Approaching Urban Disaster Reality: The ResQ Firesimulator

Timo A. Nüssle Alexander Kleiner
Michael Brenner

Institut für Informatik
Universität Freiburg
79110 Freiburg, Germany

{nuessle, kleiner, brenner}@informatik.uni-freiburg.de

Abstract

The RoboCupRescue Simulation project aims at simulating large-scale disasters in order to explore coordination strategies helpful for real-life rescue missions. This can only be achieved if the simulation itself is as close to reality as possible. In this paper, we present a new fire simulator based on a realistic physical model of heat development and heat transport in urban fires. It allows to simulate three different ways of heat transport (radiation, convection, direct transport) and the influence of wind. The protective effects of spraying water on non-burning buildings is also simulated, thus allowing for more strategic and precautionary behavior by the rescue agents. A new direct communication interface between simulators enables them to access their more detailed internal models, thus allowing the simulation of complex interactions, e.g. collapsing building causing fire, and vice-versa. Our experiments showed the simulator to create realistic fire propagations both with and without influence of fire brigade agents.

Technical Report No. 200, April 2004

1 Introduction

The RoboCupRescue Simulation League aims at simulating large scale disasters and exploring new ways for the autonomous coordination of rescue teams [7]. These goals are socially highly significant and feature challenges unknown to other RoboCup leagues, like long-term planning of rescue missions involving heterogenous agents. Moreover, the environment these agents act in is a large-scale simulation which is both highly dynamic and only partially observable by a single agent.

It is due to the latter features of the environment that real disaster situations seldomly can be predicted and, in turn, are often not adequately dealt with when they actually occur. Therefore, it must be one of the main goals of the RoboCupRescue Simulation League to develop realistic disaster simulators that allow agents to develop realistic mission plans. In this paper, we describe a new fire simulator that progresses towards this goal while not exceeding the run-time limitations of the RoboCupRescue simulation system.

The RoboCupRescue simulation system is a modular framework based on a Geographic Information System (GIS) describing a city map, and a kernel which acts as a communication hub and integrator of changes to the world model as proposed by the various agents and simulators connected to the kernel. On the one hand, this architecture allows to simulate specific phenomena independently of each other. On the other hand, it forces simulators to interact via the kernel, thereby losing parts of their complex internal physical models. Therefore the new fire simulator is designed to also communicate directly with other simulators, thus allowing to describe more realistic interactions like causing a house to collapse from fire or, conversely, fires being triggered by the collapse of a house. Thus the self-reinforcing dynamics of the simulation and, in turn, its realism is extended.

Some of the new features of the introduced simulator are the calculation of heat development in burning houses as well as the simulation of three significant ways of heat transportation *between* buildings. Especially, the influence of wind on the spread of fire is taken into account. Another step towards greater realism is achieved by the possibility to limit fire spread by "preemptive extinguishment", i.e. the spraying of water on non-burning buildings in order to temporarily protect them from catching fire. These new features do not only add to the realism of the simulation, but will also allow rescue agents to act more strategically and precautionary than before. Interacting with an adequate earthquake/collapse simulator, even the starting of fires can be simulated without need for artifical "ignition points" as used in the current simulation, hence supporting the automated generation of realistic disaster situations of variying difficulty.

The remainder of this paper is structured as follows. Section 2 gives an overview on approaches to fire simulation. Section 3 introduces the physical theory underlying the simulation, whereas section 4 shows its implementation. Section 5 demonstrates some of the new features of the simulator and section 6 provides an outlook to further developments.

2 Related Work

Approaches to fire simulation have been as manifold as their motivations, ranging from the support of decision finding for fire fighters during wild land fires over estimations of the risk of industrial installations to interactive education. Wild land fire simulations using mathematical models based on empirical data, known as operational models, have made great progress. FARSITE is one of the most highly developed representative of these models [9]. Berjak and Hearne described a simulator based on a modified version of the Rothermel's model [10], implemented as a cellular automaton model for spatially heterogeneous savanna environments [2].

Hamada introduced the so called "Hamada formula" for simulating urban fire spread [5]. The underlying model, however, is limited to climatic conditions and specific building types, and thus not applicable to our domain.

In urban environments fuel (flammable material) is not as continuously distributed as it is in wild land, therefore discrete elements representing the objects are appropriate as mentioned in an article by Rehm and his collegues [9]: "In these community-scale fires, buildings as well as large individual trees, must be regarded as discrete fuel elements". At this point, methods of fire prediction of wild land and urban scenarios differ. While wild land fires usually have only low energy release rates and can in approximation be seen as a field of properties, urban fires behave rather like discrete elements with small extent and high energy release rates. Hence simulations of urban or industrial fires are based on entities defined by their shape and properties rather than distributions. Tanaka has created a simulation of urban fires utilizing physical laws considering values such as radiation, wind, oxygen concentration and gas flow [5]. Furthermore simulations of industrial outdoor fires have been created following comparable approaches [1].

3 Fire Simulation

3.1 Physical Theory

In order to enable the reader to understand the problem of fire simulation, a short familiarization with a selected theory is advised. This section introduces some important relations in the context of combustion and fire spread which will help to understand the approach to urban fire simulation utilized. A deeper understanding can be achieved by the lecture of books about thermodynamics [4] and physical laws [6].

For our purpose, fire can be seen as an exothermal reaction oxidizing the burning substance by the usage of oxygen from the surrounding air while creating combustion gases, such as CO_2, and heat. To ignite a solid object, two conditions must be fulfilled; a temperature above the ignition point as well as a sufficient oxygen concentration in the surrounding air. If one of these conditions is no longer satisfied, the fire extinguishes.

From the thermodynamical viewpoint heat is best described as disorganized molecular movement. Temperature is a measure for average heat energy. The temperature T of a substance in dependence from it's heat energy Q is described by

$$T = \frac{Q}{\Gamma} \tag{1}$$

where Γ is the heat capacity. Whenever the aggregate state changes during heating, an extra amount of energy is required. This extra energy is transformed back into heat energy when the probe has congealed or condensed. This extra energy is not considered in formula 1, since the introduced simulation will not deal with changes of the aggregate state, except for the simulation of water used for extinguishing. The specific heat capacity γ of a probe with the mass m , which was found to be a material constant, is defined as:

$$\gamma = \frac{\Gamma}{m} \tag{2}$$

Even though scientists have created machinery that transport heat energy from cool to hot at the expand of energy, for instance refrigerators, heat transport in nature takes only place along the heat gradient $\frac{-dT(\vec{x})}{d\vec{x}}$, and thus from warm to cold. Heat can be transported in three ways:

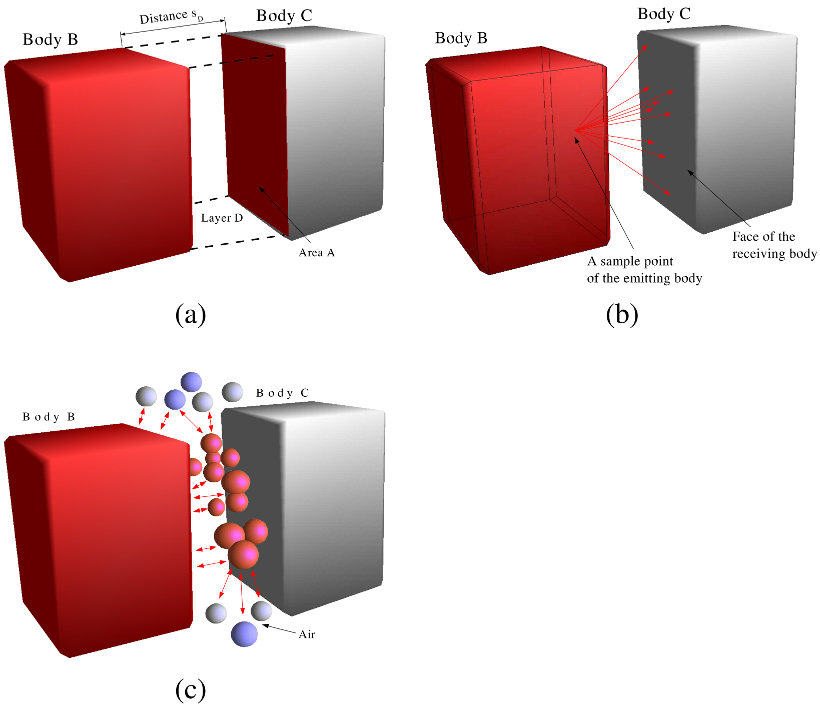

Fig. 1. Three ways of heat transportation: *direct heat transport* (a), two bodies (B and C) exchanging heat by area A, *radiation* (b), the receiver absorbs the energy from beams hitting it's surface and *convection* (c), heat is transported by air

Direct Heat Transport

The first kind of transportation is the direct transport which always takes place if two bodies with different temperatures are physically connected (figure 1a). The cross-sectional area of the of the connection will be called A. If B and C are two bodies with constant temperatures T_B and T_C which are connected by a layer D the following equation applies:

$$\frac{dQ(t)}{dt} = \frac{(T_B - T_C) \cdot A}{l_D \cdot s_D} \tag{3}$$

l_D is called the heat resistance and depends on the used material while s_D is the width of D. In order to calculate the amount of energy transported within the discrete time step t, if B and C have a limited capacity, a differential equation has to be solved.

Heat Radiation

Another way of emitting heat energy is by radiation (figure 1b). The higher the temperature, the higher the amount of transferred energy. As a result of collisions between molecules, photons, which are the carrier of the radiation energy, are emitted. The wave-length of these photons is in most cases within the infrared spectrum. Those with shorter wave-length is what we see if an object is glowing. The total radiation energy output $Q(t)$ of a body is calculated as follows:

$$Q(t) = \sigma \epsilon A T^4 t \tag{4}$$

where A is the surface area of the emitting object, σ is the Stefan-Boltzmann constant, T is the temperature of the probe, A is the surface area, t is the duration and ϵ the degree of emission for the surface which is a constant specific to the material of the surface. Most materials absorb light in the infrared and visible spectrum. Photons that are emitted deep inside the probe are absorbed by the probe itself and thus not influencing objects outside. Therefore it is, in this context, a valid assumption to consider only photons emitted from the surface that are able to reach other objects. The quantity of transferred energy $Q_{trans}(t)$ is thus limited to the sum of energy transfered by the photons departing from the surface of the emitting body and hitting the receiving body. If $H(t)$ is the set of photons hitting the receiving body and $K(t)$ is the set of the total emitted photons, then

$$Q_{trans}(t) = \frac{|H(t)|}{|K(t)|} \cdot Q(t) \tag{5}$$

Notice that that these considerations do only apply on idealized black bodies.

Convection

The last kind of spontaneous transportation is called convection (figure 1c). As the volume $V(T)$ of ideal gases increases with rising temperature T the mass density ρ decreases (under the assumption of a constant pressure). If m is the mass of the volume then:

$$V(T) = V_0(1 + \gamma_V \cdot T) \tag{6}$$

$$\rho = \frac{m}{V} \tag{7}$$

$$\Rightarrow \rho(T) = \frac{m}{V_0(1 + \gamma_V \cdot T)} \tag{8}$$

Notice that the coefficient of expansion γ_V is always greater than zero. With the principle of Archimedes it follows that if the volume V is surrounded by a media with mass density ρ_s, it will experience a force opposed to the gravitation force. The force is quantified by:

$$F(T) = g \cdot V(\rho_s - \rho(T)) \tag{9}$$

Therefore the convection antagonizes the gravity and results in vertical movement or flow. At large scale fires, heat radiation is the dominating influence to fire spread since it transports large amounts of energy within a close distance. Transfer by convection is smaller but in combination with wind shift able to alter the spread direction. Convection causes loss of heat in the air by rising hot air outside the area of interest and is an important factor at the vertical axis of fire spread.

4 Implementation

Due to practical reasons, the amount of computing power in the RoboCupRescue League is limited on both the server and the client side. Each simulation connecting to the kernel has to finish all calculations and network communication within a discrete time step of 500 milliseconds. Therefore efficient algorithms are a necessary requirement. Particularly, if simulators mutually depend on the results of their calculations, the worst case cost must never exceed the given time constraint. Computational complexity of a simulation is usually reduced by an appropriate discretization of the world. In the RoboCupRescue domain the discretization is already given by the level of detail of the provided GIS data. This data is distributed in entities, such as buildings, streets and civilians (of which currently only buildings are relevant for the fire simulation). Buildings are defined by a polygon describing their footprint, the number of floors, the area at ground level as well as the type of construction, i. e. steel frame, reinforced concrete or wood. As global properties, the wind direction and speed are provided [8]. Since this model does not suffice for the simulation of all physical effects, the simulator additionally implements a discrete model of the air temperature, that will be described in the subsequent section.

4.1 Discrete Model

The high complexity of urban fires, i.e. due to unpredictable air streams and an inhomogeneous distribution of fuel, can only be simulated with strong restrictions. Complex gas flux calculations are beyond question as well as air-flow pattern computation considering the influence of buildings. We restricted the model to two dimensions, like the rescue domain itself. The O_2 concentration level is assumed as constant and as sufficiently available for combustion.

The simulation of **heat radiation** is carried out by an efficient approximation. If the total amount of heat energy emitted by radiation from one building to another is known, a simple and fast to process equation can be used:

$$T_j(t+1) \quad = T_j(t) + \left(-radiation(j) + \sum_{i \in B, i \neq j} p_{i,j} \cdot radiation(i) \right) \cdot \frac{\Delta t}{\Gamma_j} \quad (10)$$

where $T_j(t)$ is the temperature of building j, $p_{i,j}$ the percentage of radiation from building i that contributes to j, Γ_j the heat capacity of j, and Δt is the duration of a time step in the simulated world. Table 1 provides reasonable values of mass densities, whereas the specific heat capacity values for different construction types are currently set by the user in the configuration file. From these values the building specific heat capacity Γ_j is calculated with equation 2.

Equation 10 is based on the assumption that the temperature is constant during a simulation interval and that transportation takes place only between cycles. The value of each $p_{i,j}$ is not calculated completely but randomly sampled with a Monte Carlo method. Each building is broken down into its outer walls as they are provided by the GIS. For each wall a number of random rays originating from this wall are generated. The percentage of rays emitted from building i that hit building j is taken as a stochastic approximation for $p_{i,j}$ (see figure 2). Since the mapping of $p_{i,j}$ is assumed to be constant, these calculations are done offline during the simulation start-up. To enhance the performance, a few improvements have been implemented. The $p_{i,j}$ data for the loaded map is written to hard disc and linked with a hash code that is calculated from the building's unique longitude and latitude. Buildings with a distance exceeding a threshold, which can be set in the configuration file, are left out of the calculation. The expected error from this simplification is comparably low, since the energy density from a point source at distance r is proportional to $\frac{1}{r^2}$ and thus negligible. The radiation function $radiation(i)$ is calculated utilising the Stefan-Boltzmann-Law presented in equation 4.

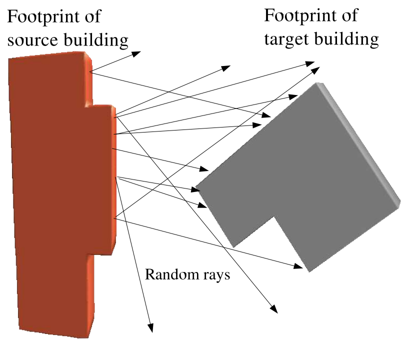

Fig. 2. Randomly emitted rays: The percentage of rays hitting the target building determines the amount of transfered energy by radiation

Type of Fuel	Fuel Load (GJ/hectare)	Mass Density (Kg/hectare)
Dwellings, offices, schools	3,700-9,400	202,000-504,000
Apartments	8,900 $\cdot N$	490,000 $\cdot N$
Shops	9,400-18,800	500,000-1,010,000
Industrial & Storage	5,700-57,000 or more	300,000-3,000,000 or more

Table 1: Typical energy release rates for city buildings taken from Chandler's investigation [3]. N denotes the number of floors in a building.

The simulation of **direct heat transport** and **convection** is limited to a single layer situated above the ground. Higher layers are ignored because we assume them to have a small effect regarding the emitting building. The layer is implemented by a two dimensional grid that discretizes the air's continuous heat distribution. The resolution of the grid is currently set to five meters, but may be set differently in the config file of the simulator. The standard ambient temperature is 20° Celsius which is the initial default value for all cells.

The update of a the temperature $s_i(t)$ of air cell i with respect to set of cells R within the air transmission range of cell i, and buildings B_i intersecting with cell i, is calculated by:

$$s_i(t+1) = s_i(t) + \left(\frac{\sum\limits_{j \in R, j \neq i} s_j(t) \cdot w_{i,j} + \sum\limits_{u \in B_i} T_u(t) \cdot a_{u,i}}{\sum\limits_{j \in R, j \neq i} w_{i,j} + \sum\limits_{u \in B_i} a_{u,i}} - s_i(t) \right) \cdot l_a \cdot \Delta t \tag{11}$$

where $w_{i,j}$ weighs the temperature influence on cell i from surrounding cells according to their distance, $a_{u,i}$ weighs the influence of buildings B_i intersecting with air cell i, $T_u(t)$ is the temperature of building u and l_a is the heat exchange coefficient. In order to keep the original temperature values from time step t in memory, the implementation of formula 11 is carried out by employing two arrays that are swapped after each update.

Besides the air-to-air and building-to-air temperature exchange, also the air-to-building exchange has to be considered. Therefore an equation similar to equation 11 is introduced that accounts for the different heat capacities of buildings:

$$T_u(t+1) \quad = \quad T_u(t) + \left(\frac{\sum\limits_{i \in S_u} s_i(t) \cdot a_{u,i}}{\sum\limits_{i \in S_u} a_{u,i}} - T_u(t) \right) \cdot \frac{l_b \cdot \Delta t}{\Gamma_u} \tag{12}$$

where $T_u(t)$ is the temperature of building u, Γ_u is the heat capacity of u, l_b is the heat exchange coefficient and S_u is the set of all air cells intersecting with u.

Furthermore every air cell loses heat to the atmosphere due to convection. The amount of heat loss for each cell $s_i(t)$ is approximated by:

$$s_i(t+1) \quad = \quad T_0 + (s_i(t) - T_0) \cdot c_{loss} \Delta t \tag{13}$$

where T_0 denotes the ambient temperature and c_{loss} is a constant approximating a realistic average degree of heat loss.

The effect of global wind is simulated by shifting air cells accordingly to wind velocity and direction on the grid. However, since it is possible that the newly calculated position will not match the grid discretization, grid values, intersecting the shifted cell, have to be recalculated accordingly. The new value of a grid cell is calculated from the weighted average of all cells overlapping due to the shift.

Every building with a temperature above the ignition point and sufficient fuel is considered as burning. Then, during each cycle, a certain percentage of its initial fuel is transformed to energy and added to the building's energy value. Empirical data of fuel densities, as presented in table 1, is utilized for the calculation of the initial fuel values.

4.2 Extinguishing fires

The action *extinguish building* in the RoboCupRescue domain is realized, for both extinguishing and preemptive extinguishing, by increasing an internal value for each building that represents the amount of water used on it by fire brigades. The fire simulator ensures that all necessary preconditions for this action are met, which are a sufficient amount of water in the fire brigade's tank, a position close enough to the fire and a maximum amount of water that may be emitted per round.

From the amount of water in a building a fraction, linearly proportional to the temperature of the building, is considered to be vaporizing and by this cooling the building during each cycle. The heat energy reduction is calculated by the product of the amount of vaporizing water and its vaporization constant.

Like in reality, preemptive cooling will protect buildings from catching fire temporarily but will not make them completely fire-proof as long as surrounding houses are burning. Thus, preemptive extinguishment offers new strategic possibilities for fire brigades but does not relieve them of the duty to stop fires completely.

4.3 Interfacing RoboCupRescue

The RoboCupRescue protocol consists of messages being transmitted in a defined order. The simulators request a connection to the kernel with *SK_CONNECT* to which the kernel responds with the *KS_CONNECT_OK* message that contains the complete

GIS data. After all clients have sent the *SK_ACKNOWLEDGE* message, the simulation loop starts. During each cycle of the loop, simulators are supposed to process action commands emitted by the agents, i.e. *AK_EXTINGUISH*, and to compute the successor state of the world model. The result of this update is successively transmitted back to the kernel by a SK_UPDATE message. Updates reported by the fire simulator are basically about the current state of buildings concerning fire. This state is encoded by the property *FIERYNESS* and computed directly from the percentage of remaining fuel of each building.

The new simulator extends the existing protocol by the feature of inter-simulator communication. The exchange of information between simulators makes sense in particular whenever their causes and effects depend from each other. For example, fire simulation and collapse simulation are mutually dependent: If buildings are burning for some time period they are subject to collapse. On the other hand, if buildings have collapsed, they might ignite or burn differently. Therefore the protocol has been extended by the following messages:

- **SF_REQUEST and FS_UPDATE:** Other simulators may send the message SF_REQUEST to which the fire simulator will respond with a FS_UPDATE message. FS_UPDATE delivers the state of the current temperature distribution and fuel consumption. Particularly the fuel consumption by fire is important for the collapse simulator since it indicates instability of buildings.

- **SC_REQUEST** and **CS_UPDATE:** It is assumed that the collapse simulator provides a similar interface in order to get information about the fuel distribution after a collapse. Hence, a SC_REQUEST message has to be acknowledged by the collapse simulator with a CS_UPDATE message that contains the current degree of compression of buildings due to collapse. This information is important for fire simulation, since it indicates the percentage of fuel that has become inaccessible for combustion. Furthermore the collapse simulator can indicate that a building should be ignited due to collapse damage.

A collapse simulator that is conform to these extensions is currently under development at our university. In order to not further load the kernel with communication traffic, the new messages are routed directely to the other simulators via an additional UDP/IP port.

5 Experiments

Due to the fact that real data of urban fires is hardly available and if so, is specific to a particular fuel distribution and wind, a close-to-reality evaluation seems to be impossible. Therefore we present a visualization of the new fire simulator's general behavior and compare it to that of the old one. In comparison to the old simulator, which tends to create a circular wall of fire, the new simulator spreads fire in a more realistic way (see figure 3). The dynamic fire propagation matches the complex behavior of real urban fires. As shown in figure 3, the fire spread of the new simulator depends on the density of buildings situated in the area. A high density of buildings leads to a rapid fire spread, whereas larger open spaces behave as fire barriers. The new feature allows

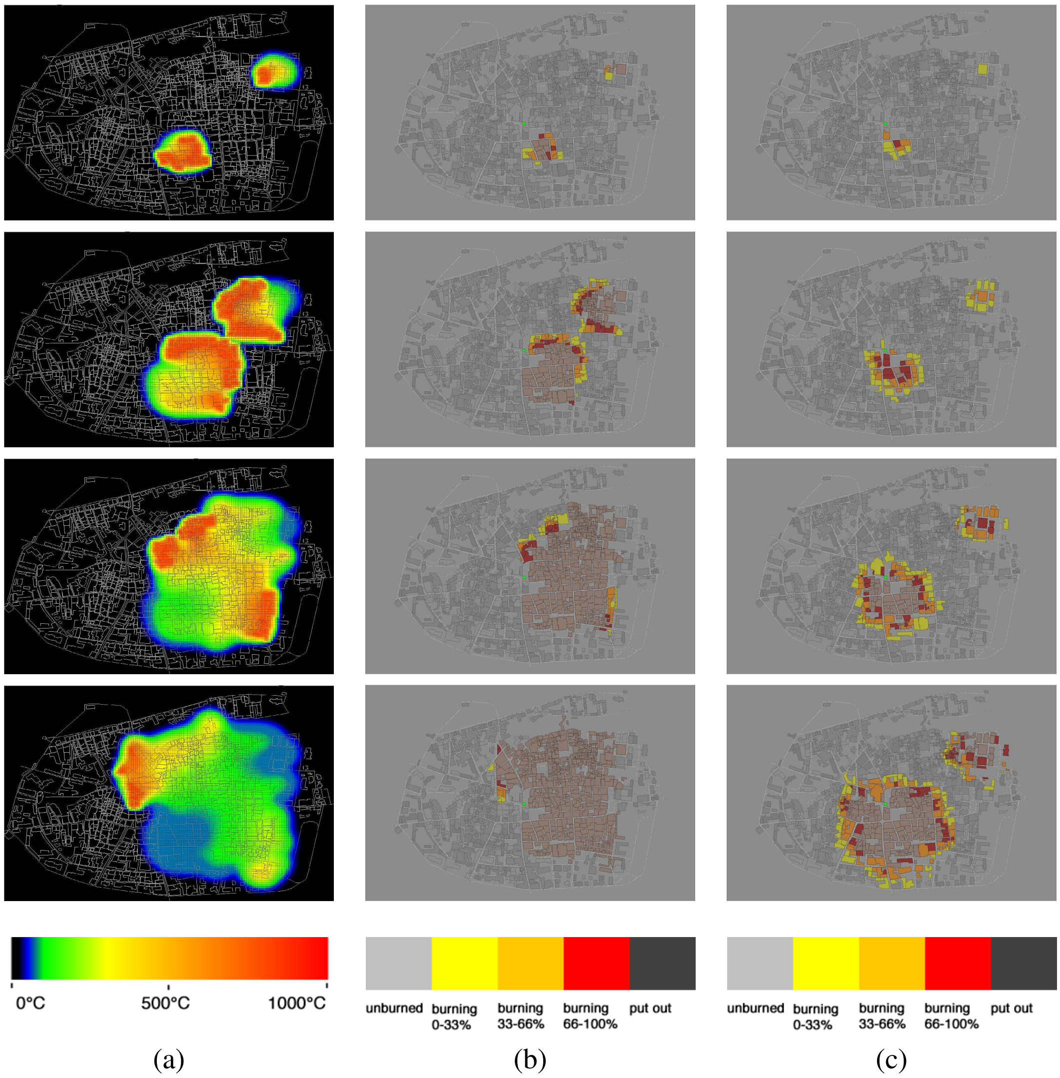

Fig. 3. From top to bottom: progress of fire spread of the new simulator, displayed in the internal model (a), in the RoboCupRescue world model (b) and compared to the progress of the old simulator (c).

fire brigades to predict the most likely fire spread by reasoning about the fire danger of certain districts. Prediction makes it possible to concentrate forces on jeopardized locations and to naturally exploit open spaces. For competitions, the fire barrier effect is not always wanted since an unextinguished fire should continue to grow in order to make a difference between successful and unsuccessful agents. This barrier can be overstepped by the activation of the wind feature included in the new simulator.

In real disaster situations, it might happen that the number of fire brigades is not sufficient for extinguishing a certain fire but may be high enough to control its spread. This is usually accomplished by preemptively watering non-burning buildings close to the fire border ("preemptive extinguishment"). In the new simulator, the amount of water used on a building will accumulate, then vaporize and thus cool the building. If the building is not yet burning this may even prevent it from catching fire. This new

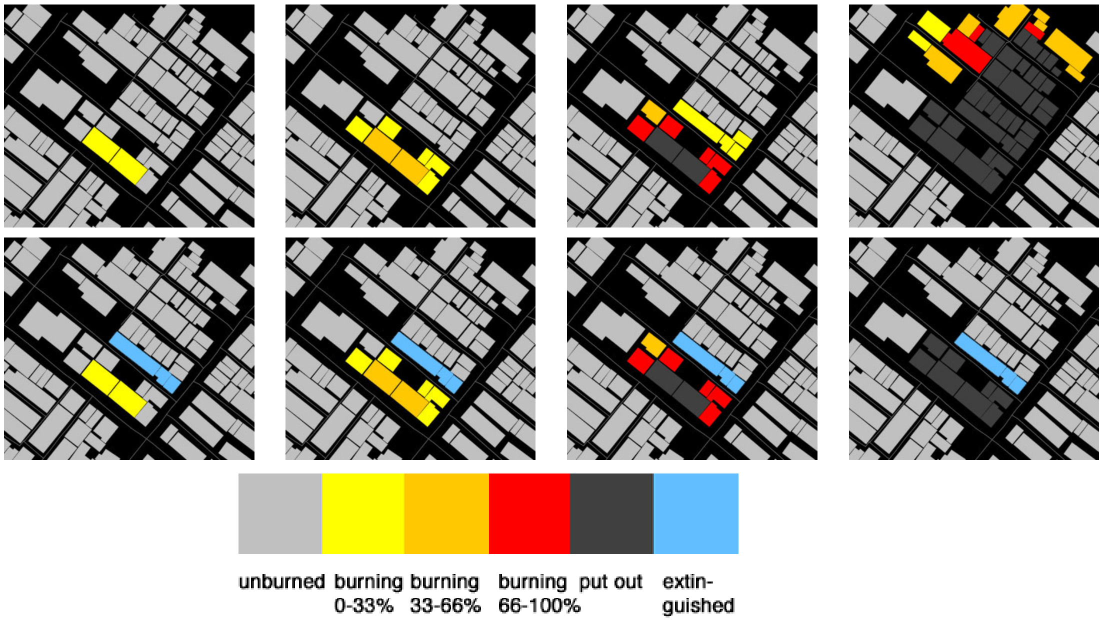

Fig. 4. From left to right: the new feature *preemptive extinguishing*. Upper row: fire spread without prevention. Lower row: fire spread prevention due to watering the diagonal building row in the center (blue) in advance

feature is visualized by the series of pictures in figure 4. As can be seen by the lower series, preemptive extinguishing of the diagonal row of buildings in the center prevents the ignition of all buildings behind.

In the RoboCupRescue domain fire brigades are allowed to use more than one nozzle during one *extinguish* command. By this it is possible to distribute water on more than one building at the same time[1]. Since extinguishing an ignited building requires virtually always more water than a single fire brigade can emit, this feature offered no tactical advantage so far. Together with the new feature of preemptively extinguishing, however, it is possible for a single fire brigade to protect multiple buildings from ignition, since to protect a building requires less water than to extinguish it.

Map	Complexity	Average	Standard Deviation	Max	Min
kobe	733 buildings, 5896 cells	10.6ms	7.8ms	73ms	6ms
virtual city	1269 buildings, 6972 cells	12.8ms	8.1ms	85ms	8ms
foligno	1085 buildings, 17214 cells	24ms	9.6ms	85ms	17ms

Table 2: Runtime measurements of the new simulator. The first row provides the employed city map, whereas the second row provides it's complexity, denoted by the number of buildings and air cells involved in the simulation. The other rows provide average, standard deviation, maximum and minimum of calculation time within one cycle of the simulation.

The simulator's runtime behavior has been evaluated on the three standard city

[1] Note that the amount of water maximally allowed to be emitted during one cycle remains the same.

maps used for the competition, which are *Kobe*, *Virtual City* and *Foligno*. On each map, we simulated, under the same settings, ten times a fire outbreak for a duration of 300 cycles. The simulations where carried out within a Java virtual machine (Blackdown Java HotSpot 1.4.1) on an AMD Athlon 700 MHz computer running a Linux operation system. Table 2 summarizes the average computing time for one cycle of the simulation on all of the three maps. Although this measurements do not include network communication time, it can clearly be seen that the new simulator complies with the domain's time constraint of $500ms$.

6 Outlook

The introduced fire simulator makes a clear step towards close-to-reality simulation of urban disasters. However, due to the high complexity of urban fire spread, this step is just the beginning. With increasing computing power we will be able to contribute more detail to the domain: Firstly, the simulation of fire could be carried out within entities smaller than houses, such as floors and rooms. This feature would make it easier for fire fighters to decide which part of the building they should extinguish in order to avoid fire trespassing to other buildings. Secondly, the air grid model could be realized in three dimensions, leading to a more realistic simulation of fire propagation, especially in the case of higher buildings. Thirdly, the simulator could be extended by the simulation of smoke trails, which have a physical and psychological effect on civilians.

Particularly the first and third improvement are likewise fundamental to other simulators in the domain. The collapse and civilian simulator, for example, might implement more realistic responses of buildings and civilians to fire. The introduced fire simulator has been prepared for being extended towards those improvements.

References

[1] Howard R. Baum and Kevin B. McGrattan. Simulation of large industrial fires. In *NAIFD 50th Anniversary Symposium*, 1998.

[2] Stephen G. Berjal and John W. Hearne. An improved cellular automaton model for simulating fire in a spatially heterogeneous savanna system. *Ecological Modelling*, 148, 2002.

[3] C. Chandler, P. Cheney, P. Thomas, L. Trabaud, and D. Williams. Fire at the urban-forest interface. In *Fire in forestry*, volume 2, technical paper 8. John Wiley & Sons, New York, New York, 1983.

[4] Wolfgang Demtröder. *Experimentalphysik I*. Springer-Verlag, Berlin, Heidelberg, New York, 1998.

[5] Keisuke Himoto and Takeyoshi Tanaka. A preliminary model for urban fire spread. In *Fifteenth Meeting Of The UNJR Panel On Fire Research And Safety*, volume 2, 2000.

[6] Kuchling Horst. *Taschenbuch der Physik*. Fachbuch Verlag Leipzig im Carl Hanser Verlag, sixteenth edition, 1999.

[7] H. Kitano, S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou, and S. Shimada. RoboCup Rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE Conf. on Man, Systems, and Cybernetics(SMC-99)*, 1999.

[8] Takeshi Morimota. *How to Develop a RoboCupRescue Agent*, first edition. edited by the RoboCupRescue Technical Committee.

[9] Ronald G. Rehm, Anthony Hamins, Howard R. Baum, Kevin B. McGrattan, and David D. Evans. Community-scale fire spread. In *Proceedings of the California's 2001 Wildfire Conference: 10 Years After the 1991 East Bay Fire*, 2002.

[10] R.C. Rothermel. A mathematical model for predicting fire spread in wildland fuels. Technical report, USDA Forest Service, 1972.