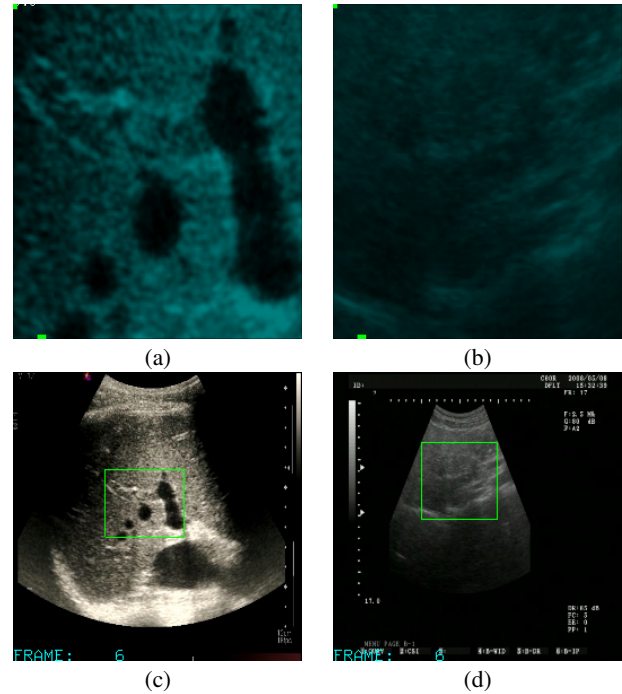


Figure 8: Test videos. (a): Video A, which is a subregion contained in ultrasound video in (c). (b): Video B, which is a subregion contained in ultrasound video in (d).

shift of either the video (movement trace image in case of CycleStack Plot) or the signal takes place. In case of an *Asynchronous Event*, the shift destroys the correlation that existed between the video and the signal so far. On the other hand, an *Opposite Event* reverses the correlation between the two, leading to a special type of asynchronism.

### 4.1 Case Study

This section explains how CycleStack provides intuitive visualization of unusually long and short cycles, and significant phase shift leading to opposite or asynchronous events. This is important because these events, whether detected online during the treatment or offline during the post-analysis, require further investigation by the professionals.

Figures 8 (c) and (d) shows the first frames of the two ultrasound videos to be used for our case study. The videos use the green rectangles to enclose the regions of interest (ROI), detailed view of which are provided in Figures 8 (a) and (b), respectively. The test video in Figure 8 (c), called Video A, contains 3240 frames which takes 180 seconds to play. The second test video in Figure 8 (d), referred to as Video B, consists of 11629 frames playable in 401 seconds.

Figure 9 presents two temporal sections of the CycleStack Plot generated from Video A. The last time step of each cycle is displayed as a label to its right. From the 3 cycle bricks visible in Figure 9 (a), it can be observed that even though the respiratory signal is noisy, the signal (the red curve) still loosely follow a white band in the movement traces image. These three bricks are examples of cycles during which the video is *synchronous* to the signal. Once the user familiarizes himself with this pattern, any deviation from this should trigger his response. For instance, Figure 9 (b) shows such deviations, namely asynchronous and opposite events. For instance, for the cycles ending at time steps 2304 and 2429, the signal (the red curve) does not align to the movement trace image (the grayscale image) in the previously described manner. These

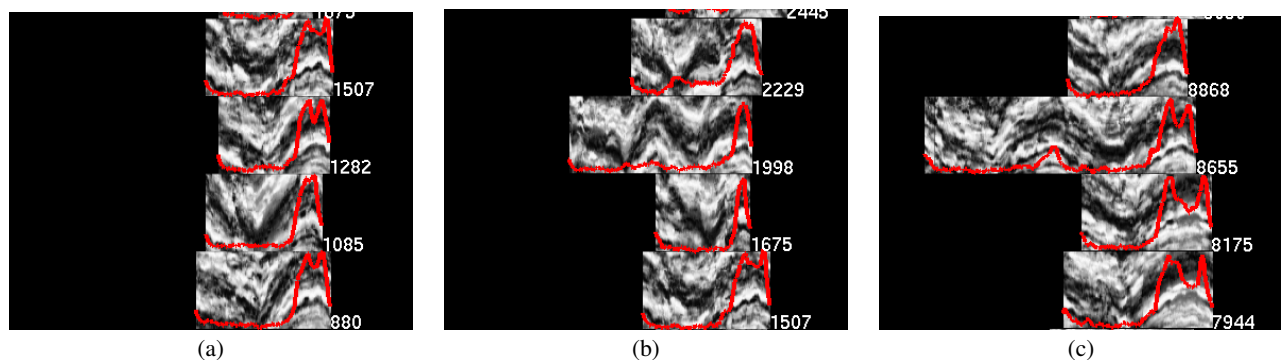


Figure 10: Relevant Portions of the CycleStack plots for Video B. (a) four cycle bricks with synchronous movement traces and respiration signals. (b): four cycle bricks from time step 1283 to 2229, including one asynchronous cycle (which ends at time steps 1998). (c): four cycle bricks from time step 7680 to 8868, including one asynchronous cycle (which ends at time steps 8655).

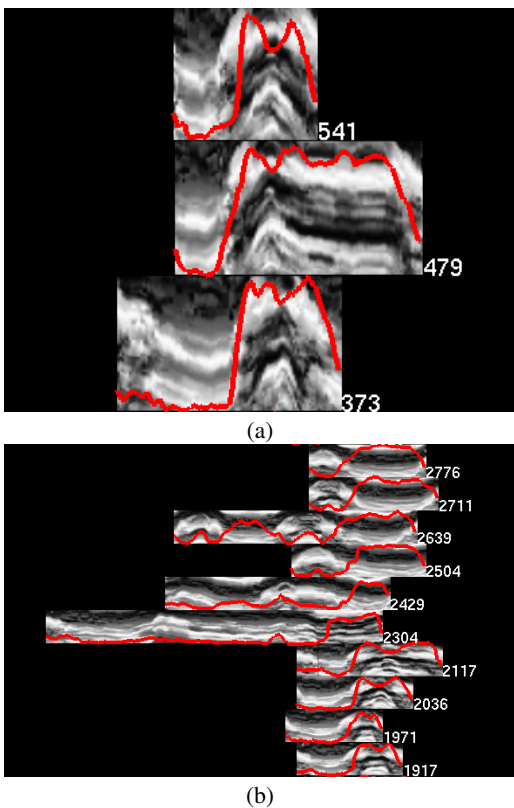


Figure 9: Relevant portions of the CycleStack plots for Video A. (a): 3 cycle bricks with synchronous movement traces and respiration signals. (b): the cycle bricks from time step 1971 to 2776, displaying synchronous cycles (end at time steps 2304 and 2409) and opposite cycles (from time step 2430 to 2776).

two comparatively long cycles rather trigger a significant change in the correlation between them. A cycle preceding these two, for example the one with label 2117, and a cycle following these two, say the one with label 2504, can be seen from Figure 9 (b) to observe that the movement trace image has almost reversed its position with respect to the signal, causing a (*Opposite Event*). This example illustrates how Cyclestack plot allows the user to mentally juxtapose different cycles for comparative analysis. In fact, a simple interaction technique such as select-and-drag can be incorporated to enable

Table 1: User Study Result for Asynchronous Events

Video	Method	Rate	Mean	Std.Dev.
Synthesized	Video/Signal	100%	1370.67	13.46
	CycleStack	100%	1388.11	25.66
Ultrasound Video B	Video/Signal	44.4%	1887.00	32.72
	CycleStack	88.9%	1966.38	141.83

Table 2: User Study Result for Opposite Events

Video	Method	Rate	Mean	Std.Dev.
Synthesized	Video/Signal	100%	1155.86	55.84
	CycleStack	100%	1171.00	51.40
Ultrasound Video A	Video/Signal	88.9%	2488.88	206.06
	CycleStack	100%	2557.00	102.36

visual juxtaposition of multiple cycles from different time zones. One more observation: in the periodic cycle that ends at time step 2639, one can observe that there are actually two periodic cycles in the grayscale rectangle. This indicates probable malfunction of the algorithm that segments the respiratory signal into cycles.

Figure 10 presents temporal snapshots of the CycleStack Plot corresponding to Video B. The four cycle bricks in Figure 10 (a) provide example of the synchronism present between the signal and the movement trace image in normal course. This example illustrates that the signal does not necessarily need to follow a bright or dark band in the movement trace image, as was the previous case; but there exists a similarity among the four displayed bricks which the user can subjectively learn without much difficulty. Figure 10 (b) shows four cycle bricks where the signal in the cycle brick that ends at time step 1998 has generated a different pattern, which is fairly obvious, with the movement trace image, leading to an *Asynchronous event*. To understand the difference from opposite event, we encourage the reader to compare cycle 1675, which precedes 1998, to cycle 2229 which follows 1998. Unlike the example of opposite event discussed in last paragraph with reference to 9 (b), these two cycles exhibit similarity which means the cycle ending at 1998 has not reversed the correlation. Another asynchronous cycle which ends at time step 8655 is detectable in Figure 10 (c).

## 4.2 User Study

The user study has been performed with 18 participants. We have divided the participants into two groups. After a brief demonstra-

tion of the technique by us, one group has watched the test videos and the signal on two separate windows. The other group has interacted with the CycleStack plots generated from the same videos. Both groups have been assigned the same set of tasks so that our hypothesis, already mentioned in Section 4, can be tested by comparing the overall responses of the two groups.

The participants of both groups have been requested to respond when they have identified either the asynchronous event or opposite event. During the test, our application displays a dialog box with two buttons, one for each event. The respective button can be pressed by the participant when he/she detects an event. For each user, the timestamp(s) when a button was pressed have been stored into a log for subsequent analysis. We have analyzed the detection rate and the accuracy of the detected timestep for each type of event. Two test videos, one synthesized and one ultrasound, have been used for each type.

For the asynchronous event, our synthesized video contains a sphere whose radius changes over time. The change indeed follows a cyclic pattern, but with a random time period. One cycle with abnormally long time period has been intentionally incorporated to initiate a phase shift. The ultrasound Video B introduced in Section 4.1 has been used as the second test video. From our case study, it is apparent that an asynchronous cycle exists between time step 1675 and 1998. Therefore, during analysis, we have checked the number of participants who have recorded an event between these two time steps to obtain the detection rate. We have also measured the average and the standard deviation of all the time steps of response from users of each group. The average and the standard deviation indicates how quickly they have responded to the event.

Table 1 presents the result from the user study. The third column (Rate) in Table 1 lists the detection rate, expressed as percentage. The asynchronous event in the synthesized video is equally detectable by either technique: watching video/signal or CycleStack Plot. But only 4 of the 9 participants watching video/signal have detected the asynchronous event between time steps 1675 and 1998 in the ultrasound Video B; whereas it has been recorded by all but one participants using CycleStack. A possible reason for this: CycleStack keeps completed cycles on the screen, allowing the viewer to detect the asynchronous cycle even after it has gone by. But the viewer of video/signal is deprived of that scope.

The fourth column (Mean) and the fifth column (Std.Dev.) in Table 1 report the average and the standard derivation of the recorded time steps respectively. For both the videos, the average time step in case of video/signal is slightly ahead of that in case of the CycleStack, whereas the video/signal method has led to smaller standard deviation than its counterpart. The same factor which led to higher detection rate of CycleStack Plot has a role to play here as well. In case of video/signal, the user either detects the event or misses it. So there is almost no possibility of delayed response. On the contrary, a CycleStack user may, in fact, wait until the beginning of the next cycle to be certain about his intuition. Because with a few more cycles on top of it, the anomaly of a cycle often becomes much more apparent that when it appeared on the top of the stack. This may have led to delayed response for a few participants leading to the higher mean and standard deviation.

For the opposite event, the synthesized video contains a sphere whose centroid moves back and forth on a horizontal line. After a while, the movement undergoes a pause during a pre-defined time step for half a period in order to have the correlation reversed. The ultrasound Video A in Section 4.1 is the other one used to test the opposite event. We have tested whether the viewer can detect the opposite cycles after time step 2000, as described in Section 4.1. Table 2 presents the result. The third column in Table 2 shows that the opposite event in the synthesized video have claimed response from all participants of both groups. Regarding the ultrasound video, only one participant who watched video/signal have

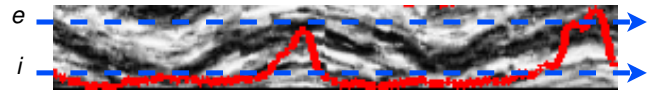


Figure 11: A failure case of our two-threshold cycle segmentation algorithm. The thresholds  $e$  and  $i$  are plotted as the two blue dashed lines. This cycle brick actually contains two periodic cycles, while the first one is not segmented since the signal value during first cycle does not exceed the threshold  $e$ .

failed to detect the opposite cycle. The comparable detection rate of the video/signal and CycleStack watchers can be attributed to the nature of the opposite event itself. This is something which affects all the timesteps that follows and that effect is observable in the video also. This eventually provides the video/signal users with more time to detect it, even if at late. From the fourth and fifth columns in Table 2, it can be seen that although the average time step of detection for video/signal users is smaller than that of the CycleStack watchers, the standard derivation of video/signal group is much larger. This again indicates that some participants who have watched video/signal might have taken late action.

Currently, our conclusion from the user study is that CycleStack Plot ensures higher detection rate of asynchronous events in the ultrasound video because of the viewer's chance of detecting a missed event later. CycleStack Plot also can help users detect the opposite events within a more precise range around the actual time step of occurrence.

## 5 LIMITATION

This section discusses the known limitations of CycleStack plot and our current method of evaluation, followed by potential improvements and possible extension of the technique to other application domains.

Limitations of the CycleStack algorithm include the following. First, layout of the CycleStack depends on the segmentation of the signal (see Section 3.2). The two-threshold segmentation scheme that has been used in our case study is parameter-dependent. Hence, to ensure correct result, tuning of the two thresholds which are the parameters is crucial and requires extra effort by the user. The algorithm may occasionally fail to segment a cycle even after choosing parameters carefully. One example is the long cycle brick in Figure 11 (a), which is clearly seen to have contained two cycles. Furthermore, the current segmentation scheme is designed mainly for respiratory signal which is of sinusoidal nature. More robust segmentation, such as scale-space analysis for time series [7], thus should be studied for accurate segmentation of various types of signal. Meanwhile, currently only movements along a line are sampled, thus requiring deeper study on more flexible sampling for other applications. Finally, CycleStack plot can only display periodic patterns from two inputs (the movement trace image from the video and the respiratory signal from the patient in our case). In order to visualize multivariate time-varying data with a similar technique, some way of displaying more than two inputs at a time is needed.

The user interface has scopes of improvement as well. For instance, with more and more cycle bricks accumulating on the stack, the height of each brick continues to shrink. Currently our interface allows the viewer to zoom in or drag the CycleStack plot to focus on a region of interest which might have already become too small to be clearly visible. As an improvement, we can either refine our layout scheme, such as deleting the earliest cycles to reuse the space for new cycles, or add more flexible user interaction widgets in the future. For example, The user can be given the right to assign different weights to all or to the important cycle bricks on-the-fly so that they maintain a desired height and aspect ratio.

To evaluate our proposed method, we have performed a user study which also has room for improvement. Since the current version of CycleStack plot is primarily designed to be applied in ultrasound-based treatment methods, a user study with trained operators will be performed in near future to obtain the real users' understanding of the problem and their feedback about the interface. In addition to comparing our technique with the conventional way of watching video and signal together, CycleStack plot also needs to be evaluated with reference to other time-series data visualization techniques to understand its effectiveness in revealing patterns and anomalies in a more precise manner. Besides, more real ultrasound videos should be collected and tested.

## 6 CONCLUSION

In this paper, we have proposed an effective visualization algorithm called CycleStack Plot which can benefit the medical professionals who deal with variety of ultrasound-based treatment methods. In general, by superimposing the movement of an object of interest in a video onto a signal, CycleStack allows the viewer to understand the underlying synchronism between the two, and organizes the individual plots in an aligned layout so that any deviation from the synchronism becomes evident. In the context of medical applications, the visualization helps analyze the relation between the periodic movement of the affected body part and the patient's respiratory signal.

In the future, our main focus will be to verify the benefit of CycleStack Plot by having it employed in an actual medical system. Collecting comments and feedback from the operators of such a system is going to be a necessary step to improve the interface. Besides, to see if the use of CycleStack Plot has positively influenced the success rate of the treatment over a considerable period of time can be an indirect way of evaluating its effectiveness.

## ACKNOWLEDGEMENTS

This work was initialized in Summer 2008 when the first author interned in Mitsubishi Electric Research Laboratories with the third author. The authors would like to thank K. Hirasawa, R. Yamakoshi, H. Okuda and K. Sumi from Advanced Technology Labs, Mitsubishi Electric Corporation, Japan for providing insightful comments and an opportunity to work on this project. The authors also wish to thank to all the participants in the case studies for their time, and the anonymous reviewers of this paper for their comment and suggestion. The first two authors are supported by NSF ITR Grant ACI-0325934, NSF RI Grant CNS-0403342, NSF Career Award CCF-0346883, and DOE SciDAC grant DE-FC02-06ER25779.

## REFERENCES

- [1] R. P. Botchen, S. Bachthaler, F. Schick, M. Chen, G. Mori, D. Weiskopf, and T. Ertl. Action-based multifield video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):885–899, 2008.
- [2] M. Chen, R. P. Botchen, R. Hashim, D. Weiskopf, T. Ertl, and I. Thornton. Visual signatures in video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1093–1100, 2006.
- [3] G. Daniel and M. Chen. Video visualization. In *VIS '03: Proceedings of the IEEE Visualization 2003*, pages 409–416, 2003.
- [4] S. S. Fels, E. Lee, and K. Mase. Techniques for interactive video cubism. In *MM '00: Proceedings of ACM Multimedia 2000*, pages 368–370, 2000.
- [5] S. B. Jiang. Technical aspects of image-guided respiration-gated radiation therapy. *Medical Dosimetry*, 31(2):141–151, 2006.
- [6] J. Lin, E. Keogh, and S. Lonardi. Visualizing and discovering non-trivial patterns in large time series databases. *Information Visualization*, 4(2):61–82, 2005.

- [7] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990.
- [8] E. Rietzel, T. Pan, and G. Chen. Four-dimensional computed tomography: Image formation and clinical protocol. *Medical Physics*, 32:874–889, 2005.
- [9] C. Saw, E. Brandner, R. Selvaraj, H. Chen, M. S. Huq, and D. Heron. A review on the clinical implementation of respiratory gated radiation therapy. *Biomedical Imaging and Intervention Journal*, 3(1):e40, 2007.
- [10] C. B. Saw, D. E. Heron, N. J. Yue, and M. S. Huq. Editorial: Cone-beam imaging and respiratory motion (igrt)part ii. *Medical Dosimetry*, 31(2):89–90, 2006.
- [11] A. Schweikard, G. Glosser, M. Bodduluri, M. J. Murphy, and J. R. Adler. Robotic motion compensation for respiratory movement during radiosurgery. *Computer Aided Surgery*, 5(4):263–277, 2000.
- [12] G. Starkschall. respiratory-gated radiation therapy. In J. D. Cox, J. Y. Chang, and R. Komaki, editors, *Image-Guided Radiotherapy of Lung Cancer*, chapter 5, pages 83–92. Informa HealthCare, 2004.
- [13] A. Tang, S. Greenberg, and S. Fels. Exploring video streams using slit-tear visualizations. In *AVI '08: Proceedings of the Working Conference on Advanced Visual Interfaces 2008*, pages 191–198, 2008.
- [14] M. Weber, M. Alexa, and W. Miller. Visualizing time-series on spirals. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001*, pages 7–13, 2001.
- [15] W. Wein, J.-Z. Cheng, and A. Khamene. Ultrasound based respiratory motion compensation in the abdomen. In *MICCAI '08: Workshop on Image Guidance and Computer Assistance for Soft-Tissue Interventions*, 2008.
- [16] R. E. Wurm, F. Gum, S. Erbel, L. Schlenger, D. Scheffler, D. Agaoglu, R. Schild, B. Gebauer, P. Rogalla, M. Plotkin, K. Ocran, and V. Budach. Image guided respiratory gated hypofractionated stereotactic body radiation therapy (h-sbrt) for liver and lung tumors: Initial experience. *Acta Oncologica*, 45(7):881–889, 2006.