

## Chapter 12

# INTERACTIVE VISUALIZATION OF INTERDEPENDENCIES AND VULNERABILITIES IN CONSTRAINED ENVIRONMENTS

Nils Lunden, Robin Sveen, Hans Lund, Nils Svendsen and Stephen Wolthusen

**Abstract** Many critical infrastructure assets from hospitals to industrial facilities rely on multiple infrastructure services whose close proximity can result in the failure of one component causing cascading failures in other assets. This effective analysis and mitigation of risks requires the consideration of numerous scenarios and input from domain experts.

This paper describes a distributed interactive visualization and analysis mechanism for constrained environments such as buildings and industrial facilities in which the physical and logical components and dependencies are considered in a three-dimensional model. Physical effects modeled include flooding, fire, smoke and gas explosions; the logical dependencies considered include telecommunications and electrical power. This allows the creation of scenarios by interactively creating events whose effects are subsequently captured by physical and logical sub-models that can be viewed and replayed from multiple angles to permit efficient analysis and review. The capabilities of the system are illustrated using a complex scenario of cascading physical and logical failures resulting from a water leakage in a hospital environment.

**Keywords:** Visualization, concurrent models, distributed simulation

## 1. Introduction

Most critical infrastructure facilities are dependent on other critical infrastructure components and sectors. Dependencies and interdependencies between components and sectors can be captured and analyzed at the logical level [19]. However, this view is often imperfect as physical dependencies and hazards must also be considered in order to obtain an understanding of the risks to

an infrastructure element or cluster and possible mitigation approaches. The simultaneous consideration of logical dependencies and physical effects permits the identification of potential hazards due to unexpected (inter-)dependencies in logically separate but physically proximate infrastructure elements. Also, it facilitates the development of dynamic scenarios for assessing the existence and severity of risks as well as the time available to take mitigation measures [18].

While such models provide valuable insights in larger, sparsely populated geospatial domains and must have limited complexity to permit effective use, their utility in confined environments such as buildings and industrial facilities is limited. Elaborate models and visualization environments exist for some types of hazards [7, 11], but these are concentrated on a single incident or hazard category. They are often unsuitable for identifying the cascading failures of interest when analyzing the robustness of a facility to disruptions and its ability to provide critical services under degraded circumstances. However, they may be utilized to assess risks and mitigation strategies. Nevertheless, given the intractable number of possible simultaneous and cascading contingencies, automated analysis is unlikely to yield significant insights. It is therefore critical to allow domain experts from different realms to create and analyze scenarios interactively while providing physical and logical models that track the evolution of the scenarios and allow for their visualization. The resulting interactive visualization paradigm, which would facilitate the viewing of event traces from different perspectives and with varying levels of detail and emphasis, can yield important insights.

## 2. Model Components

The model used for visualization incorporates multiple modules centered around the geospatial representation of the area under consideration. This is achieved using a “scene graph” representation found in most vector-based graphical editing systems and in some distributed gaming environments. The scene graph representation permits the efficient and intuitive representation of logical and spatial groupings of entities in the form of a tree.

The visualization is controlled by a “master node” that retains the full model of the scenes to be rendered. The “rendering nodes” that participate in a simulation register with the master node, download the initial scene graphs and forward any local changes that are made to the master node. A tree-structured change propagation hierarchy can be employed reduce the overhead on the master node. Communication between nodes is realized via IPSec-secured channels to minimize the latency of individual update bursts. Note that the master node may or may not have a rendering engine running on it.

Distributed operations on scene graphs may be grouped by sub-trees that, for example, contain the contents of individual rooms; the operations must be transmitted once for the basic configuration when a rendering (visualization) node joins a simulation, adds or changes rooms. Any changes made must be transmitted to all nodes. When a node performs modifications locally, these must be communicated first to the master node, which then transmits the

modifications to all affected visualization nodes. To minimize communications traffic, rendering is performed on visualization nodes.

All objects and entities in the model are characterized by three-dimensional geo-referenced coordinates. The precise representation may range from a simple point cloud description to arbitrary polygon mesh forms for complex objects, where the latter may result from tessellations of other types of object or object surface descriptions (e.g., non-uniform rational B-spline surfaces) because polygon meshes are the natural representation used by the visualization engine.

The overall architecture is modular in that it supports an extensible set of sub-models for different types of events. This also allows the replacement of sub-models with others of higher or lower fidelity depending on resource availability and other requirements. To minimize the complexity of sub-models, components may be shared and reused. This includes material properties that describe the behavior of objects with regard to flammability, water-related effects and response to pressure, in addition to visualization properties such as color and transparency. In particular, the model captures both bulk material and surface properties.

The second major component shared across models is the basic physics model. Given the choice of visualization platform (see Section 3), this is based on the jME Physics 2 interface, which, in turn, provides an adaptation layer for several physics engines, including the Open Dynamics Engine (ODE) [16] and the proprietary PhysX engine by Nvidia. The two engines run on a variety of platforms and offer rich feature sets. The PhysX engine, in particular, offers performance benefits because it can draw on hardware support in the form of a dedicated physics processing unit (PPU) or a parallel implementation on a graphics processing unit (GPU) [3]. This engine also supports our material model, especially with regard to friction effects and general collision detection for the various geometric shapes used to model scenarios.

However, even with hardware acceleration support, some of the sub-models described in this paper (especially those in Sections 2.1 and 2.3) cannot be represented by a naïve object model for individual particles. Real-time performance falls to unacceptable levels when more than 2,000 objects are used (this occurs when the frame rate of rendering drops below 30 frames/second). We have, therefore, developed a dedicated “particle model” that can draw on the physics engine to employ a “sampling strategy” to capture the underlying object behavior in addition to that dictated by sub-models using small objects that do not require individual collision modeling but aggregate, explicit models and surface collision models. Figures 1 and 2 show an example of a simple (fire) particle model using the sampling approach in conjunction with surface interactions that include friction, deflection and heat exchange.

Although this approach provides adequate fidelity at relatively low computational complexity for small volumes and time-scales, it is still undesirable to model very small particles individually over larger time and spatial ranges; this is because other influences (e.g., sub-model effects) can dominate. Consequently, we have chosen to capture meso-scale effects using a “vector field

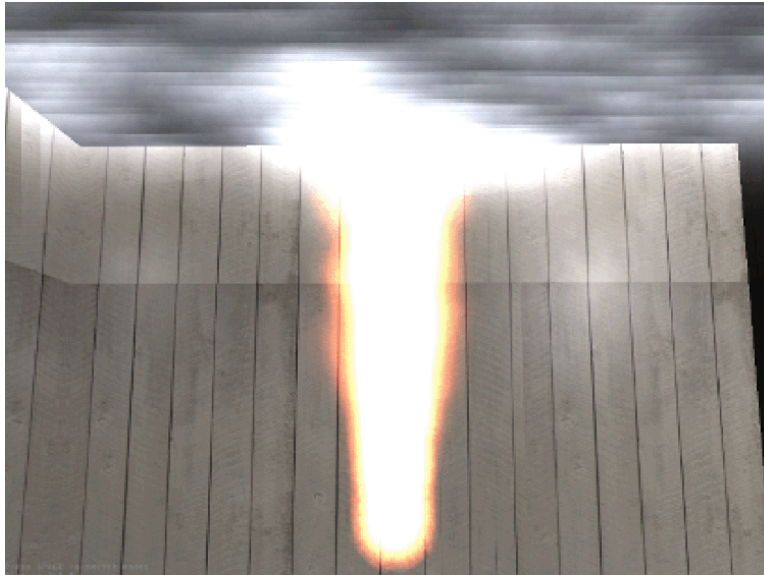
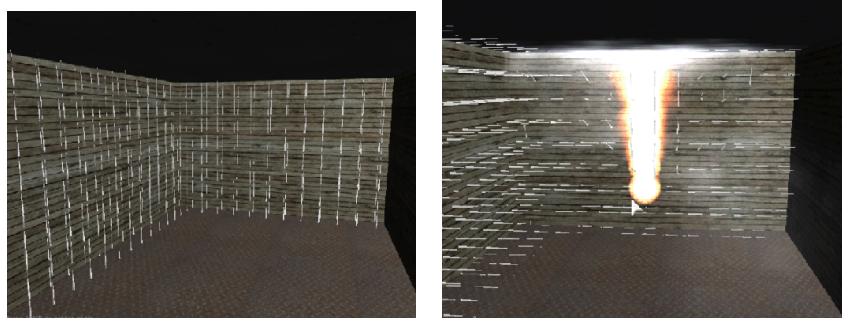


Figure 1. Interpolated model using sampling and surface interactions.



(a) Start of simulation.

(b) Updated vector field.

Figure 2. Vector field meso-scale particle model simulation.

model layer” using gradient vector fields  $\nabla f = \langle f_x, f_y, f_z \rangle(t)$  and a Hamiltonian dynamics model [1]. A number of more efficient approximations do exist, but we believe that this approach represents a straightforward “first principles” approximation (even though the reversible dynamics provided by the approximation are not used in our model). The choice of formalism restricts us to the use of smooth functions, but as we are mainly interested in modeling thermo-physical effects, this is not an issue. However, for the vector field model, it is desirable to increase the update frequency and, hence, the temporal and spatial

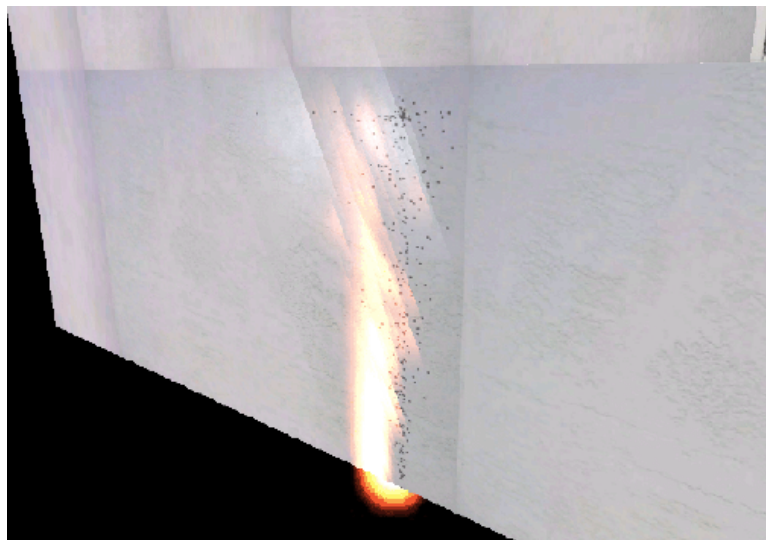


Figure 3. Soot particle aggregation and heat transfer interactions.

resolution to prevent interpolation errors from producing undesirable artifacts despite an otherwise smooth visual appearance in case of rapid dynamic effects. Note that the use of parallel models with discrete time steps can lead to artifacts, especially due to sampling effects.

## 2.1 Fire and Smoke Sub-Models

Fire and smoke are two demanding modeling and visualization domains. Several dedicated models are available in the literature (see, e.g., [11]). However, for the purposes of scenario development and visualization, a simple hydrodynamic model of heat transfer was chosen to obtain the requisite parameters for the particle model described above [20]; this resulted in a set of reference samples and a corresponding vector field for modeling heat transfer.

Of particular interest to critical information infrastructures are the adverse effects produced by soot particles that are affected by these flows. Consequently, soot particles and particle accumulations are modeled explicitly. Figure 3 shows soot particle aggregation and heat transfer interactions with a wall. The diagonal lines are superimposed artificially to enhance visibility of aggregations caused by heat transfer to the wall.

Note that our model is limited to simple scenarios and cannot accurately capture some effects found in more elaborate models. Also, the model is limited by the constraint on particle numbers to obtain support points for the vector field model. However, because we are primarily interested in observing the interactions of multiple sub-models, the limited fidelity of our model does not pose an insurmountable restriction.



Figure 4. Flooding effect simulation for unpressurized volumes.

## 2.2 Flooding Sub-Models

The scenarios considered in this paper do not engage detailed hydrodynamic models for water because dynamic effects are of limited interest in confined areas (unlike for a large-scale scenario such as a dam burst). Instead, we concentrate on spatially-constrained higher-pressure leaks and flooding effects. The former uses the aforementioned (spatially-constrained) particle model to capture the effects of leaking or ruptured pressurized pipes in a simplified fluid dynamics model [8] that considers a limited range of viscosity for the leaking material. Note that viscosity is assumed to be static in the model; this assumption may have to be adjusted for certain materials (e.g., liquefied natural gas) to consider temperature and heat transfer in the surrounding area.

The main emphasis is on flooding effects; the geo-referenced mesh and a coarser support vector mesh are used to establish whether fluids are retained in a contained volume as well as the flows between volumes and flow rates. Clearly, this is heavily dependent on an accurate mesh representation of the underlying model, especially when retrieving models from architectural drawings, which may suffer from gaps or format conversion errors and must, therefore, be validated manually. Nevertheless, the model keeps track of fluid levels in non-pressurized volumes to determine the pressure exerted on the sides of the container (see Figure 4). This, coupled with the material properties described earlier, allows leakage and rupture effects to be expressed. However, the model does not fully capture the dynamics of non-pressurized flooding and may, therefore, under-predict the dynamic effects compared with more complex

hydrological models [12]. Nevertheless, this can easily be compensated for by adjusting the parameters to obtain conservative upper bounds on the estimated effects.

## 2.3 Gas Spreading and Explosion Sub-Models

Models for gas and vapor cloud explosions include those by Alonso, *et al.* [5] for semi-open areas and by Cleaver, *et al.* [4] for confined spaces. We approximate explosions using the particle and vector field approach described above. However, as noted in Sections 2.1 and 2.2, a high-resolution sub-model is required to capture the speed and volatility of an explosion.

In the case of gas spreading, however, it is necessary to capture gas properties (e.g., relative density) in addition to flow models because aerosol formation increases the risk of an explosion. The model currently supports only a limited number of industrial gases. In the case of propane ( $C_3H_8$ ), a common industrial gas, an aerosol is flammable when the propane concentration is between 2.1% and 9.5% at temperatures exceeding 723K assuming natural oxygen levels. However, if the aerosol is formed in the presence of pure oxygen, the upper bound increases to 48%. Note that the accuracy of our model is limited by the mesh resolution and concentrations must be interpolated between individual mesh points. Also, only binary reactions are supported; this captures common redox reactions.

## 2.4 Logical Dependencies

Having specified the individual sub-models, it is necessary to cross-link the models. “Geospatial buffering” is used to identify affected entities as has previously been shown for meso-scale systems [18], albeit at a finer granularity than for our model.

After the volumes affected by events in a given simulation time step (more precisely, aggregate time step) are identified, the infrastructure components affected are identified by their geo-referenced coordinates in the scene graph of the overall model. This, in turn, can be used in a straightforward manner as a buffered multigraph in a conventional dependency analysis model [19]. Although it would be desirable to associate infrastructure components of different types with properties for physical simulations, we add these properties manually to the scenario components as discussed in Section 2.5.

## 2.5 Scenario Development

The development of simulation scenarios for the investigation of incidents or for evaluation and training requires considerable effort. The basis for the simulation is a faithful three-dimensional model, which must be of sufficient quality to ensure that all the structural elements are identified along with minor features such as ducting to ensure that the sub-models can be applied correctly. In the case of gas and smoke spreading or flooding simulations, even minor

imprecision due to volumes that are not fully convex may result in undesirable results. To minimize the effort required to generate models in the native representation of the visualization engine, the three-dimensional models may be imported in the AutoCAD Drawing Interchange Format (DXF). However, they must subsequently be analyzed to ensure correctness and for the presence of the desired properties.

Next, the model must be annotated with material and surface properties where needed. Although this is supported by AutoCAD DXF, the annotations may not reach the level of detail required for a given scenario and may, therefore, require additional manual refinement of the model or annotation. Typical examples include walls constructed of layers of different materials or walls containing ductwork and wiring. However, aggregate characteristics may be sufficient unless the properties of a given object or area are critical to scenario elaboration.

An additional step involves the placement of entities and objects that are not found in architectural drawings. In some cases, the required locations, dimensions and characteristics may be obtained from other databases (e.g., those storing cable and pipe management information), but manual placement is required for what-if scenarios. At this point, the logical dependencies also need to be specified, including geo-references associated with relevant entities and the physical representations of objects.

The development of scenarios can entail some effort, particularly when capturing details such as the placement of cable runs, pipes and ducting required to analyze complex, cascading faults and events. However, such baseline “scenes” are likely to be reused in multiple scenarios. Therefore, we concentrate on making scenario development instead of scene development interactive, especially since scene development typically requires manual validation in any case.

The final step in scenario development is the elaboration of events and activities. Here, the developer can use the interactive visualization environment to indicate where events (e.g., gas or water leakage, fire, etc.) occur and at which point in time. As these events are fed into the various interacting sub-models, the location and time of each event must be recorded to ensure that any developments from an event such as cascading failures are captured adequately. To facilitate the use of scenarios in what-if analyses and contingency training, events can be retained in a tree structure to permit interactive backtracking and the reuse of event chains in the simulation environment.

### **3. Visualization**

The models described above capture event information for a scene graph. However, it is still necessary to transform this into a visual representation.

One of our objectives in developing the simulation and visualization environment was to provide maximum flexibility for deployment. Subject matter experts and participants in training exercises are often in different locations (hence the need for a distributed simulation and visualization environment) and substantial differences may exist in the computing resources available. To



maximize availability and bypass licensing and platform constraints, we selected jMonkeyEngine 2.0.1, a free, open source visualization engine. A Java-based system underlying the BSD open source license enables the engine to run satisfactorily on most platforms. The actual rendering mechanisms available depend on the platform with both LWJGL (Lightweight Java Game Library) and JOGL (Java OpenGL) bindings available. JOGL provides a more mature and higher-performance environment in cases where OpenGL rendering is supported by the underlying platform.

The Java implementation of the visualization system and models provides a largely platform-independent architecture for all the components. However, there is a performance impact because the visualization system and models can easily reach the limits of a particular platform. The fidelity of the underlying models can be enhanced by parallelized operation, but this is not the case for visualization. Therefore, a platform-specific rendering architecture may be necessary for scenarios where very high visual quality is required in real-time environments. We do not foresee a need for this at present; however, the sub-models described in Section 2 can be transformed to benefit from the localized parallelism afforded by general-purpose graphics processing unit (GPGPU) programming, especially for the master node(s) in order to scale up to larger numbers of visualization nodes.

### 3.1 Review Mechanism

As noted in the introduction, one of our goals is to permit the joint development and analysis of scenarios. For real-time training exercises, this may be considered a “serious game” in that it forces participants to react to complex events often with limited information and limited time to consider the most appropriate course of action or adverse effects resulting from a decision. Also, following a simulation, but especially after conducting training exercises, it may be important to review the events in a given scenario with multiple experts using a joint timeline.

To this end, simulation events are tagged with a timeline and the transactions required to reach a given configuration. When a sequence of events is to be reviewed, it is necessary to configure the time and location of the events and customize other aspects of the review such as the location and viewpoint of the reviewer and additional model information (e.g., material properties and logical dependencies). Where appropriate, visualization parameters can be adjusted to add fidelity or to reduce unnecessary aspects. For example, if the etiology of an electrical fire is to be investigated, it may be sufficient to understand the patterns in which soot deposits build up; fire and smoke simulations distract from understanding these patterns.

## 4. Evaluation

In addition to constructing individual scenarios for sub-models and minimal interacting sub-models and their visualization, we have constructed a larger



Figure 5. View of hospital area.

scenario in which all the sub-models developed are used. The scenario is based on a hospital environment. It does not represent an actual facility, but uses individual rooms and components in a composition.

The hospital scenario was chosen for several reasons. Hospitals are part of the critical infrastructure and their inability to provide services may have local as well as larger-scale effects. Also, they typically use hazardous materials and equipment (e.g., oxygen delivery systems and emergency generators) and are equipped with substantial information technology assets and capital equipment. The placement of these assets and the potential cascading effects of failures or accidents are not always fully understood. At the same time, responding to and mitigating faults requires cooperation among a large number of individuals ranging from information technology and facilities management specialists to fire rescue and emergency response units and hospital staff.

Figure 5 provides a composite illustration of the scene. A server room (lower left) is located one floor below an intensive care ward (upper left). The adjacent corridors (upper and lower right-hand side) are connected on both floors through HVAC ducting (visible in the upper right-hand quadrant). Additional cabling and pipes as well as hidden ducts that are part of the model are not visible in the illustration.

Based on the scene, we have developed several scenarios primarily to demonstrate the potential cascading effects. One scenario involved an HVAC steam pipe leak that caused localized water damage and led to an electrical fire in the server room. This resulted in the loss of the affected machine and cascading damage as the soot and smoke from the electrical fire affected adjacent computer systems. Ultimately, all information technology components in the server room were lost, requiring the use of a redundant system in another location. While circuit breakers and fire control systems kept the fire from spreading and causing further electrical outages, the hot smoke and soot escaped to the adjacent corridor and entered the intensive care ward by way of HVAC ducting, which could have had severe effects on patients.

## 5. Related Work

Visualization and physics engines have been used extensively in high-fidelity environments such as surgery simulations [9, 10]. According to Rohde and Toschlog [14], the fidelity of simulations provided by common gaming engines is comparable to that provided by military simulation environments. However, in the case of visualization for end users, it is the perceptual quality that is relevant in physics simulations. Yeh, *et al.* [21] have shown that these simulations can be adapted to avoid unnecessary computations. The gaming industry has considerable expertise in creating scalable multi-party environments; these can be leveraged in critical infrastructure simulations. Interested readers are referred to [17] for a detailed review of modeling and simulation environments.

Our sub-models were chosen for their simplicity and ability to capture sufficient detail for visualization and to understand the etiology of events rather than to establish ground truth. Large bodies of work exist in individual modeling areas; most of them are not based on first principles but are hybrids incorporating experimental results. One example of this approach is work focused on fire dynamics and concomitant effects (see, e.g., [6, 13]). A model in widespread use for simulating fire and smoke propagation in enclosed spaces is the Fire Dynamics Simulator [11] from the National Institute of Standards and Technology along with Smokeview [7], its visualization system. Similarly, explosions and blast damage (especially blast overpressure damage) have been studied experimentally and using various first-principle models (see, e.g., [4, 5]).

Polygon mesh models for analyzing flooding effects have been used in hydrological and hydrodynamic models, but typically at coarser levels of individual buildings rather than within individual structures [15]. At larger scales, the size of the model must be considered as in the case of small-scale models at higher resolutions. While fluid dynamics models are typically based on rasterization [2], more elaborate models typically rely on finite element approaches.

## 6. Conclusions

Capturing the cascading effects due to interdependencies between critical infrastructure assets requires the integration of multiple models. The distributed interactive visualization and analysis architecture presented in this paper is aimed towards constrained environments such as buildings and industrial facilities. Physical effects considered include flooding, fire, smoke and gas explosions, while the logical dependencies considered include telecommunications and electrical power. The visualization system allows first responders and subject matter experts to conduct scenario analyses and training exercises in an interactive and distributed manner.

Future work will focus on the development of a federated rendering mechanism, which is desirable from the information flow and information sharing perspectives. Also, future research will attempt to integrate explicit control-law-based physical plant models that would provide higher levels of automation for scenario-based analyses.

Note that the distributed interactive visualization and analysis system described in this paper is based entirely on free and open source software. The software is available upon request from the authors.

## References

- [1] R. Abraham and J. Marsden, *Foundations of Mechanics*, Addison-Wesley, Redwood City, California, 1987.
- [2] P. Bates and A. De Roo, A simple raster-based model for flood inundation simulation, *Journal of Hydrology*, vol. 236(1-2), pp. 54–77, 2000.
- [3] A. Boeing and T. Braun, Evaluation of real-time physics simulation systems, *Proceedings of the Fifth International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pp. 281–288, 2007.
- [4] R. Cleaver, C. Humphreys, J. Morgan and C. Robinson, Development of a model to predict the effects of explosions in compact congested regions, *Journal of Hazardous Materials*, vol. 53(1), pp. 35–55, 1997.
- [5] F. Diaz Alonso, E. Gonzalez Ferradasa, J. Sanchez Perez, A. Minana Aznara, J. Ruiz Gimenoa and J. Martinez Alonso, Characteristic overpressure-impulse-distance curves for vapor cloud explosions using the TNO multi-energy model, *Journal of Hazardous Materials*, vol. 137(2), pp. 734–741, 2006.
- [6] D. Drysdale, *An Introduction to Fire Dynamics*, John Wiley, Chichester, United Kingdom, 1999.
- [7] G. Forney, Smokeview (Version 5) – A Tool for Visualizing Fire Dynamics Simulation Data, Volume II: Technical Reference Guide, NIST Special Publication 1017-2, National Institute of Standards and Technology, Gaithersburg, Maryland, 2010.
- [8] P. Kundu and I. Cohen, *Fluid Mechanics*, Academic Press, Burlington, Massachusetts, 2008.
- [9] A. Maciel, T. Halic, Z. Lu, L. Nedel and S. De, Using the PhysX engine for physics-based virtual surgery with force feedback, *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 5(3), pp. 341–353, 2009.
- [10] S. Marks, J. Windsor and B. Wunsche, Evaluation of game engines for simulated surgical training, *Proceedings of the Fifth International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pp. 273–280, 2007.
- [11] K. McGrattan, S. Hostika, J. Floyd, H. Baum, R. Rehm, W. Mell and R. McDermott, Fire Dynamics Simulator (Version 5), Technical Reference Guide, Volume I: Mathematical Model, NIST Special Publication 1018-5, National Institute of Standards and Technology, Gaithersburg, Maryland, 2009.

- [12] J. Neal, T. Fewtrell, P. Bates and N. Wright, A comparison of three parallelization methods for 2D flood inundation models, *Environmental Modelling and Software*, vol. 25(4), pp. 398–411, 2010.
- [13] S. Olenick and D. Carpenter, An updated international survey of computer models for fire and smoke, *Journal of Fire Protection Engineering*, vol. 13(2), pp. 87–110, 2003.
- [14] M. Rohde and M. Toschlog, Toward the fusion of serious simulation and video games, *Proceedings of the Spring Simulation Multiconference: Military Modeling and Simulation Symposium*, article 71, 2009.
- [15] J. Schubert, B. Sanders, M. Smith and N. Wright, Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding, *Advances in Water Resources*, vol. 31(12), pp. 1603–1621, 2008.
- [16] R. Smith, Open Dynamics Engine (v0.5) User Guide, ([www.ode.org/ode-latest-userguide.html](http://www.ode.org/ode-latest-userguide.html)), 2006.
- [17] N. Svendsen, Interdependencies in Critical Infrastructures: A Qualitative Approach to Model Physical, Logical and Geographical Interdependencies, Ph.D. Thesis, Norwegian Information Security Laboratory, Department of Computer Science, Gjøvik University College, Gjøvik, Norway, 2008.
- [18] N. Svendsen and S. Wolthusen, A framework for 3D geospatial buffering of events of interest in critical infrastructures, *Proceedings of the Second International Workshop on Critical Information Infrastructures Security*, pp. 37–48, 2007.
- [19] N. Svendsen and S. Wolthusen, Multigraph dependency models for heterogeneous infrastructures, in *Critical Infrastructure Protection*, E. Goetz and S. Sheno (Eds.), Springer, Boston, Massachusetts, pp. 337–350, 2007.
- [20] B. Weigand, *Analytical Methods for Heat Transfer and Fluid Flow Problems*, Springer-Verlag, Heidelberg, Germany, 2004.
- [21] T. Yeh, G. Reinman, S. Patel and P. Faloutsos, Fool me twice: Exploring and exploiting error tolerance in physics-based animation, *ACM Transactions on Graphics* vol. 29(1), pp. 5-1–5-11, 2009.