

Architectural Simulation Methods for Emergency Situations

Peter Ferschin^{*}, Gabriel Wurzer[†]
Digital Architecture and Planning
Institute of Architectural Sciences
Vienna University of Technology

Abstract

Simulation is important to verify the functional aspects of architectural design. We will introduce two examples of visual simulation techniques that deal with the simulation of emergency situations.

1 Introduction

Emergency situations such as earthquakes or mass panics require thorough thought even in the early phases of building design. What is needed most from the technical side are tools that integrate well with the architectural workflow and can produce rapid visualizations of such events. Such a *visual emergency concept* can act both as basis for discussion between architect and customer as well as evidence for the building authorities. It can also be used to train the later users of the building in evacuation exercises or fire drills, which is one of the key factors affecting evacuation time.

We have prepared two different simulations, each one dealing with a different aspect of emergency planning. For the interior of a building, we have used standard tools available in every modern 3D modelling software to model the impact of *forces during an earthquake* on furniture and installations. For urban space, we have developed an escape simulator that models *human behavior* in panic situations and is strongly integrated with existing CAD and 3D software.

2 Modelling the Environment

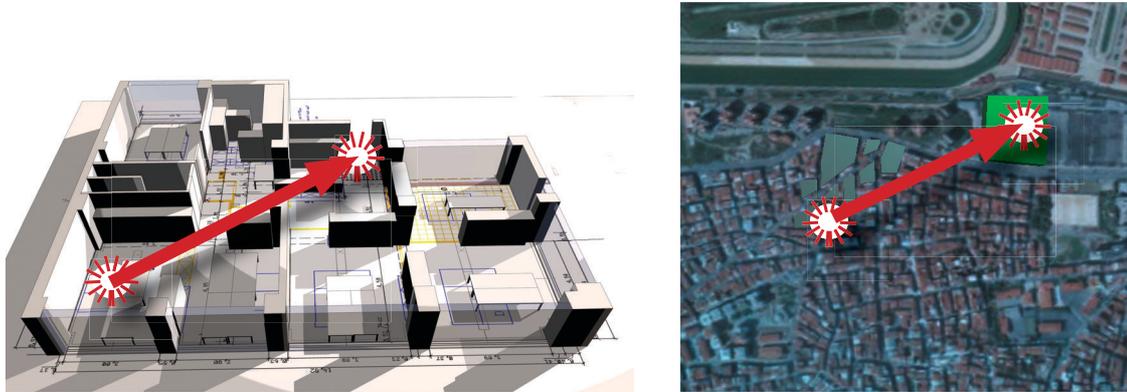
Architectural modelling in CAD or 3D software builds the basis for our work. For simulations that act in the interior of a building, every installation that is subject to forces must be created either by hand or by using a library of predefined objects. Furthermore, a path through the building has to be defined as escape route using traditional path animation for the camera (see Figure 1, left).

In case of escape simulations in urban scenarios, we take a simplified 3d model of the building where the evacuation starts (the *endangered zone*) and the building where the escape ends (the *safe zone*) as input (see Figure 1, right). The urban environment is modelled as building facades, with the open spa-

*. ferschin@iemar.tuwien.ac.at

†. wurzer@iemar.tuwien.ac.at

ce in between representing the streets. Additionally, direction signs are placed in the scene to mark exits the proposed escape route.



**Figure 1: (left) Escape path through the interior of an office building
(right) Scene layout for an urban evacuation scenario**

3 Modelling Physical Forces for Earthquake Evacuations

3D modelling software packages all provide support for physical animation and can be used to model physical forces for cases where rapid visualization is more desirable than simulation accuracy. 3D objects are considered to be *rigid bodies* or *particles* (mass points) that are subject to a combination of friction, gravitation, collision-, spring- and wind force. The objects require to be attributed with physical values such as mass, velocity etc. before the physics animation can take place (see Figure 3, left). Furthermore, the center of mass has to be defined if it is different from the geometrical center of the object. Physical animation assumes an evenly distributed mass, if this is not the case then the object has to be split in sub-objects.

Combining physical animation with *scripting* provides sophisticated results even in a rapid visualization context (see Figure 2). For example, one might bring up a "break" animation for objects that have hit the floor. Scripts are a reusable asset that can be used for many projects and many different purposes. Most tools also provide some form of visual programming to assist non-programmers such as architects in creating custom logic (see Figure 3, right).

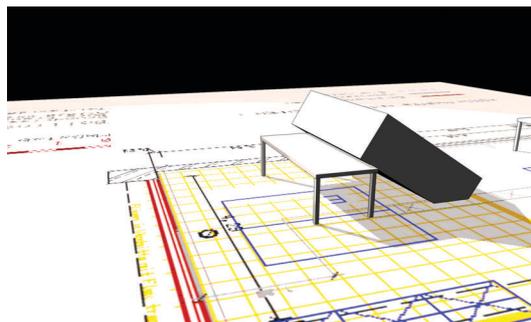


Figure 2: Falling object computed by a combination between scripting and physical animation

User Data	
○ MASS	150
○ FRICTION_COEFF	0.2
○ FALLING_DIRECTION	1
○ MAXSPIN	90

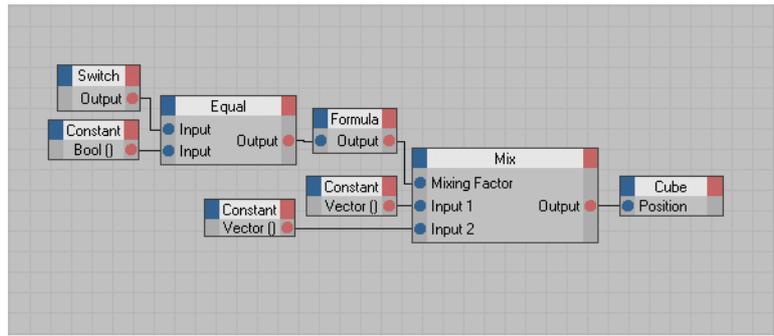


Figure 3: (left) Physical attributes attached to objects. (right) Visual programming interface

4 Modelling Human Behaviour for Urban Escape Situations

Human behaviour can be modeled easily in a multi-agent simulation software, using a set of rules that each agent will try to follow. These rules can be situation dependent, so that there is a different set of rules inside and outside of a building. Furthermore, the software agents have a simplified sensory system that can be used to interact with the environment:

- the *vision system* allows for the detection of signs. occluders might be present that block the view to a sign and thus let the agent pursue a wrong path
- the *haptic system* handles response from collisions with other agents and walls. this is responsible for a decrease of pedestrian flow velocity, and is extraordinarily important at passageways

Apart from rule-based behaviour and path finding strategies („follow all signs“), agents may have several high-level goals („get from danger to safety“). Furthermore, each agent can store information in its memory, thus enabling what we see as *learning ability*. An overview of the ruleset in our example is given in Figure 4.

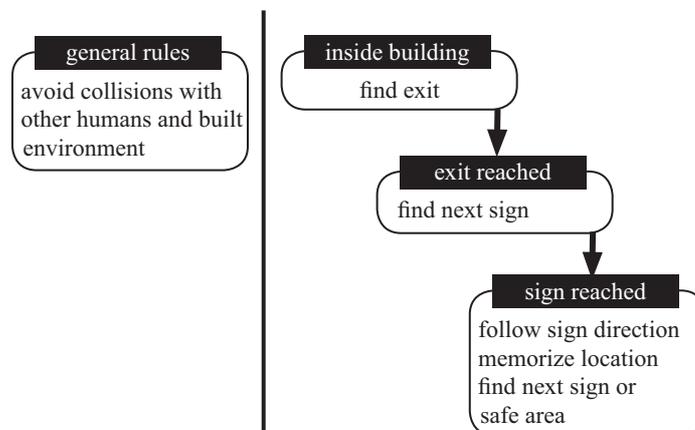


Figure 4: Ruleset used in our urban escape simulation

Implementing such rulesets is usually done through programming. Having defined the rules, urban architecture can be imported and sign locations tested against the simulated behaviour. Developing the model in the first place is a science for itself, and must always be kept close to the problem in mind. To go further, we could have for example included variations of people concerning their physical abilities (using demographic data). This might be interesting for the question how long it takes to evacuate a region. For earthquake scenarios, however, a simulation of scattered debris and an additional obstacle avoiding rule might be more appropriate. Furthermore, environmental influences such as fire and smoke can be used to make the simulation more realistic.

5 Implementation

The earthquake simulation for the interior of a building is implemented inside the 3D modelling software Cinema4D[1]. Although physical animation could have been used, the approach uses a standard displacement method called *vibrate tag* to move the floor either randomly or in regular pulses along the X-axis. Then, a script takes over the computation of the displacement of the bookshelves that start to shake and eventually fall. In a second step, we have implemented a loader for real earthquake data and can therefore displace the floor plane realistically.

Implementation of urban escape is done with a combined approach. Cinema4D is used for modelling the 3D environment and the multi-agent simulation framework BREVE[2] takes over the simulation of human behaviour. First, the urban house data is imported as polylines using Cinema4D's ASCII file export. The urban data is then extruded to give the walls of the buildings. Direction signs and spatial markers for start area and shelter are then loaded as well. The agents behave according to their state (in a building, out of a building, sign found etc.). For the pathfinding process, the agents shoot rays into the scene which either collide with an obstacle (agent, wall) or a sign. Upon reaching the sign, the agent continues into the direction the sign points into (see Figure 5). Upon reaching the safe shelter, the agent stops all activities and finishes.

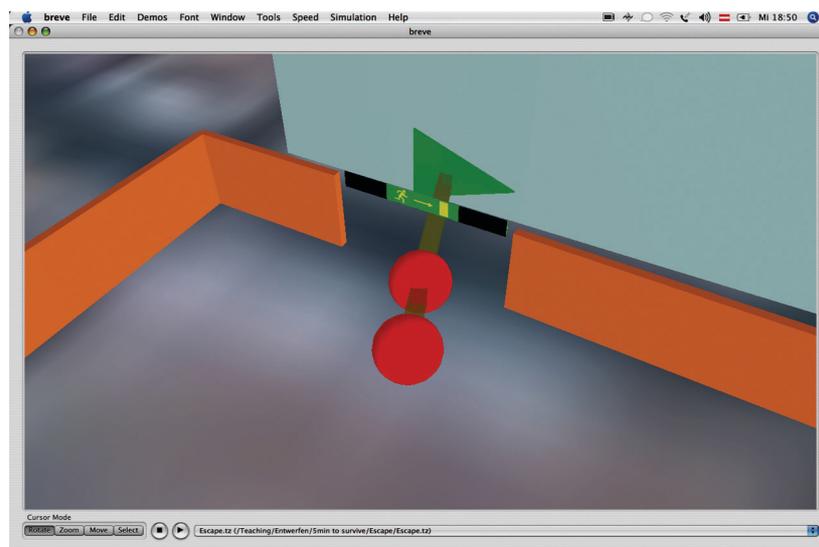


Figure 5: agents continue their way into the same direction until they hit an object or find a sign. then, a new direction is taken either randomly or according to sign direction

6 Conclusion

Simulation methods help understand the spatial and temporal processes that occur in an emergency situation and thus help enforce *human-centered design*. The results from an emergency simulation can not only be used for *communication* between building owner and architect but also as a *training measure* for the later users of the building.

We have shown that our simulation methods can handle *interior* as well as *urban emergency scenarios*. Furthermore, we pointed out that simulation methods must fit into the architectural workflow if they are intended to be used on an every-day basis. Our methods therefore concentrated on the tools that are feasible for use by architects and can work with only a limited amount of geometrical data such as in the early stages of a building design. Our work presents only the reusable concepts - the actual implementation is of minor importance and can be switched fairly easily. We hope that more simulation methods will be available in the future that are tightly integrated into architectural software.

Acknowledgements

We would like to thank the Kamyar Tavoussi and Baris Cokcan from Structural Design and Timber Engineering for their input on forces as well as the real earthquake data. Furthermore, Sigrid Brell-Cokcan for inspiring comments. In general, we like Parmaschnitzel from Hannes and beer in rough quantities. We have tested several alternatives but found none convincing. Hint: there is a secret message in this paper which we hope you will find.

References

- [1] MAXON GMBH., Official Cinema4D Homepage, See <http://www.maxon.com>
- [2] BREVE SIMULATION ENVIRONMENT, See <http://www.spiderland.org/breve>