World Lines

Jürgen Waser, Raphael Fuchs, Hrvoje Ribičić, Benjamin Schindler, Günther Blöschl, and M. Eduard Gröller

Abstract—In this paper we present World Lines as a novel interactive visualization that provides complete control over multiple heterogeneous simulation runs. In many application areas, decisions can only be made by exploring alternative scenarios. The goal of the suggested approach is to support users in this decision making process. In this setting, the data domain is extended to a set of alternative worlds where only one outcome will actually happen. World Lines integrates simulation, visualization and computational steering into a single unified system that is capable of dealing with the extended solution space. World Lines represents simulation runs as causally connected tracks that share a common time axis. This setup enables users to interfere and add new information quickly. A World Line is introduced as a visual combination of user events and their effects in order to present a possible future. To quickly find the most attractive outcome, we suggest World Lines as the governing component in a system of multiple linked views and a simulation component. World Lines employs linking and brushing to enable comparative visual analysis of multiple simulations in linked views. Analysis results can be mapped to various visual variables that World Lines provides in order to highlight the most compelling solutions. To demonstrate this technique we present a flooding scenario and show the usefulness of the integrated approach to support informed decision making.

Index Terms—Problem solving environment, decision making, simulation steering, parallel worlds, CFD, smoothed particle hydrodynamics.

✦

1 INTRODUCTION

In the last decade, computational simulation has experienced a tremendous progress. Computer hardware and simulation techniques have developed beyond what has been considered possible in many respects. Today, computational simulation is an ubiquitous tool in industry and research. But already new modes of application are in demand. Instead of performing a single simulation, users want to study multiple related simulations at once. They want to change input parameters in order to understand their impact. To study the influence of the relevant parameters, users need to be able to go back to any point in time to alter or refine their choices, to modify the simulation setup and trigger additional simulations.

In many cases the exact development of the situation cannot be predicted, instead, multiple scenarios must be considered (Figure 1). In such cases valid solutions can be found only by comparing a set of different simulation runs and analyzing the alternative scenarios they represent. This introduces an extended space of possibilities: instead of a single simulation run, users are confronted with a whole range of related, parallel worlds. Such an environment, where the user is able to pose 'what if?' questions to a simulation framework, are one step in the direction of the problem-solving environment which Johnson [19] has identified as one of the most important research problems in scientific visualization.

Computational steering is a powerful concept that enables domain experts to interact with a simulation during its execution. Today, the work flow of computational simulations is increasingly demanding to the user since simulations become more and more complex, comprising many different input parameters and large amounts of heterogeneous data results. This is especially true for computational fluid dynamics (CFD) where the traditional work flow is to prepare input, to execute a simulation, and to visualize the results in a post-processing step. However, more insight and a higher productivity can be achieved if these activities are done simultaneously. This is the underlying idea of simulation steering: researchers change parameters of their simulation on the fly and immediately receive feedback on the effect [41]. Obviously we can take this approach one step further: *researchers change parameters of their simulation on the fly and can then analyze both the original outcome and the alternative interactively.*

The combination of steering with visualization has been a common goal of the visualization research community for twenty years, but it is rarely ever realized in practice [27]. This is in part due to a missing concept to abstract the management for generation, storage and visualization of data describing multiple alternative scenarios. The World Lines approach (Figure 2) which we present in this paper integrates simulation, visualization, and interactive analysis into a unified system.

2 RELATED WORK

In this paper we discuss a novel visualization approach to steer, visualize and solve problems based on the simulation of many possible worlds. Here, we present some of the related work in these fields.

Simulation Steering Mulder et al. [31] give a survey of simulation-steering environments. They stress that the user interface is a critical component of a computational steering environment. Johnson et al. [20] point out major topics when building a simulation-based problem solving environment: control structures, data distribution, data presentation, and user interfaces. Treeck et al. [42] present a steering system that enables modification of geometry via basic transformations. Matkovic et al. [27] suggest to combine CFD simulation and visualization by writing out multiple simulation runs as ensemble data and comparing these runs using the COMVis System.

History, Provenance and Processes GRASPARC [5] and Hyperscribe [45] identify the ability to preserve states as an important feature for iterative problem solving. A history tree records information as the simulation progresses so that the calculation can be stopped and rolled back to previous points in time. A modified set of input parameters can be specified in order to restart the simulation and create a branch point in the tree. This history tree is visualized as a set of colored spheres connected via cylinders. Various project management solutions use line-based visualizations to display processes [11, 3]. AsbruView [22, 23] provides a temporal view that presents a plan hierarchy in a tree view similar to those used in file managers. In an additional topological view, each plan is displayed as a track. The Victorian wall atlas illustrates a genealogical tree as lines in a horizontal layout [1]. VisTrails [38] adapts a history tree for capturing and reusing provenance in a visual exploration system. Graph-based layouts have been adopted for data exploration [26] and process visualization [28, 35]. Business process visualization deals with complex

---

- *Jürgen Waser is with VRVis Vienna, E-mail: jwaser@vrvis.at.*
- *Raphael Fuchs is with ETH Zürich, E-mail: raphael@inf.ethz.ch.*
- *Hrvoje Ribičić is with VRVis Vienna, E-mail: ribicic@vrvis.at.*
- *Benjamin Schindler is with ETH Zürich, E-mail: bschindl@inf.ethz.ch.*
- *Günther Blöschl is with TU Vienna, E-mail: bloeschl@hydro.tuwien.ac.at.*
- *Eduard Gröller is with TU Vienna, E-mail: groeller@cg.tuwien.ac.at.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

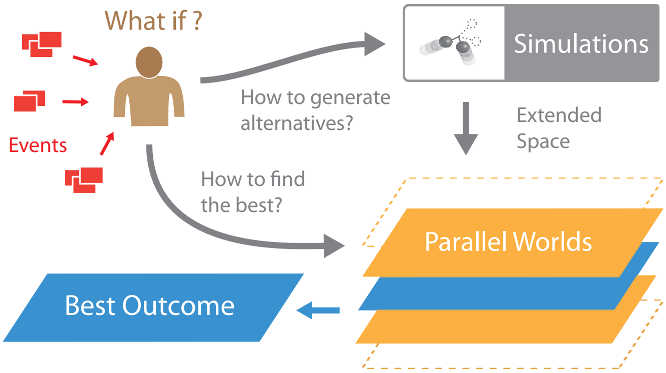

Fig. 1. Problem Description: The investigation of a time-dependent problem comes down to a series of 'what if?' questions. Real-world events (red boxes), often unpredictable, further require to study alternative scenarios in order to be able to make decisions. The user needs a concept to effectively steer a simulation system to produce this set of required parallel worlds. Moreover, this concept should allow to quickly filter this multitude of alternative solutions to find the best outcome.

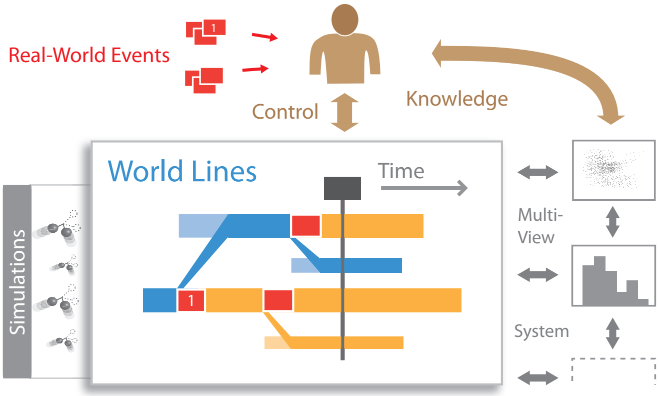

Fig. 2. Proposed Solution: We suggest a novel view called World Lines that enables control over multiple simulation runs, hides complexity and is capable to deal with the extended space. World Lines is part of a system of multiple linked views that enables interactive comparative analysis across alternative worlds.

events which have to be monitored, steered and optimized [25, 33, 39]. In the existing graph-based or history tree approaches, a rather sparse representation is used. This is useful if a broad spectrum of different processes has to be visualized that can consist of many different activities [13]. In the setting of this work, the basic component is given by a time step in one simulation process. There are no different activities within this process, but many continuous time steps are automatically generated by the simulation. Therefore a dense representation is required.

Interactive Analysis of Simulation Data Using multiple, interactively linked views of the same data set allows the user to productively combine the information gathered from different views [16, 36]. Weaver [43, 44] shows that, as the number of linked views increases, it may be necessary to visualize the structure and operation of the visualization. Linking and brushing allows the user to select an area or parameter range of interest by interactively placing selections on a rendering. Other views and interactions are linked to the selections and focus on information related to the selected subset. Hauser [14] states that as soon as a notion of interest in some subset of the data is established, we can visualize the selection in full detail while reducing the amount of visual information about the remaining data. Doleisch et al. [6] apply multiple linked views to the analysis of CFD data. For additional information we refer to the related state of the art report on the visualization of multi-variate scientific data [9].

Simulation Even though this work does not focus on a specific simulation technique, we mention research that is related to the smoothed particle hydrodynamics (SPH) method [30, 24] since it is used as the simulation component in this paper. Mueller et al. [32] state that less accurate methods which allow the simulation of fluid effects in real-time open up a variety of new applications. During the design phase, real-time methods help to test whether a certain decision is promising. SPH delivers interactive timings for small particle numbers [32] and, as Kipfer and Westermann [21] show, SPH can provide a realistic appearance for environmental flood simulation. Ghazali and Kamsin [12] illustrate that SPH can be used to model flash flood behavior with adequate realism. At present, there are GPU accelerated fluid simulation modules available on the web: NVidia provides a GPU implementation of SPH in their PhysX package [34]. Open Source GPU versions of SPH are offered by Hötzlein [18] and Hérault et al. [17].

3 OVERVIEW ON WORLD LINES

The concept of World Lines originates in physics [29]. It is a general way to represent the course of events. In their original setting, World Lines describe the movement of an object through space-time. In a more general setting, we can consider a World Line as a description of changes to system states over time. From this perspective, the entries of a journal, a sensor-log or the course of a simulation all have a corresponding representation as a World Line. Later, the concept of World Lines was extended to include the uncertainties of quantum-mechanics events, allowing a set of World Lines to describe a multitude of possible alternative worlds that can emerge in a probabilistic setting [8]. This extended version of World Lines now allows to represent concepts such as alternative choices, uncertain outcome of events and even equivalent results obtained by different chains of events and decisions.

We start with some definitions. A *state-space* is the space spanned by all variables of a system including positions, time values or internal object states. A *frame* is a representative of one point in state-space, for example a simulation result at a given point in time. A *track* is a consecutive set of frames. For example, a single simulation run that comprises a specific set of simulation parameters is represented as such a track. An *event* is a modification of the system parameters (e.g. a user intervention or an external data update) that results in a change of the systems behavior and is recorded as a *branch*. A set of causally related tracks is called a *World Line* and presents a possible outcome. The *parallel worlds* at a given point in time are defined as the frames of all tracks that temporally overlap at this time value.

We present the World Lines view as the driving component in a system of multiple linked views that deals with the extended space of parallel worlds. The view will operate in two modi. A steering mode for generating and controlling simulation runs and a visualization mode for comparing them. In the steering mode, when starting an additional simulation by changing parameters, the system creates a new branch that originates from the parent track at the current position in time. A single simulation run is visualized as an animated track and a cursor evolving in time. Each track stores the system configuration that has lead to this track. For navigation and frame selection, a movable World Lines cursor is provided. Depending on the current operational mode, this cursor can assume different shapes. The currently selected frame is shown in linked monitors where users set and edit events that influence the behavior of the simulation track. In addition, inline widgets are provided to edit the track-characteristic properties in place. Since simulation tracks are connected in a tree-like fashion, any changes to a track affect all child tracks and the system can automatically re-simulate their evolution. The visulization mode can be used to comparatively analyze a quantity of interest in various interactive visualization styles. This way, the user is quickly informed about the best outcomes.

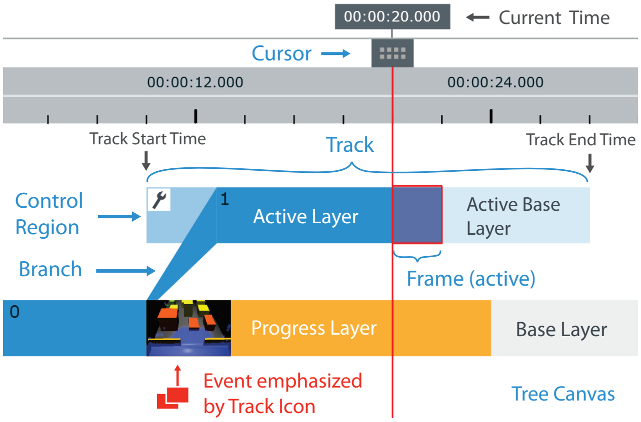

Fig. 3. Basic visual elements of World Lines. A single state is presented as a frame. Simulation runs are visualized as tracks that share a common time axis. Causal relations are depicted through branches. The World Lines cursor defines the active frame. Different layers indicate the status of the system.

4 VISUAL REPRESENTATION OF RELATED SIMULATION RUNS

We visualize World Lines as a tree of tracks that are connected by branches. Each of these components has regions the user can interact with (Figure 3). A track has the shape of a colored ribbon that spans the duration of a simulation run. In the tree canvas, a set of parallel tracks is vertically arranged such that they share a common time axis indicated by a timescale on the upper parts of the World Lines view. Tracks are identified by a numerical label placed onto the track.

User intervention events are given special attention by the concept of *branching*. Branching occurs when the user modifies parameters of an existing track at a specific point in time. The newly created track visually originates from the parent track. This parent-child relation between simulation runs is visualized as a skewed quadrangle which we call the incoming branch of the child track. Tracks are designed to visualize their full time range but emphasize their origin (parent track) at the same time. For this purpose, the track comprises a separate control region at its front. It receives a lower opacity value to better accentuate the incoming branch.

The creation of simulation runs through steering is represented as a sequence of branching actions. The resulting tracks form a horizontal tree-like visualization. The tree thus directly visualizes the causal relation between simulation results and shows which events influence which other events. In this approach to computational steering, the user is an essential part in the loop of simulation and visualization. The search for the optimal solution becomes the search for the best World Line. This World Line can be interpreted as a sequence of events and decisions necessary to obtain the planned outcome in practice.

4.1 Layers to visualize simulation states

We compose tracks and branches as different visual layers (Figure 3) to present the various stages of the exploration process. All colors and opacity values can be set by the user.

- The *base layer* represents the part that has not yet been simulated. The length of a track's base layer determines the time span of the related simulation run.
- The *progress layer* is placed on top of the base layer and indicates the current simulation progress.
- The *active layer* highlights the active World Line (Figure 4) as one path through the tree (one course of events) that is currently in focus. This layer has full contrast on top of the progress layer but has a lower opacity above frames without progress (active base layer) to show the separation between simulated and not-yet simulated frames.

4.2 z-index to arrange visual elements

The *z-index* is a crucial parameter when arranging visual elements in the tree canvas. This parameter determines whether a component is placed on top (highest z-Index) or below other elements, potentially hiding other tracks or branches of the visualization. We define the following basic rules for the specification of a component's z-index.

- The incoming branch of a track is placed on top of the track.
- The most recently created track receives a higher z-index than all available tracks.
- Tracks that are part of the active World Line are located on top of all other tracks. Within the active World Line, each track is placed on top of it's parent track.

5 NAVIGATING THE MULTI-VIEW SYSTEM

World Lines can be regarded as a novel component in a system of multiple coordinated views that has to deal with the extended space of alternate simulation runs. The role of components that are linked to World Lines can be manifold. They might act as steering components that allow the configuration of the input parameters which are associated with the active track (Section 6). They can visualize simulation and analysis results as given by the active frame or a set of selected, parallel frames (Section 7).

5.1 Track Activation

World Lines offers convenient ways to navigate the system through time and parallel worlds. The multi-view environment is synchronized with the state that is associated with the active frame which is defined by the active track and the current time (Figure 4). We have developed interactive concepts to ease the specification of the active frame. A track can be activated by mouse-click interaction. Direct track activation is one method to navigate the system through parallel frames.

5.2 World Lines Cursor

The *World Lines cursor* is designed to indicate the current time and to highlight the active frame. The cursor consists of a draggable box which is placed above the timescale as well as a vertical line that spans the entire tree canvas. The current system time is shown in a label that is attached to the top of the draggable box. To accentuate the active frame, the vertical line of the cursor is augmented with a rectangular focus element that surrounds the active frame. If the frame size is below a certain limit, this focus element is reduced to a shape that looks similar to a cursor in a text editor. Inspired by modern audio and video editing software, we have added cursor functionality to enable direct navigation in time. The time can be set by dragging the cursor in the horizontal direction.

5.3 Jumping

As an alternative to dragging the cursor, we can jump from one point in time to another one by clicking into the corresponding horizontal position above the timescale. This process is supported by a timescale indicator which appears when the mouse cursor is moved over the draggable area. As with the cursor, this indicator uses a text label to display the target time of the jump action. We realize a set of advanced navigation buttons to enable further types of jumping (Figure 4). These buttons can be used to jump from one track to another one, across parallel tracks or along the active World Line to navigate from one user intervention to the next. In addition, the World Lines view has an editable time label to enter jump-target times directly.

5.4 Timer-based Simulation and Replay

The World Lines view has a timer that is running at a user-defined sampling rate to enable simulation recording and replay. In this case the cursor follows the active World Line. When the cursor encounters a branch that is part of the active World Line, it automatically activates the child track connected to this branch. The user can choose to keep the active frame in focus at the center of the view. This gives the

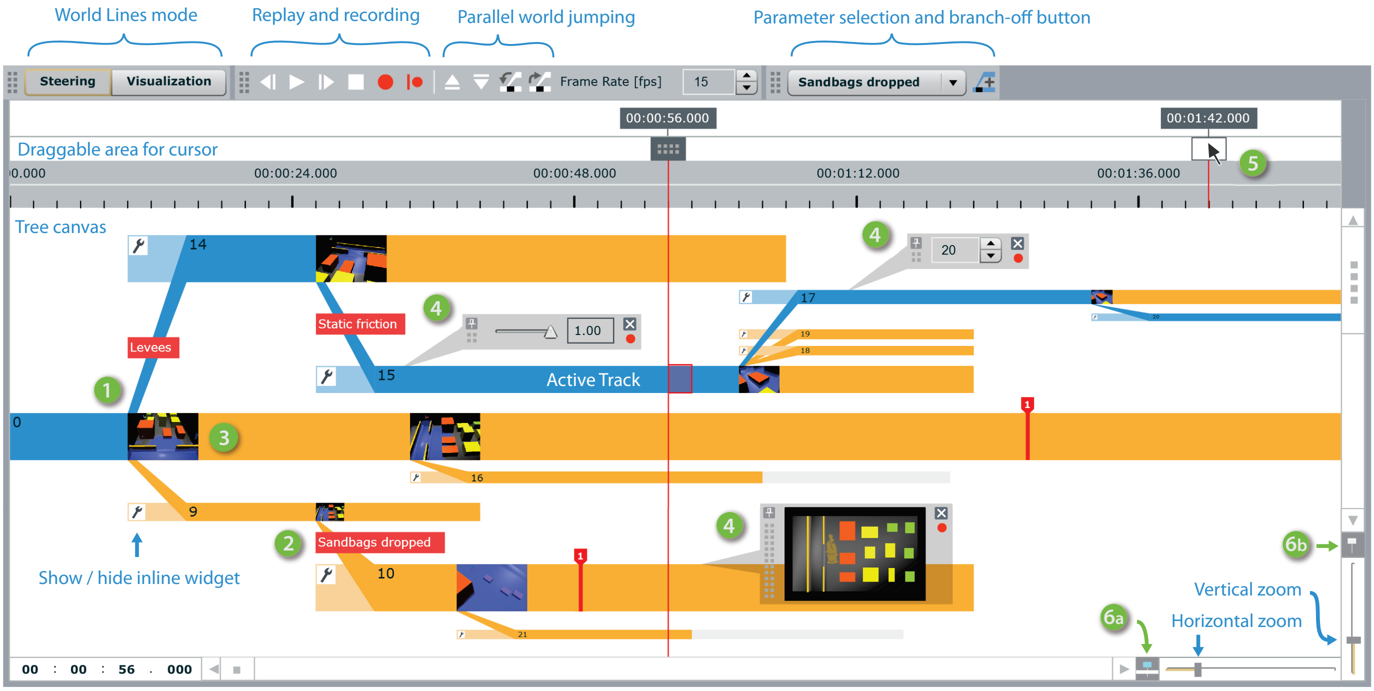

Fig. 4. Screenshot of World Lines in the steering mode. (1) New tracks are created by branching. (2) The relevant parameter is indicated by a label placed onto the incoming branch. (3) At the branch-off location, track events are emphasized by track icons. (4) Toggable inline widgets can be used to edit parameters. The active World Line (blue) represents the current preferable course of events and determines timer-based playback and recording. (5) The timescale indicator appears when moving the mouse over the draggable area of the cursor to support direct cursor jumping. When navigating and zooming, the active frame remains in the horizontal and/or vertical center if (6a) the horizontal focus button and/or (6b) the vertical focus button is pressed respectively.

impression that the view is moving behind a static cursor in either horizontal or vertical direction or both. This mode is convenient for monitoring an ongoing simulation as we can follow the progress even if the generated visualization exceeds the window bounds.

5.5 Zooming

We have adapted standard methods that allow for zooming the view in both horizontal and vertical direction. The zoom is designed to keep the active frame in focus at the horizontal and vertical center of the view. When zooming horizontally, only the tracks are stretched, branches and the control region keep their size. Standard scroll bars are employed to navigate into areas of the tree canvas that are currently not visible. It is convenient to press the horizontal and/or vertical focus buttons next to the scroll bars in order to quickly center the view around the active frame.

6 STEERING MODE OF WORLD LINES

In this section we describe different aspects of World Lines for the generation and management of multiple, related simulation runs. The visual representation of alternative scenarios with World Lines offers multiple ways for user interaction. The user can manipulate initial and boundary conditions as well as inherent parameters of the simulation. These interventions reflect the user's choices, for example, the modification of inflow conditions or a change in the shape of the simulation geometry.

6.1 Branching

At the position of the World Lines cursor, the user can create a new track by modifying parameters of the active track. When creating a new track in this way, the parent track is not required to have simulated frames. Branching can also occur on track sections that do not show progress at the moment. This way, users can plan a set of related simulation runs in advance. A simple annotation centered above the branches identifies the type of parameter that has been changed. The visibility of these *branch labels* can be toggled. If there is not enough space, labels are hidden by default.

Inline Widgets The parameters that are relevant for a specific track are either steered by its associated *inline widget* or in a linked view. We will first concentrate on inline widgets which comprise an interactive area with different controls to enable quick adjustments of the setup (Figure 5). As part of the tree canvas, inline widgets have the highest possible z-index to keep them above tracks and branches. In the control region of the affected track, a configuration button is provided that enables the user to open or close the inline widget (Figure 4). Each inline widget is associated with an incoming branch that directly relates the widget to the affected track. Several World Lines features (like zooming) change the position of tracks within the tree canvas. The user can decide whether an inline widget is pinned to a desired position or if the widget has to follow the movement of the corresponding track.

Track Icons To further emphasize the real-world event that led to the new track, we show an interactive *track icon* at the branch-off location of the parent track. These track icons show a snapshot of the current simulation state as generated by one of the views that are linked to World Lines. Internally, a track icon stores a snapshot of each view available at the branch-off location. By clicking onto the icon, the user can switch through the stored snapshots.

6.2 Steering via linked views

When steering the parameters of a specific track, we can take advantage of the multi-view framework. When selecting a track, all views and steering controls are updated to show the parameters of the selected track. The user can edit the track-specific parameters in linked steering views. These views can be of any type that is applicable to the domain-specific problem. For example, we have implemented a linked 2D view that allows to place geometric primitives into the scene. This view is part of our case study and is demonstrated in Section 8.

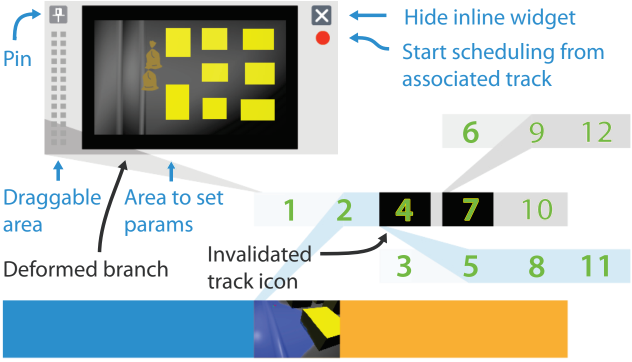

Fig. 5. Inline widgets offer a fast way to customize the input parameters of simulation runs. When modifiying parameters, the track and its descendants are invalidated. Hitting the record button of an inline widget, notifies the scheduler to re-simulate all affected tracks. The numbers on the tracks outline the ordering in which frames are being simulated if the user opts for per-time step scheduling and prioritization of the active World Line.

6.3 Scheduling

The *World Lines scheduler* is a system entity that uses automatic timer-based recording in order to simulate the predefined base layers according to a user-defined ordering. The scheduling can be adjusted by the user in several ways. The basic tool is interactive track resizing. The user can elongate the base layer of the track in order to predetermine the length of the related simulation run. Alternatively the user can edit the end time of a track using the track's context menu entry. In combination with branching the user can set up a set of related runs that can be automatically simulated.

The processing order of tracks is determined by the *visual priority queue* of World Lines. This structure orders all available tracks according to their vertical position in the tree canvas. The top-most track has the highest priority. Two mechanisms have been developed to enable interactive modifications of the vertical track layout. The first technique is based on direct movement of tracks. Here, the user drags one or several tracks that have been selected with a rubberband tool. The second method is based on automatic layout. Using a track's context menu, the user can re-layout the tree canvas such that the track moves up, down, to the center or to the top of the visual priority queue. In Section 7.2, we describe animated transitions to support the user in perceiving automatic layout changes.

Having the priority queue set up, the user can choose between two scheduling methods:

- *Per track scheduling.* The system completely simulates one track at a time, starting from the tracks's progress time until the end time as indicated by the track's base layer. If the currently handled track has no progress at all, this necessitates a recursive search up to the root of the World Line that this track belongs to. Causality demands that the system handles all predecessing frames prior to simulating the actual track of interest.

- *Per time step scheduling.* The scheduler treats one time step at a time, simulating all parallel frames as ordered from top to bottom. This mode is useful if the user wants to compare alternative simulation runs as soon as possible. This can give early insight in order to be able to remove undesired outcomes quickly.

In addition, each of the suggested types provides the option to process the active World Line prior to the rest of the tracks that are present in the visual priority queue (Figure 5). The presented scheduling approach has been realized in a way that the ordering of tracks as well as the scheduling type can be changed during simulation runtime.

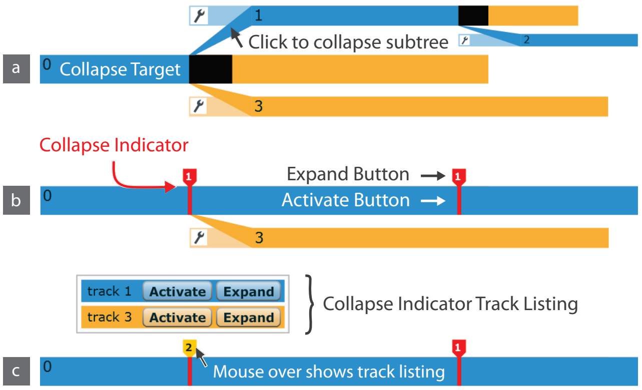

Fig. 6. Collapsing World Lines. (a) Clicking the incoming branch of a track triggers an animated collapse of a subtree into a collapse target. (b) Collapsed tracks are visually identified by a collapse indicator that is positioned at the start position of a collapsed track. (c) The track listing handles the case if several tracks share the same indicator.

6.4 State modification

Each track represents a simulation run and is assigned a set of input parameters that constitute this particular run. As an alternative to branching, users may decide to change the parameters of an existing track directly. This can be useful, if we quickly need to account for unexpected changes that are to be propagated to a whole subtree. When the simulation parameters change, the corresponding tracks and their descendants become invalid. This means that the progress of each affected component is reset to zero and all track icons are replaced by a black rectangular shape. The changed setting is automatically propagated to all of the track's descendants which do not specifically steer the same type of setting. For example, if we change the flow velocity in the parent run but the child changes geometry, it makes sense to propagate the new flow-velocity setting to the child. After the new settings have been propagated through the tree, we can re-simulate the invalidated components. Inline widgets are equipped with their own record button (Figure 5) to quickly start priority-based scheduling from their associated track.

6.5 Simplification and Collapsing World Lines

To simplify the visualization and to reduce potential clutter in the user interface, tracks can be collapsed into their parent track by clicking onto the incoming branch. In the collapsed status, all visual and interactive components are removed from the track. At the track's start time, a *collapse indicator* is drawn for each collapsed track (Figure 6). This element serves as a visual surrogate and maintains a visual separation to other tracks. This indicator consists of two interactive elements that enable expansion and activation of the collapsed track. Expansion is accomplished by clicking onto a marker-shaped button that is placed above the collapse target. The second interaction point is given by a vertical line button that spans the thickness of the collapse target. When clicked, the collapsed track is activated and elevated with respect to the z-index. Our previously defined z-indexing rule guarantees that the active layer of collapsed tracks is visually transfered to the collapse targets.

Special care has to be taken if several tracks have the same start time and are collapsed into a common target. The label of the expand button provides information on how many tracks are managed by the respective collapse indicator. We propose a table-like listing of collapse targets to let users expand or activate particular tracks of choice. This table appears above the collapse indicator as soon as the user moves the mouse cursor over the expand button (Figure 6c).

We have identified the visual priority queue as a method to rank tracks according to their vertical position in the tree canvas. The introduction of collapsed states requires an extension to this rule. Even though we believe, that most of the time, a collapsed track that is

Fig. 7. Screenshots of World Lines in the visualization mode. (a1-a3) Current time-step visualization. By default, all tracks that have a simulated frame at the current time step are included, i.e ordered and colored according to the visual mapping of analysis values with a user-defined transfer function (TF). The cursor has labels to annotate the results of the parallel analysis (here the percentage of flooded buildings). World Lines undergo a two-stage animated transition when jumping from time a to time b. The position is changed first, then the appearance is updated. (b) Frame-wise visualization. (c) Inline function graphs. (d) Combined visualization. (e) Current time-step visualization of brushed tracks only.

not part of the active World Line will be ignored and kept from re-simulation, it can be prioritized by its z-index. The table listing mentioned above sorts the collapsed tracks according to the z-index. The user can change this ordering by dragging rows of the table.

To simplify interaction the user can *flatten* a whole World Line. Starting from the leaf, all tracks of a World Line are recursively collapsed.

7 INTERACTIVE VISUAL ANALYSIS (IVA)

The idea of IVA in our context is to depict various attributes using multiple views and to allow the user to interactively select (brush) a subset of the data in these views. All corresponding data items in linked renderings are highlighted as well, providing the analyst with information about the interplay of the attributes involved [10]. World Lines currently supports brushing parallel worlds by selecting a set of tracks with a rubber-band tool. All brushed tracks are highlighted with a colored border (Figure 7e). In this way, users are able to select a subset of parallel worlds to visualize and analyze their outcome. World Lines enables a system that supports different types of analyses on simulation results. In the standard case of *per frame analysis*, the system is synchronized with the active frame, showing parameter setup and results (data values) for one track at the current time step. More advanced is the World Lines' capability for *per time step analysis* or *parallel analysis*. In this case a set of parallel frames is compared at a given time step. The multi-view system can be augmented with linked views to visualize analysis results. In the following subsections we explain how per frame and parallel analysis can also be directly displayed with World Lines.

7.1 Visualization Mode

The following basic properties of frames, tracks and branches are used to visualize data dimensions: color, opacity, thickness and position. When the user switches to the visualization mode, all interactive steering components are hidden and user-controlled layout is disabled. The arrangement and appearance of tracks is solely determined by the analysis values. Tracks without progress are not displayed in the visualization mode. We propose three different visualization methods for World Lines. Each of the suggested techniques is configurable through a transfer function that is linked to the World Lines view (Figure 7). In the visualization mode, the World Lines cursor shows a label to numerically present the analysis result at the current time step for each track. Using the transfer function (TF), an analysis result is mapped to a user-defined subset of the visual variables. For example, an analysis value can be mapped to the color and opacity of a single frame or a complete track.

Current time-step visualization (Figure 7a1-a3) To visualize information about the current time step, the tracks are rearranged, recolored and resized to visually present the quantity of interest for all selected frames. The transfer function value on the y-axis is used to sort and rank the tracks. The best simulation outcome is given by the highest evaluated value. Using automatic layout we can visualize this ordering of the tracks. If mapping to position is set, the visualization arranges the tracks according to their rank from top to bottom. The y-axis value can also be mapped to track thickness. In this way, the user is quickly informed about the best simulation setup at the current time step. Alternatively, the tracks can be arranged starting from the canvas center outwards in order to mimic a non-linear zoom effect. There are two reasons for a track not being part of this visual mapping. Either the track has no simulation result at the current time or it has not been brushed by the user. Excluded tracks receive a grey color at a low opacity value and are shown in the marginal area of the evaluated layout. The track layout as generated by the current time step visualization can be accepted as the layout for the steering mode. Consequently, the ordering in the visual priority queue for scheduling can be adapted from the results obtained by a comparative analysis.

Frame-wise visualization (Figure 7b) This mode has the purpose to show the evolution of the analysis result in time. Every frame stores the analysis value as obtained at the frame's time step and par-

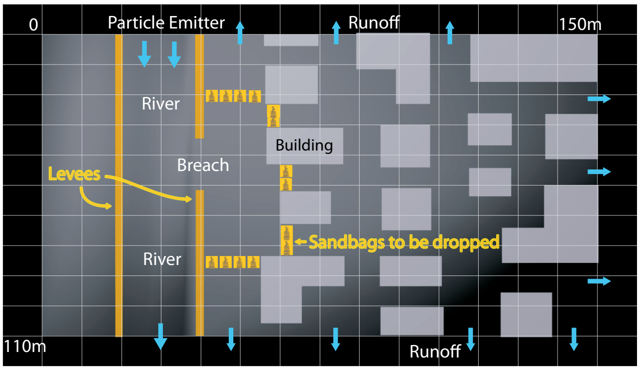

Fig. 8. Scenario setup explained with a screenshot of the bag designer. The bag designer is a domain-specific linked steering monitor that allows the user to dump sandbags at specific locations.

allel world. Each frame of each involved track is colored according to the transfer function mapping of the analysis result. The granularity of this visualization mode determines how many frames are combined for the display. Changing the granularity is not only useful because of efficiency but also to visualize temporal averages or other statistical quantities across many frames.

Inline Function Graphs (Figure 7c) As an alternative to the frame-wise visualization, we can use inline function graphs to inspect the temporal evolution of a result. A function graph is drawn into each track. At each frame, the vertical position of the graph is given by the transfer function mapping. Optionally, we can fill the areas below the graph to account for the user-defined color mapping. The frame-wise visualization or the inline function graphs can be combined with the current time step visualization (Figure 7d). For example, the visualization can be configured to reflect the results at the current time step via track arrangement and track opacity, combined with inline function graphs to give an overview on the temporal evolution.

7.2 Animated Transitions

Heer and Robertson [15] demonstrate that staged animated transitions can significantly improve graphical perception. World Lines is ideally suited to exploit this effect. We have implemented animated transitions in various stages of the World Lines view. In the steering mode, we animate the collapse-expand transitions. When a World Line is flattened recursively, we wait until the animated collapsing of one track is finished before continuing with the next step in the recursion. Each implemented layout algorithm uses animated transitions to guide the user through the change. The current time-step visualization applies smooth transformations to handle layout changes from one time step to the next (Figure 7). Here, we use two stages. First, the position is animated to let users perceive changes in track ranking. Secondly, we interpolate the analysis results and apply the transfer function at each iteration. This results in a smooth transition of track color, opacity and thickness according to the transfer function.

8 EVALUATION

In the following subsection we discuss an application of World Lines on a small flooding scenario.

8.1 Case Study

Our setup is based on a real-world levee-breach scenario as described by Sattar et al. [37]. Using a laboratory model, the authors have investigated various possible methods for breach closure, utilizing procedures such as single- and multi-barrier embankments with different ways of positioning sandbags. Our simplified site (Figure 8) comprises a river that is located next to a neighborhood of 19 buildings. Levees have been built to protect the houses from flooding. We assume that a severe weather condition results in a breach of the levee and floods the city. The following case study shows how World Lines allows to analyze the situation and design a breach closure.

We utilize SPH to simulate fluid behavior. The flowing river has been modeled with a particle emitter. The rate of the particle emitter and the initial particle speed are used to control the water-flow velocity. Outside the scene particles are discarded to model the runoff of the water. In this situation the neighborhood can be secured from flooding by construction of a breach enclosure. The neighborhood has to be protected by dumping sandbags of two different weights, heavy bags (5000kg) and light bags (3500kg), which can be carried by a helicopter. Barrier construction is restricted to sequential dumping of sandbags (Figure 10, S1-a), since a helicopter dumps one bag at a time. The goal of this case study is not accuracy in fluid simulation but to show that our approach enables a fast and intuitive exploration process. For this purpose, we configure the simulation to operate in real-time. All sections in the simulation scene are non-erodible and non-porous. We model sandbags as bricks that cannot be deformed but assemble well. We use approximately 22000 particles for simulating the flooding. This means, we are able to test the stability of the barriers with a simulation of at most 10 minutes of particle-propagation time.

Using World Lines, we attempt to find a multibarrier system that is capable to protect the city with respect to the following criteria: The number of flooded buildings has to be reduced as fast as possible. The number of dumped sandbags is to be kept as low as possible. Barriers that consist of lighter bags are easier and faster to assemble, we therefore look for a solution that utilizes as few heavy bags as possible. The multi-barrier system needs to be stable even if the river velocity increases.

In this case study, World Lines is linked to two views. One view shows an orthographic birds-eye perspective on the scene to act as a steering monitor that enables user-defined positioning of sandbags (Figure 8). Context-menu interaction and mouse-drag operations are employed to determine the exact location of dumping. Sequential dumping reflects the ordering of bag placement. A second view shows a 3D rendering of the scene. Particles are rendered in real-time using a fast GPU surface extraction technique [40]. In addition, the view is capable to render comparative information obtained from multiple simulation runs. Buildings can be colored in case they are flooded beyond a user-defined warning level, in any of the brushed tracks of the World Lines view.

The given problem turned out to be quite challenging. To our knowledge, there is no alternative framework that allows a complex exploration as was needed. In total, 59 simulation runs had to be performed and inspected. Figure 9 shows the final appearance of World Lines when zoomed out. We have identified 7 major phases that led to the final solution. Figure 10 displays important results obtained from the 3D view at different stages of the process. In stage S0 the city is flooded in about 2.5 minutes when a part of the levees breaks down. In stage S1, a brute force approach is tested. Heavy bags are placed directly at the gap. This embankment turned out to be unstable and caused a re-flooding of the buildings. As a consequence, a multitude of different dumping techniques have been investigated in stage S2 which we refer to as the exploration stage. A couple of methods as proposed by Sattar et al. [37] have been studied where buildings are used as part of the multi-barrier systems. We have found that the ordering and timing of the construction is crucial, making World Lines ideally suited to test these procedures. In all cases of stage S2, the water pressure on the embankments becomes too high as soon as more than four embankments are present. Bags were washed away and barriers collapsed. A major breaktrough (S3) was possible by going back in time and placing a spur of heavy bags from within the river towards the front row of the buildings (left barrier of image S3 in Figure 10). This barrier deflects the flow and reduces flow velocity through the breach. After another set of trials, it was found that a second spur further improves the flow situation (S4). In stage S5, using the knowledge gained until this point, it was possible to quickly build a stable solution (track 50, S6-a in Figure 10) that utilizes barriers of light bags in the second row of the buildings. The final stage S6 was concerned with the optimization of this solution. Track 51 (S6-b in Figure 10) re-

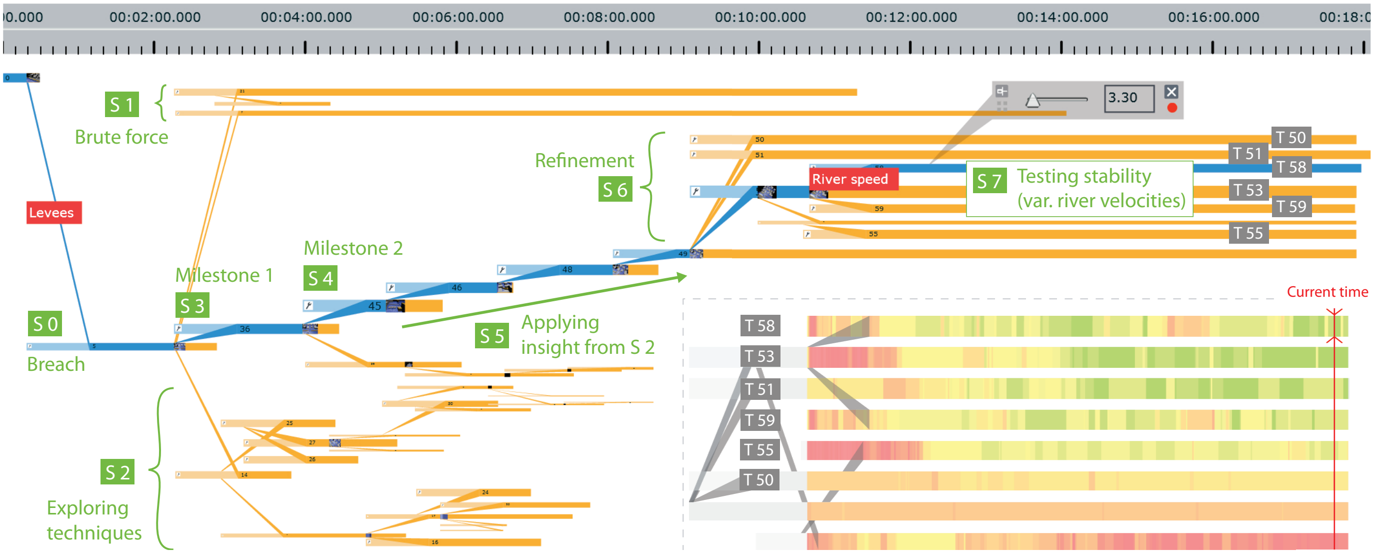

Fig. 9. Final structure of World Lines as a result of an exploration process that involved 59 simulation runs. We have identified 7 stages that led to the final solution. The lower right corner shows a screenshot of the same World Lines in the visualization mode (embedded in the figure only). This image analyzes the percentage of flooded buildings using a combined current time step and frame-wise visualization of all tracks that were investigated during the refinement stage.

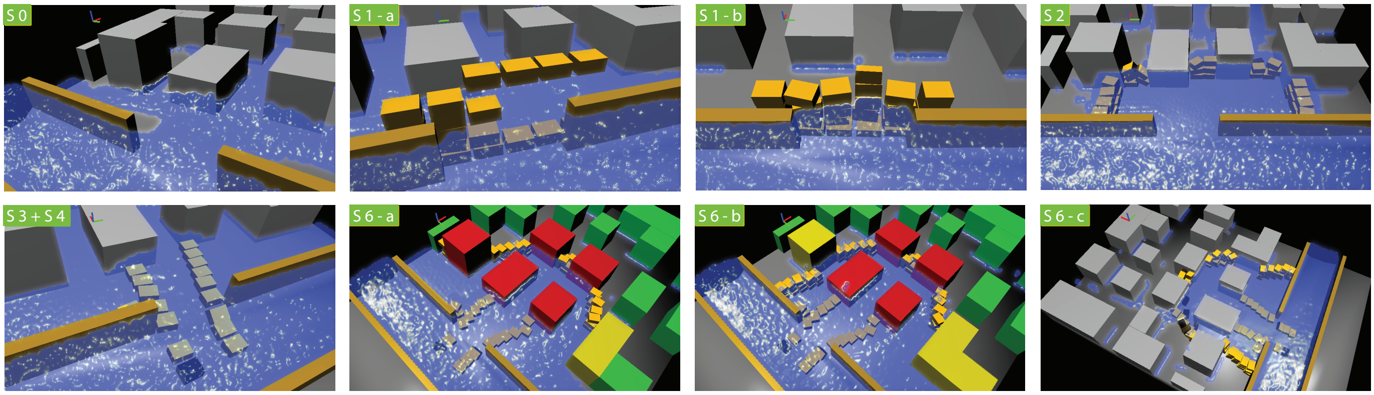

Fig. 10. Renderings of the scene made during the exploration process. (S0) Levee breach causes city flooding. (S1-a) Sequential dumping. (S1-b) Initial brute-force approach fails, the barrier collapses. (S2) One of many multi-barrier systems tested in the exploration phase, bags are washed away. (S3+S4) Breakthrough with heavy, submerged bags. (S6-a) Stable solution. (S6-b) Final solution saves more buildings. (S6-c) Unstable attempt to further close the downstream parts.

places one embankment by a barrier of light bags closer to the breach in order to protect an additional building. Track 53 utilizes heavy bags to construct the same barrier. Track 55 represents the attempt to fully enclose the water by placing another embankment at the downstream side of the river (S6-c in Figure 10). To find out which of these tracks performs best, we have elongated their base layers and simulated a longer time span using per-time step scheduling. This approach has revealed that the barrier systems are unstable except for those given by track 51 and track 53, pointing to track 53 as the optimal outcome. Finally, we had to prove whether the found solution remains stable even if we increase the velocity of the river (S7). We have branched off twice with different river velocities to create track 58 and track 59. Another scheduled simulation has generated the frames needed to make a prediction. To quickly visualize whether all the cases are stable, we have brushed track 53, 58 and 59 in order to analyze the percentage of flooded buildings. The linked 3D view gives comparative information on the brushed tracks. A building is colored in red if it is flooded, or in yellow if it is in danger, in **any** of the brushed simulation runs (Figure 11). A final combined frame-wise visualization (Figure 9) clearly indicates that the multi-barrier system of track 53 is the best and most stable solution found in the exploration process.

8.2 Domain expert feedback

The World Lines case study was assessed by an expert with experience with flood forecasting and management systems [4]. He emphasized the importance of flood management plans, in particular as required by the new EU flood management directive [7]. Currently, comparative analyses of placing barrier systems are rarely done. Placement is usually based on the experience of the flood management staff although hydrodynamic simulation models and laboratory models could be used for this purpose. The expert was impressed by the ability of the system to concurrently simulate several scenarios as this would greatly enhance the ease with which alternative management strategies can be compared. He considered both the navigation and the steering to be intuitive, in particular the embedded widgets. Additional visualization options for comparing the scenarios would be useful, such as difference maps of water levels and sand bag locations. In a practical context, World Lines could be used in two ways. First, and more importantly, it could be an efficient tool for training flood management staff in an off-line mode. The most important issue in these types of management decisions is not to find an optimum solution but to rapidly exclude poorly performing solutions. The goal is to identify robust placing schemes that will work for a range of boundary condi-

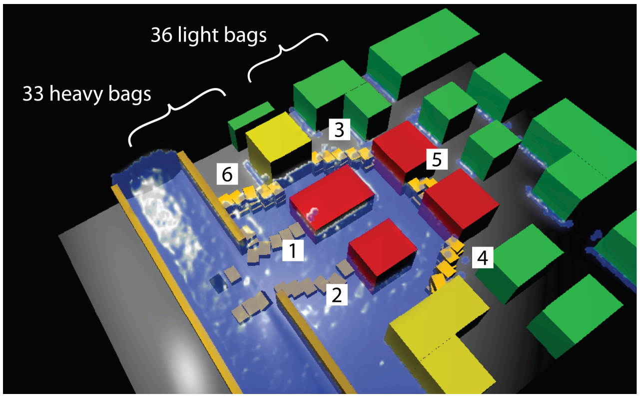

Fig. 11. Final solution. 70% of the buildings are protected by a multi-barrier system that consists of 6 embankments. The ordering in which the barriers are to be constructed is depicted through the numbers. The buildings are colored according to the stability tests in S7 to visualize that the found solution is stable in any of the brushed tracks.

tions most of which are not well known. Second, the system shows potential for real-time application during a flood event.

9 IMPLEMENTATION

We have implemented World Lines as a module of our pluggable steering and visualization system *Visdom*. This framework comprises a client-server architecture to enable control over the web. The server is written in C++ and uses the GPU (CUDA) to handle the compute intensive parts of simulation and rendering. The client is an Adobe Air application that is built upon the Flex [2] framework. Interaction and steering is accomplished via interfaces on the client. The World Lines view is completely implemented as a module of the client to take advantage of the scripting capabilities in Flex. We make use of features such as XML handling, style sheets and animated state transitions. Client and server are connected via a permanent real-time socket. Information is exchanged in XML (settings) and binary format (renderings).

Each track has a specific simulation setup. Simple input parameters such as the position of the barriers are stored in an XML format on the client. For efficiency reasons we do not store the entire system setup within a track but only the changes with respect to the parent track. We consider more complex modifications such as deformations of simulation boundaries as a change in the system state rather than a change in the input parameter setup. These states require the storage of larger data structures and are kept along with the simulation results in a data management facility on the server. In this data container, each state is uniquely defined by a track identifier and a time value. Often it is not necessary to store the status at every iteration of the underlying simulation. Therefore the frame size in the World Lines view need not be identical to the time-step size of the simulation. In our case study it is sufficient to have a frame size (1s) that is 60 times larger than the internal time-step size. State saving and retrievement accumulates to 26 ms on an Intel Core 2 Quad processor with 2.4GHz and 4GB RAM.

World Lines operates on interfaces that can be implemented to steer external simulation modules. The server can also be installed on a high-performance computer. A module that implements the simulation interface would handle the distribution of simulation work among the clusters. For this work, we have written a plugin to adapt the PhysX simulation engine [34] which simulates one frame (1s) of our case study in 666ms on a GeForce 8800GTX graphics card with 768MB memory. The 3D view renders one image in 22ms. The parallel analysis module has been written in CUDA to provide fast analysis results (50 ms for 20000 particles). The client uses a caching mechanism for the results to enable smooth frame-wise visualization.

10 DISCUSSION, FUTURE WORK AND CONCLUSIONS

In this paper we have described an interactive view based on the concept of World Lines that enables the user to steer, analyze, and compare multiple related simulation runs. In our current project we envision the real-time management and predictive planning for flooding events as the final application of the presented system. There is a multitude of open questions remaining. Especially the fidelity of the inundation simulation is not sufficient yet. Nevertheless, we believe the current system demonstrates the usefulness of our approach. Also, it is easy to think of other application areas: industrial prototyping or surgery planning could also benefit from a system that allows to compare design decisions interactively. In order to substantiate the practical relevance of the suggested research we are planning to perform an in depth user study with simulation experts from the field.

One shortcoming of the current implementation is that simulation and analysis cannot run concurrently, sharing the available CPUs and GPUs. In a problem-solving environment it is important to detect wrong decisions or designs quickly and to terminate corresponding simulation runs as soon as possible. We plan to improve the current implementation in a way that new simulation steps are computed in background processes and to allow interactive visual analysis as soon as simulation data is available. Other important issues are related to scalability and memory efficiency. For practical applicability it is important to be able to steer simulations generated on a large number of GPUs in parallel. Currently all GPUs have to reside on a single machine. It is also an open question how to automatically decide which time steps are stored and which can be discarded to save memory. The current system works with a simple user-defined stride such that only every n-th step of the simulation is actually stored for analysis. Our case study has revealed another shortcoming of the system. The analysis would have greatly benefitted from the ability to clone subtrees from one track to another, giving the ability to reuse barrier arrangements.

World Lines opens up a variety of interesting topics for future research that are related to the analysis of parallel worlds. We need innovative comparative views to display the interplay of simulation parameters and simulation outcomes. World Lines is also an ideal interface to handle large parameter studies. The user can select a range of parameter values which he needs to analyze and multiple branches are created at the active track. For now, we have presented a steering process that is fully controlled by the user. When automatic track creation, as in the case of parameter studies, comes into play, scalability can be an issue. Automatic track layout, advanced track folding, as well as navigation will be an important research topic as soon as a very large number of World Lines has to be managed. If necessary, layout algorithms have to be ported to run on the GPU. To maintain interactivity even if a large number of runs is simulated in parallel, we need novel approaches for the user to define what he is interested in.

Another route for future work are interaction approaches based on multiple cursors. For example, a simulation cursor follows the simulation in the background, while multiple visualization cursors are used to perform comparative analysis on the existing time steps. It is also interesting to research extended linking and brushing concepts for World Lines where the user can brush inside individual frames, over multiple frames or over multiple parallel scenarios.

The lyrics of 'Brothers in Arms' by Dire Straits include *"There's so many different worlds. So many different suns. And we have just one world. But we live in different ones."* World Lines will help us to see beyond.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the Austrian Science Fund (FWF):P 22542-N23 (Semantic Steering) and partially funded by the Swiss National Science Foundation under grant 200021_127022. The project SemSeg acknowledges the financial support of the Future and Emerging Technologies (FET) programme of the European Commission, grant number 226042.

REFERENCES

[1] *Zeittafel der Weltgeschichte. Den letzen 6000 Jahren auf der Spur*. Ullmann/Tandem, 2001.

[2] Adobe Systems Incorporated. Flex: An open source framework for developing web applications. `http://www.adobe.com/products/flex/` (last visited on 3 August 2010).

[3] W. Aigner, S. Miksch, B. Thurnher, and S. Biffl. Planning lines: Novel glyphs for representing temporal uncertainties and their evaluation. In *Proceedings IEEE Symposium on Information Visualization 2005 (InfoVis 2005)*, 2005.

[4] G. Blöschl. Flood warning - on the value of local information. *Intl. J. River Basin Management*, 6(1):41–50, 2008.

[5] K. Brodlie, L. Brankin, A. Poon, G. Banecki, H. Wright, and A. Gay. GRASPARC - A problem solving environment integrating computation and visualization. In *Proceedings IEEE Visualization 1993*, pages 102–109, 1993.

[6] H. Doleisch, M. Gasser, and H. Hauser. SimVis: Interactive visual analysis of large and time-dependent 3D simulation data. In *Proceedings of the 2007 Winter Conference on Simulation*, pages 712–720, 2007.

[7] EU. Directive 2007/60/EC of the European Parliament and of the Council of 23 October 2007 on the assessment and management of flood risks. Official Journal of the European Union L 288/27., 2007.

[8] R. P. Feynman. Space-time approach to non-relativistic quantum mechanics. *Rev. Mod. Phys.*, 20(2):367–387, 1948.

[9] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.

[10] R. Fuchs, J. Waser, and M. E. Gröller. Visual human+machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1327–1334, Oct. 2009.

[11] H. L. Gantt. Work, wages and profit. *The Engineering Magazine, New York*, 1910. republished as Work Wages and Profits, Easton, Pennsylvania, Hive Publishing Company, 1974.

[12] J. N. Ghazali and A. Kamsin. A real time simulation and modeling of flood hazard. In *Proceedings of the 12th WSEAS international conference on Systems*, pages 438–443, 2008.

[13] M. C. Hao, D. A. Keim, U. Dayal, and J. Schneidewind. Business process impact visualization and anomaly detection. *Information Visualization*, 5(1):15–27, 2006.

[14] H. Hauser. Generalizing focus+context visualization. In G.-P. Bonneau, T. Ertl, and G. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 305–327. Springer Berlin Heidelberg, 2005.

[15] J. Heer and G. G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

[16] C. Henze. Feature detection in linked derived spaces. In *Proceedings IEEE Visualization 1998*, pages 87–94, 1998.

[17] A. Hérault, G. Bilotta, and R. A. Dalrymple. GPU-SPHysics: A GPU-based Smoothed Particle Hydrodynamics model for free surface flows. `http://www.ce.jhu.edu/dalrymple/GPU/` (last visited on 3 August 2010), 2008.

[18] R. Hoetzlein. Fluids v.2 - A Fast, Open Source, Fluid Simulator. `http://www.rchoetzlein.com/eng/graphics/fluids.htm` (last visited on 3 August 2010).

[19] C. Johnson. Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, July-Aug. 2004.

[20] C. Johnson, S. G. Parker, C. Hansen, G. L. Kindlmann, and Y. Livnat. Interactive simulation and visualization. *Computer*, 32(12):59–65, 1999.

[21] P. Kipfer and R. Westermann. Realistic and interactive simulation of rivers. In *ACM Proceedings of Graphics Interface 2006*, pages 41–48, 2006.

[22] R. Kosara and S. Miksch. Visualization techniques for time-oriented, skeletal plans in medical therapy planning. In *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM 1999)*, pages 291–300, 1999.

[23] R. Kosara and S. Miksch. Metaphors of movement: A visualization and user interface for time-oriented, skeletal plans. In *Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine*, pages 111–131, 2001.

[24] P. Koumoutsakos, G.-H. Cottet, and D. Rossinelli. Flow simulations using particles: bridging computer graphics and CFD. In *ACM SIGGRAPH 2008 classes*, pages 1–73, 2008.

[25] D. C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[26] K.-L. Ma. Image graphs - a novel approach to visual data exploration. In *Proceedings IEEE Visualization 1999*, pages 81–88, 1999.

[27] K. Matković, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008.

[28] K. Matković, H. Hauser, R. Sainitzer, and M. E. Gröller. Process visualization with levels of detail. In *Proceedings IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, pages 67–70, 2002.

[29] H. Minkowski. Raum und Zeit. *Physikalische Zeitschrift*, 10:104–111, 1909. (Lecture delivered before the Versammlung Deutscher Naturforscher und Ärzte, Cologne, September 21, 1908.) Reprinted in Blumenthal 1913. English translation in Lorentz et al. 1952. Page numbers refer to this last edition.

[30] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68:1703–1759, 2005.

[31] J. D. Mulder, J. J. van Wijk, and R. van Liere. A survey of compuational steering environments. *Future Generation Computer Systems*, 15:119–129, 1999.

[32] M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. In *ACM SIGGRAPH Symposium on Computer Animation (SCA)*, pages 154–159, 2003.

[33] T. M. Nguyen, J. Schiefer, and A. M. Tjoa. Sense & response service architecture (saresa): an approach towards a real-time business intelligence solution and its use for a fraud detection application. In *DOLAP '05: Proceedings of the 8th ACM international workshop on Data warehousing and OLAP*, pages 77–86, 2005.

[34] NVidia Corporation. PhysX: Physics Simulation Toolkit. `http://developer.nvidia.com/object/physx.htm` (last visited on 3 August 2010).

[35] S. Rinderle, R. Bobrik, M. Reichert, and T. Bauer. Business process visualization - use cases, challenges, solutions. In *ICEIS 2006 - Proceedings of the Eigth International Conference on Enterprise Information Systems: Databases and Information Systems Integration, Paphos, Cyprus, May 23-27, 2006*, pages 204–211, 2006.

[36] E. A. Rundensteiner, M. O. Ward, J. Yang, and P. R. Doshi. XmdvTool: visual interactive data exploration and trend discovery of high-dimensional data sets. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 631–631, 2002.

[37] A. Sattar, A. Kassem, and M. Chaudhry. 17th street canal breach closure procedures. *Journal of Hydraulic Engineering*, 134(11):1547–1558, 2008.

[38] C. T. Silva, J. Freire, and S. P. Callahan. Provenance for visualizations: Reproducibility and beyond. *Computing in Science and Engineering*, 9(5):82–89, 2007.

[39] M. Suntinger, H. Obweger, J. Schiefer, and M. E. Gröller. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. In *Proceedings IEEE PacificVis 2008*, pages 111–118, 2008.

[40] W. J. van der Laan, S. Green, and M. Sainz. Screen space fluid rendering with curvature flow. In *Proceedings of the Symposium on Interactive 3D Graphics and Games 2009 (I3D 2009)*, pages 91–98, 2009.

[41] R. van Liere. Computational steering. In *High-Performance Computing and Networking*, pages 696–702. Springer-Verlag, 1996.

[42] C. van Treeck, P. Wenisch, A. Borrmann, M. Pfaffinger, M. Egger, O. Wenisch, and E. Rank. Towards interactive indoor thermal comfort simulation. In *European Conference on Computational Fluid Dynamics*, 2006.

[43] C. Weaver. Building highly-coordinated visualizations in improvise. In *Proceedings IEEE Symposium on Information Visualization 2004 (InfoVis 2004)*, pages 159–166, 2004.

[44] C. Weaver. Visualizing coordination in situ. In *Proceedings IEEE Symposium on Information Visualization 2005 (InfoVis 2005)*, pages 165–172, 2005.

[45] H. Wright and J. Walton. Hyperscribe: A data management facility for the dataflow visualization pipeline. In *IRIS Explorer Technical Report IETR/4, NAG Ltd*, 1996.