The Black Sheep Robocup Rescue Simulation Infrastructure Team Description

Daniel Bertinshaw, Jonathan Teutenberg, Cameron Skinner

Department of Computer Science
University of Auckland
New Zealand
dber021@ec.auckland.ac.nz
jteu004@cs.auckland.ac.nz
cam@cs.auckland.ac.nz

Abstract. The simulation of fire propagation and control is one of the most critical components of the Robocup Rescue environment. The current fire simulator provides only a rough approximation of actual fire spread in a real-world situation. Our new fire simulator moves away from the simple building-distance model currently used, instead calculating heat transfer based on building geometries. This has been implemented using well understood physics and in consultation with local emergency services to achieve more realistic results.

1 Introduction

We have identified four main areas in which we feel the simulation of fire spread and control in the Robocup Rescue Simulation can be improved. These areas are the burning of an individual building, the ignition of a building based on its neighbours, the extinguishing of a burning building and providing 'fair' randomness for a competition. In addition to improving these parts of the simulator we also attempt to retain, if not improve on, the efficiency of the current simulator.

This fire simulator is a continuation of work presented in [1]. Two main sources have been used in the implementation of the fire simulator. The first source is that of expert knowledge, provided by the New Zealand Fire Service [5] and by previous research into fire propagation [2].

2 Individual Building Simulation

There are two important features in the analysis of an individual burning building. One is the duration of burning. The second is the heat released by the building. Since the fire occurs in a disaster situation, we assume that internal fire prevention features such as sprinkler systems will not be functioning correctly. Given these assumptions the duration of the fire can be found by [5]:

$$t_{fire} = \frac{\left(\Delta H_c \cdot M_{fuel}\right)}{\left(A_{fuel} \cdot Q''_{fuel}\right)} \tag{1}$$

where ΔH_c is the calorific heat value, which is generally about 18MJ/kg, M_{fuel} is the mass of the fuel in the building (in kg), and $Q^n{}_{fuel}$ is the heat release values of the fuel (in MW/m^2). Some typical values for $Q^n{}_{fuel}$ are [5]:

0.5	for retail
0.25	for offices
1.0	for warehouses

A_{fuel} is the surface area of the fuel, this normally can be considered to be directly proportional to the total floor area.

For this system, ΔH_c is a global constant, with the heat release values and the mass of fuel being assigned randomly by the simulator for each building. Further discussion on this is given in section 5.

3 Fire Propagation

Fire propagation can be broken into two parts: the physical and the geometric. Physical fire propagation determines the transfer of heat between two surfaces (i.e. walls of buildings). The geometric properties cover the determination of which surfaces affect each other, based on orientation and distance.

3.1 Physical Properties of Fire Propagation

The basic physical property that causes ignition of a building is the radiant heat of neighboring buildings. This is governed by the laws of heat transfer. For two plane radiators (or walls) the intensity (in kW/m^2) on the cooler surface is given as:

$$I_r = k_1 \Phi \varepsilon \sigma \left[\left(273 + T_e\right)^4 - \left(273 + T_r\right)^4 \right] \tag{2}$$

k_1 is the “glazing factor.” For a building without fire-proof glazing, it is assumed that the windows will fall out and fire will project out through the openings. If the building uses fire proof glazing, the windows will stay in place approximately halving the fire output. It is common to ignore flame projection [2].

ε is the emissivity, a dimensionless constant, 1.0 is a conservative value.

σ is Stefan’s constant, $56.7 \times 10^{-12} (kW / M^2 K^4)$

T_e and T_r are the temperatures of the emitting and receiving surfaces in Celsius.

Φ is the configuration factor, defined as

$$\Phi=\frac{1}{90}\left(\frac{x}{\sqrt{1+x^2}}\tan^{-1}\left(\frac{y}{\sqrt{1+x^2}}\right)+\frac{y}{\sqrt{1+y^2}}\tan^{-1}\left(\frac{x}{\sqrt{1+y^2}}\right)\right)$$

where $x = H_r / 2R$, $y = W_r / 2R$ and $\tan^{-1}$ is given in degrees, W_r, H_r are the width and height respectively of the radiating surface, and R is the distance between the two radiators. However an approximation that is often used is,

$$\Phi \approx \frac{W_r H_r}{\pi R^2} \tag{3}$$

The temperature inside a building can be given by the Standard Fire Curve [3]:

$$T_e = 345\log(8t+1)+T_o \tag{4}$$

Where t is the duration of the fire, in minutes, and T_0 is the initial temperature of the surface.

The adjacent building only ignites if the incident intensity is over a material dependant value, for cellulose based material this is12.5 kW/m^2, for plastics 10 kW/m^2. In the simulator this value is based on the Type attribute of the building.

Typically the calculation of the heat intensity between two buildings relies on the calculation of vents, that is, holes in the firecell from which heat is emanating. Since we are working entirely from polygonal building data we do not have access to this information. Conservatively we may assume that there are no vents and so no projective flames [2].

3.2 Geometric Properties of Fire Propagation

To calculate the intensity of heat being applied to a given building, we need to consider each polygon of the emitting building interacting with each polygon of the receiving building.

As we have the plane equation for heat intensity, we can use the polygonal representation to create the “visually apparent area” of a building from another. We do this by checking the radial separation of each vertex in a building from the centre of the viewing building. Note that in projecting these points from Cartesian space to Polar space, there is a special case if a polygon lies partially above and partially

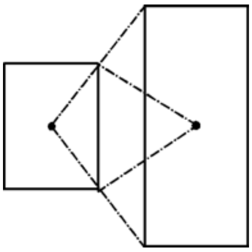

Figure1: Flat-facing buildings

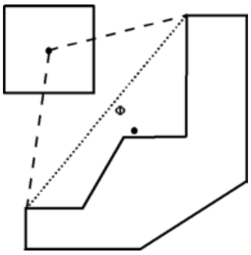

Figure2: Bad approximation for Φ

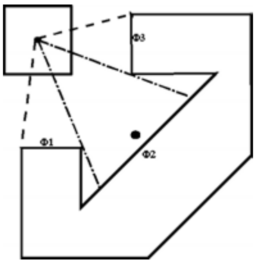

Figure3: Complicated overlapping

below the horizontal line defined by the point of view. This can be fixed by maximizing angles above the line and minimizing angles below the line.

An analysis of this method shows that if the two buildings are adjacent and facing at an angle of 0, the visually apparent area will be exact to the radiating area (Fig. 1). However, if the buildings are not flat facing or are an odd shape, the approximation begins to lose its accuracy (Fig. 2). Ultimately there is no way of calculating the worst possible case of the approximation, given that buildings may cover any real polygon.

As radiant heat is a wave phenomenon, it obeys the rule of superposition, allowing us to linearly add the intensities of multiple radiant faces. We can then add the intensities of all the faces.

The Total incident heat is then:

$$\begin{aligned} I_r &= \sum_i \Phi_i k_1 \varepsilon\sigma \left[(273+T_e)^4 - (273+T_r)^4\right] \\ &= k_1 \varepsilon\sigma \left[(273+T_e)^4 - (273+T_r)^4\right] \cdot \sum_i \Phi_i \end{aligned} \tag{5}$$

thus we need to find the configuration factor of each facing line segment between the maximally separated points.

To do this, we look at our polygon, a set of indexed points stored in a cycle. We find which of these is closer to the Point of Vision, and then traverse the polygon along that route, till we reach the last point. For each face, we calculate the configuration factor, and add it to a running total. The calculated configuration factor can then be used to find the incident intensity.

But this is still insufficient for complicated overlapping (Fig. 3). So the method of binary space partition trees (BSP-tree) [4] has been used. This provides us with a much more accurate method for calculating the configuration factor of any building, and naturally agrees with the approximation for the configuration factor given in (4).

4 Fire Extinguishing

The effect of the application of water to a building follows naturally from (5). Here we see that the intensity of heat transfer is dependent on the temperature of the surfaces of each building. When water hits the building, we assume that it evaporates almost instantaneously, draining energy from the fire. Thus we make an adjustment to the value of the temperature T, based on the energy released in vapourising the water:

$$\Delta T = -\frac{\Delta T_{water} \times M_{water} \times H_{water}}{M_{fuel} \times H_{fuel}} \tag{6}$$

This equation is based on the water initially being at room temperature (20 degrees Celsius), and instantly evaporating at 100 degrees Celsius, so $\Delta T_{water} = 80$.

M_{water} is the mass of the water, which is equal to the volume of water being applied.

H_{water} and H_{fuel} are the specific heat values for the water and the fuel. Both these are global constants.

Once the temperature is below the building's ignition temperature, the building is considered doused. This building will no longer emit heat and will not reignite. Also note that while applying water reduces the temperature of the building, it does not affect the burn time unless it is completely extinguished.

5 Randomness

So far, all of the equations for calculating fire spread and control have been deterministic. While they are a reasonable attempt at approximating the physics of fire, they do not capture any of the numerous building-internal factors that give its chaotic form. For this reason, and for competition purposes, it is necessary to introduce a random element into the simulator.

In the current simulator, randomness is implemented by using a probability of ignition at a timestep from each neighbouring burning building. In practice, this requires a new random number to be calculated for each burning building at each timestep. As the random number generator will have produced differing amounts of numbers given different actions by agents, even if the initial random seed is the same the fire simulator will produce different fire spread for each team.

Our simulator creates a 'fair' form of random fire spread. At startup, before agent's actions can affect the random number generator, every building is assigned random values for Q^{n}_{fuel}, the heat release value as defined in (1), and M_{fuel} the mass of fuel. This models the unknown information on the contents of buildings in a city, and will affect the burn time and spread of fire between buildings. In addition, as all random calculation is performed at startup, every team in a competition will be dealing with the same random disaster situation if the same random seed is used.

6 Representation and Implementation

The implementation of the model proposed at first appears quite complex. However we have found that a large proportion of the work can be performed at startup, and further results can be cached as the simulator runs.

6.1 Building Parameters

At start up the following parameters are computed / intialised for each building:

- A random value for the heat release value, Q''_{fuel}
- A random value for the mass of the fuel, M_{fuel}
- A list of neighbouring buildings based on distance between faces of buildings
- The number of timesteps this building will burn for, t_{fire} using (1)
- An array of length t_{fire} containing the change in temperature of the building for each timestep it burns using (4)
- A value for current temperature, T_e
- A count of how many timesteps the building has been burning

The neighbour list is used to limit the number of computations of configuration factors between buildings. In addition to these parameters, for each building in the neighbour list a value for the sum of configuration factors between the walls of the buildings used in (5) can be held.

6.2 Calculating Fire Spread

The most costly computation in fire spread is determining the configuration factors between buildings. As mentioned above, we can store these values with the neighbour list, however these values are initially null. When a building first becomes adjacent to a burning building, the values for the sum of their configuration factors is computed, and is then cached. This allows the simulator to start up in reasonable time, but still limits the in-simulation computations.

Other than computing configuration factors when a new building catches fire, there are also a number of actions to be performed for each burning building. First, each building updates its temperature according to the values in its array of. Next, for each unburnt neighbouring building, update the neighbours temperature using equation (2).

A buildings fieryness rating is related to its temperature, so using this simulator a buildings fieryness will decrease when firemen are extinguishing.

7 Conclusion

We have implemented a new fire simulator for the Robocup Rescue Simulation league. This simulator improves on the current simulator in four main areas, while keeping the cost of computation to a minimum.

The first area is in the representation of a burning building. The duration of burning buildings follows the recent guidelines of [5]. Temperature is now the most important parameter, and temperature change over time is calculated using the standards in [3].

The second area of improvement is the propagation of fire. This is based on the results of prior research [1] [2], using the thermodynamic laws of heat transfer. The algorithm used calculates the heat transferred between each pair of walls of the buildings, creating a system that realistically takes building geometry into account.

The third area of improvement is in the effect that water has on the fire. Here simple thermodynamics was once more used to determine the temperature change caused by the vapourisation of water. In addition, the simulator now allows fieryness of buildings to decrease as their temperature is lowered.

The final area of improvement is in the 'fair' use of randomness. This simulator ensures that for a given random seed, every run of the fire simulator gives the same fire spread irrespective of how the random number generator is used during simulation. This removes an element of luck from competition while preserving uncertainty in the simulation.

References

1. Berdinshaw, D., Guesgen, H.: Physical Approximations for Urban Fire Spread Simulations (2003)
2. Buchanan, A. H: Fire Engineering Design Guide. Centre for Advanced Engineering, University of Canterbury (2001).
3. International Standards Organisation. ISO 834-1: Fire-resistance Tests - Elements of Building Construction - Part 1: General Requirements. (1999)
4. Krishnaswamy, R., Alijani G., Su, S. On the Construction of Binary Space Partition Trees, In Proceedings of the 1990 ACM annual conference on Cooperation, ACM - Association for Computing Machinery (1990).
5. New Zealand Fire Service. Fire Fighting Code of Practice (2003).