

# Automatic Animation for Time-Varying Data Visualization

Li Yu <sup>†</sup> Aidong Lu <sup>‡</sup> William Ribarsky <sup>§</sup> Wei Chen <sup>¶</sup>

<sup>1</sup>University of North Carolina at Charlotte, USA

<sup>2</sup>Zhejiang University, China

## Abstract

*This paper presents a digital storytelling approach that generates automatic animations for time-varying data visualization. Our approach simulates the composition and transition of storytelling techniques and synthesizes animations to describe various event features. Specifically, we analyze information related to a given event and abstract it as an event graph, which represents data features as nodes and event relationships as links. This graph embeds a tree-like hierarchical structure which encodes data features at different scales. Next, narrative structures are built by exploring starting nodes and suitable search strategies in this graph. Different stages of narrative structures are considered in our automatic rendering parameter decision process to generate animations as digital stories. We integrate this animation generation approach into an interactive exploration process of time-varying data, so that more comprehensive information can be provided in a timely fashion. We demonstrate with a storm surge application that our approach allows semantic visualization of time-varying data and easy animation generation for users without special knowledge about the underlying visualization techniques.*

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics—Animation

## 1. Introduction

Animation has been used extensively in visualization. Studies from both research and education have reported that users often describe animation as fun and exciting, which elicits the usage of animation for effective visualization. Especially for time-varying data research, where temporal evolutions are often focused, animation becomes a natural way to represent and analyze the characteristics of data changes across time. For example, storm surge research studies the behavior of hurricanes and their impact on water volumes, which are two important events over the entire duration of the storm. It is obviously not efficient for users to analyze data features and relationships with static visualization for each time step. Therefore, exploring suitable ways to generate animation and incorporating animation in the interactive

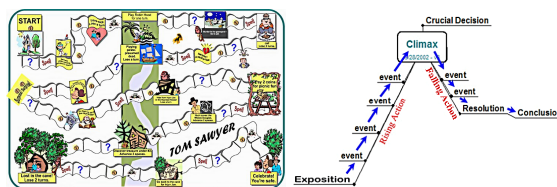
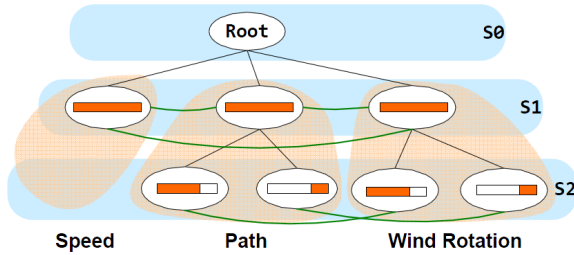


Figure 1: Example representations of narrative structures.

exploration process are valuable for time-varying data visualization.

However, existing visualization systems only provide limited support to generate animations. They are often produced with limited capabilities through adjusting rendering configuration, such as allowing users to record snapshots, adjust rendering parameters, and blend time segments. It is time-consuming for users to determine and modify all these parameters before reaching a satisfying animation result. Also, since these rendering configurations do not imply event semantics directly, current approaches can be very difficult for non-professional users. Therefore, useful storytelling techniques should be considered to describe event features in the

<sup>†</sup> lyu8@uncc.edu<sup>1</sup>  
<sup>‡</sup> aidong.lu@uncc.edu<sup>1</sup>  
<sup>§</sup> ribarsky@uncc.edu<sup>1</sup>  
<sup>¶</sup> chenwei@cad.zju.edu.cn<sup>2</sup>



**Figure 2:** An event graph for the behavior of hurricane eye. The event is described from three feature aspects at different scales. The orange bar inside each node represents its time duration. The scale 0 contains a virtual root that connects all the feature aspects. The black edges are tree links which indicate child/parent relationships of features belonging to the same aspect. The green edges are relation links which describe similarities of event features from different aspects.

visualization, so that the produced animations are easy to understand and follow. Up to now, there has not been such an approach to connect the generation of animation with the description of temporal events.

This paper presents an approach to generate animated visualization as digital stories, which can effectively demonstrate temporal events by mimicking the composition and transition of storytelling techniques. Our approach considers general tips of both storytelling and story writing. For example, as shown in Figure 1, a good story may include a plot with several stages, such as exposition, conflict, rising action, climax, a falling action and final resolution [Obs]. Stories often describe a subject from different aspects and switch topics smoothly. Similarly, our approach describes a temporal event from different aspects and scales through the concept of an event graph (Figure 2), which abstracts a time-varying dataset as a graph of related event features. Animations are then generated by searching for similar narrative structures shown in Figure 1. Our approach also allows interactive generation and modification of digital stories based on the semantics of event features.

Specifically, animations are designed for both summary visualization and interactive exploration of time-varying datasets. Given an event description, we first automatically construct an event graph that abstracts event features from the entire time duration as nodes and their relationships as links. The event graph is built by exploring suitable segmentation of event durations according to different data features, thereby embedding a tree-like hierarchical structure. Narrative structures are built by exploring suitable starting points and search strategies in the event graph. We provide automatic graph search algorithms as well as interactive modification methods. According to different stages of a narrative structure, we automatically determine all the rendering parameters to highlight event features and construct story transitions. We also integrate this animation generation approach into the interactive exploration process of time-

varying datasets, so that suitable animations can be synthesized promptly to provide more comprehensive event visualization. We demonstrate with a storm surge application that our approach allows semantic visualization of time-varying data by generating suitable animations to describe various event features.

The main contribution of this paper is our automatic animation generation approach, which employs animation as a summary and exploration tool to enhance time-varying data visualization. The key concept of our approach is the event graph, which characterizes temporal patterns from different aspects and scales in a time-varying sequence. Closely related to the event graph, several options of constructing narrative structures are explored, ranging from automatic graph search strategies to interactive modification methods. Our approach allows semantic generation of animations, as users can easily produce or modify animations by selecting feature aspects, event details, and options of digital storytelling. We also provide a composition solution for animations by determining optimal parameters automatically with a narrative structure.

## 2. Related Work

This section presents the related work of animation and storytelling in visualization. Our approach is also related to semantic visualization [RBG07], focus+context [VFSG06], automatic composition [LME06], and movie creation methods for videos [FJS96], computer games [HM03] and virtual environments [FFSD04]. For clarity, we focus on the animation and storytelling issues here.

### 2.1. Animation in Visualization

Many visualization systems provide a certain degree of animation support, ranging from recording keyframes to producing animations automatically. For example, Akiba et al. [AWM09] developed a template-based animation tool for volume visualization. Gershon [Ger92] presented methods for visualizing fuzzy data, including displaying a series of blurred images in an animation loop. Viola et al. [VFSG06] presented a method to focus viewpoints automatically on features of a volumetric dataset. Other types of animation include animation of 2D steady vector fields [LJL04], animation of orthogonal texture patterns for vector field visualization [BW08], and 3D interactive animation in information visualization [RCM93].

Animation has also been applied to enrich static visualization. For example, Lum et al. [LSM02] presented kinetic visualization, which visualized animated particles over an object surface to enhance the visual perception. Correa and Silver [CS05] produced animations that highlighted data features by traversing the volume along a path specified via transfer function. Woodring and Shen [WS07] highlighted

objects in a volume dataset by applying animations with positional motions and opacity variations. Also, Blumenkrants et al. [BSS06] created narrative algorithm visualization with the algorithm graph based on information like a central plot or a story. Results of their study indicated that the narrative visualization appeared effective.

The effectiveness of animation is closely related to the data type. For example, Robertson et al. [RFF\*08] evaluated the effectiveness of animation in trend visualization and draw the conclusion that small comparable visualization was the most effective approach. However, Lundström et al [LLPY07] presented a different case. We believe that for data with meaningful 3D structures, like our time-varying datasets, animation can visualize temporal events effectively.

Different from previous approaches of animations, our approach generates animations automatically by simulating storytelling techniques and integrates automatic composition algorithms to produce smoothly transitioned animations. Similar to the multiresolution video method [FJS96], our approach allows animation generation with different levels-of-detail. Our approach is also relevant to the algorithm visualization method [BSS06] on building narrative structures through tree search strategies.

## 2.2. Storytelling in Visualization

Storytelling technique in visualization has been seldom studied. Gershon and Page [GP01] discussed the usage of stories in information visualization, especially for cases when data characteristics were abstract and could not be visualized in the form of a picture in a satisfying way. Wohlfart and Hauser [WH07] presented a method to use storytelling to represent a volume dataset through the processes of story authoring and story telling. They demonstrated the potential of their approach with medical visualization examples. Similarly, our approach can also use animations to represent data information. Different from previous approaches, our approach exploits the automatic generation of animations for time-varying datasets through the detection of event features and construction of narrative structures. Our approach also allows semantic visualizations, which are more flexible and meaningful for domain users.

## 3. Our Approach

In this section, we first present the concept of event graph. Then, we describe our approach to construct an event graph and build narrative structures given a user-specified event. We further present our automatic methods to determine all the required rendering parameters and generate animations with smooth transitions. In addition, we introduce two usages of animations for time-varying data visualization.

### 3.1. Event Graph

The purpose of the event graph is to abstract various features of an event in a time-varying dataset. We choose the form of a graph, which represents event features as nodes and node relationships as links, since it is close to the narrative structures used in storytelling techniques, as shown in Figure 1. With such an event graph, we can generate animations with narrative structures built by exploring suitable starting points and graph search strategies. Since most of our narrative structures are composed of linked node pairs, which correspond to semantic transitions between related event features, we can convey different feature aspects or scales of an event smoothly in the resulting animations.

For example, Figure 2 shows an event graph of a hurricane eye in a time-varying storm surge dataset. The behavior of the hurricane eye is described from the aspects of moving path, moving speed, and wind rotations around the eye region. A reasonable narrative structure is to describe these three feature aspects respectively, like introducing an event from different viewpoints; and each aspect from scales of low to high, like starting with an overview and gradually getting into the details.

We can represent an event graph  $G$  as follows:

$$G = \{\{nodes\}; \{tree\ links\}; \{relation\ links\}\} \quad (1)$$

where an event graph consists of three sets of information: nodes for representing event features from several aspects and at different scales, tree links for indicating the child and parent relationships of nodes belonging to the same feature aspect; relation links for indicating the similarities of time durations of nodes from different feature aspects.

Specifically, the nodes without any children correspond to basic events, such as a straight moving path or a constant moving speed. The nodes with children correspond to complex events which are composed of multiple basic events. Every event graph also contains one virtual node that connects all the features aspects. The number of nodes in an event graph is related to both the event complexity and detail levels according to user interest. Each node has several components: feature aspect, time range, a score measure to indicate the significance of this event feature, a list of tree links, and a list of relation links.

There are two types of links: tree links and relation links. The tree links represent the relationships of complex events and basic events belong to the same feature aspect. The duration of a parent node always contains the durations of its children nodes. The relation links describe the temporal similarities of nodes that are on the same scale but from different feature aspects. Each relation link has two components: a related node and a similarity value, which uses 0 to represent "the same", 1 to indicate that the time range of the current node includes the related node,  $-1$  vice versa, and the ratio of overlapped time duration to the time range of the current

node for other cases. To avoid confusion, we use relation links with value 0 most of the time.

An important feature of the event graph is that each event graph embeds a tree-like structure that organizes all the relevant event features. As shown in the graph example in Figure 2, each feature aspect is represented as a tree branch. The scale 0 represents a virtual node as the tree root. The scale 1 always represents event features from the entire duration of a time-varying dataset. Starting from the scale 2, the nodes are calculated by segmenting the time range of parent nodes according to the characteristics of data changes. With this tree structure, we can locate different levels of event details according to the node scales.

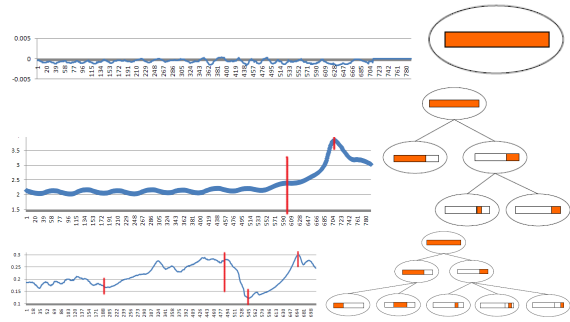
### 3.2. Constructing Event Graphs

We construct an event graph by following the embedded hierarchical tree structure. Initially, given an event and its feature descriptions, we can first build a basic tree structure with the virtual root at scale 0 and a node for representing each feature aspect from the entire time duration at scale 1.

Starting from scale 2, we segment the duration of each node on a parent scale until reaching basic events. As shown in Figure 3, the moving speed of the hurricane eye is reducing constantly, therefore it is a basic event and only contains one node. The river surface elevation and the wind rotation of the hurricane eye contain more variation. They are represented by a sub-tree in the event graph. Our approach to time segmentation is through measuring the dissimilarities of time steps automatically. For every feature aspect, we compare each pair of adjacent time steps and quantify their differences as one value, which can be shown as the curves in Figure 3. We detect if this node describes a basic event by testing if all the dissimilarity values are below a user-specified threshold. The segmentation process stops when reaching a basic event; otherwise it continues to higher scales.

The segmentation is achieved by searching for the cuts that divide a time duration into several segments, where the dissimilarity values are more evenly distributed. Specifically, we calculate the gradients of dissimilarity values, smooth them with a Gaussian operator, and apply Sobel edge detection [TV98]. The locations with the maximum results are the cuts to segment a time duration. For storytelling effects, we limit the number of children for each node to 3, so that we can keep an evenly distributed event graph.

For example, for the event of the hurricane eye, we first locate the path of the eye as the weighted center of wind magnitudes for each time step. The moving speed is measured as the shift of eye location in adjacent time steps. For the wind rotation, we first identify a region around the hurricane eye with high wind magnitudes, which is shown as the blue region. Then we sum the wind angle at each vertex of all the triangles within this region. Since the hurricane speed



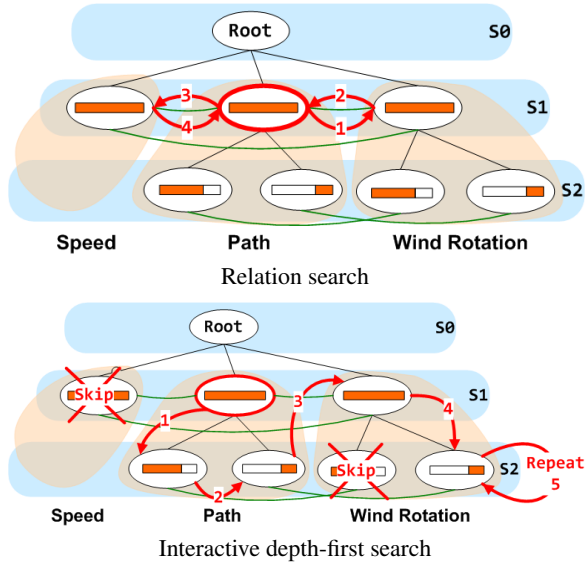
**Figure 3:** (left) The temporal dissimilarity curves for moving speed of the hurricane, river surface elevations, and wind rotation of the hurricane eye. Our segmentation results are shown as the red lines. Longer lines indicate segmentation on lower scales. (right) Constructed sub-trees.

is roughly constant during the entire duration, this branch only has a leaf on scale 1, while the moving path and wind rotations are further divided to higher scales.

To match event features from different aspects, we combine them during the construction process of the event graph. Specifically, we determine if two nodes with the same duration, but from different feature aspects, can be segmented jointly. We achieve this by measuring the distributions of their dissimilarity values. The dissimilarity values from the involved time durations are concatenated and treated as one high-dimensional vector. We combine two feature aspects if the dot product of their normalized dissimilarity vectors is smaller than 10% of the time duration. We further merge their dissimilarity values for the segmentation process. Otherwise, the conclusion is that these two aspects are very different and should be segmented separately. For example, the moving path and wind rotation of the hurricane eye are segmented jointly on scale 2.

The relation links are constructed for each scale to describe the relationships of node durations from different feature aspects. For scale 1, since the durations of all the nodes are the same, there is a relation link with value 0 between every node pair. Starting from scale 2, we measure the relationships of node durations and assign corresponding values. For example, there are two relation links between the two nodes under moving path and wind rotation aspects in Figure 2.

We also measure a significance score of each node based on the multiplication of its temporal duration, region size and dissimilarity variance of an event within the node duration. The region size is measured as the accumulated area that covers the event for the entire duration. For example, in Figure 2, the moving path covers all the regions where the hurricane passes, while the wind rotation is only related to the small region of hurricane eye. The dissimilarity variance is calculated through the dissimilarity curve, shown in Figure 3. For each scale, we further normalize all the scores



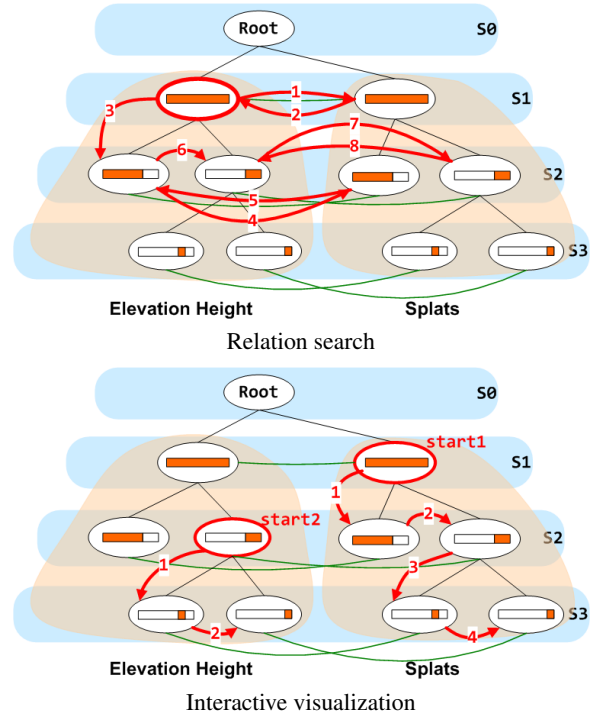
**Figure 4:** The event graph of the hurricane eye and 2 narrative structures, shown as the red links. The node with red boundary is the starting point. The top shows the result of a relation search algorithm on scale 1. The bottom shows the result of a depth-first search algorithm below scale 3. Interactions are also applied to skip the feature aspect of speed and one node of wind rotation, as well as repeating the end segment of wind rotation aspect.

by dividing the largest value, so that they can suggest the significance portion on this scale. The virtual root is always assigned a score value 0, meaning that this node should be skipped. We also balance different feature aspects by multiplying a user-assigned scale with the dissimilarity variance. For example, we magnify the scores of wind rotation by 10 to make them comparable to the scores of other feature aspects. The significance results describe the events in the example of Figure 2 well. On scale 1, the moving path has the highest score value, 1.0, since it describes an event that involves the entire data region. The score value of moving speed is the lowest, 0.1, since it has to do with details which are not obvious when we consider the entire time duration.

Scientists often have some domain knowledge of the data under study. For example, in the storm surge study, scientists are interested in the behavior of hurricane and effects of the hurricane on ocean and river volumes. Therefore, we can build a set of event graphs before visualization. Figure 5 shows another example result of an event graph built from a storm surge dataset: bumps and splats in the river.

### 3.3. Building Narrative Structure

A narrative structure is a plot that determines the sequence of event features in an animation. It is built by searching for a suitable node as the beginning and a search strategy that



**Figure 5:** The event graph of bumps and splats and 2 narrative structures, shown as the red links. The node with red boundary is the starting point. The top shows the result of relation search below scale 3. The bottom shows the result of interactive visualization, where users first choose to visualize the splat feature globally and then select the feature of elevation height close to the end of the time range.

traverses the event graph and satisfies user-assigned storytelling options. Several automatic schemes as well as interactive modification methods are explored to generate narrative structures.

The starting node is preferably the most significant node on the lowest available scale, which can serve as an overview to introduce the major event feature. For example, for the event of the hurricane eye in Figure 4, we compare the scores of the three nodes on scale 1, and select the moving path node as the starting point. This is the same beginning for the two narrative structures built from different strategies in Figure 4.

The path of a narrative structure traverses the rest of the relevant nodes in the event graph. A path is composed of tree links and relation links in a specific order. Traversing along the tree links is identical to describing additional event details. Traversing along the relation links equals the description of related event features from other aspects. These two links help us to achieve smooth semantic transitions in the storytelling.

To build suitable narrative structures, we design three criteria for the graph search strategies.

1. An event is composed of an overview at the beginning and smooth transitions to event details.
2. Related features are introduced sequentially according to their score values or temporal order.
3. No node is repeated unless specified by the user.

We can utilize some existing algorithms of graph search, such as breadth-first search and depth-first search [CLRS01], to generate our narrative structures. These two search algorithms meet our design criterion 3 directly. If we always start from a node on the lowest available scale, we can also meet our criterion 1. According to criterion 2, the only addition we make is to determine the search sequence of children nodes given a parent node. We can choose to follow the order of their temporal durations or the order of their score values.

Since we introduce the relation links to the event graph, we also develop a search strategy called relation search. Specifically, given a starting node, we follow the depth-first search algorithm. The main difference is that if the current node has relation links, we first traverse the nodes connected by relation links instead of exploring tree links. This means that we do not perform depth-first search on relation links. Examples with the hurricane and splat events are provided in Figures 4 and 5.

The design of our event graph allows easy user interaction and modification on narrative structures. For example, we provide the following interaction methods.

**Selecting feature aspects** Users can select different event aspects to visualize and the branches that are not selected are ignored in the searching process.

**Selecting of level-of-details** Users can adjust the levels-of-detail of animations and only the nodes on involved scales are traversed.

**Skip** Users can choose to ignore some feature aspects at certain levels (e.g., if they are already familiar with this data portion). These nodes are not considered during the graph traversal.

**Repeat** A portion of a narrative structure can be repeated for emphasis. We add the repeated portion to the narrative structure.

**Selecting starting point** Users can select the starting node. The branch containing the starting node will be treated as mostly important.

### 3.4. Automating Animation as Storytelling

To avoid unnecessary user interaction, our approach determines all the rendering parameters automatically. The parameters are calculated for each node in the narrative structure. We also provide a transition between adjacent nodes to indicate feature changes.

Our rendering parameters include: time steps, viewpoints,

and data resolution. The other rendering settings are fixed in an animation to avoid confusion. For example, the region-of-interest is always highlighted in green, as shown in Figure 7. All the other regions are visualized in the same color design: terrain always in grey and surfaces always colored according to their elevation levels with a blue-to-red colormap. All these detailed rendering parameters are automatically determined based on our experiments. For professional users, an interface is also provided for adjustments.

The number of time steps selected for each node is proportional to its score value. The data loading and rendering times for each time step are approximately the same, assuming we use the same data resolution for all the time steps belonging to this node. Therefore, the number of time steps is linear to the allocation of time duration. To emphasize important features, a small number of time steps is used for basic events, while a rather large number is used for major events like the hurricane moving path. We set the maximum allowed number of time steps  $N_{max}$ , which can also be adjusted. The number of time steps for a node  $n$  with score  $score(n)$  is calculated as  $N_{ideal}(n) = score(n)^p \times N_{max}$ , where  $p$  is a parameter to adjust the effect of score values. We treat an event evolving gradually in each node duration, so we sample time steps evenly. Assume the number of time steps contained in a node is  $N(n)$ . If  $N(n) \geq N_{ideal}(n)$ , we select a time step for every  $N_{ideal}(n)/N(n)$  time steps. Otherwise, we set a sleep time proportional to  $N_{ideal}(n)/N(n) \times rendering\ speed$  in the program.

Several options for viewpoints are explored for our animation: sky view, local view, dynamic view, optimal view, and user-selected views. Generally, we prefer to put the event close to the center of the screen and avoid having the interesting region occluded by surroundings.

**Sky view** A sky view is often used in TV programs. It points to the event center, is vertical to the main event moving direction, and has a 60 degree elevation. The view zooms into a region whose size is 1.6 times the bounding box of the interested region in this time duration.

**Local view** A local view produces the effect that viewers are standing close to the ground, which can be found by reducing the elevation degree of the sky view to the range of [0, 30].

**Dynamic view** The dynamic view follows the path of the event and points to the object center at each time step. We smooth the path to avoid the shaking camera effect. Similar to the sky view, the view zooms into a region whose size is 1.6 times of the bounding box of the interesting region in this time step.

**Optimal view** The optimal view can be selected according to the object shapes and overlap relations, such as through maximizing data variations [LME06]. It is the same for each event node. We simplify this step by running 2D principal component analysis (PCA) [WEG87] on the object center locations from the entire node duration. The

optimal view is calculated as rotating the second axis of PCA to an elevation of 60 degree. The view zooms in the same way as the sky view.

**Selected view** Our system allows users to choose their desired viewpoint by rotating and zooming the scene.

Data resolution is selected according to the performance requirement. Low resolution is used for interactive exploration. In our experiment, our machine can support a reasonable performance (around 0.5 second per time step) when using data contains 520114 polygons.

To provide smooth transition between adjacent nodes, we introduce three additional renderings of the event location, duration and graph. They are rendered automatically to provide useful information about an event feature. As shown in Figure 7, we use the top right corner to show the location of an event, which can be calculated with the camera setting. The time duration is shown on the bottom of the screen, which renders the duration of the current event node in pink. The current time step is also specified with a red arrow above the time window automatically according to data names. A simple event graph is rendered on the top with current node or link in red. These two renderings change automatically according to the node information. When the node switches, we change the scene by linearly interpolating the ending viewpoint of the previous node and the starting viewpoint of the next node, and then start the animation for the next node. We also blend the data from the two transiting nodes in the rendering to produce a simultaneous rotating and fading effect. Figure 6 provides such an example.

### 3.5. Animated Visualization

We provide two options of digital storytelling using animations: summary and interaction modes. The summary mode automatically produces an animation to summarize the relevant event features in a time-varying dataset, while the interaction mode is designed to generate and show animations in real-time to provide more comprehensive event information than static visualization for interactive exploration.

The summary mode does not have a performance constraint, and thus we can include all the nodes of an event graph in the animation results. Any of the three searching strategies can be used to build narrative structures. We can also render datasets in their original resolution and utilize any provided methods to determine our rendering parameters. Figures 4 and 5 (top) provide examples of summary animations using relation search with different levels-of-details.

The interactive mode allows users to apply animations to enrich static visualization. Instead of requiring users to select their interesting data region from every time step, we render a time-varying dataset according to a narrative structure automatically. This setting allows users to replay an animation, stop an animation anytime, and most importantly generate a specific animation that describes their interested

event features. Such interactive exploration provides more relevant information to users efficiently. Since the construction of the event graph takes time, especially for large time-varying datasets, we require users to specify the event under exploration before interaction. The event graph of any event only needs to be built once, therefore we can quickly collect all the event graphs of major events. In the storm surge application, the behavior of the hurricane and its impact on the water are two important events. We also perform data reduction before the exploration process by equally down-sampling the grid meshes for all the wind field, terrain and water surfaces.

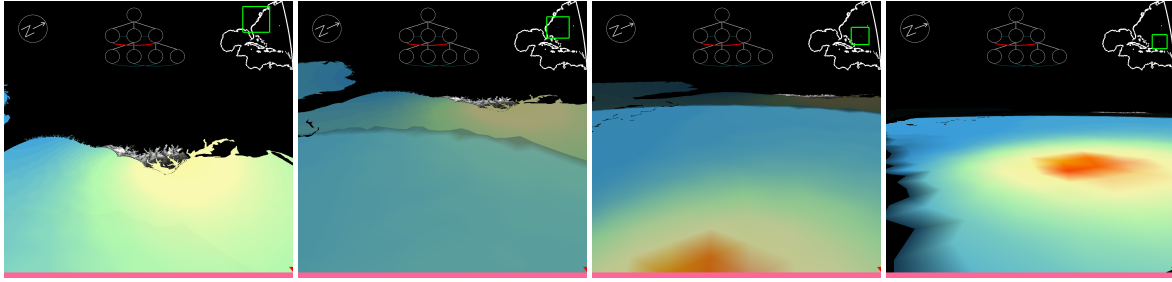
The challenges of real-time animation generation are to find a suitable starting node based on user selection and satisfy performance requirement. During interaction, users can specify their interesting event aspect and adjust the time step and viewpoint to visualize a specific data portion. With this information, we locate the matching node on the highest scale from the corresponding branch in the event graph. The time range of this node should include the selected time step and the node region covers the visualized data portion. Among our search strategies to generate narrative structures, we often choose the depth-first search and relation search algorithms, as they provide most relevant information for user interaction. Since a narrative structure is built very quickly, we can start the resulting animation without any delay. To satisfy the interactive performance requirement, we simplify choices of rendering parameters. For example, we use sky view, local view, and dynamic view instead of optimal view. We also use data in lower resolutions. Figures 4 and 5 (bottom) provide examples of animations under the interactive exploration mode.

## 4. Results and Discussions

The time-varying datasets used in this paper are generated by a storm surge/inundation model, ADCIRC (<http://www.adcirc.org/>), which has been approved for storm surge studies by Federal Emergency Management Agency (FEMA) and applied extensively for modeling storm surge in the Southern Louisiana and New Orleans areas. Specifically, we use the data of hurricane Isabel to the Outer Banks of North Carolina, USA, in 2003 [WL10]. Each dataset contains 792 time steps, 520,000 triangles, and 260,000 vertices. There are three attributes, terrain elevation, water elevation, and wind vector on every vertex.

### 4.1. Example Results

The attached multimedia files provide our animation results, which correspond to the narrative structures in Figures 4 and 5. These four examples demonstrate different automatic narrative structures and interactive modification results. The event graphs of two events, hurricane eye and splats, are constructed automatically based on event descriptions. In all the



**Figure 6:** A transition example. Four snapshots are provided from the transition of wind rotation to the moving speed on scale 1, which is indicated by the red curve on the event graph on the top. During the transition, the viewpoint is smoothly rotated and the rendered data are automatically blended to produce a rotating and fading effect.

renderings, the terrain surfaces are rendered in gray. The water and hurricane surfaces are illustrated with a blue to red colormap, with red corresponding to the highest elevation value. The options of viewpoints are randomized in the generated animations to avoid uniform rendering effects. Example snapshots are provided in Figure 7.

The narrative structure in Figure 4 (top) is generated with the relation search strategy with the lowest details. The resulting animation provides an overview of the three event features. It first shows the moving path of the hurricane eye, which is close to a straight line, since it is the most significant feature. Second, the animation visualizes the wind rotation around the hurricane eye, which mostly increases time range. At the end, it introduces the moving speed, which is constantly reduced.

The narrative structure in Figure 4 (bottom) is generated with the depth-first search strategy. The resulting animation provides more details of the event than the first animation. It has the same starting point, which overviews the moving path of the hurricane eye. The animation then visualizes each segment of the moving path. Then, the animation overviews the wind rotation and visualizes the second half time range for the hurricane eye. Since the second half is nearer to the coastal terrain, which is more important to scientists, users can choose to skip the first half range and repeat the second half. Different parameters are selected to provide more comprehensive information on the wind rotation.

Our second example event is the bumps and splats in the rivers. The narrative structure in Figure 5 (top) is generated with the relation search strategy with middle level of details. The animation first overviews the water surface elevation height, which is the dominant feature of this event. It continues to introduce the bumps and splats in the river, which are highlighted as green dots. Then, the animation visualizes the elevation height and splats respectively, for both time ranges, where we can see a periodic feature during the first time range and the effect of the hurricane during the second part.

The narrative structure in Figure 5 (bottom) is generated with interactive exploration. We first choose to visualize the

bumps and splats, so the animation starts with an overview on scale 1 and continues to visualize the data from the first and second time ranges respectively. We then choose to view the elevation height over the second time range, and the animation locates the starting node on the scale 2 and generates an animation with the depth-first search algorithm.

#### 4.2. Quantitative Results

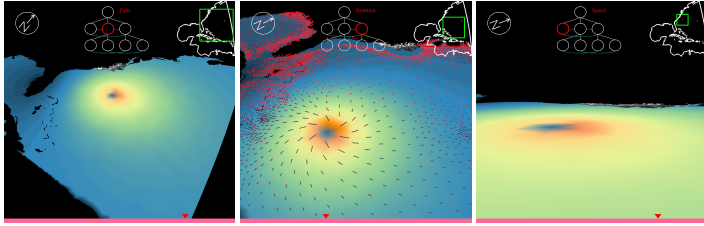
We run our algorithms on a computer with an Intel Core2 CPU 6600 at 2.40GHz and 2GB RAM. Most of the processes in our approach, construction of the event graph, calculation of rendering parameters, and data reduction, are preprocessed before interactive visualization. The running time for constructing an event graph depends on the selected event features. The event graphs in Figures 4 and 5 take around 7 minutes. We also determine parameters for the rendering at this stage, which is within 5 minutes. Data reduction can be achieved within an hour and it is only required once for a dataset.

The only operation during interactive visualization is the generation of narrative structures. The performance of all our search algorithms is interactive. Therefore, once an event graph is built, our approach allows users to generate and modify animations interactively.

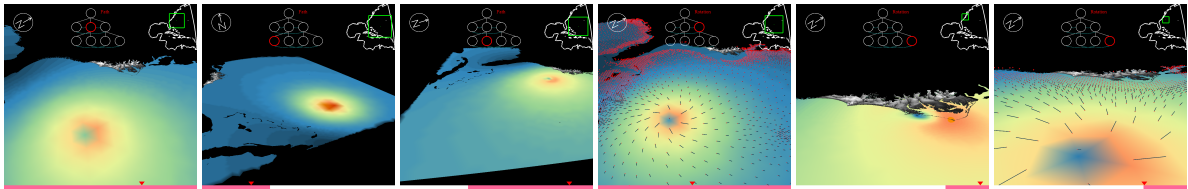
#### 5. Conclusions and Future Work

This paper presents an automatic animation approach to visualize temporal events by mimicking the composition and transition of storytelling techniques. This is achieved with the concept of an event graph, which abstracts event features from different aspects and scales, and an automatic animation process with a series of event detection, time segmentation, graph search, and rendering composition methods. Our approach also allows interactive modifications of both narrative structures and parameter selections. The resulting animations can be used in summary visualization as well as interactive exploration, which provides more comprehensive information of a time-varying dataset in a timely fashion. We demonstrate with a storm surge application that our approach

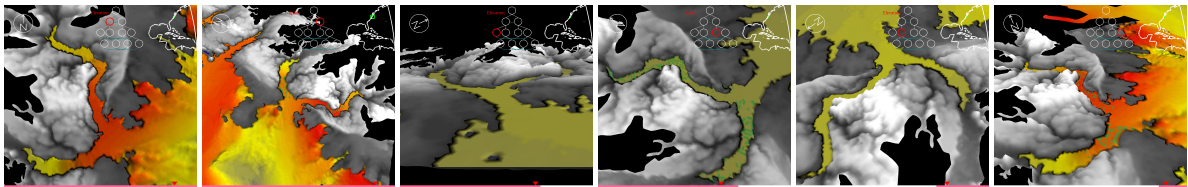




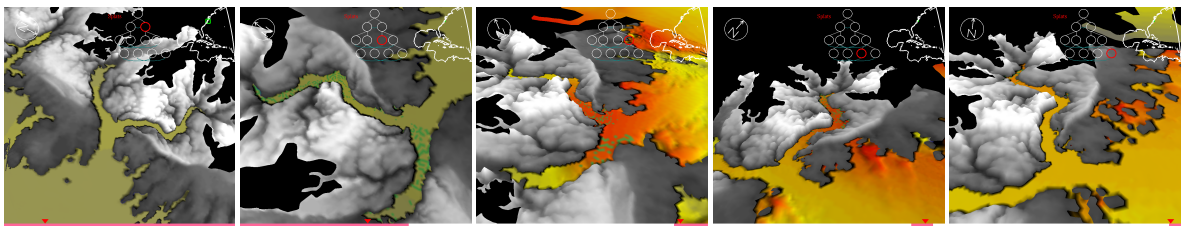
moving path (s1) -> wind rotation (s1) -> moving speed (s1)  
Hurricane eye - relation search (narrative structure is shown in Figure 4 (top))



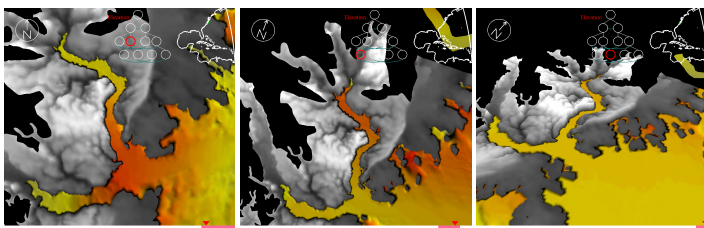
moving path (s1) -> movingpath 1 (s2) -> movingpath 2 (s2) -> wind rotation (s1) -> windrotation 2 (s2) -> windrotation 2 (s2)  
Hurricane eye - interactive depth-first search (narrative structure is shown in Figure 4 (bottom))



elevation (s1) -> bumps-splats (s1) -> elevation 1 (s2) -> bumps-splats 1 (s2) -> elevation 2 (s2) -> bumps-splats 2 (s2)  
Bumps and splats - relation search (narrative structure is shown in Figure 5 (top))



bumps-splats (s1) -> bumps-splats 1 (s2) -> bumps-splats 2 (s2) -> bumps-splats 1 (s3) -> bumps-splats 2 (s3)  
Bumps and splats - interactive exploration - start1 (narrative structure is shown in Figure 5 (bottom) marked with start1)



height 2 (s2) -> height 1 (s3) -> height 2 (s3)  
Bumps and splats - interactive exploration - start2 (narrative structure is shown in Figure 5 (bottom) marked with start2)

**Figure 7:** Example renderings from the animation results. Each row corresponds to a narrative structure in Figures 4 or 5. The event location of each node is illustrated on the right top corner and the node duration is rendered as the red bar on the bottom of each snapshot. The little red triangle above the duration bar indicates the specific time step of each snapshot. The texts under each row suggest the node sequence in each animation. Please refer to the descriptions in section 4.1 (example results) for more details of these animations.

allows semantic visualization of time-varying data and easy animation generation for users without special knowledge about the underlying visualization techniques.

Our approach provides a framework to generate animations to describe various temporal events. As shown in our example results, our approach only requires users to identify some high level key aspects of the time-varying phenomena and how to characterize the associated features, such as motion path, eye, and wind speed for the hurricane eye event. This information can be obtained through discussion with domain experts. As a result, in this case we have a structure that can be applied to *all* hurricane and storm surge simulations, and which also can be used to classify and correlate these different simulated phenomena. For our storm surge application, scientists have pointed out that the bumps and splats are directly related to the storm modeling technique (and may be unphysical artifacts); therefore they are very interested in characterizing and understanding these events. Thus we have a process to add important new features in a straightforward manner. This overall process of developing a rich narrative framework by considering a small number of key high level aspects can be applied generally to a range of time-varying phenomena.

Our future work includes exploring the effectiveness of different narrative structures of the same event given summary or interactive exploration tasks. This result can provide useful information to design general animation techniques. We are also interested in extending this approach to volumetric time-varying data visualization, where we expect to integrate more feature tracking and data reduction methods.

## 6. Acknowledgements

We wish to thank the editors and reviewers for their valuable comments. This research is supported by DHS Center of Excellence - Natural Disasters, Coastal Infrastructure and Emergency Management (DIEM), DOE DE-FG02-06ER25733, and NSF 0633150.

## References

- [AWM09] AKIBA H., WANG C., MA K.-L.: Anviz: A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications* 99 (2009). 2
- [BSS06] BLUMENKRANTS M., STAROVISKY H., SHAMIR A.: Narrative algorithm visualization. In *SoftVis '06: Proceedings of the 2006 ACM symposium on Software visualization* (2006). 3
- [BW08] BACHTHALER S., WEISKOPF D.: Animation of orthogonal texture patterns for vector field visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 741–755. 2
- [CLRS01] CORMEN T. H., LEISERSON C. E., RIVEST R. L., STEIN C.: *Introduction to Algorithms*. The MIT Press, 2001. 6
- [CS05] CORREA C. D., SILVER D.: Dataset traversal with motion-controlled transfer functions. *Visualization Conference, IEEE V* (2005), 46. 2
- [FJSD04] FRIEDMAN D., FELDMAN Y. A., SHAMIR A., DAGAN T.: Automated creation of movie summaries in interactive virtual environments. In *VR '04: Proceedings of the IEEE Virtual Reality 2004* (2004), p. 191. 2
- [FJS96] FINKELSTEIN A., JACOBS C. E., SALESIN D. H.: Multiresolution video. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996). 2, 3
- [Ger92] GERSHON N. D.: Visualization of fuzzy data using generalized animation. In *VIS '92: Proceedings of the 3rd conference on Visualization '92* (1992), pp. 268–273. 2
- [GP01] GERSHON N., PAGE W.: What storytelling can do for information visualization. *Commun. ACM* 44, 8 (2001), 31–37. 3
- [HM03] HALPER N., MASUCH M.: Action summary for computer games: Extracting action for spectator modes and summaries. In *International Conference on Application and Development of Computer Games* (2003), pp. 124–132. 2
- [LJL04] LEFER W., JOBARD B., LEDUC C.: High-quality animation of 2d steady vector fields. *IEEE Transactions on Visualization and Computer Graphics* 10, 1 (2004), 2–14. 2
- [LLPY07] LUNDSTRÖM C., LJUNG P., PERSSON A., YNNERMAN A.: Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1648–1655. 3
- [LME06] LU A., MACIEJEWSKI R., EBERT D. S.: Volume composition using eye tracking data. In *Proceedings of EuroVis* (2006), pp. 115–122. 2, 6
- [LSM02] LUM E. B., STOMPEL A., MA K. L.: Kinetic visualization: a technique for illustrating 3d shape and structure. In *VIS '02: Proceedings of the conference on Visualization '02* (2002), pp. 435–442. 2
- [Obs] OBSTFELD R.: *Fiction First Aid: Instant Remedies for Novels, Stories and Scripts*. 2
- [RBG07] RAUTEK P., BRUCKNER S., GROLLER E.: Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1336–1343. 2
- [RCM93] ROBERTSON G. G., CARD S. K., MACKINLAY J. D.: Information visualization using 3d interactive animation. *Commun. ACM* 36, 4 (1993), 57–71. 2
- [RFF\*08] ROBERTSON G., FERNANDEZ R., FISHER D., LEE B., STASKO J.: Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1325–1332. 3
- [TV98] TRUCCO E., VERRI A.: *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, 1998. 4
- [VFSG06] VIOLA I., FEIXAS M., SBERT M., GROLLER M. E.: Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 933–940. 2
- [WEG87] WOLD S., ESBENSEN K., GELADI P.: Principal component analysis. *Chemometric and intelligent Lab. Sys.* 2 (1987), 37–52. 6
- [WH07] WOHLFART M., HAUSER H.: Story telling for presentation in volume visualization. In *Proceedings of EuroVis* (2007), pp. 91–98. 3
- [WL10] WEAVER R. J., LUETTICH R. A.: 2d vs. 3d storm surge sensitivity in adcirc: Case study of hurricane isabel. In *Estuarine and Coastal Modeling XI* (2010). 7
- [WS07] WOODRING J., SHEN H.-W.: Incorporating highlighting animations into static visualizations. In *Proceedings of SPIE Electronic Imaging* (2007). 2