BRANDFORSK
2018:5

Visualization of fires in virtual reality

Jonathan Wahlqvist and Patrick van Hees

Brandforsk

Keywords
Fire, smoke, light scattering, virtual reality, VR

This report constitutes a final working manuscript for the headlined project.

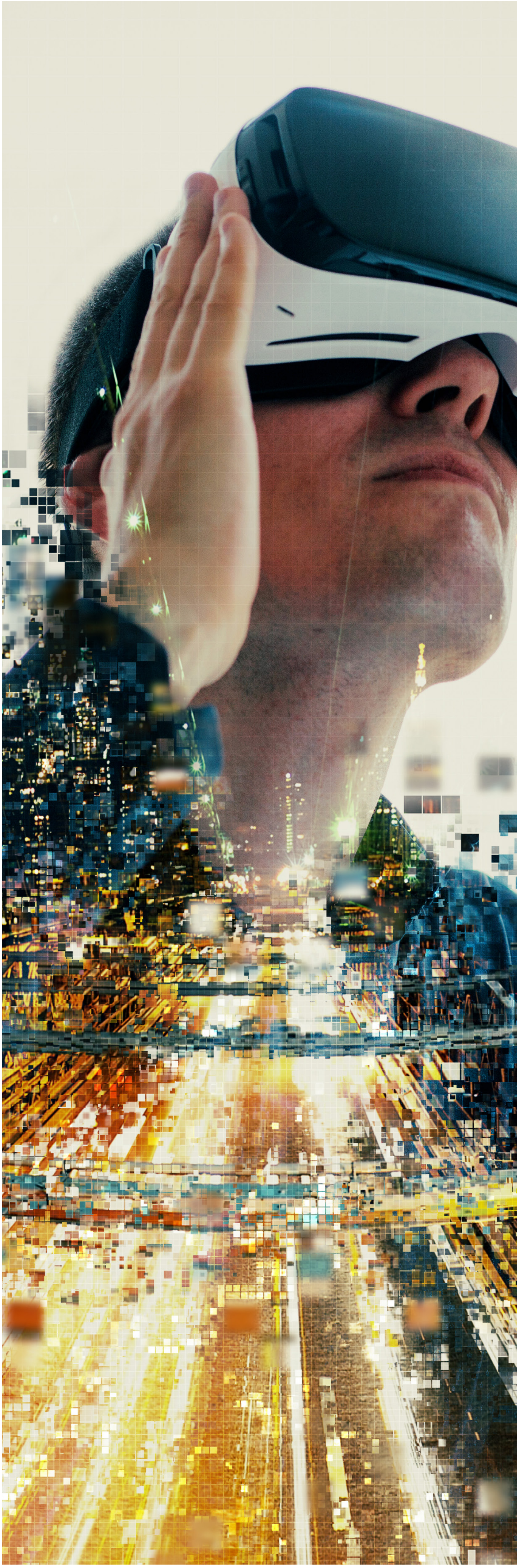
The official project report, to which reference should be made, can be found on the LTH´s website.

"Visualization of fires in VR"

www.lth.se

ISRN: LUTVDG/TVBB-3222-SE

BRANDFORSK 2018:5



Visualization of fires in VR

Jonathan Wahlqvist
Patrick van Hees

Report 3222
ISRN: LUTVDG/TVBB—3222--SE

Number of pages: 30
Illustrations: Jonathan Wahlqvist

Keywords
Fire, smoke, light scattering, virtual reality, VR

Sökord
Brand, brandgaser, ljusspridning, virtuell verklighet, VR

Abstract
Virtual reality can be a useful tool within fire safety science based on the fact that people can be exposed to hazardous environments without being in any actual danger. But to do so tools are needed that can accurately simulate such hazardous environments, specifically smoke laden rooms and burning objects. This report presents a novel method for visualizing smoke in virtual reality, as well as implementing a simplified two-zone model in a game engine which can react to user input. The combination of the two creates a tool which can be useful in many different areas, the most obvious one being different kinds of training in smoke laden environments. The presented results produce similar results to traditional visibility calculations while offering more complex visualization options, such as light interactions with the environment, which is not possible using traditional calculations.

Avdelningen för Brandteknik
Lunds tekniska högskola
Lunds universitet
Box 118
221 00 Lund
brand@brand.lth.se
http://www.brand.lth.se
Telefon: 046 - 222 73 60
Telefax: 046 - 222 46 12

Division of Fire Safety Engineering
Lund University
P.O. Box 118
SE-221 00 Lund, Sweden
brand@brand.lth.se
http://www.brand.lth.se/english
Telephone: +46 46 222 73 60
Fax: +46 46 222 46 12

Table of Contents

1. Acknowledgements

This work was made possible thanks to financial support by Brandforsk (The Swedish Fire Research Board).

2. Introduction

The use of virtual reality (VR, virtual reality) as a research tool in fire technology and evacuation has recently proved to be very useful (Kinateder et al., 2014a, 2014b; Cha et al., 2012, Xu et al., 2014). However, there is still a need for need for physically based visualization of fire and smoke. This especially includes interaction between light sources and smoke in a 3D environment which can be applied to an interactive virtual reality for use in several different areas. Examples of applications include evacuation drills, training of emergency personnel and interactive public information on how a fire develops and reacts depending on how a person act.

The interaction between smoke and light sources is a critical part of a fire and a factor that, in principle, is always simplified due to its complexity and requirements on computer hardware. However, developments have gone a long way in the last decade in this area, and it is now partially possible to accomplish this in real time on modern graphics cards. NIST has begun to implement volume rendering (same technology used in magnetizing x-ray visualization) in Smokeview (Forney, 2013) which is one part of the visualization. However, Smokeview does not implement light sources in the normal sense (only artificial lighting based on camera position). The visibility of an object is instead calculated with a simplified equation (Mulholland, 2002) where different constants are used based on the nature of the emergency exits; light-reflective or light-emitting.

This project has focused on simulating the interaction between emitting light sources and particles in the air (mainly soot) which, in turn, determines the visibility (see Figure 1 for examples of light/particle interactions). Similar attempts to visualize actual visibility has previously been made by Rubini et al. (2007). However, in this application the fire simulation was done separately first and then the view was rendered (calculated) from a given point in the space separately in so called offline mode, i.e. not in real time, thus allowing no interactive environment. This project aims to do all calculations in real-time so that a user can experience and influence the fire driven environment.

In addition to being a visualization tool, in order to offer interactive environments with fires present, a tool for calculating the physical aspects of the fire process, such as the spread of hot and sooty gases, is required. This can be done in two ways; The first option is to write a simpler code directly integrated in the 3D engine that drives the visuals. The advantage here is that there is a direct link between both parts and the exchange of data between visualization and calculation is trivial. The alternative is to write an interface between an existing simulation software and the game engine, thus utilizing the large resources put into in the development of those tools. Due to the requirement that both the visuals and the fire simulation has to run in real time (or faster) only one- or two-zone models are relevant at with presently available computer resources. However, it is not excluded to implement a coupling to more advanced models such as Computational Fluid Dynamics software (CFD), such as Fire Dynamics Simulator (FDS) (McGrattan et al., 2017) in the future. Such a coupling will not offer user/fire interactivity unless modified and simplified. This project has opted to implement a basic two-zone model within the game engine as a first step towards more advanced visualization of fires in virtual reality.

Figure 1 Example of varying degrees of smoke around light sources and the resulting scattering of light. Picture taken by Karl Fridolf.

3. Model theory

3.1. Light extinction and visibility

According to McGrattan et al. (2017) the most useful quantity for assessing visibility in a space is the light extinction coefficient, K (Mulholland, 2002). The intensity of light (wavelength independent, or monochromatic) passing a distance L through a participating media (e.g. cloud, mist, smoke) is attenuated according to

$$I / I_0 = e^{-KL}$$

Where I_0 is initial light intensity, I is the intensity after the distance L and K is the light extinction coefficient. K is a product of a mass specific extinction coefficient, K_m, which is fuel dependent, and the density of smoke particulate, $\rho \cdot Y_S$:

$$K = K_m \cdot \rho \cdot Y_S$$

Where ρ is the density if the hot gasses and Y_S is the fraction of soot in the hot gasses. The default value of K_m used by FDS is 8700 m^2/kg (McGrattan et al., 2017). This is based on the fact that Mulholland & Croarkin (2000) suggested that the value for K_m most flaming fuels is 8700 m2/kg ± 1100 m2/kg at a wavelength of 633 nm.

The light intensity along a path in participating media (such as smoke) is changed due to three main processes (shown in Figure 2):

- Absorption – the reduction in intensity due do the conversion of light to another form of energy, such as heat. (Pharr & Humphreys, 2010)
- Emission – intensity that is added along the path from luminous particles (Pharr & Humphreys, 2010), such as fire embers or a flame.
- Scattering – how light heading in one direction is scattered, or bounced, in other directions due to collisions with particles (Pharr & Humphreys, 2010). This is often divided into two sub-parts; in-scattering, light that is added along the past, and out-scattering, light that is lost along the path.

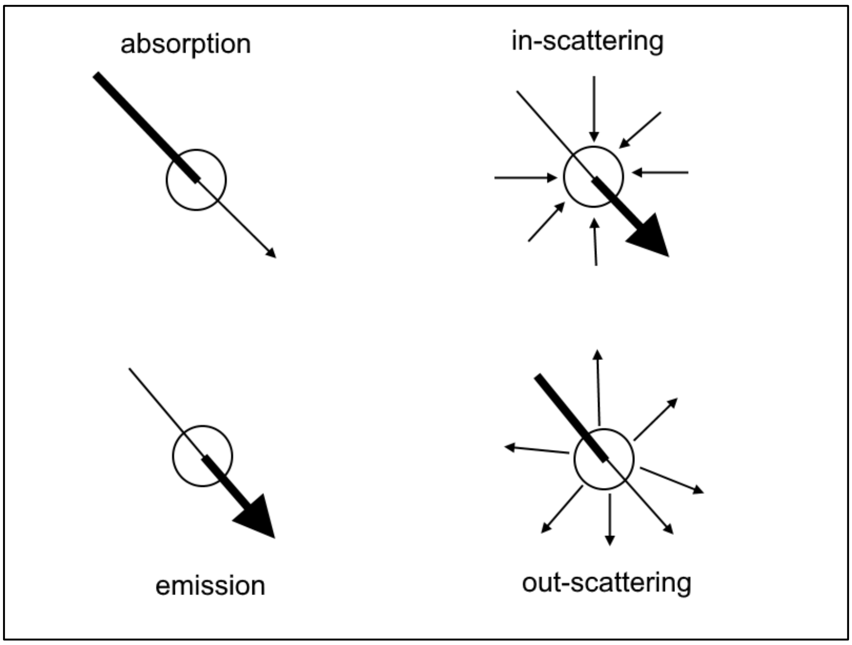


Figure 2 Main processes involved in changing the intensity of light along a path.

Extinction is the total reduction along the path, a combination of absorption and out-scattering.

According to Mulholland (2002) estimates of visibility through smoke can be made by using the equation

$$S = C / K$$

where C is a non-dimensional constant based on the nature of the observed object through the participating media (smoke). The default values used in FDS (McGrattan, 2017) is C = 8 for a light-emitting sign and C = 3 for a light-reflecting sign based on Mulholland (2002). However, Jin (1997) recommends a range C=2-4 for a light-reflecting sign and C=5-10 for a light-emitting sign, indicating that these values might be hard to determine.

Limitations

Only so called single scattering is considered due to performance reasons. This means that light is only scattered once while in reality scattered light bounces many times. But so-called multiple scattering is generally considered too complex to resolve in real-time rendering. However, there are implementations to "fake" to visual effect by adding depth based blur (Narasimham & Nayar, 2003; Unity, 2018a). These have however not been included in this work as they are not physically "correct" but more a visual effect. The limitation of using single scattering is however lessened as the smoke gets more absorbent, which might be argued is the case for "black" smoke.

3.2. Calculation of fire and smoke spread

3.2.1. Two-zone fire modelling

A simplified two-zone model was implemented in the project and was based on the structure and equations used in CFAST (Peacock et al., 2017). CFAST is described by the authors of the software as:

CFAST is a two-zone fire model that predicts the thermal environment caused by a fire within a compartmented structure. Each compartment is divided into an upper and lower gas layer (zone in the term zone fire model refers to the layers being modeled). The fire drives combustion products from the lower to the upper layer via the plume. The temperature within each layer is uniform, and its evolution in time is described by a set of ordinary differential equations derived from the fundamental laws of mass and energy conservation. The transport of smoke and heat from zone to zone is dictated by empirical correlations. Because the governing equations are relatively simple, CFAST simulations typically require a few tens of seconds of CPU time on typical personal computers. (Peacock et al., 2017)

To learn more about the underlying equations and algorithms the curious individual is recommended to read the CFAST technical reference guide, in particular chapter 2 and 3 (Peacock et al., 2017). As the time-step used in the simplified version is very small due to being updated every rendered frame (between 11-17 milliseconds depending on the type of VR-headset being used), and due to the fact that high accuracy is not the main goal of the project, the equations are directly solved explicitly instead of being iterated to a specified tolerance like in CFAST (Peacock et al., 2017).

Limitations

CFAST contains a lot of commonly used functionality such as mechanical ventilation and sprinkler activation which was not of importance within the project where mainly smoke spread/filling was of interest. Hence very limited features were implemented from CFAST. Only smoke transport within one room (connected to ambient through openings) is considered, thus not including other functionality commonly present in two-zone models such as:

- Radiative heat transfer
- Heat transfer to ceiling/walls
- Species transport other than soot

However, more features can be added in future iterations of the software, radiative heat transfer to the player is of high interest in future iterations.

4. Model implementation

The model was implemented in Unity through both traditional code (Unity scripts) and shaders (code run on the graphics processing unit (GPU)).

4.1. Unity

Unity is a multipurpose game engine that supports 2D and 3D graphics, simple ways of adding and changing functionality of game objects and a relatively simple coding API using C# (Unity, 2018g). The engine targets multiple platforms such as Windows, macOS, Linux, Android, iOS (Unity, 2018e).

Unity has a flexible rendering pipeline which supports the use of custom vertex, fragment (pixel), compute and surface (unique to the Unity pipeline) shaders using Cg, a modified version of Microsoft's High-Level Shading Language (HLSL) (Unity, 2018f)

Shaders are programs that run on the graphics processing unit (GPU) of the computer. All shaders were written in Cg using available Unity macros (Unity, 2018b) and variables (Unity, 2018c) where applicable to increase performance and simplify the code.

4.1. Visualization of fire and smoke

The model consists of two major parts, calculation of fire and smoke spread and visualization of fire and smoke. The visualization is further split into three parts; volumetric scattering and light extinction in smoke, absorption of light from light sources to surfaces (walls, ceiling, floor etc.) and visualization of flames.

4.1.1. Volumetric scattering and light extinction

The basic algorithm used is based on the fundamental theory described in section 3.1. The basic features of the algorithm are:

- Absorption – included and integrated along each step length along the view ray (based on total extinction, i.e. out-scattering is included).
- Emission – not included due to complexity of tracking light from the flame/embers which is a volumetric object. This would involve discretizing the flame into several volumes that then would have to be included along all other light sources in all calculations. The effect of lighting from the flames can be approximated by adding a point light source to the burning object, which is then treated similar to all other light sources.
- Scattering – single in-scattering is included from all point light sources. Out-scattering is included in the extinction coefficient being used provided by the two-zone model.
- All light sources are treated as infinitely small point sources as this is mathematically much simpler and computationally much faster (volumetric lights pose similar problems as light from flames)
- Distances are discretized is smaller steps, so called ray-marching steps, over which each part of the algorithm is solved and the accumulated over the full distance between observer and object, see Figure 3 for an overview.

Extinction (absorption and out-scattering) along the view ray is calculated by calculating I/I_0 for each ray-marching step and accumulate it over the full distance according to:

$$I / I_0 = \prod_0^L e^{-K \Delta L}$$

Where L is the total distance travelled along the view ray (distance between observer and object), K is the extinction coefficient at a specific position in space and ΔL is the step length.

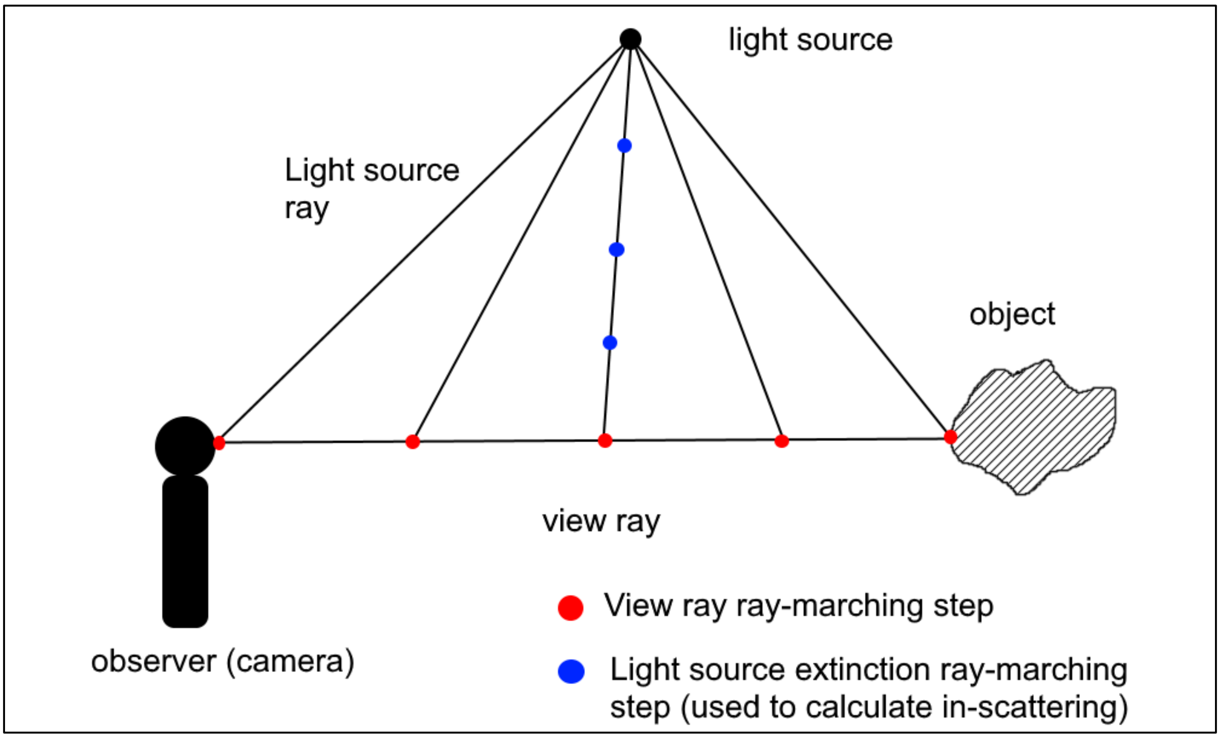


Figure 3 Visualization of the algorithm used to solve the scattering and extinction of light along a ray.

In-scattering is calculated at each view ray ray-march step. There are several things to consider however:

- Light attenuation due to distance from the light source. The further away from a light source the more light is spread out, similar to radiation (which is light at another wavelength). For point lights the attenuation is $1/d^2$, where d is the distance from the point in space of interest to the light source. The attenuation factor is then multiplied with the original point source light color and intensity to get the actual light received at a point in space (without smoke present).

- Light extinction along the path from the light source to the point of interest. This is done by doing a similar ray-march stepping from the light source to the point of interest (point along the view ray that is currently being treated) in the same manner as extinction is calculated along the view ray. The step length is set by the user and is generally a compromise between accuracy and speed. No in-scattering from other light sources is considered along this path.

- In-scattering phase function. Phase functions describes the angular distribution of light reflected from a particle when illuminated from a specific direction (Pharr & Humphreys, 2010). In other words, the phase function describes in what direction the light bounces when hitting a smoke particle in relation to the original light source. This in turn determines how much of the scattered light is actually added along the view ray. The simplest (and computationally cheapest) scattering function is the isotropic scattering function which scatters the light equally in all directions. An anisotropic scattering model has also been implemented and is selectable by the user. This model enables so-called forward or backward scattering where light scatters relatively more in the same or opposite direction of the ray from the light source. This might fit better with experimental measurements of smoke produced from burning objects, though this information may not be easily obtainable in many cases. Several such functions exist (Pharr & Humphreys, 2010), but a relatively computationally efficient model called Schlick (Blasy et al., 1993) has been implemented.

- Isotropic scattering: $1/4\pi$ (Pharr & Humphreys, 2010)
- Anisotropic scattering (Schlick): $\frac{1}{4\pi}\frac{1-k^2}{(1-k\cdot\cos(\theta))^2}$, where $k = 1.55g - 0.55g^3$.

  g is a constant used in the Henyey-Greenstein anisotropic scattering model (Henyey & Greenstein, 1941) which is adjusted to fit experimental scattering data (Pharr & Humphreys, 2010).

With this in mind, the in-scattering contribution S along the view ray at a view ray step is then:

$$S = L \cdot LA \cdot LT \cdot \mu_s \cdot p$$

Where L is the light intensity and color, LA is the light attenuation factor, LT is the light transmittance (remainder of light not being extinct along the path), μ_s is the scattering coefficient and p is the phase function. However, since light is actually being extinct along the step length the scattering contribution will get over-estimated and must therefore be corrected by solving the integral along the step length (Hillaire, 2015). To get the corrected in-scattering contribution, S_c, the following equation (Hillaire, 2015) is solved:

$$S_c = \frac{S - S \cdot e^{-K\Delta L}}{K}$$

Where K is the extinction coefficient along the step distance ΔL.

This full procedure has to be done for each light present in the scene (and for each view ray step as mentioned earlier).

Practically, the volumetric scattering of light in smoke particles is done using so called ray-marching. This is implemented in a so called post-image effect, which means that the whole scene from the view of the camera is drawn to a buffer on the graphics card containing color and depth information for each pixel. This image/buffer information is then used in the following way for each pixel:

- If desired, down-sample the image to a lower resolution in order to do the actual ray-marching on fewer pixels. This might however introduce artefacts close to sharp edges (Figure 19).
- For each pixel a ray (line) is cast along the current view vector with a total length corresponding to the depth buffer (distance to closest opaque object, information taken from the depth-buffer/z-buffer). The ray/line is then traversed by taking small steps (user definable, typically 5-20 cm) along the path and in the process accumulate absorption, out-scattering (combined as extinction) and in-scattering from light sources (Figure 3). In this step it is possible to add two-dimensional noise (hash based for computational efficiency) on top of the layer height to add a more “organic” look to the layer interface (Figure 6).
- This information is then saved to another buffer/image on the graphics card containing the scattered light information for each pixel in the RGB-channel (channels containing color information separated into red, green and blue) and the extinction (I/I_0) in the alpha-channel (channel normally containing transparency information) along the ray (Figure 4).
- If needed, the buffer from the last step is resized to match the native output of the display being used (the same size as the initial image). An optional Gaussian blur (Figure 20)can be

added in this step to reduce artefacts when scaling image size. This might however introduce artefacts close to sharp edges while not doing so might introduce a wave pattern in the smoke based on the chosen step length.

- The RGB-channel of the initial image is then multiplied by the calculated extinction saved in the second buffers alpha-channel, thereby applying extinction along the ray.
- Lastly, the RGB-channel from the secondary buffer is added to the is then added to the initial image with extinction applied, effectively adding the scattering along the ray.

Figure 4 gives a visual representation of each step described, Figure 5 shows a similar output but using a colored light (any light color is possible) which affects the RGB-channel of the secondary buffer. Figure 6 demonstrates the effect of adding a random hash based noise on top of the two-zone layer interface to enable more visually pleasing smoke layer.

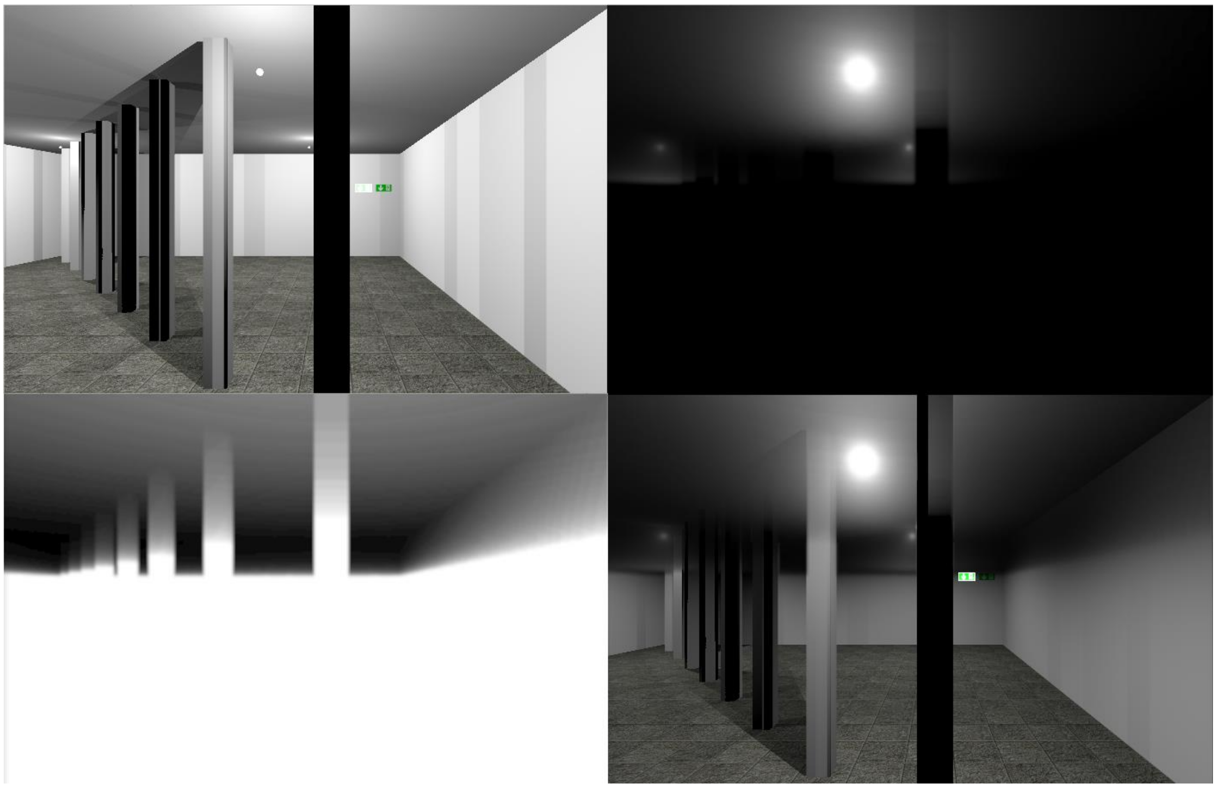


Figure 4 Top left: initial image. Top right: RGB-channel of secondary image buffer containing scattering information. Bottom left: alpha-channel of secondary image buffer containing extinction for each pixel. Bottom right: the composited image combining the information from all buffers.

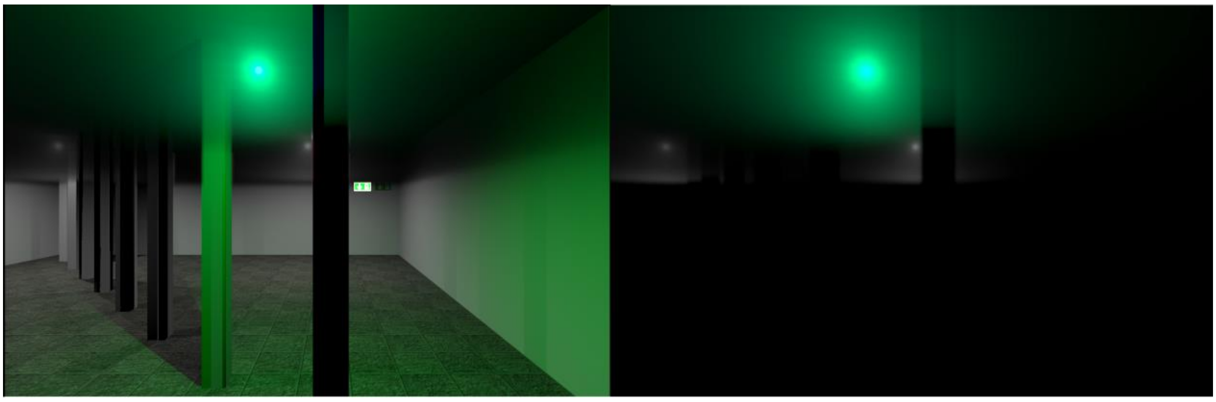


Figure 5 Left: composed final image using a colored light source. Right: RGB-channel of secondary image buffer containing scattering information.

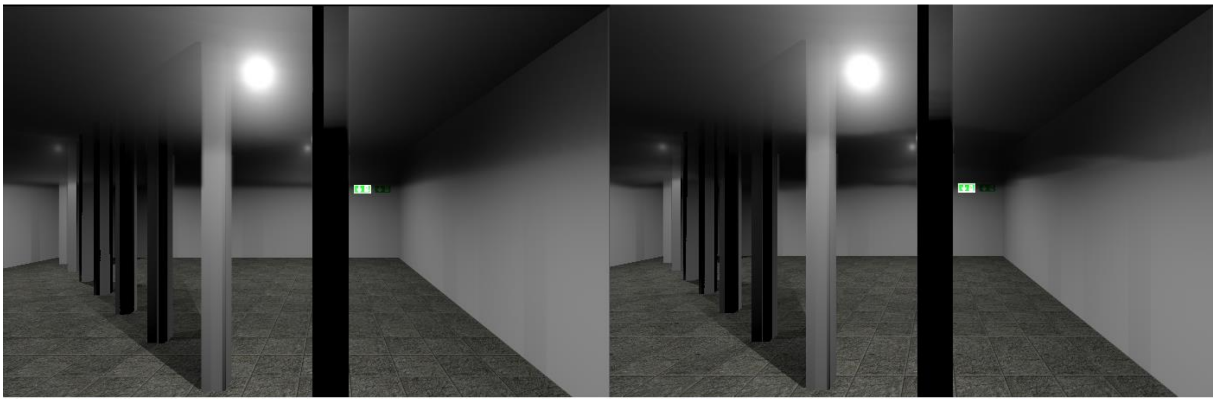


Figure 6 Left: normal “hard” two-zone layering. Right: added hash noise on top of the two-zone layer height to create a more visually pleasing layer interface.

4.1.2. Light absorption between light sources and surfaces

Light that is emitted from light source is partially or entirely extinct along the path to a surface. This means that the actual amount of light that hits a surface has to be re-calculated when introducing smoke compared to the same scenario without smoke present. Figure 8 visualizes the consequences of not including extinction between light sources and surfaces, resulting in surfaces that are too bright. Figure 7 visualizes the problem in relation to the light scattering and extinction algorithm.

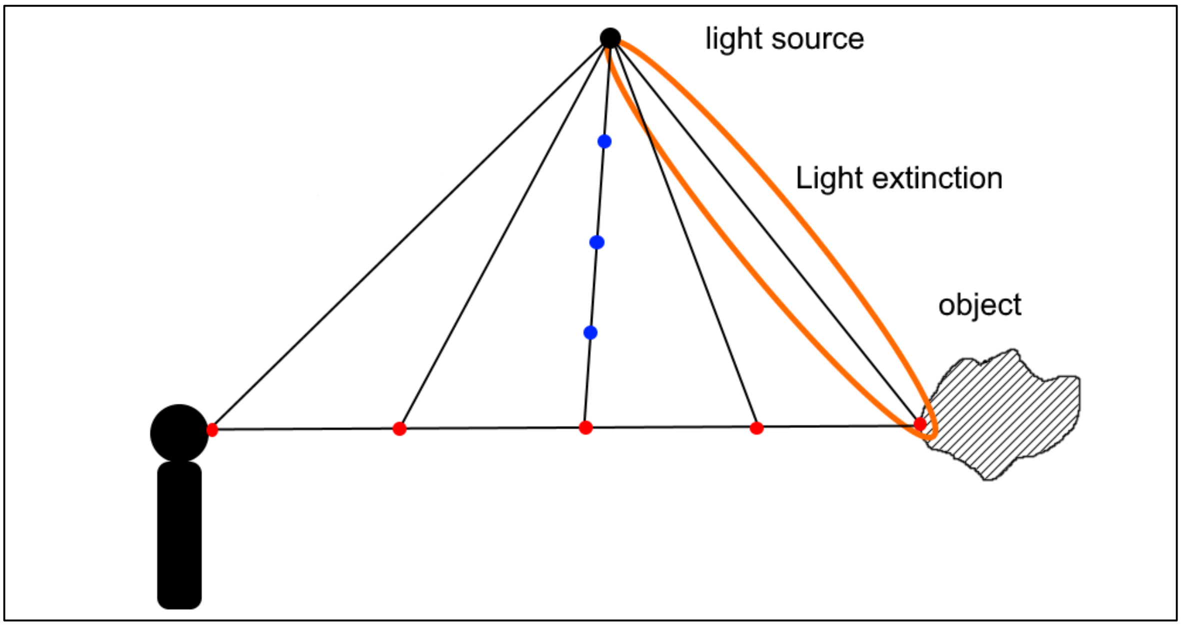


Figure 7 Visualization of the problem.

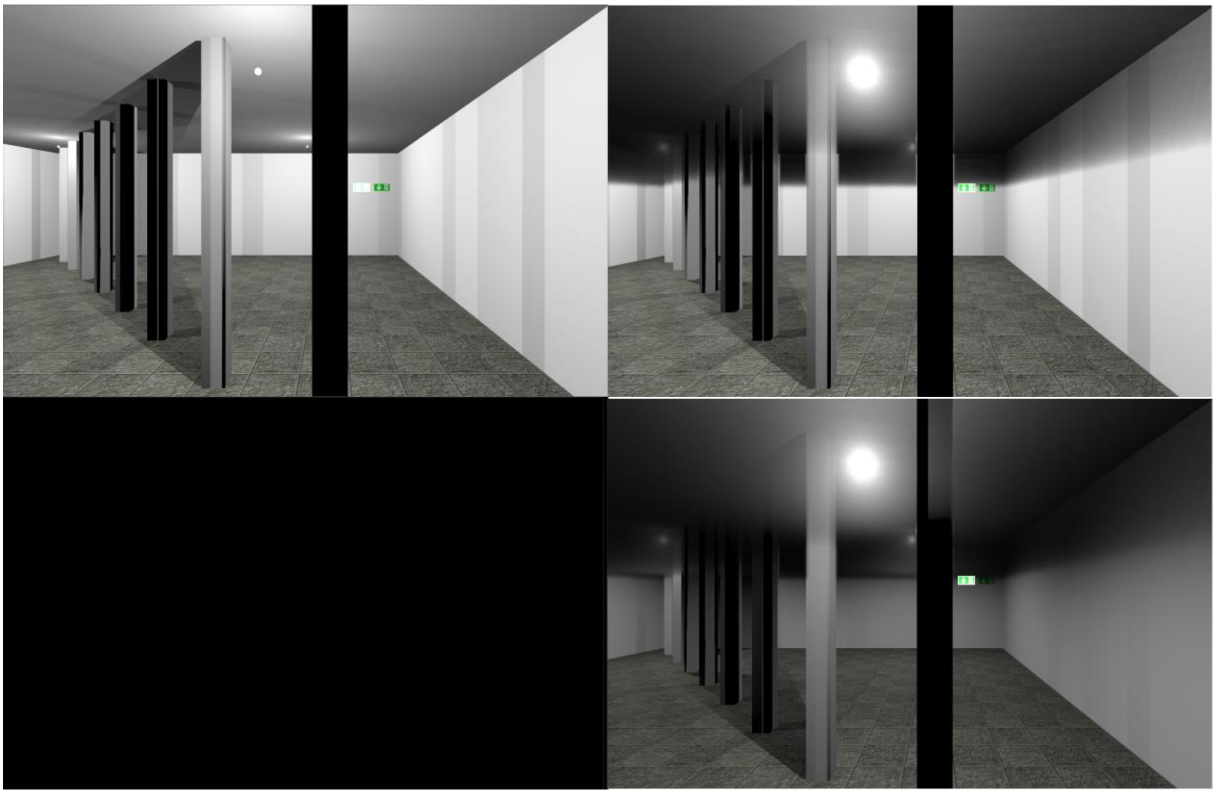


Figure 8 Top left: normal scene without smoke. Top right: scene with smoke added, including extinction and in-scattering along the view ray but no extinction of light between light sources and surfaces. Bottom right: scene with smoke, including extinction and in-scattering along the view ray as well as extinct light from light sources to surfaces.

To calculate the extinction of each light to a surface two different methods can be chosen:

- Ray marching in the exact same manner as the extinction along the view ray or along the light ray (used for in-scattering calculation).
- Use a simplified method described by Lengyel (2007) which only works with two-zone models and one room due to how the algorithm works. This method directly calculates the exact distance within the smoke from a point in space to a light (mathematically the distance above or below a specified plane, in this case the plane is specified by the smoke layer height).

This second method is significantly faster but is not future proof since it has several limitations:

- No varying smoke height across a room (so not compatible with e.g. data sets from FDS in future versions of the software).
- No varying smoke density within a room (again, a limitation for data sets not calculated by a zone model).
- Can only take the smoke in the specific room into account, so light that spreads to several rooms will only affected by the smoke layer that the light is inside.

In this initial version of the software the second method has been used since the limitations are compatible with the currently implemented capabilities and features and due to the fact that it is computationally less expensive.

4.1.3. Flame visualization

The flame visualization is done using a so called particle system (collection of particles) combined with data provided from the two-zone fire modelling to change the properties of the particle system. Particles are described in the Unity manual (Unity, 2018d) as:

Particles are small, simple images or meshes that are displayed and moved in great numbers by a particle system. Each particle represents a small portion of a fluid or amorphous entity and the effect of all the particles together creates the impression of the complete entity. Using a smoke cloud as an example, each particle would have a small smoke texture resembling a tiny cloud in its own right. When many of these mini-clouds are arranged together in an area of the scene, the overall effect is of a larger, volume-filling cloud. (Unity, 2018d)

The two-zone model changes the flame height and width according to the current heat release rate using the Heskestad flame height correlation (Heskestad, 2008).

However, there is a limitation to this approach as the actual particles does not emit light and will hence not scatter any light. The light from the flames is therefore approximated using a single point-light source added to the same location as the center of the flames. A flickering effect can be added for a visually enhanced effect; the flickering frequency is however not physically based. It would be desirable to add emission of light from the flame particles, but unfortunately this is not feasible in real-time with the currently available computational power.

Figure 9 Example of output in the test scenario with the couple two-zone model, note that the point-light source adds color to the walls (orange tint).

Figure 10 Example of likeness between a real fire (right) and a reproduced fire in Unity using particle systems (left). (Rosero, 2017)

4.1. Calculation of fire and smoke spread

4.1.1. Two-zone fire modelling

Four different fire growth models were implemented in Unity:

- Constant heat release rate (HRR)
- Linear
- Alpha-t^2 (NFPA, 1985)
- User defined curve

The user has to choose growth model, max heat release rate and time to max heat release rate. The user also sets the following parameters:

- Fire diameter
- Heat of combustion
- Radiative fraction
- Soot yield

Both the Heskestad flame height model and plume model (Heskestad, 2008) are used the flame height and mass flow at the layer interface respectively. The calculated values then update the particle system for the flames to correspond to the actual flame height that the user see.

The plume model determines the amount of smoke that gets transported to the upper layer, and the soot yield in combination with the heat of combustion and heat release rate determines the amount of soot that gets transported to the upper layer (heat release rate divided by the heat of combustion determines the amount of fuel that is burnt which is then multiplied by the soot yield to get the amount of soot). Flow through doors is determined by implementing the equations detailed in chapter 4.1 in the CFAST technical reference guide (Peacock et al., 2017).

A very simplified fire suppression model is implemented by allowing the user to specify a flow from a device such as a hand extinguisher. If the flow hits the fire source, the heat release rate is reduced based on a user defined suppression capability of the extinguishing agent (kW/s).

5. Results

5.1. Reference

To be able to compare all of the results a base scenario with no smoke present is important to establish. Figure 12 shows the test scene lit by four points lights (one in each corner offset by 2.5 meters, located at a height of 2.8 meters) with intensity of 1 and max distance of 15 meters (parameters set in Unity). The room itself is 10 x 10 meters with a ceiling height of 3.0 meters. The columns are located 1 meter apart (parallel to the walls, diagonally they are about 1.41 meters apart). The right side of the back wall has to emergency exits (one light-emitting and one light-reflecting) located 2 meters above the floor and 10 meters away from the camera which is located on the right side of the back wall. Figure 11 gives an overview of the test scene.

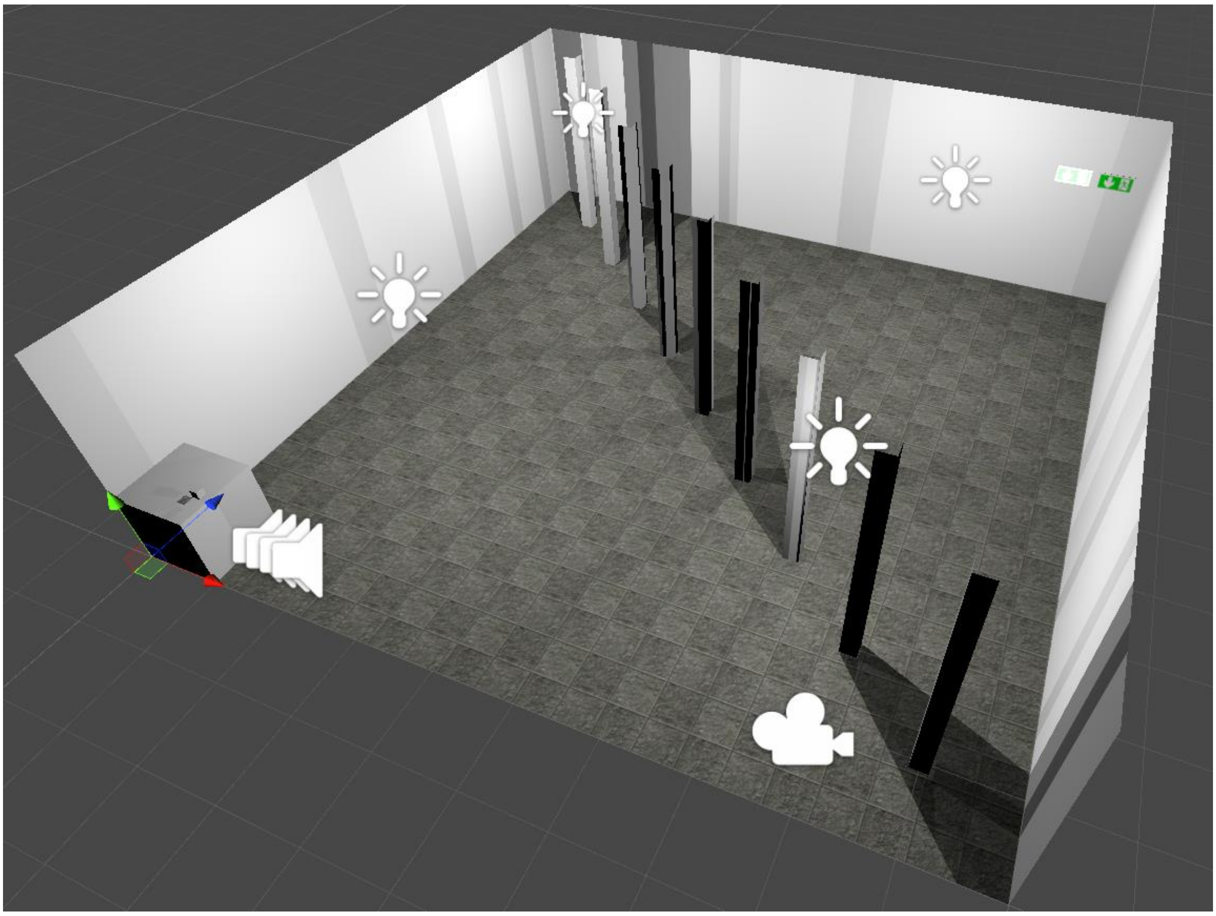


Figure 11 Overview of the test scenario.

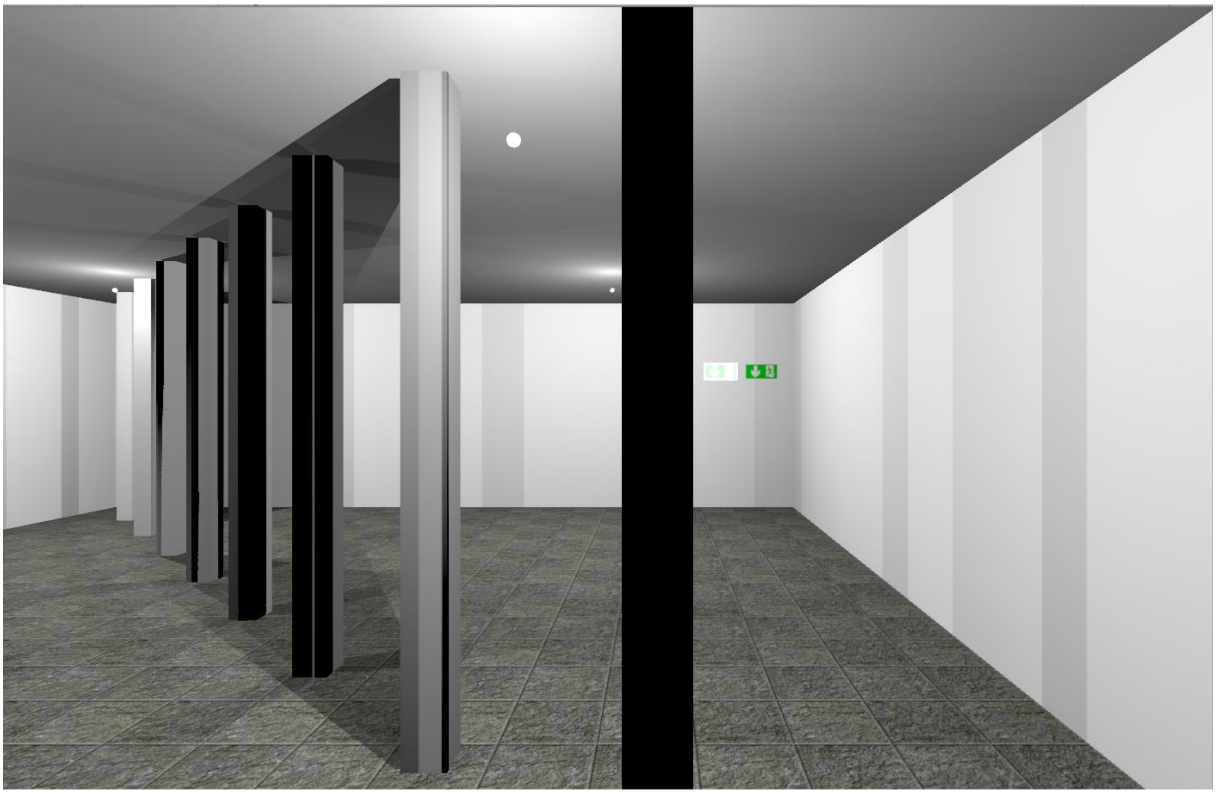


Figure 12 Base scenario with no smoke present, scene lit by four points lights with intensity 1 and max distance of 15 meters (parameters set in Unity).

5.2. Visibility

Using the traditional approach for estimating the visibility in smoke (Mulholland, 2002) the following result is expected for specific values of K (light extinction coefficient):

K	"Visibility emitting sign" (C = 8) [m]	"Visibility reflecting sign" (C = 3) [m]
0.27	30	11.25
0.32	25	9.375
0.40	20	7.5
0.53	15	5.625
0.80	10	3.75
0.89	9	3.375
1.00	8	3
1.14	7	2.625
1.33	6	2.25
1.60	5	1.875
2.00	4	1.5
2.67	3	1.125
4.00	2	0.75
8.00	1	0.375

These values are used as reference throughout the results section, but the reader is asked to keep in mind that these values are not the true value of the visibility.

To investigate the visibility using the implemented model, the whole room is filled with homogenous smoke in all cases.

With the set conditions described in section 5.1, the light-emitting sign is visible with a light extinction coefficient of 0.61 (see Figure 13), corresponding to a 13 meter visibility range using the equation presented by Mulholland (2002) while the light-reflecting sign is visible with a extinction coefficient between 0.32 and 0.27 (see Figure 14), corresponding to a visibility range of 9.375 and 11.25. The light-reflecting sign is technically visible at a value of 0.4, but the lack of contrast against the wall is very poor which would likely mean that a person not knowing that the sign is there would not see it.

However, the light-reflecting sign can be seen at much lower extinction coefficients provided that the ambient light is manipulated. Figure 15 demonstrates conditions where the light-reflecting sign is visible at a extinction coefficient of 0.61 (the same value that was the visibility limit for light-emitting sign under reference conditions) by increasing the intensity of the closest light source in the room from 1 to 5. A very similar result was achieved when moving the light source right above the light-reflecting sign since very little light is the extinct along the path to the light-reflecting sign.

Overall it can be concluded that the produced visibility is in agreement with previous simplified calculations, but also demonstrates that the subject is complex and is dependent on several factors such as proximity to light sources.

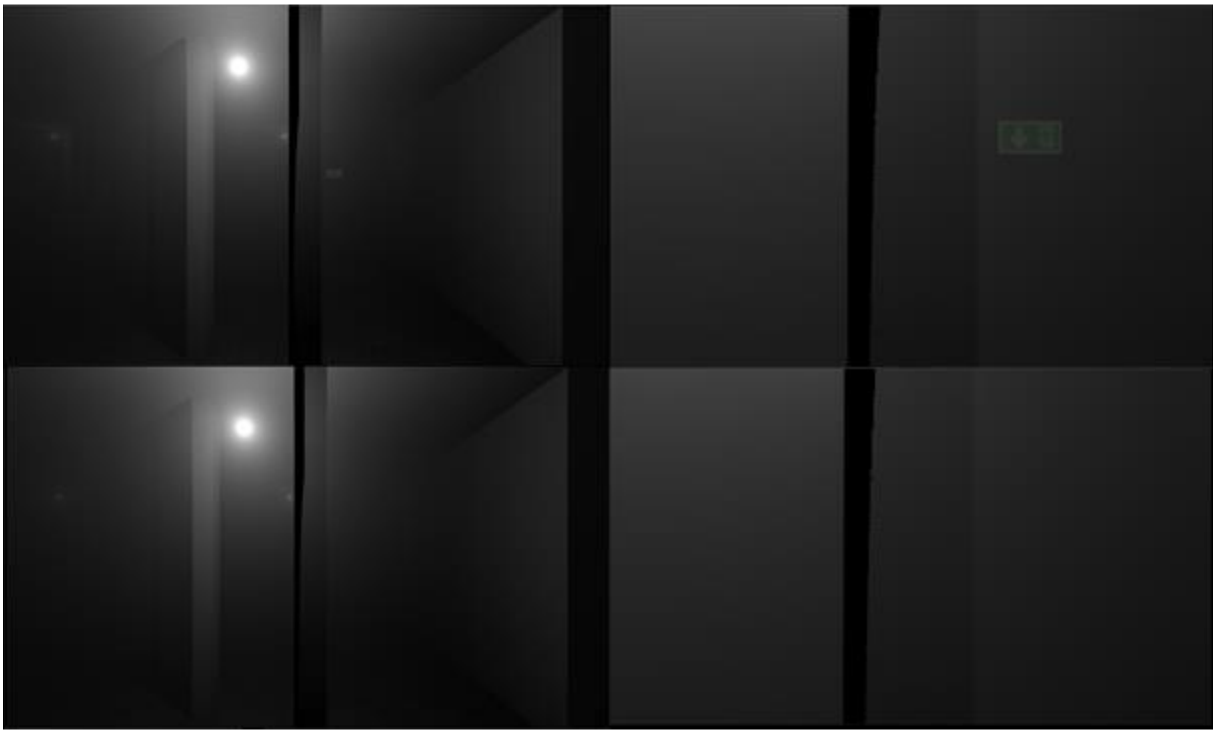


Figure 13 Barely visible light-emitting sign using K=0.61 (13 meters visibility light-emitting sign) on the top (zoomed in on bottom right side) and non-visible sign using K=0.66 (12 meters visibility light-emitting sign) on the bottom.

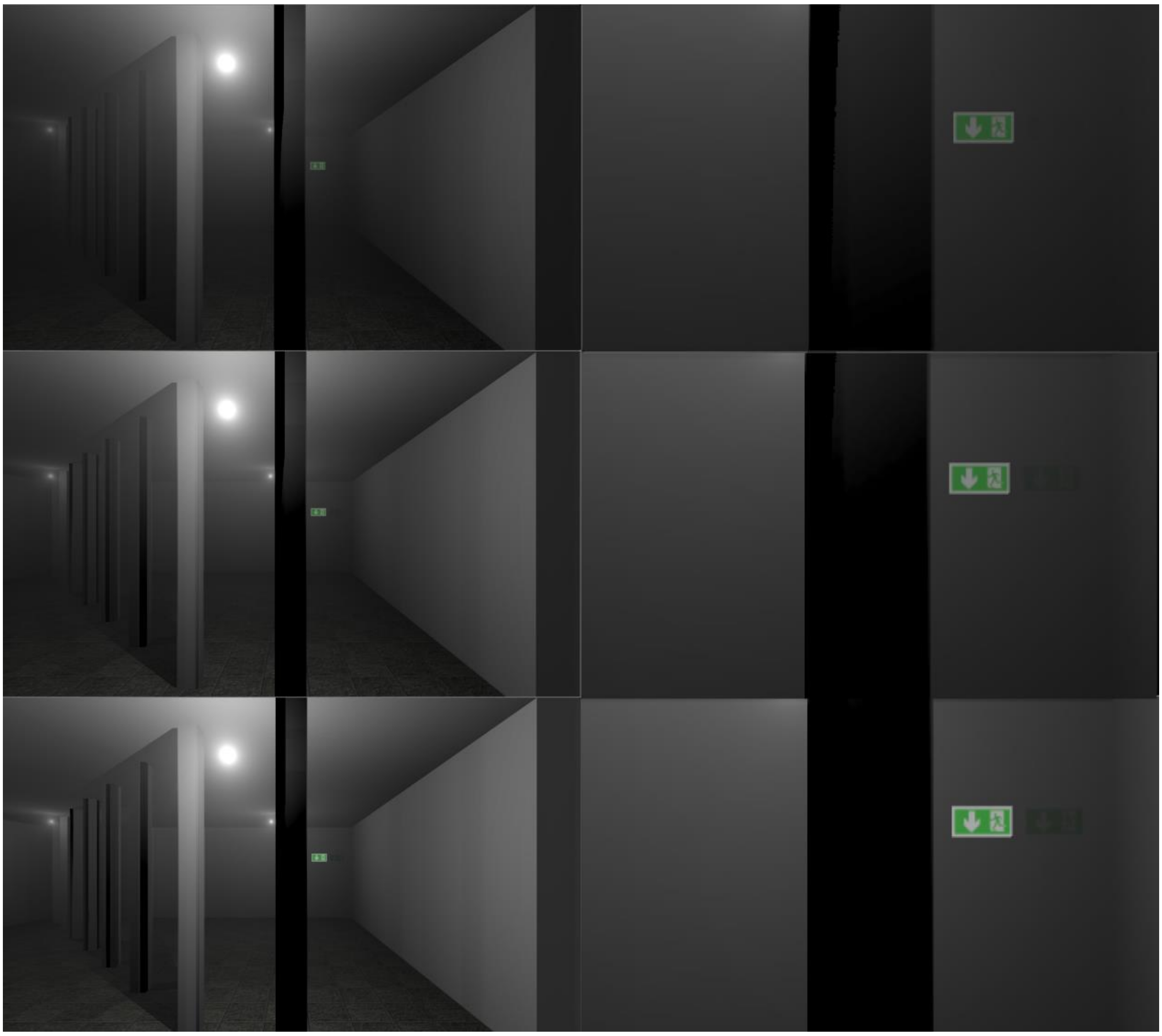


Figure 14 Barely visible light-reflecting sign at different K values (from top to bottom: 0.4, 0.32, 0.27).

Figure 15 Example of the effect of lighting on the reflecting sign which is now visible at K=0.61 where previously only the emitting sign was visible. This was achieved by increasing the light intensity for the closest point light to 5 from 1.

5.2.1. Effect of ambient light

The importance of ambient light conditions is impossible to evaluate using the simplified calculation for visibility (Mulholland, 2002) but are vital to actually determine visibility. Figure 16 shows a slightly altered scenario where the smoke layer is located 2 meters above the floor, slightly covering the top of both exit signs. All light sources are however inside the smoke layer, and as the extinction coefficient increases more and more light is extinct to the point where the room is completely dark. At this point both signs are technically "visible" since no smoke is obstructing the view, but as seen in Figure 16 the light-reflecting sign becomes completely indistinguishable while the light-emitting sign can still clearly be seen, perhaps more so than before due to the increased contrast compared to the surrounding environment.

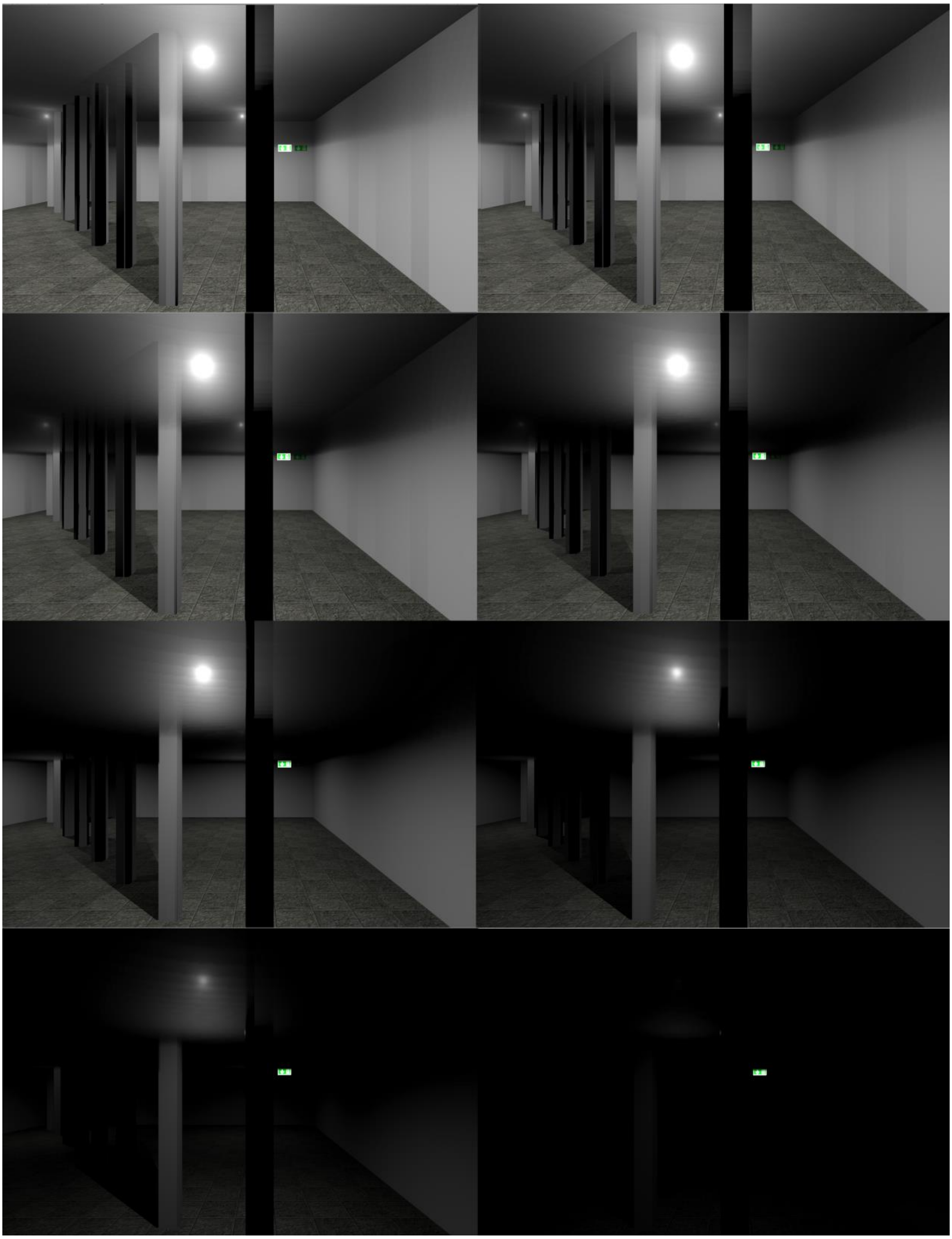


Figure 16 Change in ambient caused by increasing K (as a result of increased smoke density). Increasing K from top left to bottom right: 0.4, 0.53, 0.8, 1,14, 1.6, 2.67, 4.0, 8.0.

However, by moving the light sources to a lower level the lower part of the bottom part of the room gets lit together with the light-reflecting sign, making it visible once again. This demonstrates that the visibility of an exit sign can be manipulated by more than just the smoke itself which has to be kept in mind even though it is not included in the traditional calculations.

Figure 17 Lights located at 1.5 meters (instead of 2.8 meters) while the smoke layer (K=8) is located at 2 meters.

5.3. Sensitivity of parameters

As the user can control several parameters that influences the final image it is important to investigate the effect each parameter has. The following major parameters that can influence the final image has been identified:

- Ray-marching step length.
- Resolution scale of secondary buffer.
- Use of Gaussian blur on secondary buffer.

Figure 18 shows the effect the ray-marching step length has on the visual fidelity (using 100% resolution scale on the secondary buffer). A smaller step produces a cleaner image overall, but the difference is acceptable bearing that a doubling in step length decreases the computational cost of 50%. With the reference scenario being used it would seem though that a step length above 20 cm should not be used as the banding is becoming quite obvious. This might however be very scenario based, e.g. if the smoke is further way from the camera (player) the banding might not be obviously visible.

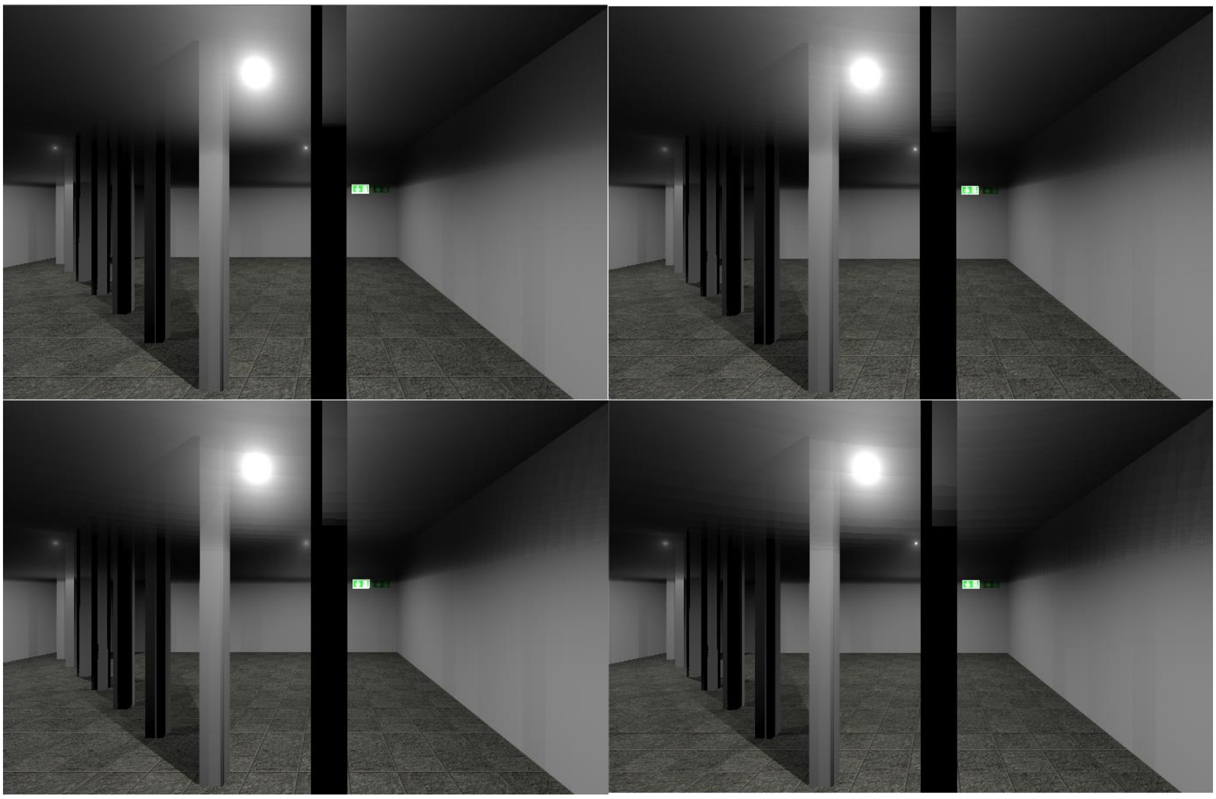


Figure 18 Visual fidelity using different ray-marching step lengths (longer steps means faster execution) using 100% resolution scaling. Notice the increase in banding artefacts as the step length increases. No blurring (Gaussian) is added to the secondary buffer. Top left: 5 cm steps. Top right: 10 cm steps. Bottom left: 15 cm steps. Bottom right: 20 cm steps.

Figure 19 shows the effect of using different resolution scales on the secondary image buffer while using 10 cm step lengths. As the resolution scale decreases the reduction in visible banding can be seen. This is due to the bi-linear interpolation that is used to up-scale the reduced image to full resolution. However, it comes at the cost of "blurriness" and artefacts around edges with large depth differences, e.g. edge of pillar close to the camera compared to the wall behind it. This is however not visually disturbing until a resolution scale of 12.5% is being used. It should also be kept in mind that a resolution scale of 50% means that number of pixels (and thereby calculations) in each direction is reduced by 50%, resulting in reduction of 75% on the total amount of pixels (1 - 0.5 * 0.5 = 0.75), effectively reducing the computational cost to 25% of full scale rendering. This also applies to the lower resolution scales, making this the factor that can impact the computational cost the most.

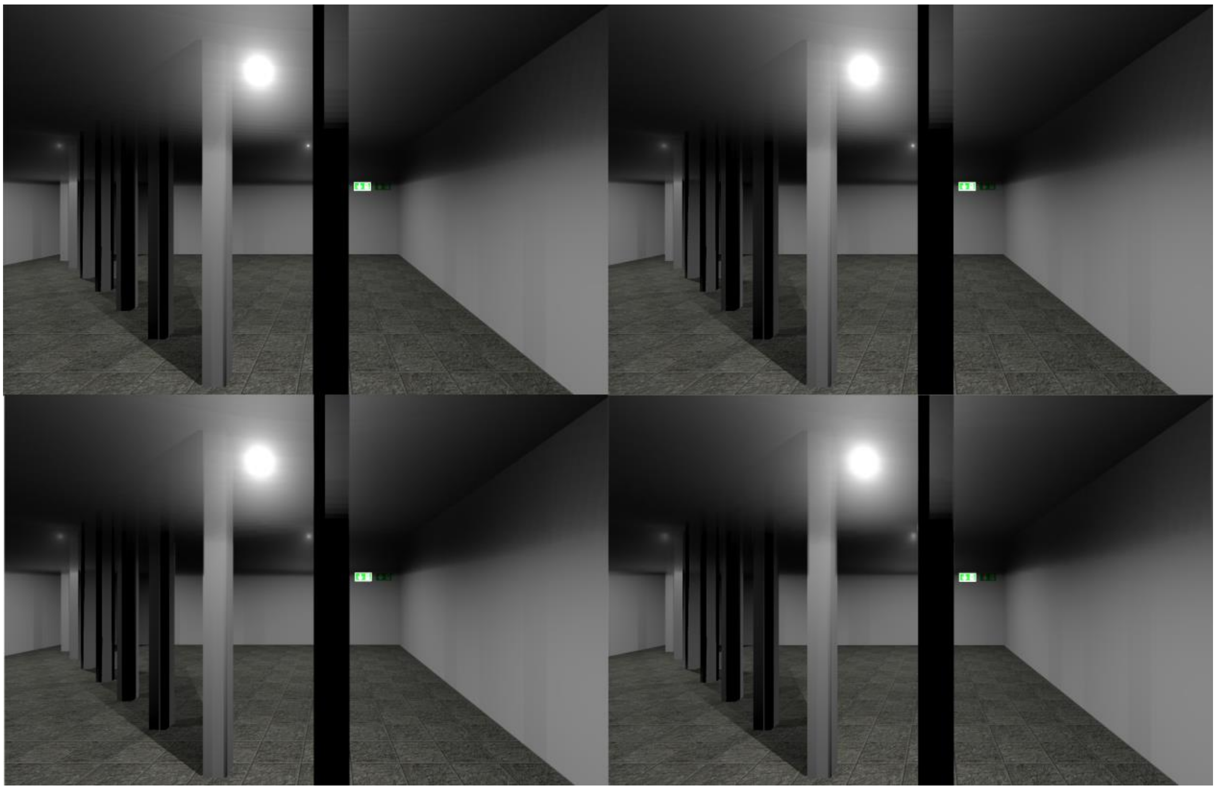


Figure 19 Effect of rendering the secondary buffer (scattering and extinction calculation data) in different resolution scales using 10 cm step length. Notice the reduction in visible banding as the resolution scale decreases at the cost of “blurriness” and artefacts around edges with large depth differences (e.g. edge of pillar close to the camera compared to the wall behind it). Top left: 100% scale. Top right: 50% scale. Bottom left: 25% scale. Bottom right: 12.5% scale.

Figure 20 shows the effect of using Gaussian blur on the final secondary buffer before being merged with the primary buffer using different resolution scales. An overall reduction in visible banding can be observed at the cost of “blurriness” (e.g. light sources in the background) and artefacts around edges with large depth differences (e.g. edge of pillar close to the camera compared to the wall behind it). These artefacts are increased as the resolution scale decreases, but the final image look very smooth without banding.

In general, it is recommended to use options that are acceptable given application. In most cases it is likely that a step length of 10 cm combined with a 50 or 25% resolution scale and using Gaussian blur produces a visually pleasing image while being computationally efficient.

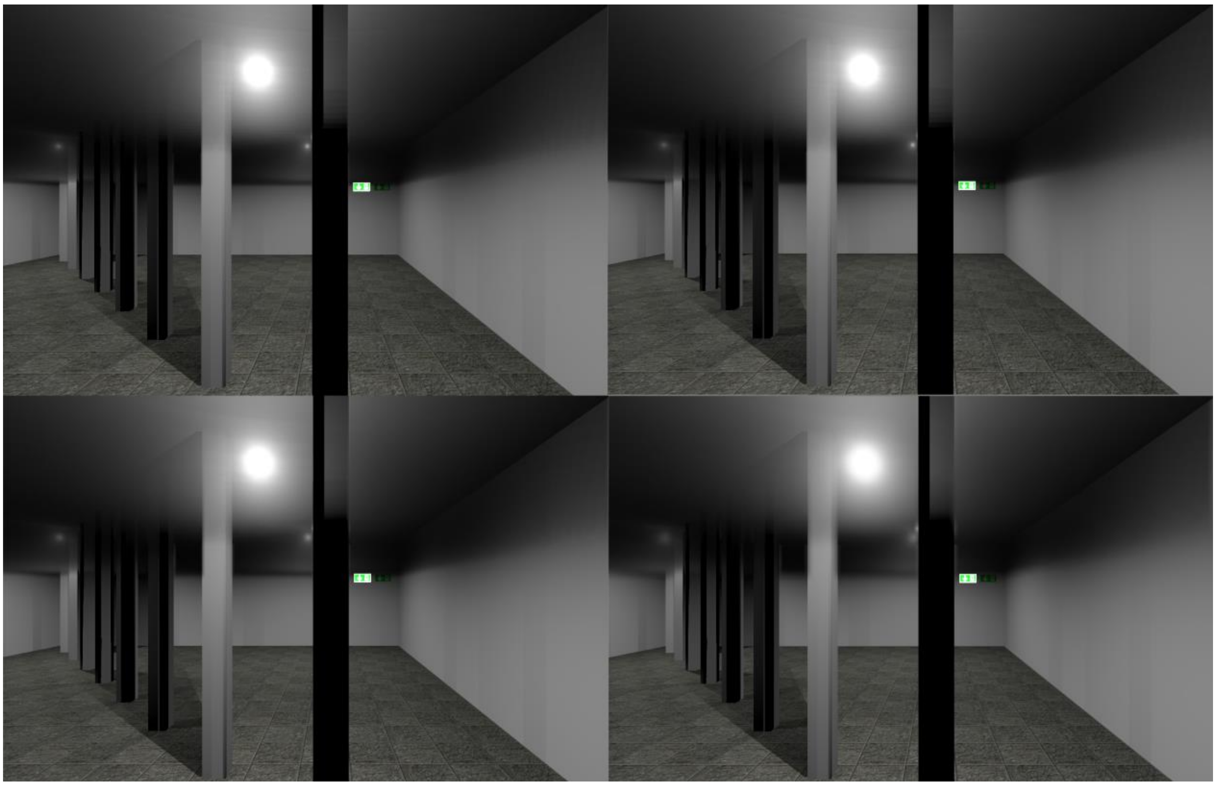


Figure 20 Effects of blurring the secondary buffer (scattering and extinction calculation data) using Gaussian blur using 10 cm steps. . Notice the overall reduction in visible banding at the cost of “blurriness” (e.g. light sources in the background) and artefacts around edges with large depth differences (e.g. edge of pillar close to the camera compared to the wall behind it). Top left: 100% scale. Top right: 50% scale. Bottom left: 25% scale. Bottom right: 12.5% scale.

6. Related work

The following work has incorporated parts of the work presented in this report:

- Rosero, F. (2017) Assessment of People's Perception of Fire Growth: A Virtual Reality Study, Lund University, LUTVDG/TVBB-5545-SE.
- Arias, S., Ronchi, E., Wahlqvist, J., Eriksson, J., Nilsson, D. (2018) ForensicVR: Investigating human behavior in fire with Virtual Reality, LUTVDG/TVBB-3218.
- Brandskyddsföreningen Fire VR game (2018) – interactive game in VR where the participants must put out a fire on a couch using a motion controller.

7. Conclusions

A novel implementation of visualization of smoke and fire in virtual reality has been demonstrated, as well as an implementation of a two-zone fire model in a game engine which allows for interactivity between the user and fire scenario. The resulting visibility is in agreement with previous simplified calculations, but also demonstrates that the visibility is complex and is dependent on several factors such as proximity to light sources, ambient light and contrast against walls.

The implemented algorithm is flexible in terms of visual quality and thereby computational cost, making it suitable for several different applications with different visual requirements.

The implemented two-zone model is basic but provides the user of interactivity that has previously not been possible. This opens up the possibility for using virtual reality in a range of applications, most obvious different kinds of training in smoke laden environments.

8. Future research

The presented work is just an initial attempt at using several new techniques and tools. Even though the work presented fulfils the objectives there is certainly room for future improvement.

8.1. Froxel implementation

Recent implementations of volumetric scattering using so called froxel (frustum view aligned volumetric pixel) grid using compute shaders has been observed in cutting edge game releases. Implementing froxels would require a total re-implementation of the visualization code but would likely make it more future proof and scalable.

8.2. True volumetric shadows

Current implementation of volumetric shadows calculates the distance in smoke between each surface and light and reduces the light intensity accordingly. This does not however take into account any total blockage in the ray path due to objects. To get true volumetric shadows, a 3D-texture containing light depth in light space is required for each light which adds significant complexity but is technically possible.

8.3. Light emission shapes

Using froxels could enable other light source shapes other than point lights, such as volumetric lights and spotlights (which are similar to point lights but have limited spread angle). This could be a first step in including more accurate light distribution from the flames.

8.4. Wave length dependent absorption

Current implementation is using a uniform extinction coefficient due to two reasons; performance and lack of experimental data. In future work it would be of interest to measure extinction coefficients of smoke using lasers at three different wavelengths corresponding to red, green and blue colors. This data retrieved could then be used to implement wavelength dependent extinction and scattering and examine the influence on visibility.

8.5. Expansion of two-zone model

As mention in the conclusions the implemented two-zone model is very simplified, and it would therefore be highly desirable to add more features in future iterations. One of the main features that would be of interest is a simple radiation model that can track the radiation received by the user. Another addition of interest would be tracking oxygen levels. Both of these additions would be used to evaluate the conditions of the user as they traverse the virtual environment.

8.6. Integration of advanced smoke data sets

Being able to import more advanced smoke data sets, e.g. from FDS, is of high interest since that would enable smoke and fire visualization in more complex scenarios where a zone model might not apply. This will come at a cost of interactivity with the fire and smoke but would be used as a complement to the presented work.

References

Blasi, P., Saec, B. L., Schlick, C. (1993) A Rendering Algorithm for Discrete Volume Density Objects, Computer Graphics Forum, doi.org/10.1111/1467-8659.1230201

Cha, M. et al (2012) A virtual reality based fire training simulator integrated with fire dynamics data, Fire Safety Journal Volume 50, Pages 12–24

Forney, G. P. (2013) NIST Special Publication 1017-2 Smokeview, A Tool for Visualizing Fire Dynamics Simulation Data Volume II: Technical Reference Guide, NIST.

Forney, G. P. and Moss W. F. (1994) Analyzing and Exploiting Numerical Characteristics of Zone Fire Models. Fire Science and Technology, 14(1 and 2):49–60. 7, 10

Henyey, L. G., Greenstein, J. L. (1941) Diffuse radiation in the Galaxy, Astrophysical Journal, vol. 93, p. 70-83

Heskestad, G.(2008) SFPE Handbook of Fire Protection Engineering, chapter Fire Plumes, Flame Height, and Air Entrainment. National Fire Protection Association, Quincy, Massachusetts, fourth edition, 15, 16, 17

Hillaire, S. (2015) SIGGRAPH 2015 – Advances in Real-time Rendering, SIGGRAPH

Jin, T. (1997). Studies on Human Behavior and Tenability in Fire Smoke, Fire Safety Science, Proceedings of the Fifth International Symposium, pp 3-21

Kinateder, M., et al. (2014) Virtual Reality for Fire Evacuation Research, Federated Conference on Computer Science and Information Systems pp. 319–327

Kinateder, M., et al. (2014) Social influence on route choice in a virtual reality tunnel fire, Transportation Research Part F, Traffic Psychology and Behaviour

Lengyel, E. (2007) Unified Distance Formulas for Halfspace Fog, Journal of Graphics Tools, Vol 12, No. 2

McGrattan, K., et al. (2017) Fire Dynamics Simulator User's Guide, NIST Special Publication 1019, Sixth Edition, NIST

Mulholland, G.W. (2002) SFPE Handbook of Fire Protection Engineering, chapter Smoke Production and Properties. National Fire Protection Association, Quincy, Massachusetts, 3rd edition.

Mulholland G.W. and Croarkin. C. (2000) Specific Extinction Coefficient of Flame Generated Smoke. Fire and Materials, 24:227–230, 235

Narasimhan, S.G. and Nayar, S.K. (2003) Shedding Light on the Weather, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. I, pp. 665-672.

NFPA, Guide for Smoke and Heat Venting (1985) NFPA 204M, National Fire Protection Association, Quincy, MA

Peacock, R. D., McGrattan, K. B., Forney, G. P., Reneke, P. A. (2017) NIST Technical Note 1889v1 CFAST – Consolidated Fire And Smoke Transport (Version 7) Volume 1: Technical Reference Guide, NIST.

Pharr, M,. Humphreys, G. (2010) Physically Based Rendering – From Theory to Implementation, Elsevier, ISBN 978-0-12-375079-2

Rosero, F. (2017) Assessment of People's Perception of Fire Growth: A Virtual Reality Study, Lund University, LUTVDG/TVBB-5545-SE

Rubini, P. A., Zhang, Q., Moss, J.B. (2007) Simulation of Visibility in Smoke Laden Environments, InterFlam 2007, 11th International Conference on Fire Science and Engineering, 3-5 Sept. 2007, London, UK

Unity (2018a) Screen Space Multiple Scattering for Unity, https://github.com/OCASM/SSMS, downloaded 2018-08-28

Unity (2018b), Unity Shader Macros, https://docs.unity3d.com/Manual/SL-BuiltinMacros.html, downloaded 2018-04-02

Unity (2018c) Unity Shader Variables, https://docs.unity3d.com/Manual/SL-UnityShaderVariables.html, downloaded 2018-04-02

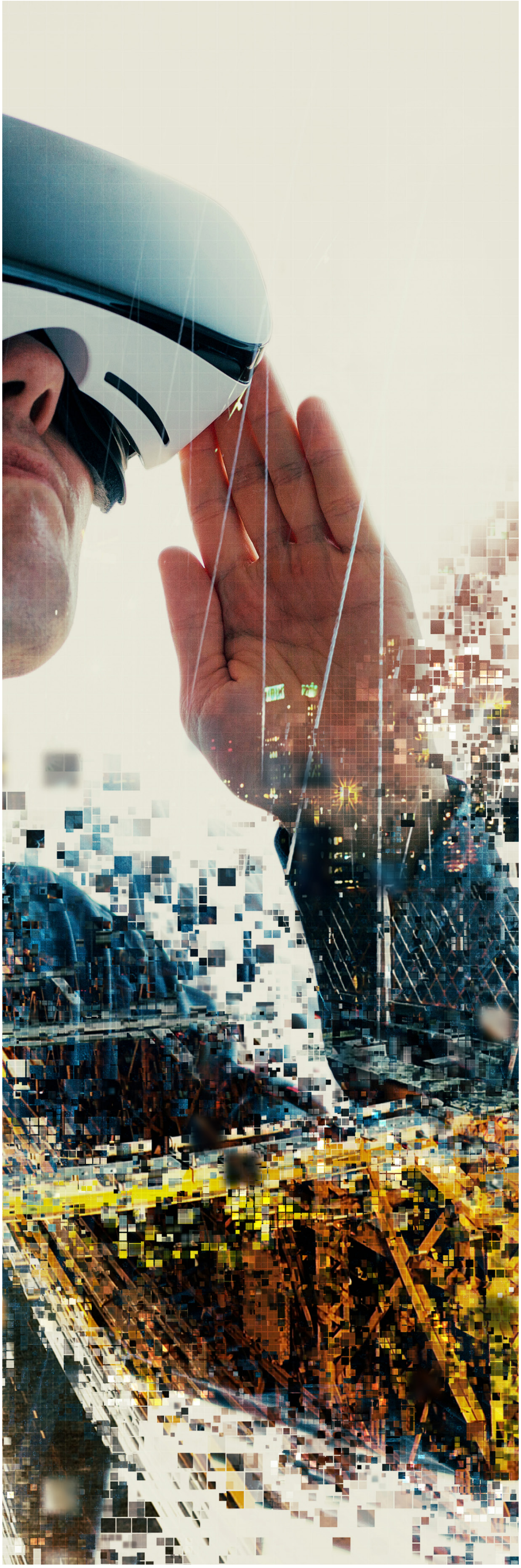
Unity (2018d) Unity What is a Particle System (2018c), https://docs.unity3d.com/Manual/PartSysWhatIs.html, downloaded 2018-04-05

Unity (2018e) Build once, deploy anywhere, https://unity3d.com/unity/features/multiplatform, downloaded 2018-04-05

Unity (2018f) Writing shaders, https://docs.unity3d.com/560/Documentation/Manual/ShadersOverview.html, downloaded 2018-04-03

Unity (2018g) A feature-rich and highly flexible editor, https://unity3d.com/unity/editor, downloaded 2018-04-9

Xu, Z. et al. (2014) A virtual reality based fire training simulator with smoke hazard assessment capacity, Advances in Engineering Software Volume 68, Pages 1–8



Department of Fire Safety Engineering
Lund University
P.O. Box 118
SE-221 00 Lund, Sweden
brand@brand.lth.se
http://www.brand.lth.se/english
Telephone: +46 46 222 73 60
Fax: +46 46 222 46 12

OUR SPONSORS & PARTNERS:

Brandforsk

BRANDFORSK REPORTS FROM 2018:

2018:1 Fire and explosion hazard of alternative fuel vehicles in tunnels

2018:2 Engineering methods for structural fire design of wood buildings– structural integrity during a full natural fire

2018:3 Kvantifiering av variationer vid beräkning av strukturers respons vid brandexponering

2018:4 Risk-Informed Fire Safety Design of Structures

2018:5 Visualizaton of fires in VR

The Swedish Fire Research Board, Brandforsk, is a non-profit body, formed in collaboration between insurance companies, industry, associations, government agencies and local municipalities. The purpose of Brandforsk is to initiate and fund research and knowledge development within the field of fire safety in order to reduce the negative social and economic impact of fire.

The work is under the leadership of the board of directors and is undertaken in the form of projects at universities, institutes of technology, research organisations, government agencies and industrial enterprises. The Secretariat of Brandforsk shares the pre¬mises of the Swedish Fire Protection Association, SFPA, which is also the principal organization.

Brandforsk

Årstaängsvägen 21 c
P.O. Box 472 44, SE-100 74 Stockholm, Sweden
Phone: 0046-8-588 474 14
brandforsk@brandskyddsforeningen.se
www.brandforsk.se