Visual Analysis and Steering of Flooding Simulations

Hrvoje Ribičić, Jürgen Waser, Raphael Fuchs, Günter Blöschl, and Eduard Gröller, *Member*, *IEEE Computer Society*

Abstract—We present a visualization tool for the real-time analysis of interactively steered ensemble-simulation runs, and apply it to flooding simulations. Simulations are performed on-the-fly, generating large quantities of data. The user wants to make sense of the data as it is created. The tool facilitates understanding of what happens in all scenarios, where important events occur, and how simulation runs are related. We combine different approaches to achieve this goal. To maintain an overview, data are aggregated and embedded into the simulation rendering, showing trends, outliers, and robustness. For a detailed view, we use information-visualization views and interactive visual analysis techniques. A selection mechanism connects the two approaches. Points of interest are selected by clicking on aggregates, supplying data for visual analysis. This allows the user to maintain an overview of the ensemble and perform analysis even as new data are supplied through simulation steering. Unexpected or unwanted developments are detected easily, and the user can focus the exploration on them. The solution was evaluated with two case studies focusing on placing and testing flood defense measures. Both were evaluated by a consortium of flood simulation and defense experts, who found the system to be both intuitive and relevant.

Index Terms—Data aggregation, problem solving environment, interactive visual analysis, decision making, uncertainty visualization, simulation steering

1 Introduction

When uncertainty is present in simulations, ensemble simulations are often used to examine what might happen depending on unknown parameter values [1], [2]. In flood simulations, the uncertainty might come from unknown initial parameters, e.g., where a dam will break, or from the parameters under the user's control, e.g., the way protection barriers are placed. Sampling the expected parameter space yields multiple simulation runs, each of which represents one sequence of events. Understanding such an ensemble data set is vital to finding robust solutions that work regardless of the uncertainty involved.

As Fig. 1 illustrates, what makes understanding difficult is the sheer quantity of data present. While understanding a single simulation run presents its own set of challenges (Fig. 1a), an ensemble simulation only multiplies these (Fig. 1b). A promising approach to reduce the quantity of data is interactive steering. Interactive steering allows the user to select the next set of parameters to be simulated based on conclusions drawn from already present data, incrementally exploring the parameter space. Two examples of interactive steering are World Lines [3] and ComVis [4].

Visdom [5] uses the World Lines metaphor to control and integrate simulation and visualization. The parameter space is shown as a 2D tree of gray-colored lines called tracks, each of which represents a simulation run with a different set of simulation parameters (see Fig. 1). The horizontal dimension of the tree corresponds to the time that has passed, while the tracks are distributed vertically. The tree begins with one root track, which represents the initial parameters of the simulation. All other tracks are created when a change in simulation parameters is introduced to an existing simulation state. The change spawns a new simulation run with different parameters, which continues from the previous state.

The simulation states are shown as frames—rectangles that are the building blocks of tracks. By selecting a frame, the simulation data are forwarded to views, which render and display a representation of the simulation state. While this mechanism serves well for examining individual states, World Lines offers only limited support for analyzing the relationships of states. The visualization mode allows the coloring of frames according to some measure extracted from matching simulation states, but a very limited amount of information can be shown in this way. More advanced displays catering to ensemble simulations have been presented in Nodes on Ropes [1], but they are geared toward monitoring a simulation rather than analyzing it.

On the other hand, ComVis has impressive analysis capabilities provided by its information-visualization views. Linking and brushing are used to discover correlations between output and input variables. Once an interesting subset of parameters has been identified, it can be sampled in higher resolution. The necessary simulation runs are

- *H. Ribičić and J. Waser are with the VRVis Forschungs-GmbH, Donau-City-Straße 1, 1220 Wien, Austria. E-mail: {ribicic, jwaser}@vrvis.at.*
- *R. Fuchs is with the Scientific Visualization Group ETH-Zentrum, Information Technology and Education, ETH Zürich, CAB G 66.2, Universitätsstrasse 6, 8092 Zürich, Switzerland. E-mail: raphael.fuchs1@gmail.com.*
- *G. Blöschl is with the Institut für Wasserbau und Ingenieurhydrologie, Technische Universität Wien, Karlsplatz 13/222, A-1040 Wien, Austria. E-mail: bloeschl@hydro.tuwien.ac.at.*
- *E. Gröller is with the Institut für Computergraphik und Algorithmen, Favoritenstrasse 9-11 / E186, A-1040 Wien, Austria. E-mail: groeller@cg.tuwien.ac.at.*

Manuscript received 27 Oct. 2011; revised 8 May 2012; accepted 31 July 2012; published online 22 Aug. 2012.
Recommended for acceptance by D. Weiskopf.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-10-0263.
Digital Object Identifier no. 10.1109/TVCG.2012.175.

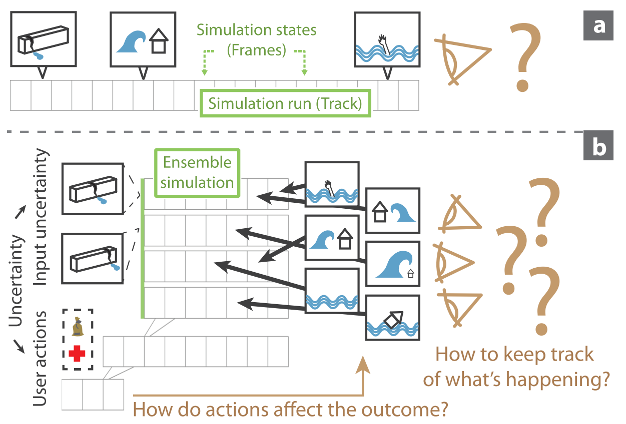

Fig. 1. (a) The analysis of a single simulation can be a challenging task, as the user must identify where and when key events happen. (b) In ensemble simulations, finding the interesting events and relating them to the parameters that caused them is even more difficult.

performed offline, and loaded into the system alongside existing data at a later time. The many available views allow for a better understanding of the effects of different parameters. However, the tool lacks flexibility when compared to World Lines. The parameter space allows only for isolated numeric parameters and results, discarding spatial and connectivity information. More importantly, parameter changes cannot be introduced during a run, complicating the interactive steering of simulation scenarios.

Both of these tools have their advantages and disadvantages. Visdom provides a more intuitive exploration of a greater variety of parameter spaces, while ComVis allows the user to perform an in-depth analysis of the parameter space. The goal of this paper is to unite the two approaches, allowing interactive visual analysis to be performed on ensemble data in real time. Our solution is shown in Fig. 2. We follow Schneiderman's mantra of visual exploration: overview first, zoom and filter, then details on demand [6]. The steering is done using World Lines. To show an overview, multiple simulation states are combined to create aggregate renderings (Fig. 2a). An aggregate rendering is a single-state rendering that is enriched with information aggregated from multiple states. The information is represented visually, allowing the user to understand trends more easily. One example of an aggregate rendering is coloring the fluid according to the standard deviation of the velocity in all simulation states.

Once an aggregate is in place (Fig. 2b), the user can examine it to find areas of interest, such as areas with especially high deviation. We provide a selection mechanism for the user to designate these areas. Detailed information about the selected areas can be shown in information-visualization views. Through linking and brushing the user can investigate the relationships of values, and identify the parameters that produced them. For example, it is possible to find out which parameters result in a high velocity by brushing a histogram, and seeing the related states brushed in World Lines.

As the user learns from the results of the exploration, other approaches and solutions emerge. They can be explored immediately thanks to the steering capabilities of the system (Fig. 2c). To aid the user in understanding what has changed, the views update instantly. The information-visualization views show how the points of interest have been affected by the change. Side effects in the form of unexpected developments can still be spotted using aggregate renderings.

To demonstrate our approach, we first use a synthetic case-study scenario in which a city is threatened by a levee breach. This case study is modeled after an actual flooding event and the authorities' response. Its simplicity serves to illustrate the principles of our solution well. Uncertainty is present with respect to the initial breach location, and the defenses that can be deployed. Our approach is used first to understand the uncertainty present in the simulation scenario, and then to examine the effects of possible user

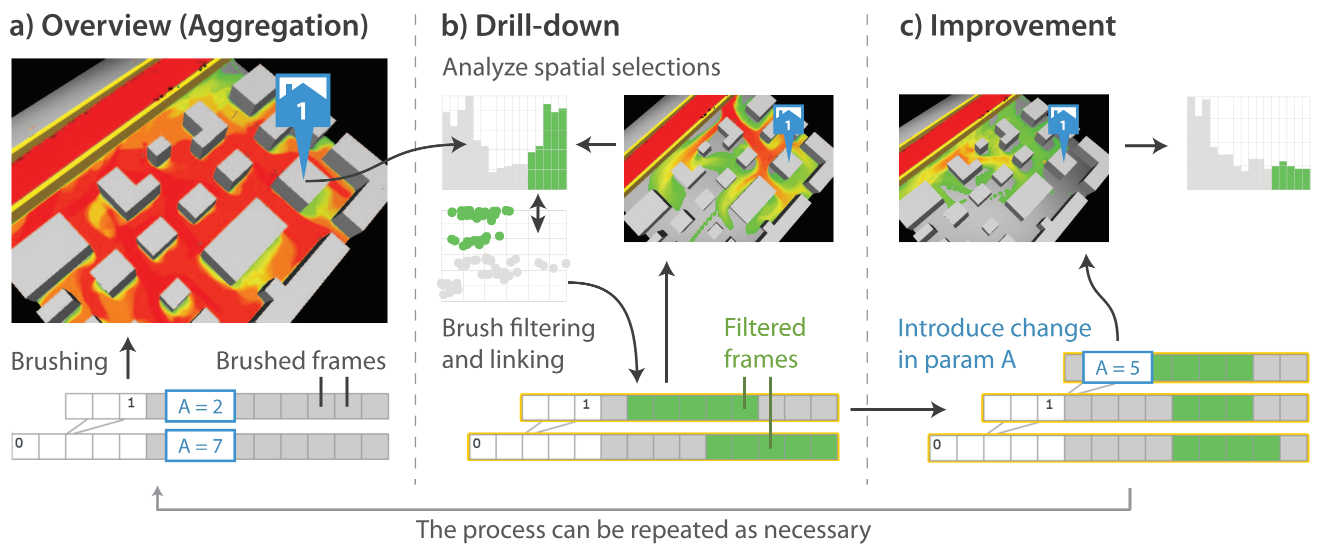

Fig. 2. The proposed solution for performing an analysis of an ensemble flooding simulation. (a) The aggregations help the user understand what is happening in several selected simulation states (shown as dark gray) at once. (b) When details are needed, selections (blue icon in the image) provide data to the information-visualization views. By brushing values of interest, the states associated with these values are highlighted, becoming green. (c) The views allow the user to see the effects of any changes introduced. The user can continue filtering and introducing changes as necessary.

actions. By using the filtering mechanisms, we find a solution that is robust enough for all possible breach locations.

The synthetic case study serves as a useful introduction to the properties of our solution, which is why it was presented to a group of experts in the field of hydrology for an evaluation. They provided comments which allowed us to improve the system, but they also supplied us with the data and models necessary to construct a second case study. In this flood-protection related study, which we will refer to as the real-world case study, a town is protected from floods by mobile walls. The uncertainty again lies with the location, where these defenses fail. However, the goal is not to design defenses, but to find out which areas should be evacuated and cleared to prevent most of the damage. This case study was presented to an extended group of potential users, whose comments we describe in some detail.

In summary, the scientific contributions of this paper are:

- Aggregate renderings which allow information from multiple states to be shown in one rendering.
- The selection mechanisms necessary to bridge aggregation and analysis, and matching information-visualization views.
- An integrated steering enviroment supporting a dynamic drill-down process throughout which new data can be generated using improved parameters.
- An evaluation of the methods by a group of experts and potential users.

2 Related Work

As the development of hardware has made the execution of ensemble simulations feasible, the visualization community has produced various accompanying solutions. However, none of them appear to be geared toward interactive steering. In this section, we explore the qualities of similar solutions and the works that inspired our approach.

The visualization of multifield scientific data is an extensive field of research. As an ensemble might be thought of as a collection of fields, there exists a similarity between the two problems. For a general overview, we refer to a recent state-of-the-art report [7].

One of the traditional approaches to visualize time series involves locating and showing interesting and representative timesteps. Bruckner and Möller [8] extend this approach to ensembles. Simulated states are clustered according to a similarity measure. The ensemble is represented as a directed graph of keyframes. In the context of interactive steering, the disadvantage of the approach is that the structure derived from user choices cannot be preserved when ordering states according to similarity.

Interactive visual analysis is a common and successful way of approaching simulation problems. Doleisch et al. [9], [10] suggest linking and brushing in information-visualization views to analyze computational fluid dynamics (CFD) simulation data. To aid the user in understanding the data, multiple coordinated information visualizations [11] are linked to the three-dimensional rendering. This displays the spatial arrangement of the data in the simulation. Data derivations and brushing are used in linked windows to allow the user to filter data elements flexibly. The previously mentioned ComVis [4] applies these techniques to ensembles.

The general concept of using aggregates to summarize data is not new. Database systems and data mining systems [12] support aggregation of information stored in tables. The aggregate manipulator [13] performs aggregation by grouping data elements in a database on their attributes. In this sense, aggregation by grouping is directly supported by the SQL query language. The Data Cube (also OLAP Cube) [14] describes multidimensional data as a multidimensional cube, where data can be selected by choosing one or multiple axes of the cube. It is then possible to use SQL aggregate functions [15] on the selected dimensions of the cube. Stolte et al. [16] apply this approach for high-dimensional data visualization.

Other types of data reduction have been used to create novel visualizations. Malik et al. [17] compare CT scans made at different resolutions by interleaving the results within the same image. Woodring and Shen [18] visualize a volume time series by applying a transfer function independently to all time steps and using projection along the time dimension to produce a single color volume or chronovolume. Balabanian et al. [19] refine the approach by allowing different styles and composition methods to be chosen. Woodring and Shen [20] employ user-specified expressions to combine values from different fields.

Akiba et al. [21] aggregate histogram information over time and highlight minimum and maximum values. Meyer et al. [22] use grouping and aggregation for gene expression data. Correspondence is defined by a nearest neighbor criterion and the aggregation uses a minimum operator.

Potter et al. [2] discuss the visualization of ensemble data and suggest spatial aggregation using mean, standard deviation, minima, and maxima for each position over the whole ensemble in multiple views. This work is perhaps the biggest inspiration for this paper, as both spatial selection and additional charts are present in conjunction with aggregation. However, the paper deals with 2D fields and histograms only.

The discussion of related work shows that while many components used in this paper have been presented before, there have been no attempts to integrate them into a greater whole. In this paper, we suggest a generalization of aggregation mechanisms so that they can be used with selection and filtering mechanisms.

3 Aggregate Renderings

As the previous section has shown, aggregation is commonly used to reduce or simplify the data. However, it is used in either information-visualization views, or dedicated visualizations. In a simulation, such as the flooding scenario, there are multiple objects and elements of interest given, represented by different data structures. While it would be possible to use some of the previously mentioned techniques to create linked views with separate aggregations, it would be hard to imagine such a complex layout functioning as an intuitive overview. Instead, we visualize aggregations within the existing 3D view.

There are many different aspects of the simulation, which are interesting to aggregate. These include raw data such as the simulated fluid attributes, derived measures such as the risk to buildings, and visualizations themselves,

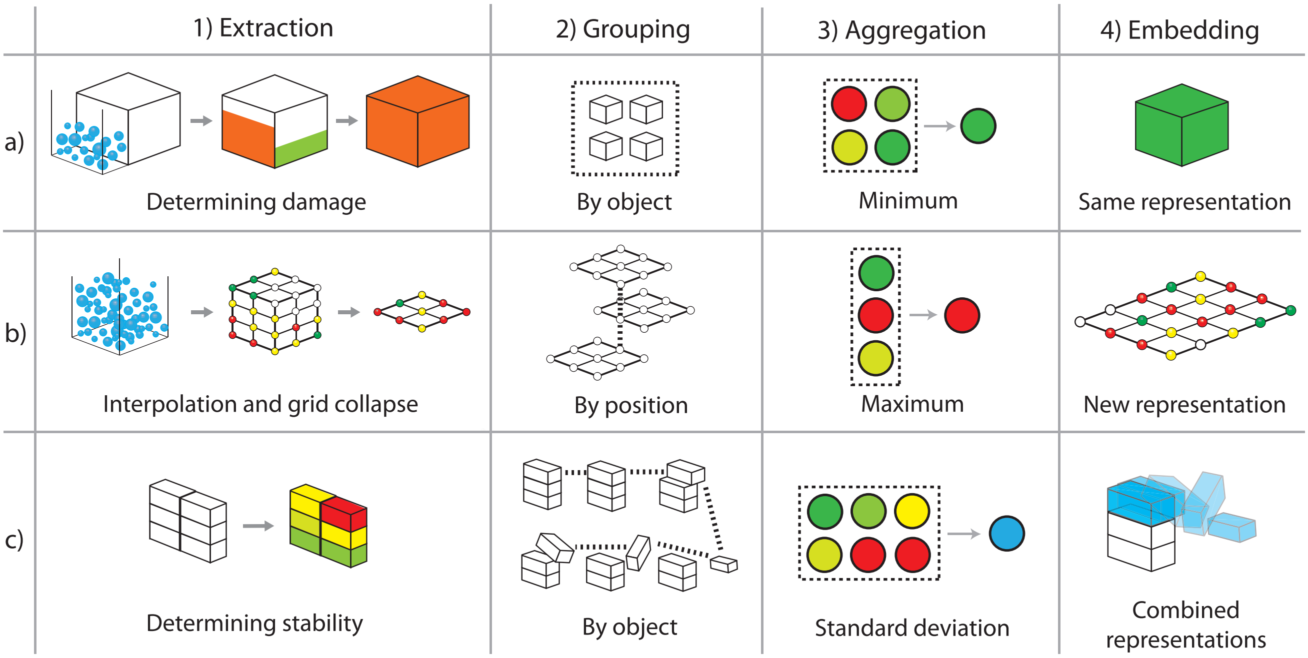

Fig. 3. The various types of aggregation: (a) Object aggregation (buildings). (b) Field aggregation (fluid). (c) Instance aggregation (bags). The types of aggregation are shown progressing through the four phases of an aggregated rendering: (1) Extraction, (2) Grouping, (3) Aggregation, (4) Embedding. The spectrum of colors from green to red indicates increasing values.

such as glyphs. Despite their diverse nature, aggregate renderings representing these aspects can be created using a unified approach. The approach can be divided into four phases (columns of Fig. 3):

1. Extraction, where the simulation data of each state is processed to create the data to be aggregated.
2. Grouping, where groups of corresponding values from different states are formed.
3. Aggregation, where the values within groups are combined, using an aggregation operator.
4. Embedding, where the per-group values are translated into renderable objects or object properties.

The creation of an aggregated rendering begins with the extraction phase (phase 1 in Fig. 3). Its goal is to produce a description of the simulation state with respect to what is aggregated. The description contains a data domain (e.g., grid, particles) and corresponding values. The values may be provided by the simulation, such as velocity or position. Alternatively, they may be derived from the provided values to describe a property of the scenario, such as risk of flooding, building damage or barrier stability.

After the descriptions are produced, the grouping phase determines how they should be combined. It describes which elements from different states can be considered similar enough to be merged into groups of related values. In phase 2 of Fig. 3, these groups are shown using dashed lines. The similarity can be based on position (values at the same location), identity (belonging to the same object), or any other criterion. As this also defines which input elements are taken into consideration and how many values are produced, the extraction and embedding phases depend on the chosen type of grouping.

On the other hand, the aggregation phase (phase 3 in Fig. 3) is performed in the same way regardless of what is aggregated. The aggregation operator is a function that, when given multiple values as input, returns a single value describing a statistical property of its input. The minimum, maximum, standard deviation, mean or count are all valid aggregation operators. As the operators are not exclusive to any particular aggregate rendering, the user can change the operator at will. This causes new values to be produced, presenting a different overview of the data.

To show the results of the aggregation to the user, in the embedding phase (phase 4 in Fig. 3) an aggregate rendering is created. As we do not aggregate entire simulation states, but only certain aspects of them, we cannot create a rendering based only on the aggregated data. A user-selected simulation state is used to provide a context, supplying all of the simulation data that is not aggregated. The final aggregate rendering is created as a combination of the state rendering and the aggregate representations. This also allows aggregate renderings to work as comparative visualizations. If an interesting aspect of the simulation can be represented in two ways, the nonaggregated representation can be contrasted against the aggregated one.

3.1 Object Aggregation

To further clarify what aggregated renderings are, it is best to continue with a simple example shown in Fig. 3a. The goal of the synthetic case-study scenario is to protect buildings from flooding. To find out how successful the current setup is, a building damage measure can be used. The height of the fluid surrounding the building determines the damage. Afterwards, the danger values can be used to color the buildings. The user can visually inspect buildings, and easily see which ones are in danger. However, suppose that the user wants to see the worst outcome for a building in a variety of simulation runs. This can be done with an aggregate rendering, which uses the maximum aggregation

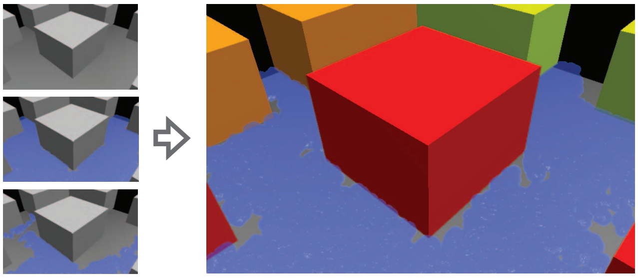

Fig. 4. Building aggregation. The water levels from various states are combined, and the building is colored according to the highest one. The color red signifies that the building is flooded in at least one state.

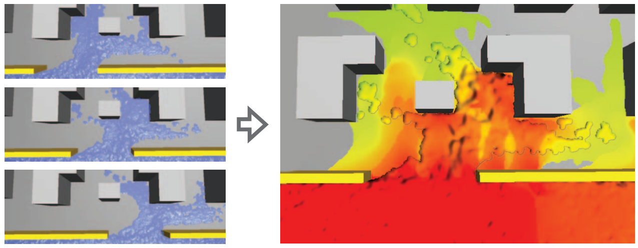

Fig. 5. Field aggregation. The height of the fluid is aggregated with the maximum operator, producing an aggregated rendering that uses one of the input states as context. The color red signifies the highest water level reached by the river, while green signifies the lowest significant level of water.

operator. Each group that is formed contains all values related to a single building. The maximum operator is applied to them, and the aggregated value is again mapped to the building color, as shown in Fig. 4.

To make this aggregation, we use a property unique to the buildings in our simulation scenario. Every building is represented by the same shape in all simulation states. This property makes it easy to show the aggregated value, as the building needs only to be colored appropriately. To generalize, whenever we are dealing with an object that has the same representation in all simulation states, we can aggregate its properties and apply them to its appearance. As we aggregate the properties of an object, we call this type of aggregation an *object aggregation*.

An object can be any type of entity, not just a shape. This makes it possible to apply this aggregation type to many phenomena. For example, a 2D bitmap showing values as color can be aggregated if pixels are thought of as objects. However, as the next two sections will show, other types of aggregation must be introduced for more complex simulation entities.

3.2 Field Aggregation

As the simulation scenarios we discuss revolve around flooding and water, there is a need to visualize the properties of the simulated fluid. In turn, there is also a need to understand how they change in the ensemble, and, thus, a need for an appropriate aggregation.

The problem of visualizing a fluid with a number of attributes is similar to the problem of visualizing a field. If a grid whose structure is constant in time is used to represent the field, each cell may be treated as a separate object to aggregate. However, in a free-surface fluid simulation problem, such as the one we use, the same grid cell may or may not contain fluid depending on the state of the simulation. It is possible to look at the field in two ways: as a collection of cell objects whose appearance changes in time, or as a field object whose appearance changes in time. Either way, there is no consistent representation that can be used to show the aggregate.

As can be seen in Fig. 3b, the idea of *field aggregation* is to first aggregate all the data and then produce a new appropriate representation capable of showing all of them. The correspondence used for this aggregation type is based on position. During the grouping phase, the size and resolution of the aggregated field is calculated in advance so that every input field is contained within it. The resolution can be fixed, or based on the input fields' resolutions. In the aggregation phase, the values are interpolated at the desired positions, and combined to create a new field. Groups created in such a way may vary in size, as there is no guarantee that all of the fields are going to be able to provide a value for the aggregation position and group.

Within the simulation scenario, we apply the field aggregation to the height, velocity, and pressure fields of the fluid. The simulation system uses the particle-based SPH model [23]. As it is impractical to aggregate individual particles and create a new particle set, we use SPH interpolation to produce a property grid of a sufficient resolution.

To avoid the need for volume rendering or for a more complex method of displaying the grid, the property grid is compressed to a 2D form. An aggregation operator matching the one to be used in the aggregation phase is applied to all the values projected onto the same location in the XY plane (phase 1 of Fig. 3b). Displaying the 2D grid on the surface of the terrain allows us to reduce visual clutter while sacrificing only a part of the available information. If a context fluid is present, it can also be shown with the grid projected onto it. This allows a comparison of the aggregated and the context state. A result can be seen in Fig. 5.

3.3 Instance Aggregation

The barriers present in the simulation scenarios are constructed from small blocks, each representing one bag of sand used to seal a breach. As water rushes in, these bags ideally retain their position, but more often end up getting swept away. To understand when and why the barriers break, we wish to aggregate individual bags to see their movement and the properties associated with them. However, the bags do not satisfy the requirements of the previously mentioned aggregation types. They change position and orientation, resulting in a similar but possibly different representation in each simulation state.

As with the field aggregation, a new type of aggregation must be used to depict these objects. Shown in Fig. 3c, the *instance aggregation* takes as input both the representations of objects and the values associated with them. Its name is derived from the fact that it operates on the instances of the same object stemming from different simulation states. For an object such as a bag, each instance is its representation in a simulation state, defined by the position and orientation of the bag.

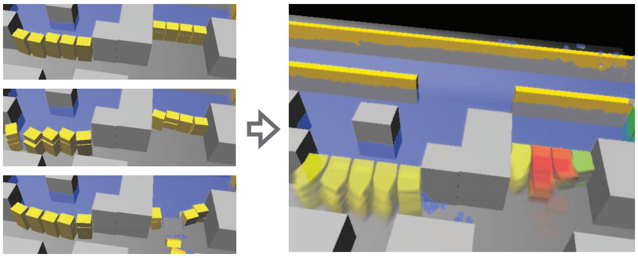

Fig. 6. Instance aggregation. Multiple states, three of which are shown to the left, are combined to create an aggregated rendering. The states are colored according to how much they have moved from their initial position, with red signifying the greatest movement.

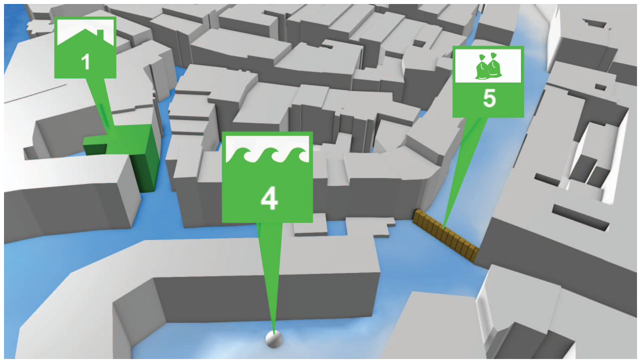

Fig. 7. The selection icons, showing what is chosen in the rendering. In this image, a building, a spatial position, and a bag have been selected.

The aggregation groups the instances according to the object they are representing. The values associated with the instances, such as the bag movement, are aggregated, and forwarded with the representations to the embedding phase. All representations belonging to the same group are modified to show the aggregated value. They are drawn with a small level of transparency applied to them. As a result, the bags are displayed with a slight ghosting effect, showing their movement while the color encodes the aggregated value. The effect can be seen in Fig. 6.

The effectiveness of this aggregation depends on whether the user can see which objects individual instances belong to. Too many instances cause visual clutter, reducing the user's ability to perceive the movement and rotation of individual objects. Too few instances do not provide the user with enough information to determine the movement of objects. To allow this visualization to work properly, regardless of the number of instances used, we provide three mechanisms meant to improve its efficacy. Duplicate elimination removes instances which are too similar in position and orientation to already created instances. This reduces visual clutter by culling instances that do not provide enough new information. Instance interpolation fills the gap between instances with new ones created by interpolating the existing states' positions and orientations. Finally, a transfer function can be used to modulate the transparency of the instances based on the aggregation values. In this way, the user can focus on instances with certain properties. The three mechanisms do not solve the problem of visual clutter completely, but can improve the perception of the direction and the extent of the object movement.

4 Points of Interest

In the previous section, we have shown how aggregation can be applied to a number of interesting phenomena in a consistent way. The primary purpose of the aggregated renderings is to provide an overview and allow unusual behavior to be spotted quickly. Since aggregation compresses the available information, we supply interactive selection mechanisms to retrieve the full information. Selection allows the user to designate exactly which areas are interesting in an intuitive way. By clicking on the aggregated rendering, the user can specify points of interest. In the rendering, these are represented as floating icons, visible in Fig. 7. As new points of interest may emerge during an exploration, the user can select additional points, or remove old ones by clicking on the icons in the rendering.

Regardless of what is selected, the retrieval of information can be done consistently in an aggregate rendering, as shown in Fig. 8. A click on an aggregate rendering can be mapped to a representation of an aggregate (Fig. 8a). Based on the primitive or position that has been selected by the click, it is possible to match it to one or more of the groups created in the grouping phase (Fig. 8b). The values associated with the groups can then be forwarded to an information-visualization view for further analysis (Fig. 8c). The benefit of the unified model is simple: As long as a user's click can reliably be mapped to a representation of an aggregate, the selection mechanism can extract data from any type of aggregation.

To map the click, it must be possible to determine if a representation of an aggregate has been selected, and if so, what additional information must be retrieved. In the case of instance aggregation, the object identifier must be found, while the field aggregation requires the exact spatial position of the selection. Either way, some information about the structure of the scene equivalent to a scene graph must be present. If the scene structure is known, standard approaches to selection such as ray picking or color picking can be used to retrieve the data.

The data produced by the selection mechanism consist of the exact values given to the aggregation operator before. This data can greatly differ in size and elements present. As an example, a bag may be placed within the simulation at a certain point in time. When the simulation states before and after that point in time are used to create an aggregate rendering, the bag will be present within it. If the bag is selected, however, the group related to it will be missing some values (see Fig. 8b). This is because no information about the bag can be retrieved from the states where it does

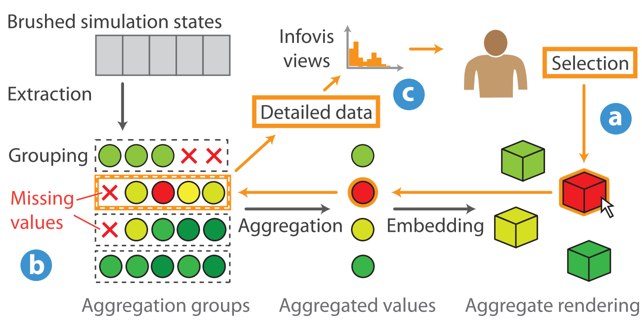

Fig. 8. The selection mechanism. The black arrows describe the flow of aggregation, while the orange arrows pointing in the opposite direction describe selection. The selected elements are depicted with an orange border.

not exist. Another example concerns field aggregation. Depending on the spread of the water and the barrier configuration, some areas might not contain water. The aggregation groups related to these areas will all be missing values, signifying a lack of flooding.

To allow data from different selected points to be compared despite the differences in the present data, we introduce the concept of *missing values*. When extracting data, the selection mechanism must produce one value for each simulation state. If a value cannot be retrieved from a simulation state, it is substituted by a missing value. As will be discussed in the next section, this change requires that the information-visualization views are modified to be able to handle missing values.

5 Filtering and Exploration

As the user selects points of interest, every new selection produces an n-dimensional vector of values, where n is the number of states used to create the aggregate rendering. If the number of points selected is k, we can view the resulting data as a collection of n k-vectors. The task of finding states with desired properties amounts to finding suitable vectors in this data set. To support the user in this filtering task, information-visualization views are provided.

5.1 Information-Visualization Views

In Visdom, any number of information-visualization views can be created. Every view has access to the data created by selection, and can show a certain number of the data dimensions. All the views are linked, and a brushing action in one view is automatically reflected in all the others.

Five of the available views are shown in Fig. 9. The line graph (Fig. 9a) and the histogram (Fig. 9b) are both capable of showing only one data dimension. The line graph, however, embeds additional temporal and input parameter-space information by drawing different parameters as different time-dependent lines. Two dimensions can be handled by a scatterplot (Fig. 9c). For cases where more dimensions should be shown, a parallel coordinates view (Fig. 9d) is available. World Lines (Fig. 9e) are also considered to be an information-visualization view. To make World Lines more useful for the purposes of this solution, we introduce a new analysis mode. It focuses on showing selected frames and tracks by eliminating most of the progress and frame coloring information. The frames can also be brushed and linked in the same fashion as the other views.

The major difference between common information-visualization views and the ones present in our system is the support for missing values. In the views where only one dimension of the data is shown, no special care needs to be taken to handle the missing values, as they simply result in a smaller data set. While it might be desirable to be notified that the values are not present, their absence does not conceal information. The same principle cannot be applied to the scatterplot and the parallel coordinates. Opting not to show any data entries which have a missing value would mean that potentially interesting parts of the data are hidden. Therefore, both the scatterplot and the parallel coordinates support the placement of values outside the axes. By drawing points outside of the grid for the scatterplot, and below the vertical lines for the parallel coordinates, we

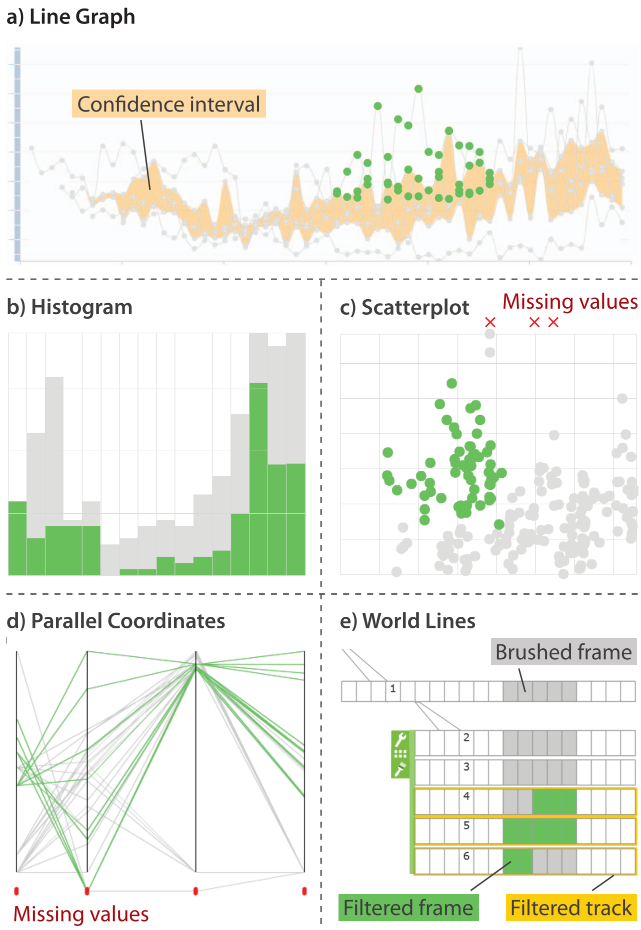

Fig. 9. Different information-visualization view types, with brushed values. The green color denotes the brushed values serving as a filter of World Lines states.

signify that these values are missing. To further highlight this, a shade of red is used (see Figs. 9c and 9d).

While there is a disadvantage to introducing new visual elements that the user might not be familiar with, there is a large benefit to showing the missing values. When they are visually present, the user can brush them, and pose queries that could not be answered otherwise. An example is the following query: "Show the velocity of other barriers in the states where this barrier does not exist." By using a scatterplot and brushing the missing values, the user can see the velocities and the states where the barrier is missing. Queries related to the absence of data would prove very difficult to pose without the missing values mechanism.

5.2 The Exploration Loop

World Lines maintains two brushes—the aggregation brush of all the states used in the aggregation, and the brush of filtered states. Any brushing action performed in the other views results in a change of the filter brush. Additionally, the aggregate is updated, using only the filtered states to form the aggregation. The overview's update allows the user to see the common properties of the filtered states.

Once a filter brush with the desired properties has been found, the user can perform a contraction action. A contraction causes all the nonfiltered states to become unselected, removing any data related to them from the information-visualization views. The user can also expand

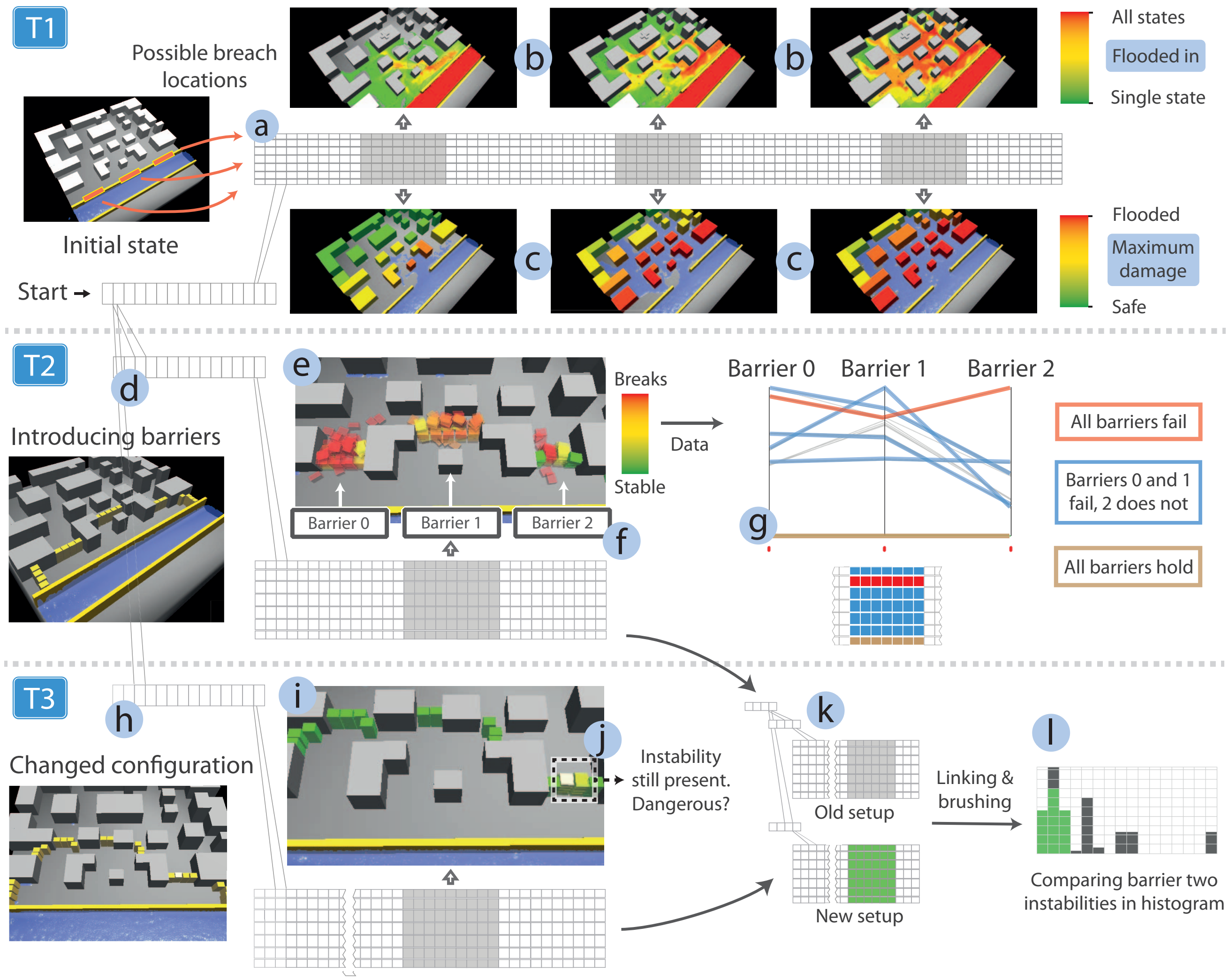

Fig. 10. The case study used to evaluate the approach. The first task (T1) involves aggregation to examine uncertainty. The second task (T2) is the application of a solution, and the exploration of its effects using aggregation and information-visualization views. Based on the results, task three (T3) is enhancing the solution according to what has been learned, and finding whether the new solution is stable.

both of the brushes. At any point, new parameters can be introduced, and additional data can be generated on the fly. The aggregation brush can be expanded with these new states, allowing them to be examined and compared against the results generated so far.

At the beginning of the user's interaction with the simulation, only the starting state is present. By introducing new parameters and performing simulation, the number of the states grows. The aggregation brush can be thought of as the focus of the exploration. As the number of states rapidly increases, the ability to follow developments with the brush allows the user to navigate the ensemble simulation, seeking out properties and states of interest. With the aggregates and information-visualization views serving as guidance, the intertwined contractions and expansions form a loop. This loop of analysis and experimentation enhances World Lines, where the analysis is manual, and ComVis, where experimentation is strenuous.

6 Case Studies and Expert Feedback

To evaluate whether the techniques presented in the previous sections are interesting, useful, and usable, we presented two case studies to experts working in various positions in the area of flood management. To some extent, this process was iterative. The initial motivation for developing the techniques came from working with World Lines and seeing their shortcomings when dealing with ensemble simulations and more complex analyses. The synthetic case study demonstrated our ideas, and allowed us to present them to a small number of experts. While the experts suggested no major changes, their feedback and data allowed us to make a second, real-world case study. Based on the second case study, we presented our work to a wider audience of experts.

6.1 Synthetic Case Study

To show the work process and the benefits related to using our solution, we reintroduce the flooding scenario from the Nodes on Ropes paper [1]. The scenario and its exploration is shown in Fig. 10. A breach in the levees protecting a city from a flooding river is likely to occur, and the authorities wish to explore the flood defenses that can be deployed to minimize the damage done to the buildings. The breach event has not yet occurred, and the location where it may occur can be predicted with only a limited degree of certainty. We have identified three relevant parameter groups in the case study:

- *Bag properties*. For each individual bag: bag position, orientation, and mass. The number of bags is unbounded and can easily exceed 100.
- *River properties*. The velocity and the water level of the river can be altered.
- *Protection properties*. These include the height of the protection barriers, whether a breach exists, and if so where it is and how wide it is.

As the river properties can be assumed to be constant, we focused on the interplay between bag and protection properties. Fig. 10 depicts the three tasks that were identified, illustrated on a World Lines structure created during the exploration. Given the complexity of the figure, throughout this section we will use letters (a-l) to identify the various steps, and numbers to identify the tasks (T1-T3).

The first task is understanding the uncertainty of possible breach locations (T1). A number of tracks is created, each corresponding to a breach location likely enough to warrant examination (Fig. 10a). To understand the area at risk from a breach, two aggregations are used. The fluid is aggregated using the count operator, showing the area where water may spread, and how likely it is that an area will be flooded (Fig. 10b). The building damage is aggregated using a maximum operator, showing the risks for various buildings (Fig. 10c). By brushing the simulation runs at various points in time, it is possible to see that the endangered area spreads quickly.

To contain the water, bag barriers need to be placed between the buildings and the levee. Task two (T2) involves determining where they should be placed, and examining their performance. Their placement (Fig. 10d) is guided by the fluid aggregation, which shows the extent of the breaches. The states of the bags after some time has passed are aggregated according to their stability and the maximum operator (Fig. 10e). The stability of bags is measured by how much they have moved from their original position in the barrier. A rather unfortunate result is that all the barriers protecting the center of the city seem to break over time, as evidenced by the red bags that have changed position. To check if there are some consistently bad barriers, the operator is changed to minimum. All of the bags are colored green, showing that for each one of them there is at least one breach position where they are stable.

The lack of consistency suggests that it might be interesting to explore the relationship of barrier failures. For every barrier, a bag within it is selected, creating three points of interest (Fig. 10f). When shown in a parallel-coordinates plot (Fig. 10g), two groups of outlier states stand out. The first one, colored brown, is related to an unlikely position of the breach, causing no barrier breaks. If this outlier case is eliminated, it can be seen that barriers 0 and 1 break consistently. In the second group of outliers, colored red, barrier 2 breaks as well. However, the second group contains states stemming from only a single simulation run, suggesting that the barrier's placement is rather good. Based on these results, we decide to slightly reinforce barrier 2, and to replace barriers 0 and 1 with a new set of barriers.

Task three (T3) involves applying the new barrier configuration (Fig. 10h), and checking its performance. In the new configuration, the fluid aggregation suggests that the flooding is contained. When replaced with the bag aggregation (Fig. 10i), it can be seen that while the new topmost barriers are stable, the remaining barrier from the original configuration shows a slight instability (Fig. 10j). To see whether this development is significant, both configurations are brushed in World Lines to provide data to compare (Fig. 10k). We create a linked histogram (Fig. 10l) that shows the instability of barrier 2, and apply a filter brush to the values originating from the new configuration. As the figure shows, the instability is actually lower than in the first setup, and nowhere near the values that have come to signify danger.

Given that all the barriers are stable, we can conclude that the current solution successfully stops the flooding. By refining an unsuccessful attempt guided by what we have learned, we have managed to find a solution that works well enough for our purposes.

6.2 Initial Expert Feedback

Our first source of feedback was a coauthor of this paper who is an expert in the area of hydrology, specializing in flood simulations. He saw potential for the techniques to be applied in planning and training activities. The ability to quickly examine and analyze many alternatives is an asset while planning, especially in the area of hydrology. Input-parameter uncertainty is always present, and the plan must take this into account. Aggregated renderings can show the statistical properties of a plan, with different aggregation operators describing the reliability, expected values, and the uncertainty involved with a plan. As far as the information-visualization views and filtering are concerned, although the expert required some clarification, he concluded that he considers them useful as well. One problem he would have liked them applied to is the elimination of unnecessary parameters in complex ensemble simulations. Apart from that, he concluded that it would be interesting to analyze real-world data gathered from previous flooding events in combination with simulations for training purposes.

After gathering the initial feedback, we decided to contact additional experts to gain more feedback related to the areas the expert mentioned. We found two persons who fit this description. The first one was another simulation expert specializing in ensemble flooding-simulations who could give us feedback on the solution's analysis capabilities. The second one was an engineer from a consulting agency specializing in the development of flood-protection measures. Both of them were given a one-hour demonstration of the system. While they found the ideas attractive, the synthetic case study did not match the problems that they were working on. In cooperation with them, we developed a new case study focusing on issues they identified, using real-world GIS data and flood-defense specifications.

6.3 Real-World Case Study

The new case study begins with the same basic problem as the previous one. As can be seen in Fig. 11, a river passing through a town is prone to overflowing. The defense planners wish to examine the effects of the flood given a certain amount of uncertainty with regards to failing

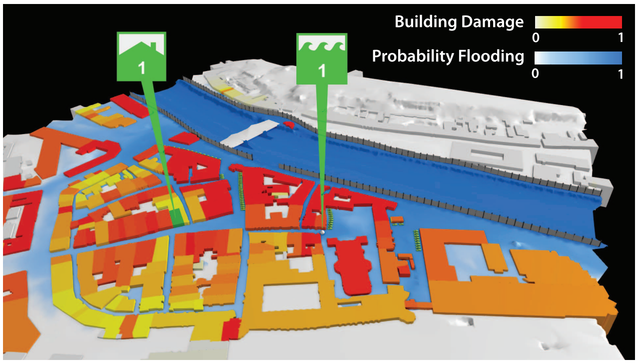

Fig. 11. Overview of the effect of uncertain breach locations in a real-world case study scenario. The properties of buildings and the probability of flooding are aggregated to show the consequences of the flooding.

defense mechanisms. However, the protections in this case are static rather than dynamic. The goals of the case study are derived from the situation in Köln, a town in Germany. Köln employs mobile walls to prevent water from spreading into the town. These walls have been built recently, and with no flood that tested their reliability, a breach is still possible. The breach position is then uncertain by definition. As the evacuation and response priorities have to be determined based on the breach location, a simulation-powered analysis is required to develop corresponding plans.

Apart from the building damage and flooded areas, which were important output variables in the previous case study as well, a new input and output parameter was added. The streets of a town are at any time filled with parked cars, which can be seen in Fig. 12a. If a failure of the defenses is anticipated, these cars can and should be removed before any flooding occurs. If this is not done, they represent a great danger. Cars carried by water can slam into objects causing further damage. Should they collide with flood defenses, the additional breaches can increase the flooding. While all these factors make it highly advisable that cars should be removed whenever a flood is pending, the removal requires time and resources. If these are limited, the removal must be prioritized according to the danger the cars pose. The goal of this case study is to see how effective the presented system is at answering questions related to risk and danger with respect to building damage, flooding, and debris such as cars. In total, the parameters of this case study are:

- The river properties, e.g., velocity and water level.
- The location of defenses. The position and height of a mobile wall can be defined as a 3D curve. Any number of them can be present.
- The location of breaches. Any number of breaches can be present. They are defined by the wall they belong to, the position on that wall, and by their width.
- The position and quantity of cars. As the cars are parked along a street, they can be described by a 2D curve and a spacing parameter. The car mass can also be adjusted.

We began our analysis of the case study by examining the building damage and flooded areas through object aggregation applied to buildings and field aggregation

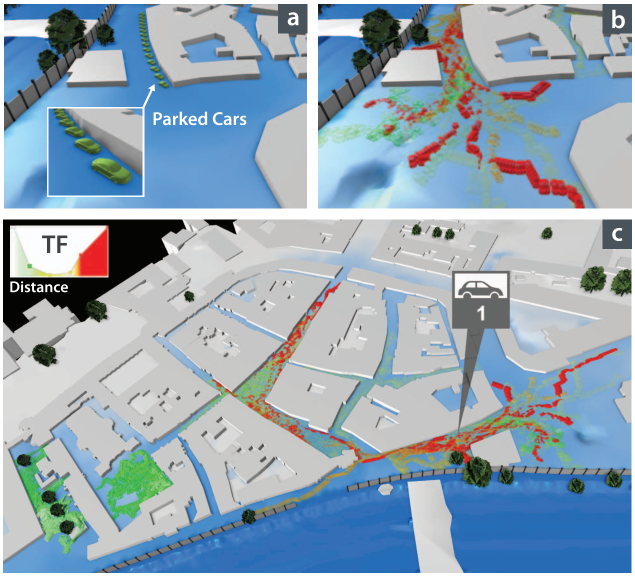

Fig. 12. Instance aggregation applied to cars. (a) The initial state of the cars. (b) A close-up of the car aggregation. The car positions are interpolated to compensate for the large time step size. (c) An overview of the car movement. The cars on the left are mostly stationary, while the cars to the right move the most.

applied to the extracted velocity field. The maximum building damage and the probability of area flooding turned out to provide the most useful overview. Fig. 11 shows the aggregated rendering, adjusted to show the minimum amount of detail necessary and make the aggregation more visible. Using these two aggregates and the selection and filtering mechanisms, we were able to pose multiple queries about building danger and evacuation priority. As these do not differ significantly from the previous case study, they will not be described here in detail.

To examine the car movement, we applied the instance aggregation to the cars, coloring them according to the distance they have travelled. Figs. 12b and 12c show the instance aggregation combined with the field aggregation. We discovered that only the parked cars positioned closer to the breaches to the right moved a significant distance, and decided to focus on them. By adjusting the transparency function (TF) to show primarily stationary and very mobile vehicles, a clearer picture emerged. While the cars further away from the mobile wall moved even more than the closer ones, their movement was contained by the street layout. The closer cars exhibited a chaotic movement, which was quite dangerous as the aggregation showed some cars veering dangerously close to the barriers. Based on this, we used an area selection tool to pick the cars near the breach. The area selection tool performs an aggregation of its own on all groups that it affects. The produced data, thus, showed the maximum movement of any car initially parked within the selected zone.

After extracting a measure of the danger posed by the cars, we used information-visualization views to analyze how the various outputs were correlated. In the parallel coordinates shown in Fig. 13c, we brushed the damage inflicted to the selected building. For the purposes of the case study, we consider it to be a school—a high-priority

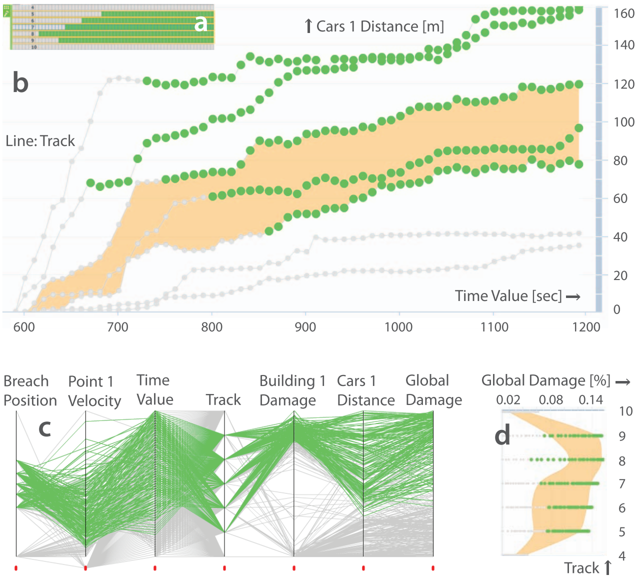

Fig. 13. The analysis of the ensemble data performed using information-visualization views. World Lines (a) shows frames where an important building is at risk, while the line graph (b) provides additional details about the dangers posed by washed-away cars. (c) The parallel coordinates show the relationships of the variables, while another rotated line graph (d) shows the confidence interval of the global damage.

evacuation zone. When this building is in danger, some people who would otherwise be working on removing the cars have to help with evacuating the school instead. First, we consider the brush in World Lines (see Fig. 13a), giving us an idea when the danger happens. To examine it in more detail, we use a line graph showing car movement in the ensemble (see Fig. 13b). To see how the entire city is faring, we examine the brush in another line graph (see Fig. 13d) which plots the global-damage confidence-interval.

By using aggregated renderings, we were able to locate the source of the risk quickly. The filtering and selection mechanisms then allowed us to find coexisting risks, which would have to be considered in the making of the emergency plan.

6.4 Final Expert Feedback

After developing the real-world case study, we collected opinions from the two experts again, but also attempted to evaluate the solution with a greater number of potential users. For this purpose, we provided a questionnaire in which the users were asked to grade the intuitiveness and the relevance of the presented techniques and employ ratings on a scale of one to four. The evaluation started with the two simulation experts, who were invited for another session, lasting between 1 and 2 hours. The experts found the overviews generated by the aggregations to be quite useful, and thought the same of the filtering mechanisms. However, they needed time to adjust to the information-visualization views.

We received more detailed feedback from the consulting engineer. He sees the tool as having great potential as a planning device, but noted that further reworking is needed for it to be used for a specific purpose. He considered the aggregations to be quite intuitive, and suggested a level-of-detail approach. While an engineer appreciates the ability to fine-tune parameters and perform detailed analyses, the average user would benefit from a reduced interface containing aggregations as an overview and little else.

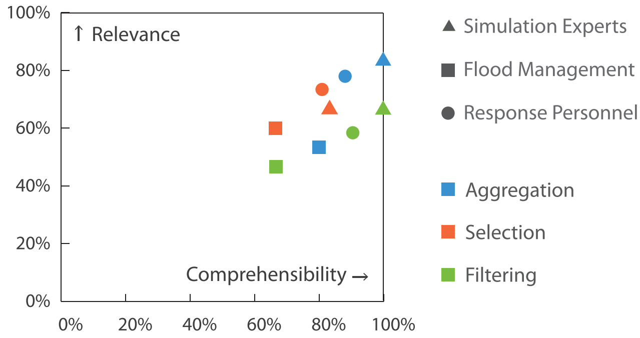

Fig. 14. The responses of the experts who had participated in the second evaluation.

Apart from the simulation experts, we presented the real-world case study to the people in charge of the Köln flood protection. Seven response personnel and five flood-protection planners took part. Our experience with the simulation experts suggested that at least half an hour was necessary for a user to be introduced to the system, with an hour needed to perform an evaluation. Given that their gathering was a part of a greater meeting, we had only a day in which to demonstrate the capabilities of the system. As we could only send a single person capable of performing the evaluation, we had to take a different approach to collect feedback from this group. Each part of the questionnaire was illustrated by a video showing the system in action. The videos, accompanied by live commentary by the evaluator who also answered questions, were presented to the group. After this 2-hour long session, informal conversations provided more information.

The questions posed can be divided into three categories: aggregation, information visualization and filtering, and selection. All three were received well, as can be seen in Fig. 14. Interestingly enough, the response personnel consistently considered the components more intuitive and relevant to their line of work than the flood-management staff. Given that the tool is meant to be a planning tool, and that we had received positive feedback from people involved in planning earlier, we are not able to completely understand this result. The simulation experts had the best opinion of the tool. This is probably a result of their earlier involvement and the more engaged evaluation.

7 Implementation

The aggregations shown in this paper are meant to help users create an understanding of a very large and varied set of data. It is safe to assume that the number of states used in an aggregation exceeds a hundred, and is only limited by the number of simulated states. The design questions are related to performance. Each aggregation requires data from multiple simulation states to be processed. Since the data are produced and aggregated according to the states the user brushes, it is not possible to predict which data an aggregation will need in advance. Instead, we gradually produce and store the data to suit the user's requests.

Whenever the user introduces a change into the simulation, or brushes a set of states, the necessary calculations are determined and performed. All of the intermediate results such as simulation states or extracted information are stored. If possible, these results are reused. For example, whenever an aggregation operator is changed, the extraction and grouping phases are skipped, as the change could not have affected them. The algorithm used in Visdom that makes such caching possible is described by Schindler et al. [24]. The functionality required for aggregation is decomposed into various data flow nodes, and the system takes care of data generation and reuse automatically.

The amount of time needed to create the aggregations depends on the input type used. Tests were performed on an Intel I7-equipped computer with an NVIDIA GTX 480 card. For the synthetic case-study scenario with around 60,000 particles, the extraction of the building and bag data takes a relatively short time to complete, requiring 20 ms per state for the buildings and 30 ms per state for the bags. The fluid-property extraction is computed on the GPU and takes around 60 ms per state, primarily because of the SPH interpolation. The extraction phase is the computationally most expensive part of the presented technique. Both the grouping and the aggregation phase are fast, requiring only basic operations to be performed on previously prepared data. The embedding phase is limited primarily by the renderings used as a basis for the embedding. The real-world case study uses around 500,000 particles, but the nature of the SPH interpolation means that the increase in time needed is linear. The particles analysis requires approximately 400 ms per state with such a simulation.

The first aggregation the user performs is usually the most time consuming one and incurs a noticeable delay depending on the number of states selected. All the subsequent aggregations reuse at least part of the data, and require less time. The brushing of the information-visualization views deals with small amounts of data and can be performed interactively. If the user brushes a view, linked window updates are done as the selection changes, allowing the analysis to run smoothly.

All the timings mentioned above are insignificant in comparison to the simulation time, which we have found to be at least ten times greater than the analysis time. While the results are not delivered instantly, the calculations are performed fast enough to allow for experimentation and analysis. The analysis results could also be precomputed immediately after the simulation states have been calculated, hiding the overhead.

8 Applicability in Other Problem Domains

While we have performed our evaluation on flood simulations, during the inception and development of the tool we were careful not to make any assumptions about the underlying type of simulation or data. Steerability is not a necessity for using the tool. Both the tool and the framework it resides in are meant to handle steerable simulations. However, they can also be used as a standard data flow system as well, with the ability to integrate modular components. This tool could be applied to other problems. The question is how efficient it would be in other domains.

The basic problem that this system could solve involves a large number of multidimensional data instances, which has to be filtered in some way. As many tools already attempt to tackle this problem using information-visualization views, the strengths of our tool dictate which problems might be interesting. The selection mechanisms work well for problems with an infinite or extremely large parameter space. The size of the parameter space can come from the complexity of the underlying data. For example, genetic data can be seen as having a very large parameter space, many instances, and a need to identify instances with certain properties. Given that tools using similar principles already exist in this domain [22], the usefulness in this case would depend on how well an aggregation could be used to detect interesting regions of the parameter space.

A parameter space may be enlarged dynamically not only by interactively steering a simulation, but also, e.g., by using a feature-detection algorithm on data instances. In the case of volume rendering, considering every voxel as a parameter would be impractical, but detected features could be taken as additional parameters to analyze. The difficulties here would lie with the grouping of detected features and with the visual clutter in an aggregated rendering. If instance aggregations could be additionally improved as to include only features that are selected, adding an additional level of filtering, the tool could be useful in this context.

Steerable simulations are still the best fitting area of application. Two interesting domains are traffic simulation and climate-data analysis. Traffic simulations are quite similar to our current scenarios in terms of rendering requirements, and the usefulness of introducing new simulation parameters in the middle of a run, such as crashes. The experts from Köln in charge of traffic management during crises have stated that applying the problem to this domain would be interesting. Climate-data analysis could also benefit from the introduction of changes simulating (e.g., policy changes in various states). The application of similar techniques in this domain [2] shows that selection and aggregation would be useful as well.

9 Conclusion and Future Work

In this paper, we present an aggregation, selection, and filtering-based solution meant to help the user navigate an evolving simulation ensemble. We have found the aggregate renderings useful for detecting unusual behavior and monitoring of a simulation. The main benefit of our work is the extension of the area where interactive visual analysis mechanisms can be applied. With selection as an intermediate step, the dimensions of the data to be analyzed need not be known or shown in advance. This allows our tool to be used in a steering environment where changes create new data or require the focus of the analysis to shift.

In our future work, we would like to concentrate on two aspects of the system. The analysis mode requires an amount of time proportional to the number of analyzed states, which might not be available in time-critical situations. Some filtering operations could instead be performed using a selection operator. A selection could have different meanings—find all states with a similar value at this point, show contributing states, exclude similar states, and so on. By reducing the need to create or manipulate information-visualization views, the system could become easier and faster to use for domain experts.

Another aspect is the use of automatic data processing in the system. In some aggregations, such as the instance aggregation, many instances related to a common value can occlude interesting events in other states. By clustering and displaying only representative instances, the clarity of the visualization can be improved. Feature-selection techniques combined with random sampling could be used to suggest points of interest that the user had not noticed, with respect to brushed states. These approaches cannot replace the human operator, but could be useful as a means of improving the usability of the system.

We feel that the strength of our approach lies in its flexibility and heterogeneity. The ability to quickly switch between various aggregation types allowed us to change the focus of the visualization as we switched from exploration to problem solving to solution verification. Different simulation primitives that would normally require to be visualized in different ways were presented in the same visual context. Despite the diversity, the selection and filtering mechanisms secured uniform access to the generated data. Visualization solutions often concentrate on one problem and solve it well. We hope that our approach is a step toward a more general problem-solving environment, more suited to the complex situations encountered in real life, but still harnessing the power of visualization, simulation, and interactive visual analysis techniques in a single tool.

Acknowledgments

This work was supported by grants from the Austrian Science Fund (FWF):P 22542-N23 (Semantic Steering) and the Vienna Science and Technology Fund (WWTF):ICT08-040 (Scale-VS). The authors would like to thank the Hochwasserschutzzentrale der Stadtentwässerungsbetriebe Köln, Geoconsult, Michael Greiner, and Jürgen Komma.

References

[1] J. Waser, H. Ribičić, R. Fuchs, C. Hirsch, B. Schindler, G. Blöschl, and M.E. Gröller, "Nodes on Ropes: A Comprehensive Data and Control Flow for Steering Ensemble Simulations," *IEEE Trans. Visualization and Computer Graphics,* vol. 17, no. 12, pp. 1872-1881, Dec. 2011.

[2] K. Potter, A. Wilson, P.-T. Bremer, D. Wiliams, C. Doutriaux, V. Pascucci, and C. Johnson, "Visualization of Uncertainty and Ensemble Data: Exploration of Climate Modeling and Weather Forecast Data with Integrated Visus-Cdat Systems," *J. Physics: Conf. Series,* vol. 180, 2009.

[3] J. Waser, R. Fuchs, H. Ribičić, B. Schindler, G. Blöschl, and M.E. Gröller, "World Lines," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 6, pp. 1458-1467, Nov./Dec. 2010.

[4] K. Matković, D. Gracanin, M. Jelovic, and H. Hauser, "Interactive Visual Steering - Rapid Visual Prototyping of a Common Rail Injection System," *IEEE Trans. Visualization and Computer Graphics,* vol. 14, no. 6, pp. 1699-1706, Nov./Dec. 2008.

[5] "Visdom - An Integrated Visualization System,"http://visdom.at, 2013.

[6] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," *Proc. IEEE Symp. Visual Languages,* pp. 336-343, 1996.

[7] R. Fuchs and H. Hauser, "Visualization of Multi-Variate Scientific Data," *Computer Graphics Forum,* vol. 28, no. 6, pp. 1670-1690, 2009.

[8] S. Bruckner and T. Möller, "Result-Driven Exploration of Simulation Parameter Spaces for Visual Effects Design," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 6, pp. 1467-1475, Nov. 2010.

[9] H. Doleisch, M. Gasser, and H. Hauser, "Interactive Feature Specification for Focus+Context Visualization of Complex Simulation Data," *Proc. IEEE Symp. Visualization (VisSym '03),* pp. 239-248, 2003.

[10] H. Doleisch, M. Mayer, M. Gasser, P. Priesching, and H. Hauser, "Interactive Feature Specification for Simulation Data on Time-Varying Grids," *Proc. Conf. Simulation and Visualization '05,* pp. 291-304, 2005.

[11] J.C. Roberts, "State of the Art: Coordinated & Multiple Views in Exploratory Visualization," *Proc. Fifth Int'l Conf. Coordinated and Multiple Views in Exploratory Visualization (CMV '07),* pp. 61-71, 2007.

[12] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques.* Morgan Kaufmann, 2005.

[13] J. Goldstein and S.F. Roth, "Using Aggregation and Dynamic Queries for Exploring Large Data Sets," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI '94),* pp. 23-29, 1994.

[14] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub-Totals," *Data Mining and Knowledge Discovery,* vol. 1, pp. 29-53, 1997.

[15] J. Groff, P. Weinberg, and A.J. Oppel, *SQL The Complete Reference.* McGraw-Hill, 2009.

[16] C. Stolte, D. Tang, and P. Hanrahan, "Multiscale Visualization Using Data Cubes," *IEEE Trans. Visualization and Computer Graphics,* vol. 9, no. 2, pp. 176-187, Apr. 2003.

[17] M.M. Malik, C. Heinzl, and M.E. Gröller, "Comparative Visualization for Parameter Studies of Data Set Series," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 5, pp. 829-840, Sept./Oct. 2010.

[18] J. Woodring and H.-W. Shen, "Chronovolumes: A Direct Rendering Technique for Visualizing Time-Varying Data," *Proc. Eurographics/IEEE TVCG Workshop Volume Graphics (VG '03),* pp. 27-34, 2003.

[19] J.-P. Balabanian, I. Viola, T. Möller, and M.E. Gröller, "Temporal Styles for Time-Varying Volume Data," *Proc. Fourth Int'l Symp. 3D Data Processing, Visualization and Transmission (3DPVT),* pp. 81-89, June 2008.

[20] J. Woodring and H.-W. Shen, "Multi-Variate, Time Varying, and Comparative Visualization with Contextual Cues," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 909-916, Sept./Oct. 2006.

[21] H. Akiba, N. Fout, and K.-L. Ma, "Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram," *Proc. Eurographics Visualization Symp.,* pp. 1-8, 2006.

[22] M. Meyer, T. Munzner, A. DePace, and H. Pfister, "Multeesum: A Tool for Comparative Spatial and Temporal Gene Expression Data," *IEEE Trans. Visualization and Computer Graphics,* vol. 16, no. 6, pp. 908-917, Nov./Dec. 2010.

[23] J.J. Monaghan, "Smoothed Particle Hydrodynamics," *Reports on Progress in Physics,* vol. 68, pp. 1703-1759, 2005.

[24] B. Schindler, J. Waser, R. Fuchs, and R. Peikert, "Multiverse Data-Flow Control," Technical Report 720, ETH Zürich Computer Science, 2010.

Hrvoje Ribičić received the graduate degree from the Faculty of Electrical Engineering and Computing at the University of Zagreb, in 2010. He is working toward the PhD degree at the VRVis Research Center for virtual reality and visualization. His current research topics involve the visualization and analysis of interactively steerable simulation scenarios.

Jürgen Waser received the graduate degree in 2008 from the General Physics Institute at Vienna University of Technology, Austria, and the PhD degree in computer science from the Vienna University of Technology, in 2011. He is a core developer and cofounder of the Visdom framework. His research fields of interest include simulation steering and web-based visualization systems.

Raphael Fuchs received the graduate degree in computer science, in 2006, from Philipps-University Marburg, Germany, and the PhD degree in computer science, in 2008, from the Vienna University of Technology, Austria, under the supervision of Prof. Helwig Hauser and Prof. Eduard Groller. He is currently with the chair of Computational Science (CSElab) at the ETH Zurich, Switzerland. His fields of interest include scientific visualization and applications of machine learning to visualization.

Günter Blöschl received the graduate degree from the Vienna University of Technology. His international experience includes appointments as a research fellow in Vancouver, in 1989, Canberra in 1992-1994, and Melbourne in 1993, 1997. In 2007, he was appointed a chair of Hydrology and Water Resources Management at the Vienna University of Technology. His research interests are in the areas of hydrology and water resource systems.

Eduard Gröller received the PhD degree from the Vienna University of Technology. He is a professor at the Institute of Computer Graphics and Algorithms (ICGA), Vienna University of Technology. His research interests include computer graphics, flow visualization, volume visualization, medical visualization, information visualization, and visual analytics. He is heading the visualization group at ICGA. He is a member of the IEEE Computer Society.