Progress Report 2

Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires)

Wahlqvist, Jonathan; Rohaert, Arthur; Rubini, Philip; Ronchi, Enrico

2023

Document Version:
Publisher's PDF, also known as Version of record

Progress Report 2: Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires)

Jonathan Wahlqvist

Arthur Rohaert

Philip Rubini

Enrico Ronchi

Department of Fire Safety Engineering

Lund University, Sweden

Lund 2023

Report 3253

Progress Report 2: Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires)

Jonathan Wahlqvist

Arthur Rohaert

Philip Rubini

Enrico Ronchi

Lund 2023

Progress Report 2: Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires)

Jonathan Wahlqvist, Arthur Rohaert, Philip Rubini, Enrico Ronchi

Report 3253

ISRN: LUTVDG/TVBB-3253-SE

Number of pages: 34

Keywords: wildfires, fire safety, evacuation, smoke, simulation, wildland-urban interface

Abstract. This is the second progress report of the international project funded by the National Research Council of Canada called Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires). In this second phase, the research performed included two main tasks: 1) developments concerning the modelling of smoke and 2) development of analysis methods concerning validation datasets for wildfire evacuation. Visibility in smoke is a key aspect in terms of safe evacuation in wildfire scenarios. As valid results of evacuation modelling tools would rely on an accurate representation of the impact of smoke on people, physical accuracy is required. Therefore, the rendering of smoke needs to be physically based while still being computationally inexpensive so that it can be run in a multi-physics tool in real-time. This report presents an approach for rendering smoke with a single in-scattering term which allows for smoke and light interaction over multiple wavelengths. In addition, analysis methods concerning validation datasets for wildfire evacuation models are presented and discussed. This includes both traditional regression methods as well as approaches based on machine learning.

Avdelningen för brandteknik

Lunds universitet

P.O. Box 118

221 00 Lund

brand@brand.lth.se

Department of Fire Safety Engineering

Lund University

P.O. Box 118

SE-221 00 Lund, Sweden

brand@brand.lth.se

Acknowledgements

The authors wish to acknowledge the support from the Collaborative R&D Initiative of the National Research Council of Canada (grant #STR2-0201) for contributing to funding this research. The authors are also thankful to the National Institute of Standards and Technology for financially contributing to the development of the WUI-NITY tool. The authors wish to acknowledge all collaborators in the RACE Wildfire project, in particular Dr Steve Gwynne who reviewed this report.

Table of Contents

1. Introduction

This report presents the second part of the interim results of the Resilience and Adaptation to Climatic Extreme Wildfires (RACE Wildfires) project, funded by the National Research Council of Canada. The project included different activities, this report being focused on the development of modelling tools to assess community evacuation protocols.

The reason for initiating the project was due to the potentially severe negative impacts caused by wildfires spreading towards urbanized areas, which may require mass evacuations. Wildfires may occur in Wildland-Urban Interfaces (WUI), where structures and vegetation are integrated into a fire-prone environment (Mell et al., 2010). The National Research Council of Canada (NRCC) and the National Resources Canada (NRCan) have identified the need for research to reduce the impact of WUI fires. This research led to the development of a Canadian National Guide for WUI fires (Benichou et al., 2021). Modelling tools may be useful for WUI fire evacuation, providing support both during planning and real-time management (Beverly & Bothwell, 2011). Typically, these tools consist of various modelling layers, such as wildfire spread modelling, human response and movement simulations, and traffic evacuation modelling (Ronchi et al., 2019, 2020; Ronchi & Gwynne, 2019).

The present interim report mainly focuses on the representation of smoke and also presents initial work performed on the treatment of data that can be used for the validation of wildfire evacuation modelling tools. The activities performed included a general review of approaches and methods for the representation of smoke along with the development of a new approach to smoke modelling in virtual reality. In addition, methods for the analysis of large wildfire evacuation data have been proposed including both traditional methods as well as machine learning.

The use of virtual reality (VR) has large a potential for modelling applications, as shown by the wildfire evacuation model WUI-NITY (Wahlqvist et al., 2021) and its associated modelling layers (including wildfire and smoke) and for experimental studies, e.g., driving behaviour in smoke (Wetterberg et al., 2021). One aspect which is of prime interest is to enhance the received visual stimuli of smoke and consequently visibility. This is particularly important in the evacuation domain as reduced visibility may impact human response (Ronchi & Gwynne, 2019) – both in the routes used and the speeds that can be maintained - and the whole evacuation process (Intini et al., 2022). The interaction between smoke and light is a critical part of visibility but has thus far been simplified due to its physical complexity and thereby requirements on computer hardware. For example, the WUI-NITY tool which is developed with the game engine Unity is capable of two-way interaction between different modelling layers. Nevertheless, there is a dire need for further developments in order to fully being able to model the impact of smoke on evacuation.

An accurate representation of fire smoke in VR would allow the collection of evacuation data through dedicated experiments aimed at investigating driving behaviour in smoke. For instance, it could allow the accurate representation of vehicle headlights during evacuation on the road, thus enabling the collection of data concerning headways during evacuation in wildfire scenarios. This is a clear research gap which has been identified in previous research in this domain (Rohaert et al., 2023). Such data are ultimately needed to systematically investigate the fundamental relationships between speed, flow and densities. Those relationships are the backbone of current macroscopic traffic simulations and are typically used for validation of microscopic traffic models (Ronchi et al., 2021). Therefore, the study of treatment and analysis of such data is a natural consequential step for the current project, and it is addressed in this report as well. The methods explored for data treatment and analysis include both traditional regression approaches as well as

novel methods based on machine learning (e.g., kernel ridge regression, support vector regression, and Gaussian process regression).

1.1 Aim and objectives

This work investigates developments needed in evacuation modelling tools for wildfire scenarios aimed at improving the accuracy of their results.

The first objective includes a review of the core theory and modelling approaches concerning modelling of smoke and representing visibility conditions. This information is used as a starting point to develop a new approach for the representation of smoke in virtual reality. The present work aims to build upon the approach presented in (Wahlqvist, 2018) by applying techniques used in the gaming industry while focusing on physically based smoke characteristics and the specific application of fire safety engineering.

The second objective includes paving the way for the collection, treatment and analysis of large wildfire evacuation datasets. In particular, the main focus has been on the relationship between three key variables: speed, flow and density, which can be available in traffic databases and can also be studied in Virtual Reality experiments (e.g., using driving simulators (Shibata & Sakuraba, 2019)). Therefore, this work provides guidelines to select, treat and analyse data from real-world wildfire scenarios as well as virtual reality evacuation experiments. The data selection process is linked to some key issues, such as collection methods, measurement uncertainty, and assumptions adopted in the data treatment. In other words, inclusion/exclusion criteria can play a key role in the definition of the final dataset obtained and used for development and validation purposes. In addition, this work reviews different methods of obtaining the relationships between speed, flow and density. This expands on the discussion on common mistakes in the statistical analysis of movement data from (Bode & Ronchi, 2019). First, traditional statistical methods (e.g., curve fitting) are reviewed and analysed, for the specific case of movement dynamics. Second, the benefits and drawbacks of common machine learning regressions are explored (e.g., kernel ridge regression, support vector regression and Gaussian process regression). This work presents a review of existing approaches and guidelines on how to apply them to real data sets from traffic evacuation dynamics in case of wildfires. These datasets might vary in size, distribution, and homogeneity.

1.2 Report overview

The first chapter of this report briefly introduces the project, the overall aim and objectives of this part of the work and the report structure. Chapter 2 first presents the theoretical foundation concerning modelling fire smoke along with existing modelling approaches and then presents the method employed for the development of a new fire smoke sub-model in a game engine. Chapter 3 presents a brief overview of existing approaches adopted for the treatment and analysis of data concerning the fundamental relationships between speed, flow and densities which are typically used in traffic evacuation modelling. Chapter 4 presents a general discussion on the sub-model developed, their possible uses and further steps needed for their full implementation in existing multi-physics wildfire evacuation modelling tools.

1.3 Publication outputs

At the time of publication of this second interim report, three publications have been published or submitted for publication based on the activities conducted in the project. Two publications [1] and [2] refer to the work performed on the development of a fire smoke sub-model and its impact

on evacuation. The third publication [3] refers to the treatment and analysis of traffic evacuation data concerning the speed, flow and density relationships.

[1] Intini, P., Wahlqvist, J., Wetterberg, N., & Ronchi, E. (2022). Modelling the impact of wildfire smoke on driving speed. International Journal of Disaster Risk Reduction, 80, 103211.

[2] Wahlqvist, J., Rubini, P. (2023). Real-time visualization of smoke for fire safety engineering applications. Symposium of the International Association for Fire Safety Science, Tsukuba, Japan.

[3] Rohaert, A., Wahlqvist, J., Najmanova, H., Bode, N., Ronchi, E. (2023). The evaluation of data fitting approaches for speed/flow density relationships. Proceedings of the 11th International Conference on Pedestrian and Evacuation Dynamics (PED2023) Eindhoven, The Netherlands – June 28-30, 2023.

2 Development of a sub-model of fire smoke

This section first introduces the base theory concerning the modelling of fire smoke, including visibility, smoke properties and phase function. This section is followed by a description of existing modelling approaches and a new modelling approach developed in a game engine (Unity) which makes use of both traditional code and shaders, e.g., code that runs on the graphics processing unit (GPU).

2.1 Visibility in Smoke

The light intensity along a path in a participating media (such as smoke) is changed due to three main processes (Pharr et al., 2016) (shown in Figure 1):

- Absorption – the reduction in intensity due to the conversion of light to another form of energy, such as heat.
- Emission – intensity that is added along the path from luminous particles 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. This is often divided into two sub-parts; in-scattering, light that is added along the path, and out-scattering, light that is lost along the path.

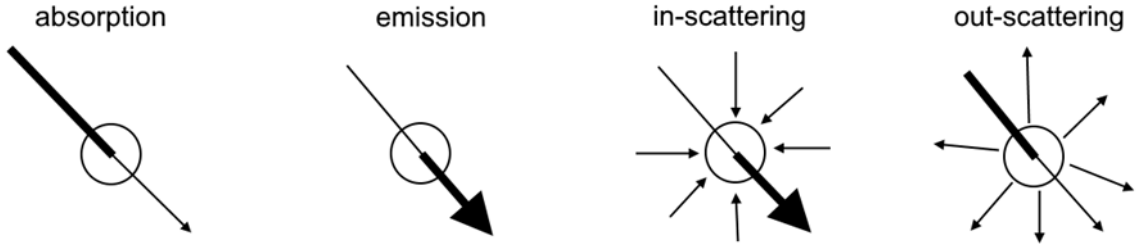


Figure 1. Main processes involved in changing the intensity of light along a path.

As described by (Forney, 2013), the change in radiance in the direction ω at any one instant and wavelength may be expressed as in Equation 1.

$$(\omega \cdot \nabla)C(x,\omega) = -\underbrace{\sigma_a(x)C(x,\omega)}_{absorption} - \underbrace{\sigma_s(x)C(x,\omega)}_{out-scattering} + \underbrace{\sigma_a(x)C_e(x,\omega)}_{emission} + \underbrace{\sigma_s(x)\int_{4\pi} p(x,\omega,\omega')\,C_i(x,\omega')d\omega'}_{in-scattering} \quad \text{[Eq. 1]}$$

where $\mathrm{C}(x,\omega)$ represents the radiance at x along a direction ω. Extinction is the total reduction along the path, a combination of absorption and out-scattering. A simplification can be made to Equation 1 by combining the absorption and out-scattering into the so-called extinction coefficient σ_t (see Equation 2).

$$\sigma_t = \sigma_a + \sigma_s \quad \text{[Eq. 2]}$$

Equation 1 can also be simplified by ignoring any in-scattering as that is an additive term ($\sigma_s(x) = 0$); both these simplifications are done in Smokeview for example (Forney, 2013). The resulting simplified equation can be useful to evaluate visibility from smoke density using experimental data from (Jin, 1978), one of the seminal and mostly used works related to visibility in smoke. According to (Mulholland, 2002) the most useful quantity for assessing visibility in a space is the light

extinction coefficient, K. 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 Beer-Lambert's (or Bouguer's) law, as shown in Equation 3.

$$\frac{I}{I_0} = e^{-KL} \quad \text{[Eq. 3]}$$

Where I_0 is the 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 and wavelength dependent (Suo-Anttila et al., 2005), and the density of smoke particulate, $\rho \cdot Y_S$ (see Equation 4).

$$K = K_m \cdot \rho \cdot Y_S \quad \text{[Eq. 4]}$$

Where ρ is the density of the hot gasses and Y_S is the mass fraction of soot in the hot gasses. The default value of K_m in the widely used Fire Dynamics Simulator (FDS) is 8700 m^2/kg (McGrattan et al., 2017) which passes these values on to Smokeview for rendering. This is based on (Mulholland & Croarkin, 2000) which suggests that the value for K_m most flaming fuels is 8700 $m^2/kg \pm 1100$ m^2/kg at a wavelength of 633 nm (a monochronic red light).

According to (Mulholland, 2002) the visibility through smoke can be estimated by dividing a non-dimensional constant C, based on the nature of the observed object through the participating media (smoke) and the light extinction K (see Equation 5).

$$S = \frac{C}{K} \quad \text{[Eq. 5]}$$

The suggested values by (Mulholland, 2002) for C are 3 for light-reflecting signs and 8 for light-emitting signs. However, (Jin, 1997) recommends a range of C between 2-4 for a light-reflecting sign and a range of C between 5-10 for a light-emitting sign, indicating that these values might be hard to determine. Recent experimental work also questioned the suggested values by Jin for higher values of the extinction coefficient and that the visibility constant under white smoke conditions is higher than what Jin suggested (Elhokayem, 2022). There are some limitations to this approach as visibility is treated as a property entirely based on the observed object being light emitting or light reflecting. To date, taking into account that it cannot be determined in a deterministic way how much light is actually transported to the observed object from the surroundings, making it especially hard to evaluate the visibility of light-reflecting objects. Another limitation is the lack of inclusion of any interaction between the light and the smoke besides the extinction from an observed surface and observer, such as in-scattering which could affect the visibility. It is also not possible to evaluate any wavelength-dependent properties of either the smoke or the light without additional data.

Although not necessarily focused on visibility, other areas such as computer graphics have been active in researching rendering of participating media for four decades, going from early works such as (Blinn, 1982) to featuring algorithms fast enough for real-time rendering in games (Hillaire, 2016; Kniss et al., 2002; Szirmay-Kalos et al., 2005; Tatarchuk et al., 2014; Ummenhoffer & Szirmay-Kalos, 2005). Worth noting however is that the use of real-time approaches often means sacrificing some accuracy along the way, e.g. multiple scattering and global illumination. General approaches are specifically used in off-line rendering, even though the advances in graphics hardware compute power can close the gap between real-time and off-line rendering techniques (Pérez et al., 1997), e.g. enabling real-time path tracing (Clarberg et al., 2022).

2.2 Smoke properties and phase function

As a starting point for the review of the existing modelling approach, the work by (Bond & Bergstrom, 2006) is used to assemble data from different fields which describe the optical properties of soot which are generally the main component of fire-induced smoke. This summary shows both the complexity and variety of fire-induced smoke, making it clear that each application requires different parameters. To add to this complexity, experimental measurements and Mie theory (Wriedt, 2012) can be combined to present radiative properties for representative wildfire smoke at 550 nm (green) which showed that different combustion phases might produce significantly different smoke properties (Patterson & McMahon, 1984). They report a single-scattering albedo of 0.97 during the smouldering phase while it was 0.66 during the flaming combustion phase. The single-scattering albedo is the ratio between the scattering coefficient and the extinction coefficient and can be used to roughly determine the importance of multiple-scattering (Adamson, 1975). A value of 1.0 means that any extinction is due to out-scattering and that light will scatter around until it exits the participating medium. A value of zero means extinction is only due to absorption and light will enter and exit in the same direction, making it possible to completely omit any scattering effects.

Widmann (2003) analysed data for flaming fires and found a correlation between wavelength and extinction coefficient which can be useful, but unfortunately, it does not treat scattering and absorption separately. More recently, (Koch et al., 2021) obtained data on burning plastics and investigated the wavelength dependency and its effect on visibility but again focused on extinction rather that scattering and absorption. However, (Suo-Anttila et al., 2005) reported both wavelength dependency and absorption and scattering coefficients individually for three different fuels: suitcase, resin and Jet-A. Table 1 shows averaged data for all three fuels, and it can be seen that the normalized wavelength-dependent absorption and scattering ratios are different; this means that the wavelength dependency is stronger for scattering compared to absorption. Therefore, scattering and absorption need to be treated separately and not as ratios of the total extinction coefficient.

Table 1. Smoke property data extracted and averaged from (Suo-Anttila et al., 2005).

Wavelength	650 nm (red)	530 nm (green)	460 nm (blue)*
Mass-specific scattering coefficient [m²/kg]	1800	2900	4000
Normalized specific scattering coefficient [-]	1	1.61	2.22
Mass-specific absorption coefficient [m²/kg]	4600	6000	7300
Normalized mass-specific absorption [-]	1	1.30	1.59
Mass-specific extinction coefficient [m²/kg]	6400	8900	11300
Normalized mass-specific extinction coefficient [-]	1	1.39	1.77
Single-scattering albedo coefficient [-]	0.28	0.33	0.35

* Data has been extrapolated.

The phase function describes the relative angular distribution of light intensity when scattering occurs at a given wavelength. As shown by (Weinert et al., 2003), different fuels can produce smoke with quite varied phase functions and they are generally quite complex. This creates problems from both an experimental point of view, as it might be hard to obtain values for all angles and wavelengths of interest, but it also creates issues when trying to model the complex scattering phase function, a problem not unique to fire and smoke. Due to this reason, some analytical phase functions applicable to Mie-scattering have been developed for practical use:

- Isotropic scattering – all light scatters equally in every direction.
- Henyey-Greenstein phase function (Henyey & Greenstein, 1941) – a phase function that enables forward- or backward-scattering based on a single parameter g.
- Schlick phase function (Blasi et al., 1993) – an approximation of the Henyey-Greenstein phase function created to be more computationally effective.

Combinations of two phase-functions with different properties can be done to emulate e.g. double lobe scattering with both forward- and backward-scattering which might be more suitable for some fuels.

2.3 Existing approaches for smoke modelling and visualization

Wildfire predictions have been performed for decades and one of the seminal mathematical models (Rothermel, 1972) is still widely used today. The most prevalent computational models implementing the so-called Rothermel equations are FARSITE (Finney & others, 1998) and Prometheus (Tymstra et al., 2010) which share similar implementations. As they solve the fire spread on a two-dimensional plane, they can be considered quite computationally effective, and can thereby be used to perform simulations of wildfire spread over large areas. However, they both have a significant weakness as neither can do any form of smoke dispersion. More advanced models that are "more" physics-driven have large drawbacks in terms of computational efficiency.

To predict gas dispersion and gas concentrations, many different models have been developed in the past decades (Blackmore et al., 1982; Siddiqui et al., 2012). These models can be classified in accordance with the mathematical approach in use (Assael & Kakosimos, 2010). It is possible to distinguish different categories with increasing complexity:

- Gaussian models
- Box models
- Lagrangian particle and puff models
- Computational fluid dynamics (CFD) models (Rodean, 1996).

In the past decades, the increment in computational power has allowed the spread of the use of CFD models to investigate a number of indoor (Siddiqui et al., 2012) and outdoor accident scenarios such as railway accidents (Manca & Brambilla, 2010), gas dispersion in urban areas (Hanna et al., 2006; Lovreglio et al., 2016; Pontiggia et al., 2009, 2010, 2011) or rail-car releases (Hanna et al., 2009). These types of models have been in some instances more effective in representing scenarios with complex geometries, such as urban areas (Pontiggia et al., 2010). However, the selection of the gas dispersion model can be affected by several other factors, such as the scale of the problem.

Smoke models are developed based on atmospheric transport and dispersion theory and chemical mechanisms or statistical relationships. Various types of smoke models are available to simulate the rise, dispersion, transport and deposition of smoke particles and gas and chemical reactions for generation of ozone and secondary organic carbon (Goodrick et al. 2012). Lagrangian models such as CALPUFF (Scire 2000), Hybrid Single-Particle Lagrangian Integrated Trajectory (HYSPLIT) (Draxler and Rolph 2003), FLEXPART (Stohl and Thomson 1999) and Daysmoke (Achtemeier et al. 2011) predict variations of individual moving smoke, which appears either as a collection of independent 'puffs' or as infinitesimal air containing a fixed mass of pollutant particles. Eulerian models such as the Community Multiscale Air Quality (CMAQ) model (Appel et al. 2017), Comprehensive Air Quality Model with Extensions (CAMx) (Ramboll Environ 2016),

and the ECMWF Integrated Forecasting System (IFS) (Wedi et al. 2015) predict variations of smoke particle and gas concentrations at spatial grid points.

Smoke models such as WFDS, FIRETEC, FIRELES, FIRESTAR, WRF-SFIRE, Daysmoke, CAWFE (Clark et al. 2004; Coen 2013), ARPS-DEVS (Dahl et al. 2015), and MesoNH-ForeFire (an atmospheric and fire spread model) (Filippi et al. 2011) and dynamic plume rise models (Freitas et al. 2010; Grell et al. 2011) explicitly resolve the plume rise.

Many CFD models and frameworks exist that are capable of simulating dispersion, e.g., OpenFOAM, but only a few are directly aiming at simulating specifically smoke dispersion from wildfires. One such model is Fire Dynamics Simulator (McGrattan et al., 2013) that has multiple sub-models to model wildfire spread, both physics-driven and Rothermel-driven (Rehm & McDermott, 2009).

One of the most commonly used smoke visualizers, Smokeview (Forney, 2013), has implemented volume rendering by performing ray-marching over the volume containing the smoke data (Forney, 2013). Other work using ray-marching exists (Kang & Macdonald, 2005), though it does not consider light/smoke interactions other than pure light extinction. Additionally, the properties of the smoke are considered wavelength independent, assigning the same extinction coefficients to all wavelengths. As no general concept of light exists in these approaches, it can be difficult to accurately assess visibility as this has to be done using correlations based on experimental data (Jin, 1978) which were done under specific light conditions. Smokeview also provides a slice rendering technique with the same limitations as the ray-marching approach, but similar techniques have been developed by both (Staubli et al., 2005) and (Carlsson et al., 2007) with the inclusion of more complex light models. However, the included light model in (Carlsson et al., 2007) only interacts with opaque surfaces, not the participating media itself, while (Staubli et al., 2005) does offer per vertex light extinction from the smoke itself but still relies on empirical models for visibility.

More physically accurate approaches for rendering smoke to evaluate visibility do exist (Rubini et al., 2007; Zhang, 2010), and the rendered results show great improvements compared to the simplified approaches mentioned previously. This approach included multiple scattering of light in smoke based on physical properties (Zhang & Rubini, 2011) as well as global illumination, rendering realistic output images which can be used to evaluate visibility. The main drawback however is the computational cost of these techniques, resulting in slower than real-time rendering making it unsuitable for interactive applications such as virtual reality evacuation experiments or real-time wildfire management.

A real-time approach that incorporated single-scattering was presented by (Wahlqvist, 2018), but that approach had some drawbacks such as no wavelength-dependent smoke properties, as well as using scaling of the native image resolution to influence the performance, a technique that easily leads to image artefacts around sharp edges. As computational efficiency is of prime interest for wildfire applications, given the possible large scales under consideration, approaches that can utilize graphical processing units (GPU) and use efficient algorithms were explored.

2.4 A new smoke modelling/visualization approach

The model was implemented in the game engine Unity (Unity Technologies, n.d.) utilizing both traditional code (Unity scripts in C#) and shaders (code that run on the graphics processing unit (GPU)). 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 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). Most importantly, Unity has been used for the development of WUI-NITY, one of the few multi-physics wildfire evacuation modelling tools which is available to represent such scenarios (Wahlqvist et al., 2021).

2.4.1 Rendering of smoke

The general approach for the rendering of smoke is based on the work presented by (Tatarchuk et al., 2014), later improved by Hillaire (Hill et al., 2016; Tatarchuk et al., 2015). This method uses froxels (frustum aligned voxels) stored as volumetric textures. For each froxel, a calculation is made for the extinction coefficient and the view-aligned in-scattering term from every light source. The froxels are then ray-marched from the camera origin to accumulate transmission values and in-scattering terms along the view path. The final output image then reads data from the froxel corresponding to the depth buffer for the current pixel and multiplies the pixel colour with the transmission value in the froxel and adds the accumulated in-scattering colour. This approach is similar to older work such as (Han et al., 2007) with the distinction of using froxels instead of pixels.

The algorithm can be divided into four main parts:

- Data injection; collection of smoke data (absorption and scattering coefficients) and calculation of view-aligned in-scattering term per froxel.
- Surface light extinction; calculate the reduction of light due to extinction from all light sources to each surface (pixel).
- Froxel ray-marching; calculation of camera-aligned accumulated transmission and in-scattering terms.
- Final image; combining normal pixel output with froxel data.

Using similar definitions as Hillaire, Equation 6 describes a monochromatic light ray with single scattering as received by an observer.

$$\underbrace{L_i(x_i,\omega_i)}_{\substack{received\\ light}} = \underbrace{Tr(x,x_s)}_{\substack{extinction\\ along\ view\\ ray}} \underbrace{L(x_s,\omega_i)}_{\substack{reflected\\ light\ on\\ surface}} + \int_{t=0}^{s} \underbrace{Tr(x,x_t)}_{\substack{extinction\\ from\ light\\ source}} \underbrace{Lscat(x_t,\omega_i)}_{in-scattering} \underbrace{\sigma_s(x)}_{\substack{scattering\\ coefficient}} \; dt \qquad \text{[Eq. 6]}$$

where:

$$L(x_s,\omega_i) = \underbrace{Vis(x_t,L)}_{\substack{visibility\\ function}} \underbrace{L(x_t,\omega_i)}_{\substack{surface\\ function}} \qquad \text{[Eq. 7]}$$

$$Lscat(x_t,\omega_i) = \sum_{i=0}^{lights} \underbrace{p(\omega_i,L)}_{\substack{phase\\ function}} \underbrace{Vis(x,L)}_{\substack{visibility\\ function}} \underbrace{L_i(x,L)}_{\substack{light\\ intensity}} \qquad \text{[Eq. 8]}$$

$$Tr(x,x_t) = e^{-\int_0^s \sigma_t(x)dt} \qquad \text{[Eq. 9]}$$

Equation 9 is discretized by converting integral terms into Riemann sums and is practically solved using discrete steps in space (ray-marching). The discretization is the same approach used by Smokeview (Forney, 2013), as shown in Equation 10.

$$e^{-\int_0^S \sigma_t(x)dt} \approx e^{-\sum_{j=1}^{N-1} \sigma_t(x_j)\Delta t} = \prod_{j=i}^{N-1} e^{-\sigma_t(x_j)\Delta t} \quad \text{[Eq. 10]}$$

Where N is the amount of steps distance S is divided into, and Δt is the step length.

The reflected light on a surface is determined by the "visibility function" and the selected surface model in Unity. The standard surface model in Unity is a physically based bidirectional reflectance distribution function (Duvenhage et al., 2013), but a built-in Lambertian model or any other user-made surface model can also be applied. The "visibility factor" in this context is a function of whether a position in space is inside a shadow umbra combined with the reduction of luminance due to extinction along the path between a light source and surface.

$$Vis(x, L) = \underbrace{shadowMap(x, L)}_{visibility\ factor} \underbrace{Tr(x, x_L)}_{\substack{extinction \\ from\ light \\ source}} \quad \text{[Eq. 11]}$$

The data injection uses frustum-aligned voxels to inject and store data. This is done to easily control the computing resources required regardless of the screen resolution while still maintaining good image quality without edge discontinuities. The froxels are created by discretizing the cut off-pyramid between the camera near and far clip plane (the camera frustum) (see Figure 2). Depending on the computational resources available, the user can change the resolution to fit their current performance target, bearing in mind that lower resolutions will make the final rendered smoke lose details and might also introduce banding on surfaces. Temporal sample-upscaling can be used to improve image quality (Tatarchuk et al., 2015) but might introduce ghosting artefacts and has not been implemented in the current work.

Injection volumes are used to pass data to the froxels. In the current implementation, these volumes are axis-aligned and discretised using a cartesian grid. The volumes contain the following data:

- Individual cell smoke density.
- Volume mass-specific scattering coefficient per wavelength/colour (red, green, blue).
- Volume mass-specific absorption coefficient per wavelength/colour (red, green blue).

A compute shader calculates the current scattering and absorption coefficient per colour (wavelength) based on smoke density using Equation 4 for each froxel.

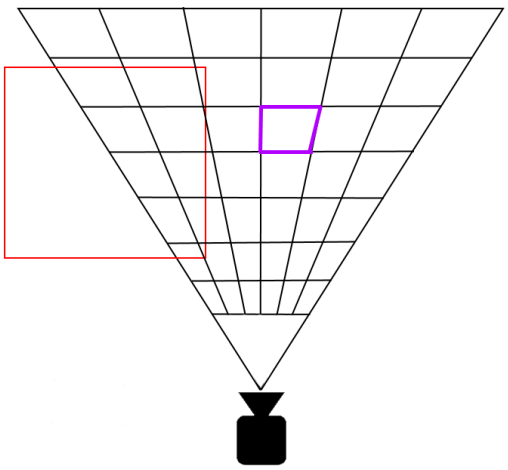


Figure 2. Top-down image of the frustum-aligned voxels, froxels, used to inject and store data. The red rectangle represents an injection volume containing smoke data, the purple outline highlights a single froxel.

Injection of the view-aligned in-scattering is a bit more resource intensive as the arriving light intensity from each light source to the current froxel centre must be calculated. Normally in computer graphics, this is done using attenuation functions, such as the inverse square law for a point source. But as the amount of arriving light is affected by the smoke, so-called volumetric shadows, ray-marching has to be performed to calculate the extinction due to the smoke. The total extinction along the path is then multiplied by the attenuation factor. Once the light intensity in the froxel is known a phase function is used to calculate the in-scattering along the view-aligned ray. In this step, the emission term, such as light from a flame, can also be injected along the path if desired. It will however not contribute light to all other voxels as that would require ray-marching each froxel against every other froxel that has an emission term which is computationally too expensive with current hardware.

Once the extinction coefficients and the in-scattering are calculated and injected, a second froxel buffer is used to accumulate transmission and in-scattering as seen from the camera. This is simply done by starting from each froxel in the XY-plane closest to the camera (Unity uses a Y-up, Z-forward coordinate system) and then march along the ray pointing from the camera origin accumulating transmission and in-scattering terms.

However, an issue when calculating the in-scattering term can easily be introduced in this step since the accumulated transmission changes over the step length, resulting in either over- or under-shooting the in-scattering term if using the initial or final transmission respectively. This becomes especially noticeable when taking long ray-marching steps. To solve this, an improvement was proposed (Tatarchuk et al., 2015) by integrating the in-scattered light along the step length. To get the corrected in-scattering contribution, S_c, Eq. 13 is used:

$$S_c = \int_{x=0}^{d} e^{-\sigma_t x} S dx = \frac{S - S \cdot e^{-\sigma_t d}}{\sigma_t} \qquad \text{[Eq. 12]}$$

Where S is the view aligned in-scattering term of the current froxel, σ_t is the extinction coefficient of the current froxel and d is the ray-march step length (the length of travel of the view-aligned ray inside the froxel). The extinction coefficient is still considered to be constant along the step length.

To calculate the change in light intensity on surfaces a very similar ray-marching step to the one that is done for the volumetric shadows is performed. This is again done using Equation 9 to calculate the extinction along the path between a light source and the world position of a pixel. The light intensity is then simply multiplied by the extinction calculated in the ray-marching step.

The current implementation is done while calculating regular surface shadows in the deferred rendering pipeline in Unity. This is done per pixel, but other approaches such as creating volumetric textures for each light and performing ray-marching on those textures can reduce the computational cost using lower resolutions of the volumetric texture (but will introduce shadow aliasing)

The rendering of the final image is straightforward and only contains two operations; using the volumetric texture created in the froxel ray-marching step for each pixel and sampling it at the world position of the pixel (graphics hardware provide "free" interpolation). The following step is to multiply the obtained transmission value with the pixel colour and add the accumulated in-scattering colour. As this requires depth information, transparent objects, such as glass, do not render correctly without extra attention.

2.4.2 Verification testing

In practice any smoke dataset can be imported and displayed, but in this first application case, the Fire Dynamics Simulator (FDS) together with Smokeview have been used to generate the presented data. Visual comparisons were done against Smokeview.

A visualization of a smoke plume in open atmosphere has been used for verification testing. This plume case is of interest to highlight how smoke properties, and thereby fuel sources, can have a large impact on how smoke is rendered.

The verification test does not include any light sources but rather uses artificial light conditions to enable exact calculations of a theoretical case. However, the main focus of the presented work is the result of the presence of smoke and light sources, either artificial or natural, simultaneously. To further investigate the results of the different rendering assumptions a simple open atmosphere plume was simulated in FDS to generate the data using a 0.1x0.1x0.1m grid size. A directional light to represent the sun (assumed to emit perfectly white light) was added above and behind the smoke plume, and a phase function with a mainly forward-scattering lobe was used (g=0.7). The smoke properties (see Table 2) were then changed in a way to highlight meaningful physics. Three cases were selected for comparison:

- Case 1, Smokeview assumptions: no scattering and no wavelength-dependent absorption.
- Case 2, Smokeview assumptions with scattering: the extinction coefficient is divided into 60% absorption and 40% scattering; a single scattering albedo close to the one found in (Suo-Anttila et al., 2005) at 650 nm (red).
- Case 3, using values found in (Suo-Anttila et al., 2005): each wavelength has unique extinction coefficients and single scattering albedos.

Table 2. Overview of used smoke properties for plume visualization.

	Mass-specific scattering coefficient [m^2/kg]			**Mass-specific absorption coefficient [m^2/kg]**		
Wavelength [nm] (colour)	650 (red)	530 (green)	460 (blue)	650 (red)	530 (green)	460 (blue)
Case 1	0	0	0	8700	8700	8700
Case 2	3480	3480	3480	5220	5220	5220
Case 3	1800	2900	4000	4600	6000	7300
Case 4 (Smokeview)	0	0	0	8700	8700	8700

As can be seen in Figure 3 the selected parameters outlined in Table 2 can have a large visual impact on the rendered smoke, and care must be taken to represent a suitable fuel, as well as representing fuel behaviour over time, for the application. For example, wildfires might produce vastly different smoke in the flaming region compared to the smouldering region, and this might influence evacuation-driving behaviour. To make it more clearly visible, the two cases which include in-scattering, an additive term, were taken with black backgrounds while the two other cases were taken with white backgrounds. Otherwise the smoke would completely merge with the background.

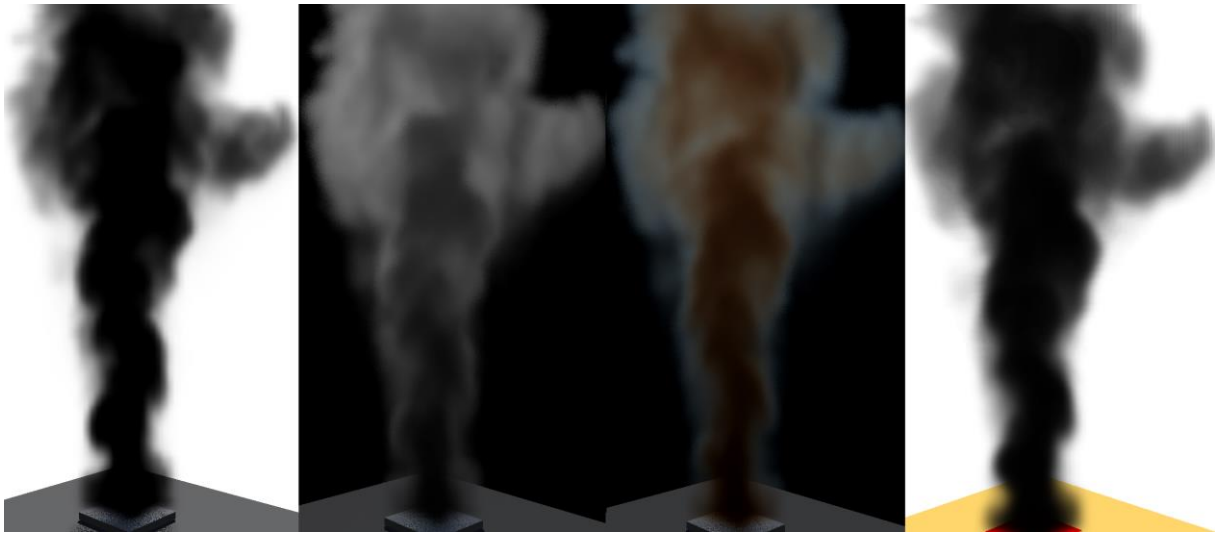


Figure 3. Comparison between different smoke characteristics using the presented work and rendering in Smokeview. From left to right: case 1, case 2, case 3, Smokeview as described in Table 2.

3 Treatment and analysis of validation data for wildfire evacuation simulations

Concerning the traffic evacuation modelling layer of a multi-physics wildfire evacuation modelling tool, understanding the fundamental relationships between speed, flow, and density is crucial. Macroscopic traffic evacuation modelling tools utilize these relationships to predict evacuation times (Ronchi & Gwynne, 2019) and ultimately develop effective evacuation plans and trigger buffers (Mitchell et al., 2023) that ensure the safe and efficient movement of people and vehicles from an area affected by a wildfire. Microscopic traffic evacuation models may use these relationships for validation purposes.

Speed, flow, and density are interrelated factors that define the behaviour of traffic on the road network. The fundamental relationship between these three factors demonstrates that as density increases, flow decreases, and at a certain point, the flow reaches its maximum capacity, known as the capacity flow rate. Beyond this point, as density continues to increase, flow decreases sharply, leading to a congested flow state where the speed of the vehicles on the road decreases significantly.

The obtainment of those fundamental relationships can be based on different sources of data. This includes data from real wildfire events (Rohaert et al., 2023; Ronchi et al., 2021) or dedicated experiments generally performed in virtual reality or driving simulators (Wetterberg et al., 2021). While the former presents the advantage of a higher ecological validity, the latter allows for high experimental control and repeatability, thus being ideal for the study of driving behaviour in specific conditions such as reduced visibility.

In the following sections, we discuss parametric and non-parametric regression.

3.1 Parametric regression

Parametric regression is a type of regression analysis in which the relationship between the dependent variable and the independent variables is modelled using a pre-specified functional form or equation, with a fixed number of parameters. This means that the shape and form of the regression curve are determined by the chosen functional form and the estimated parameter values.

In this context, the functional form refers to the speed-density relationship (the macroscopic model). Here, we first discuss some well-known traffic models and then discuss how to fit the models to evacuation data.

3.1.1 Functions (speed-density relationships)

Here, a set of well-known traffic models (see Equation 13 to Equation 18) have been fitted to the data to allow for a comparison between the different scenarios (evacuation and routine).

Equation 13 presents the parabolic model by Greenshield (Greenshields, 1936) :

$$v = v_f\left(1 - \frac{k}{k_j}\right) \qquad \text{[Eq. 13]}$$

Equation 14 presents the exponential model by Underwood (Underwood, 1961):

$$v = v_f e^{\left(-\frac{k}{k_c}\right)} \quad \text{[Eq. 14]}$$

Equation 15 presents the North-Western model by Drake et al (Drake et al., 1965):

$$v = v_f e^{\left(-\frac{1}{2}\left(\frac{k}{k_c}\right)^2\right)} \quad \text{[Eq. 15]}$$

Equation 16 presents the bi-linear model by Daganzo (Daganzo, 1994):

$$v = \begin{cases} v_f, & 0 \le k \le k_c \\ q_c\left(\frac{1}{k} - \frac{1}{k_j}\right), & k_c \le k \le k_j \end{cases} \quad \text{[Eq. 16]}$$

Equation 17 presents the model by Van Aerde and Rakha (van Aerde and Rakha, 1995; Wu and Rakha, 2009):

$$k = \frac{1}{a + \frac{b}{v_f - v} + c\,v} \; with \begin{cases} a = v_f\left(2v_c - v_f\right)/k_j v_c^2 \\ b = v_f\left(v_c - v_f\right)^2/k_j v_c^2 \\ c = 1/(v_c k_c) - v_f/k_j v_c^2 \end{cases} \quad \text{[Eq. 17]}$$

Equation 18 presents the model by Castillo & Benítez (del Castillo and Benítez, 1995):

$$v = v_f\left(1 - e^{\left(\frac{c}{vf}\left(1 - \frac{k_j}{k}\right)\right)}\right) \quad \text{[Eq. 18]}$$

3.1.2 Parameter optimisation

The methodology proposed here includes a dedicated procedure for model fitting which accounts for the peculiarities of the data under consideration. The macroscopic models $v = f(k)$ can be fitted to traffic data by minimizing the sum of squares (S) of the residuals (error on the speed predictions). The residuals are weighted as proposed by Qu et al. (2015) (see Equation 19) to make sure the model fits well over the entire range of densities (Qu et al., 2015), in which all m data points are ordered so that $k_i \le k_j$ when $i < j$). Without weighting, the models would typically fit poorly in the high-density region due to the imbalance that is often observed in traffic data.

$$S = \sum_i^m w_i\left(v_i - f(k_i)\right)^2 \quad \text{[Eq. 19]}$$

where

$$w_i = \begin{cases} (k_2 - k_1), & i = 1 \\ (k_{i+1} - k_{i-1})/2, & i = 2, 3, \dots, m-1 \\ (k_m - k_{m-1}), & i = m \end{cases}$$

To solve this optimisation problem, we applied the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963). The Levenberg-Marquardt algorithm is an iterative method that combines the steepest descent method and the Gauss-Newton method. At each iteration, it computes a step direction using a combination of the gradient and the Hessian matrix, and it adjusts a damping

parameter to control the step size. The advantage of the algorithm is that it is efficient and robust, and can handle non-linear problems with a large number of parameters.

3.1.3 Illustration of the parametric regression models

To illustrate the regression models, we used the data that was collected from the 2019 Kincade wildfire evacuation and consisted of traffic detector data collected from 24 locations, totalling 69116 data points (Rohaert et al., 2023). The data is openly available online (Rohaert et al., 2022).

Note that higher-density traffic conditions are rare in this dataset: 98 % of all data points lie in the density region of 0 to 20 veh/km/lane. This demonstrates the need to weigh the residuals, as discussed above.

Figure 4 shows the previously mentioned models with parameters optimised to fit the dataset.

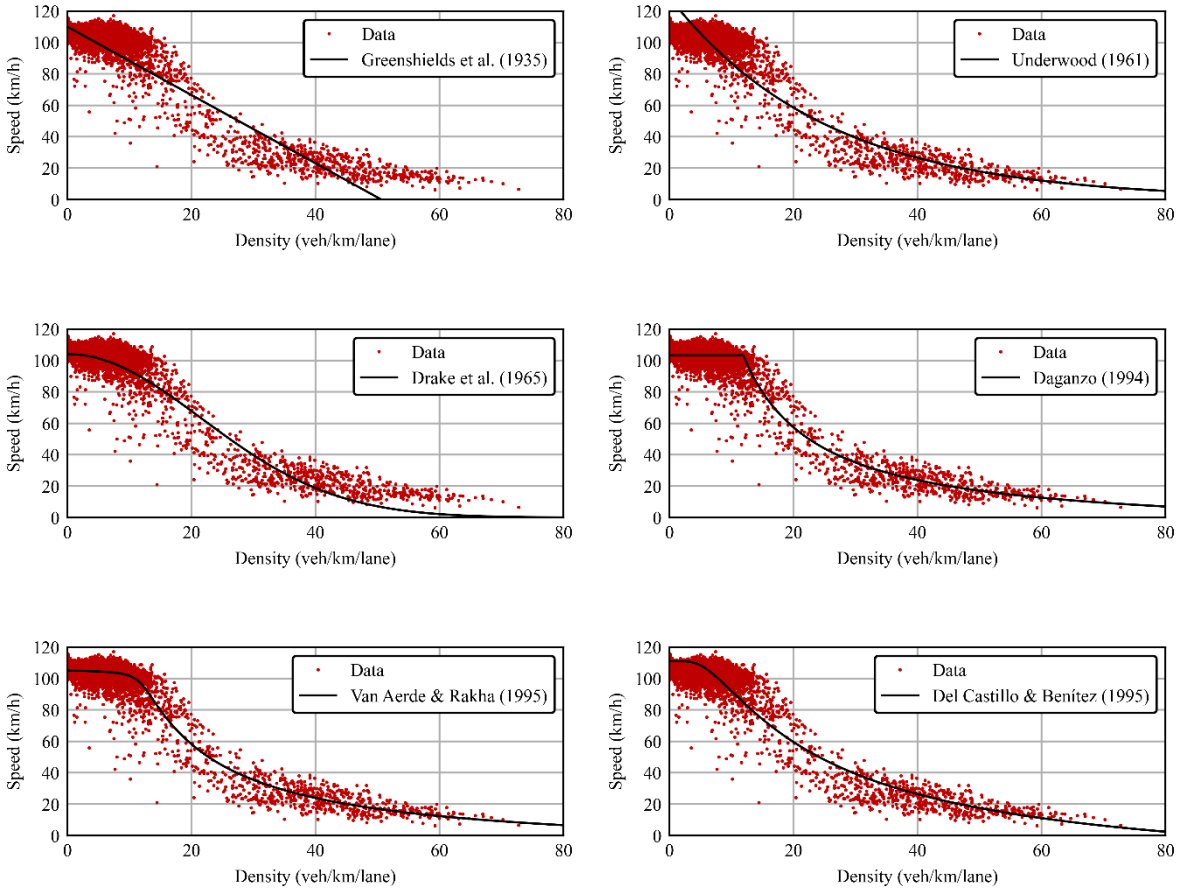


Figure 4. The parametric models, fitted to the dataset of the 2019 Kincade Fire.

A visual inspection of the fits shows that the models by Greenshields et al., Underwood, and Drake et al. fit poorly. Note that these models also have only little flexibility with only two parameters. The models by Van Aerde and Rakha provides a better fit but is more complex with four. The model by Daganzo also fits well, despite only having three parameters. The same conclusion can be drawn more objectively from the weighted root-mean-square error, presented by Rohaert et al. (2023).

Since these predefined speed-density relationships (parametric regressions) assume a specific functional form, they might not accurately capture the true relationship in the data. This is most strikingly illustrated by the fit of the model by Greenshields et al. The results are biased estimates and inaccurate predictions. Nonparametric regression, on the other hand, does not assume a specific functional form and allows for more flexibility in modelling complex relationships.

3.2 Non-parametric regression

Non-parametric regression is a type of regression that does not assume a specific functional form for the relationship between the input variables and the output variable. Instead, it estimates the function directly from the data, which makes it more flexible and less prone to bias. Non-parametric regressions allow to compare different curves and can provide better estimations of certain key parameters, such as the free-flow speed, the critical density or the capacity. The values of these parameters are hard to determine from parametric regressions. For instance, the free-flow speeds (fitted speed values at 0 veh/km/lane) in Figure 4 clearly depend on the ability of the macroscopic relationship to fit the data well in the low-density region.

Machine learning algorithms have become increasingly important in recent years because they can learn from data without being explicitly programmed. In this section, we will explore some popular non-parametric regression algorithms. We will also discuss hyper-parameter optimization, which is an important process, required to improve the performance of the algorithms.

3.2.1 Machine learning algorithms

There are several machine learning algorithms that can be used for non-parametric regression. In this section, we will discuss three popular algorithms: kernel ridge regression, support vector regression and gaussian process regression.

Kernel Ridge Regression (KRR)

Kernel Ridge Regression (KRR) is a non-parametric regression method that combines Ridge Regression with kernel methods (Pedregosa et al., 2011). In Ridge Regression, the goal is to minimize the sum of squared errors between the predicted output ($\beta_0 + \sum_j^p \beta_j x_{ij}$) and the actual output y_i, subject to a penalty term that shrinks the magnitude of the regression coefficients β_j towards zero. The penalty term is controlled by a hyperparameter, often denoted as λ (Hoerl & Kennard, 1970). The optimization process is shown in Equation 20 (Hoerl & Kennard, 1970).

$$\min_{\beta_j} \left(\sum_{i=1}^{N} \left(y_i - \left(\beta_0 + \sum_j^p \beta_j x_{ij} \right) \right)^2 + \lambda \sum_j^p \beta_j^2 \right) \quad \text{[Eq. 20]}$$

where y is the output (speed), x is the input matrix and β_j are the regression coefficients. N is the number of data points and p is the number of dimensions of the input.

In KRR, a kernel function is applied to the input features (here, the traffic density) to transform them into a high-dimensional space, where the linear relationship between the input features and output is replaced by a nonlinear relationship. This is called the kernel trick. Then, we apply Ridge Regression to the transformed features. The kernel function is usually a radial basis function (RBF) kernel, which is a popular choice due to its flexibility and ability to capture complex nonlinear relationships. It has an infinite dimensional feature space, which means that it can theoretically capture any complex relationship between data points (Pedregosa et al., 2011).

Here, the KRR is obtained using the *KernelRidge* class of the Scikit-learn library (*sklearn*) in Python (Pedregosa et al., 2011). Two hyperparameters needed to be determined:

- *alpha* is a regularization parameter that controls the amount of regularization applied to the weights in the regression model, often referred to in the literature as λ (see Equation 20). A larger value of alpha corresponds to stronger regularization and a simpler model, while a smaller value of alpha allows the model to fit the training data more closely.
- *gamma* is a scaling parameter that controls the spread of the kernel function and controls the range of influence of each training point on the predictions. A larger value of gamma corresponds to a narrower kernel and a more localized influence of each training point, while a smaller value of gamma results in a wider kernel and a more global influence of each training point.

Support Vector Regression

Support Vector Regression (SVR) is a type of non-parametric regression algorithm that uses the kernel trick, just like the KRR. The goal of SVR is to find a function that approximates the relationship between the input features and output variables while simultaneously minimizing the prediction error. In other words, the goal of SVR is to find a function that fits the data as closely as possible while still being generalizable to new, unseen data (Cortes & Vapnik, 1995; Müller et al., 1997).

After applying the kernel trick, SVR tries to find a hyperplane that maximizes the margin between the data points and the hyperplane. The margin is defined as the distance between the hyperplane and the closest data point. The hyperplane is chosen such that it is as far away from the data points as possible while still classifying them correctly. The hyperplane in SVR is used to make predictions on new data points. The predicted output is the value of the hyperplane at the corresponding point in the higher-dimensional space, which is then transformed back to the original input space using the inverse of the kernel function.

One key difference between SVR and KRR is that SVR tries to minimize the prediction error subject to a constraint on the margin, whereas KRR tries to minimize the prediction error subject to a constraint on the magnitude of the model parameters. Another difference is that SVR is more robust to outliers than KRR since the hyperplane is only influenced by the subset of the data points (the support vectors) that lie closest to the hyperplane. In contrast, KRR is influenced by all of the training data points.

Here, the SVR is obtained using the *SVR* class of the Scikit-learn library (*sklearn*) in Python (Pedregosa et al., 2011). In order to obtain a non-parametric regression, the Radial Basis Function (RBF) kernel is used. Two hyperparameters needed to be determined:

- *C*: is a regularization parameter that controls the trade-off between achieving a low training error and a low testing error. It determines the penalty for violating the margin of the support vector. A small value of C will lead to a larger number of support vectors, while a large value of C will lead to a smaller number of support vectors.
- *epsilon* the margin of tolerance, which determines the width of the epsilon-insensitive zone in which no penalty is associated with errors. Any prediction error that falls within the margin of tolerance is considered acceptable and does not incur a penalty.

Gaussian Process Regression

Gaussian Process Regression (GPR) is a probabilistic approach to regression that models the relationship between input and output variables as a Gaussian process. Like KRR and SVR, GPR uses a kernel function to define the similarity between pairs of input variables.

The basic idea behind GPR is to model the distribution of possible functions that could fit the training data, rather than just finding a single best-fit function. GPR assumes that the output variable follows a multivariate Gaussian distribution, with a mean function that represents the expected output value at each input point, and a covariance function that captures the uncertainty in the predictions (Rasmussen & Williams, 2005).

GPR estimates the mean and covariance functions that best fit the training data, based on a prior distribution over functions. This prior distribution is typically a zero-mean Gaussian process with a kernel function that captures assumptions about the smoothness and shape of the underlying function. To make predictions on new data points, GPR uses Bayes' rule to compute the posterior distribution over functions, given the observed data. The mean function of the posterior distribution represents the predicted output value at each input point, while the covariance function captures the uncertainty in the predictions (Rasmussen & Williams, 2005).

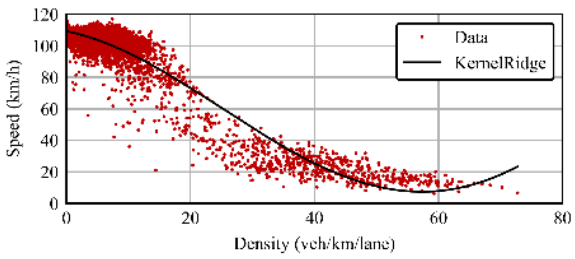
3.2.2 Hyperparameter optimisation

Hyperparameter optimization is an important aspect of non-parametric regression. In order to avoid overfitting, it is important to find the right balance between model complexity and generalization (Bode & Ronchi, 2019). Overfitting occurs when the model is too complex and fits the noise in the data rather than the underlying signal. Unfitting, on the other hand, occurs when the model is too simple and fails to capture the complexity of the underlying signal.

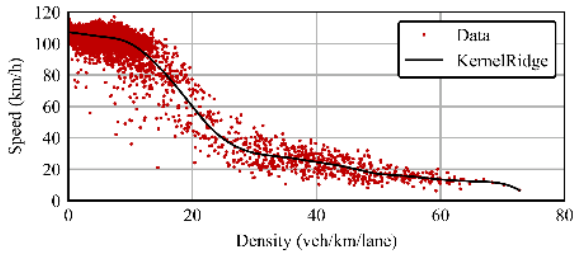
Overfitting and unfitting (variance and bias)

Overfitting and underfitting, also known as high variance and high bias, respectively, are common problems in machine learning regression models. Both of these issues can lead to poor regressions (Hastie et al., 2013).

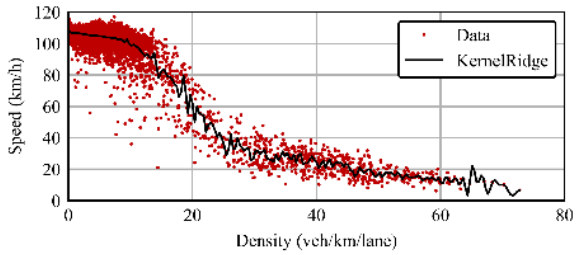
In the context of regression, an overfit model can result in a very low training error (i.e., the model predicts the training data very accurately), but a high test error (i.e., the model performs poorly on new data). On the other hand, an underfit model can result in a high training error and a high test error. To address these issues, illustrated in Figure 5, it is important to strike a balance between model complexity and the ability to capture the underlying patterns in the data. One approach is to use cross-validation.



a) underfitting



b) adequate fitting



c) overfitting

Figure 5. An example of underfitting (a), good fitting (b) and overfitting (c) kernel ridge regressions. The under-fitted regression fails to capture the trend of the data accurately, while the over-fitted model has incorporated noise of the dataset in the regression.

Cross-validation

Cross-validation is a widely used technique in machine learning that is used to evaluate the performance of a model and to select the best hyperparameters. The basic idea behind cross-validation is to split the data into multiple subsets, or folds, and to train the model on some data and test it on the remaining data. This process is repeated for all possible combinations of training and testing sets, and the performance metrics are averaged across all folds (Hastie et al., 2013).

The most common type of cross-validation is k-fold cross-validation, where the data is divided into k equal-sized folds. The model is trained on k-1 folds and tested on the remaining fold. This process is repeated k times, with each fold serving as the testing set once. The performance metrics are then averaged across all k folds to obtain an estimate of the model's performance (Hastie et al., 2013).

This method provides a way to estimate the model performance on unseen data and can therefore be used to optimise the hyperparameters. Here, a five-fold cross-validation is performed, as illustrated in Figure 6. The weighted sum of squared errors is used as a performance metric.

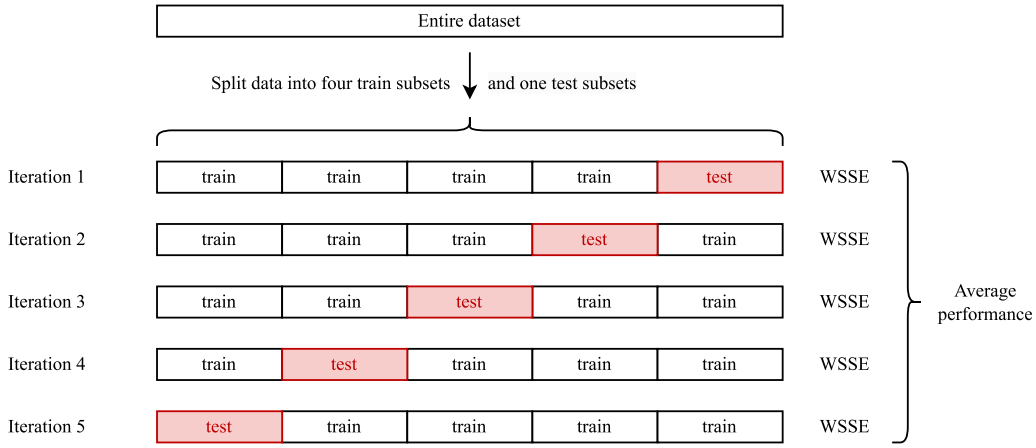


Figure 6. Schematic representation of the five-fold cross-validation as applied in this report. The performance metric is the average of the five weighted sums of squared errors (WSSE).

To obtain the optimal parameters, we have applied a grid search. Grid search is a hyperparameter tuning technique that involves defining possible hyperparameter values and simply trying all combinations. The combination that results in the best performance is selected as the optimal one. Grid search is a simple and exhaustive method for hyperparameter tuning, but it can be computationally expensive for large hyperparameter spaces.

3.2.3 Illustration of the non-parametric regression methods

In this section, we first discuss the optimal hyperparameters of the methods and then demonstrate the obtained regressions.

The three methods above are employed to find a regression for the evacuation traffic dynamics that occurred during the 2019 Kincade Fire (Rohaert et al., 2023). The data is openly accessible on Zenodo (Rohaert et al., 2022).

Before fitting the regression curves, the optimal values for the hyperparameters have been determined. Both for KRR and for SVR, a grid search has been performed. Table 3 shows both the considered values and the optimal values.

Table 3. Optimisation of the hyperparameters of KRR and SVR through a grid search

Regression technique	Hyperparameter	Values in the grid search	Optimal values
KRR	*alpha*	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1} and 10^{0}	10^{-3}
	gamma	10^{-3}, 10^{-2}, 10^{-1}, 10^{0} and 10^{1}	10^{-2}
SVR	*C*	10^{2}, 10^{3}, 10^{4}, 10^{5} and 10^{6}	10^{3}
	epsilon	10^{-7}, 10^{-6}, 10^{-5}, 10^{-4} and 10^{-3}	10^{-5}

The GPR is obtained using the *GaussianProcessRegressor* of the Scikit-learn library (*sklearn*) in Python (Pedregosa et al., 2011). When using this class, the kernel parameters are optimised internally by maximising the maximize the log-marginal likelihood. Therefore, no grid search is required.

From the grid search, it can be concluded that the regression fits best when *alpha* equals 0.001 and *gamma* equals 0.01. The result of the regression (using the entire database and the optimal hyperparameters) is shown in Figure 7.

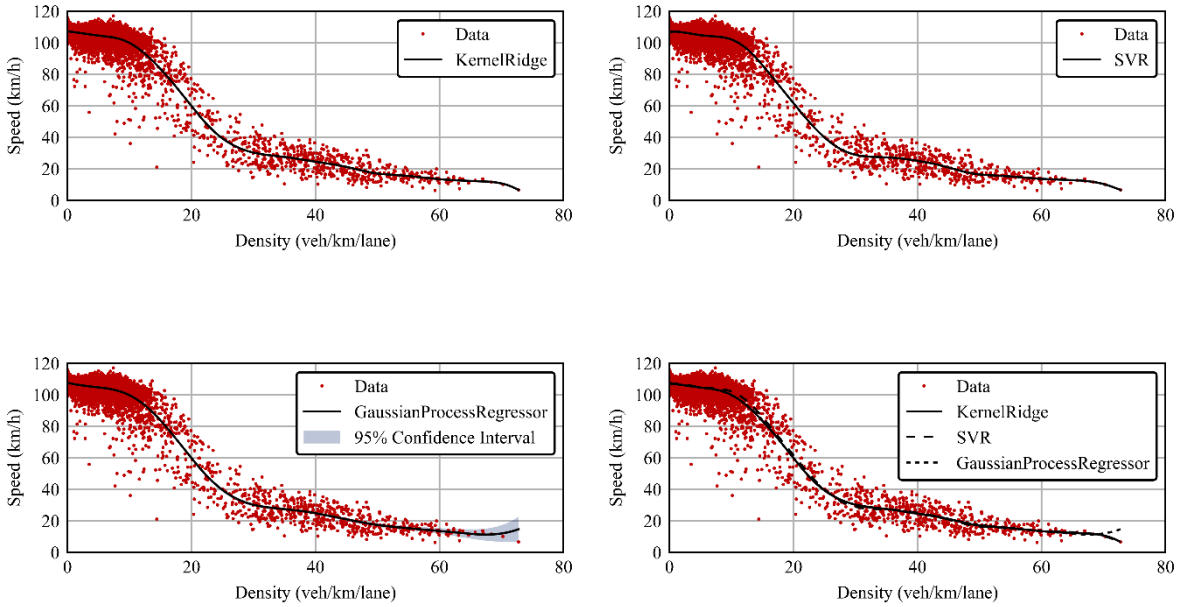


Figure 7. Kernel Ridge Regression (upper left), Support Vector Regression (upper right), Gaussian Process Regression (lower left) and a comparison of the three methods (lower right) illustrated on the traffic dynamics data from the evacuation during the 2019 Kincade Wildfire.

The three methods lead to very similar regressions, as shown in Figure 7. The agreement between the different curves could potentially be improved even further by refining the grid search for the hyperparameters of KRR and SVR.

Since GPR is a probabilistic approach to regression, it can estimate the uncertainty in its predictions by computing the covariance function between the predicted output and the observed output at each input point. This covariance function can be used to construct a confidence band around the mean predicted output (see Figure 7). The width of the confidence band depends on the uncertainty of the predictions and is larger in regions where the data is sparse or noisy. However, GPR is considerably more computationally expensive than the other two methods. All regressions are performed on one computer. For this dataset, the KRR took 10.2 seconds, SVR took 7.7 seconds and GPR took 4 hours and 41 minutes. Note that for KRR and SVR, a cross validation process was required to predefine the optimal value of the hyperparameters. Also note that calculation time is not a perfect measure of computational cost.

4 Discussion

This interim report presents two main contributions, a physically based real-time approach to visualize smoke and a review of data treatment concerning the relationships between speed, flow and density.

Regarding smoke modelling/visualization, the benefit of the presented approach is less simplifications to the radiation transport equation in terms of the interaction between light and fire induced smoke, specifically when treating absorption and scattering as separate physical processes and not as one extinction process. The results demonstrate that smoke properties play a significant role in how light and smoke interacts.

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, including wildfire scenarios. By using the froxel approach the computational cost is screen resolution independent while avoiding visual discontinuities at hard edges like screen-based approaches often show. Further relatively cheap enhancements to visual quality can be made using temporal upscaling techniques, something intended for future work.

In terms of validation, major work still lies ahead. More data is needed, both in terms of experimental smoke properties data, but also in terms of experiments involving humans and subjective visibility. The use of virtual reality in combination with realistic rendering of smoke will hopefully be one important piece in gaining more knowledge, as that enables data gathering in relatively safe environment.

There are two main limitations to the presented algorithm. Firstly, only single scattering of light is considered; light that is not scattered along a view aligned ray is "extinct". This is not energy conserving and will cause large visual discrepancies when the participating media has a high single scattering albedo and high optical density, such as clouds or mist. The single-scattering approximation can however be useful in cases with either low optical density or low single - scattering albedo.

Secondly, scattering of light on opaque surfaces, so called global illumination, does not occur except for the direct contribution towards the viewer. Again, this is not energy conserving and can result in under-illumination of the smoke, e.g., from a secondary bounce on the floor that would illuminate the smoke from below.

Previous related work (Rubini et al., 2007; Zhang, 2010) did consider both of these phenomena. However, this was only possible due to the non-real-time nature of the renderer. While it is true that available computational resources have increased significantly since, the techniques used are still considered too resource intensive for real-time applications such as virtual reality where the target framerate is often 90 frames per second or above. Global illumination approximations suitable for real-time applications are in use (Kaplanyan, 2009), they are still however an active topic of research. The light propagation volumes used for global illumination approximation (Kaplanyan, 2009), which are similar to finite volume radiation methods, could also be used to include multiple scattering, though computational resources might limit the amount of iterations performed per frame and temporal accumulation might be necessary.

Another current limit is the fact that only point, spot and directional (sun) lights are implemented. This is due to the fact that the light intensity from these light sources to a point in space can efficiently be calculated, while arbitrarily shaped area or volumetric lights are significantly more complex to calculate. This means that objects such as light from firebrands or flames will not

produce in-scattering, only extinction. This is also the reason why the current implementation cannot consider the flame as a light source correctly. However, this is something that could potentially also be solved in the future using light propagation volumes or similar techniques.

Concerning the reported methodologies to analyse traffic evacuation data, this document provides an overview of two types of regression analysis, parametric and non-parametric regression. It explains that parametric regression requires a pre-specified functional form or equation with a fixed number of parameters, while non-parametric regression estimates the function directly from the data, making it more flexible. The document also discusses three popular machine learning algorithms for non-parametric regression: Kernel Ridge Regression, Support Vector Regression, and Gaussian Process Regression. One of the key findings is the importance of hyperparameter optimization in non-parametric regression to improve the performance of these algorithms. Overall, the document highlights the differences between these approaches and their applications in analysing complex data. Non-parametric regressions allow to compare different curves and can provide estimations of certain key parameters, such as the free-flow speed and the capacity.

The functionality developed represents an important step as it allows the model to better represent a key difference between wildfires and other major incidents – the secondary impact of smoke upon locations remote from the fire front. The development has two elements – the representation of the smoke and then its impact on evacuating vehicles. The primary focus here is on the obscuration of the smoke on the driver, rather than addressing any other emotional or cognitive impacts. The inclusion of this alone may provide model estimates more sensitive to wildfire conditions and produce more naturalistic visualisations within virtual environments – enhancing the model projects and experimental/training applications.

References

Adamson, D. (1975). The role of multiple scattering in one-dimensional radiative transfer. In *Unknown*. https://ui.adsabs.harvard.edu/abs/1975rmso.rept.....A

Assael, M. J., & Kakosimos, K. E. (2010). *Fires, explosions, and toxic gas dispersions: Effects calculation and risk analysis*. CRC Press.

Benichou, N., Adelzadeh, M., Singh, J., Gomaa, I., Elsagan, N., Kinateder, M., Ma, C., Gaur, A., Bwala, A., & Sultan, M. (2021). *National Guide for Wildland-urban interface fires: Guidance on hazard and exposure assessment, property protection, community resilience and emergency planning to minimize the impact of wildland-urban interface fires*. National Research Council of Canada.

Beverly, J. L., & Bothwell, P. (2011). Wildfire evacuations in Canada 1980–2007. *Natural Hazards*, *59*(1), 571–596.

Blackmore, D. R., Herman, M. N., & Woodward, J. L. (1982). Heavy gas dispersion models. *Journal of Hazardous Materials*, *6*(1–2), Article 1–2. https://doi.org/10.1016/0304-3894(82)80036-8

Blasi, P., Le Saec, B., & Schlick, C. (1993). A Rendering Algorithm for Discrete Volume Density Objects. *Computer Graphics Forum*, *12*(3), 201–210. https://doi.org/10.1111/1467-8659.1230201

Blinn, J. F. (1982). Light reflection functions for simulation of clouds and dusty surfaces. *ACM SIGGRAPH Computer Graphics*, *16*(3), 21–29. https://doi.org/10.1145/965145.801255

Bode, N. W. F., & Ronchi, E. (2019). Statistical Model Fitting and Model Selection in Pedestrian Dynamics Research. *Collective Dynamics*, *4*, A20. https://doi.org/10.17815/CD.2019.20

Bond, T. C., & Bergstrom, R. W. (2006). Light Absorption by Carbonaceous Particles: An Investigative Review. *Aerosol Science and Technology*, *40*(1), 27–67. https://doi.org/10.1080/02786820500421521

Carlsson, A.-K., Wadman, C., Lönnermark, A., Ingason, H., & Andersson, P. (2007). *Visualization of Measurement Data from Tunnel Fire*. http://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-4763

Clarberg, P., Kallweit, S., Kolb, C., Kozlowski, P., He, Y., Wu, L., Liu, E., Bitterli, B., & Pharr, M. (2022). *Real-Time Path Tracing and Beyond*.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Duvenhage, B., Bouatouch, K., & Kourie, D. G. (2013). Numerical verification of bidirectional reflectance distribution functions for physical plausibility. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, 200–208. https://doi.org/10.1145/2513456.2513499

Elhokayem, L. (2022). *Visibility in Smoke With Different Extinction Coefficients* (No. 5665). Departement of Fire Safety Engineering, Lund University.

Finney, M. A. & others. (1998). *FARSITE, Fire Area Simulator–model development and evaluation* (Vol. 3). US Department of Agriculture, Forest Service, Rocky Mountain Research Station Ogden, UT.

Forney, G. P. (2013). Smokeview (Version 6) - A Tool for Visualizing Fire Dynamics Simulation Data Volume II: Technical Reference Guide. In *NIST*. Glenn P. Forney.

Han, J., Ki, H., & Oh, K. (2007). GPU Ray Marching for Real-Time Rendering of Participating Media. *2007 International Conference on Computational Science and Its Applications (ICCSA 2007)*, 499–504. https://doi.org/10.1109/ICCSA.2007.46

Hanna, S. R., Brown, M. J., Camelli, F. E., Chan, S. T., Coirier, W. J., Hansen, O. R., Huber, A. H., Kim, S., & Reynolds, R. M. (2006). Detailed Simulations of Atmospheric Flow and Dispersion in Downtown Manhattan: An Application of Five Computational Fluid Dynamics Models. *Bulletin of the American Meteorological Society*, *87*(12), 1713–1726. https://doi.org/10.1175/BAMS-87-12-1713

Hanna, S. R., Hansen, O. R., Ichard, M., & Strimaitis, D. (2009). CFD model simulation of dispersion from chlorine railcar releases in industrial and urban areas. *Atmospheric Environment*, *43*(2), 262–270. https://doi.org/10.1016/j.atmosenv.2008.09.081

Hastie, T., Tibshirani, R., & Friedman, J. (2013). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Henyey, L. G., & Greenstein, J. L. (1941). Diffuse radiation in the Galaxy. *The Astrophysical Journal*, *93*, 70–83. https://doi.org/10.1086/144246

Hill, S., McAuley, S., Jover, C., Lachambre, S., Pesce, A., Wu, X.-C., Cordes, R., Hery, C., Hillaire, S., Hoffman, N., Jiménez, J., Karis, B., Lagarde, S., Lobl, D., & Villemin, R. (2016). Physically based shading in theory and practice. *ACM SIGGRAPH 2016 Courses*, 1. https://doi.org/10.1145/2897826.2927353

Hillaire, S. (2016). *Physically Based Sky, Atmosphere and Cloud Rendering in Frostbite*. Electronic Arts.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1), 55–67. https://doi.org/10.1080/00401706.1970.10488634

Intini, P., Wahlqvist, J., Wetterberg, N., & Ronchi, E. (2022). Modelling the impact of wildfire smoke on driving speed. *International Journal of Disaster Risk Reduction*.

Jin, T. (1978). Visibility through fire smoke. *Journal of Fire and Flammability*, *9*(2), Article 2.

Jin, T. (1997). Studies On Human Behavior And Tenability In Fire Smoke. *Fire Safety Science*, *5*, 3–21.

Kang, K., & Macdonald, H. (2005). Modeling smoke visibility in CFD. *Fire Safety Science*, *8*, 1265–1276. https://doi.org/10.3801/IAFSS.FSS.8-1265

Kaplanyan, A. (2009). Light Propagation Volumes in CryEngine 3. *SIGGRAPH 2009*.

Kniss, J., Premoze, S., Hansen, C., & Ebert, D. (2002). Interactive translucent volume rendering and procedural modeling. *IEEE Visualization, 2002. VIS 2002.*, 109–116. https://doi.org/10.1109/VISUAL.2002.1183764

Koch, N., Niewiadomski, A. P., & Wrona, P. (2021). Influence of Light Wavelengths on Visibility in Smoke during a Tunnel Fire. *Sustainability*, *13*(21), 11599. https://doi.org/10.3390/su132111599

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, *2*(2), 164–168. https://doi.org/10.1090/qam/10666

Lovreglio, R., Ronchi, E., Maragkos, G., Beji, T., & Merci, B. (2016). A dynamic approach for the impact of a toxic gas dispersion hazard considering human behaviour and dispersion modelling. *Journal of Hazardous Materials*, *318*, 758–771. https://doi.org/10.1016/j.jhazmat.2016.06.015

Manca, D., & Brambilla, S. (2010). Complexity and uncertainty in the assessment of the Viareggio LPG railway accident. *Journal of Loss Prevention in the Process Industries*, *23*(5), 668–679. https://doi.org/10.1016/j.jlp.2010.07.007

Marquardt, D. W. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030

McGrattan, K. B., McDermott, R. J., Weinschenk, C. G., & Forney, G. P. (2013). Fire Dynamics Simulator Users Guide, Sixth Edition. *NIST*. https://www.nist.gov/publications/fire-dynamics-simulator-users-guide-sixth-edition

Mell, W. E., Manzello, S. L., Maranghides, A., Butry, D., & Rehm, R. G. (2010). The wildland–urban interface fire problem – current approaches and research needs. *International Journal of Wildland Fire*, *19*(2), 238. https://doi.org/10.1071/WF07131

Mitchell, H., Gwynne, S., Ronchi, E., Kalogeropoulos, N., & Rein, G. (2023). Integrating wildfire spread and evacuation times to design safe triggers: Application to two rural communities using PERIL model. *Safety Science*, *157*, 105914. https://doi.org/10.1016/j.ssci.2022.105914

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

Mulholland, G. W., & Croarkin, C. (2000). Specific extinction coefficient of flame generated smoke. *Fire and Materials*, *24*(5), 227–230. https://doi.org/10.1002/1099-1018(200009/10)24:5<227::AID-FAM742>3.0.CO;2-9

Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., & Vapnik, V. (1997). Predicting time series with support vector machines. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *Artificial Neural Networks—ICANN'97* (Vol. 1327, pp. 999–1004). Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0020283

Patterson, E. M., & McMahon, C. K. (1984). Absorption characteristics of forest fire particulate matter. *Atmospheric Environment (1967)*, *18*(11), 2541–2551. https://doi.org/10.1016/0004-6981(84)90027-1

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pérez, F., Pueyo, X., & Sillion, F. X. (1997). Global Illumination Techniques for the Simulation of Participating Media. In J. Dorsey & P. Slusallek (Eds.), *Rendering Techniques '97* (pp. 309–320). Springer. https://doi.org/10.1007/978-3-7091-6858-5_28

Pharr, M., Jakob, W., & Humphreys, G. (2016). *Physically Based Rendering: From Theory to Implementation* (3rd ed.). Morgan Kaufmann Publishers Inc.

Pontiggia, M., Derudi, M., Alba, M., Scaioni, M., & Rota, R. (2010). Hazardous gas releases in urban areas: Assessment of consequences through CFD modelling. *Journal of Hazardous Materials*, *176*(1), 589–596. https://doi.org/10.1016/j.jhazmat.2009.11.070

Pontiggia, M., Derudi, M., Busini, V., & Rota, R. (2009). Hazardous gas dispersion: A CFD model accounting for atmospheric stability classes. *Journal of Hazardous Materials*, *171*(1), 739–747. https://doi.org/10.1016/j.jhazmat.2009.06.064

Pontiggia, M., Landucci, G., Busini, V., Derudi, M., Alba, M., Scaioni, M., Bonvicini, S., Cozzani, V., & Rota, R. (2011). CFD model simulation of LPG dispersion in urban areas. *Atmospheric Environment*, *45*(24), 3913–3923. https://doi.org/10.1016/j.atmosenv.2011.04.071

Qu, X., Wang, S., & Zhang, J. (2015). On the fundamental diagram for freeway traffic: A novel calibration approach for single-regime models. *Transportation Research Part B: Methodological*, *73*, 91–102. https://doi.org/10.1016/j.trb.2015.01.001

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press. https://doi.org/10.7551/mitpress/3206.001.0001

Rehm, R. G., & McDermott, R. J. (2009). Fire-Front Propagation Using the Level Set Method. *NIST*. https://www.nist.gov/publications/fire-front-propagation-using-level-set-method

Rodean, H. C. (1996). Stochastic Lagrangian Models of Turbulent Diffusion. *Meteorological Monographs*, *26*(48), 1–84. https://doi.org/10.1175/0065-9401-26.48.1

Rohaert, A., Kuligowski, E. D., Ardinge, A., Wahlqvist, J., Gwynne, S. M. V., Kimball, A., Bénichou, N., & Ronchi, E. (2023). Traffic dynamics during the 2019 Kincade wildfire evacuation. *Transportation Research Part D: Transport and Environment*, *116*, 103610. https://doi.org/10.1016/j.trd.2023.103610

Rohaert, A., Kuligowski, E. D., Ardinge, A., Wahlqvist, J., Gwynne, S. M. V., Kimball, A., Noureddine Bénichou, & Ronchi, E. (2022). *Dataset of traffic dynamics during the 2019 Kincade Wildfire Evacuation* (Version 1) [Data set]. Zenodo. https://doi.org/10.5281/ZENODO.7410114

Ronchi, E., & Gwynne, S. (2019). Computational Evacuation Modeling in Wildfires. In S. L. Manzello (Ed.), *Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires* (pp. 1–10). Springer International Publishing. https://doi.org/10.1007/978-3-319-51727-8_121-1

Ronchi, E., Gwynne, S. M. V., Rein, G., Intini, P., & Wadhwani, R. (2019). An open multi-physics framework for modelling wildland-urban interface fire evacuations. *Safety Science*, *118*, 868–880. https://doi.org/10.1016/j.ssci.2019.06.009

Ronchi, E., Wahlqvist, J., Gwynne, S., Kinateder, M., Rein, G., Mitchell, H., Benichou, N., Ma, C., & Kimball, A. (2020). *WUI-NITY: a platform for the simulation of wildland-urban interface fire evacuation* (p. 80). Fire Protection Research Foundation.

Ronchi, E., Wahlqvist, J., Rohaert, A., Ardinge, A., Gwynne, S. M. V., Rein, G., Mitchell, H., Kalogeropoulos, N., Kinateder, M., Benichou, N., Kuligowski, E., Westbury, A., & Kimball, A. (2021). *WUI-NITY 2: The integration, verification, and validation of the wildfire evacuation platform WUI-NITY*. Fire Protection Research Foundation.

Rothermel, R. C. (1972). A mathematical model for predicting fire spread in wildland fuels. *Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Intermountain Forest and Range Experiment Station. 40 p.*, *115*. https://www.fs.usda.gov/research/treesearch/32533

Rubini, P. A., Zhang, Q., & Moss, J. B. (2007). Simulation of visibility in smoke laden environments. *5th International Seminar on Fire and Explosion Hazards*, 23–27.

Shibata, Y., & Sakuraba, A. (2019). Driving Simulator System for Disaster Evacuation Guide Based on Road State Information Platform. *International Conference on Network-Based Information Systems*, 377–384.

Siddiqui, M., Jayanti, S., & Swaminathan, T. (2012). CFD analysis of dense gas dispersion in indoor environment for risk assessment and risk mitigation. *Journal of Hazardous Materials*, *209–210*, 177–185. https://doi.org/10.1016/j.jhazmat.2012.01.007

Staubli, O., Sigg, C., Peikert, R., Gubler, D., & Gross, M. (2005). Volume rendering of smoke propagation CFD data. *VIS 05. IEEE Visualization, 2005.*, 335–341. https://doi.org/10.1109/VISUAL.2005.1532813

Suo-Anttila, J., Gill, W., Gritzo, L., & Blake, D. (2005). An evaluation of actual and simulated smoke properties. *Fire and Materials*, *29*(2), 91–107. https://doi.org/10.1002/fam.875

Szirmay-Kalos, L., Sbert, M., & Umenhoffer, T. (2005). Real-Time Multiple Scattering in Participating Media with Illumination Networks. *Rendering Techniques*, 277–282.

Tatarchuk, N., Hillaire, S., Stachowiak, T., Bowles, H., Wang, B., Silvennoinen, A., & Timonen, V. (2015). Advances in real time rendering, part I. *ACM SIGGRAPH 2015 Courses*, 1. https://doi.org/10.1145/2776880.2787701

Tatarchuk, N., Valient, M., Brainerd, W., Wronski, B., Sikachev, P., & Bucci, J.-N. (2014). Advances in real-time rendering in games, part II (full text not available). *ACM SIGGRAPH 2014 Courses*, 1. https://doi.org/10.1145/2614028.2615456

Tymstra, C., Bryce, R., Wotton, B., Taylor, S., Armitage, O., & others. (2010). Development and structure of Prometheus: The Canadian wildland fire growth simulation model. *Natural Resources Canada, Canadian Forest Service, Northern Forestry Centre, Information Report NOR-X-417.(Edmonton, AB).*

Ummenhoffer, T., & Szirmay-Kalos, L. (2005). *Real-Time Rendering of Cloudy Natural Phenomena with Hierarchical Depth Impostors*. https://doi.org/10.2312/egs.20051025

Unity Technologies. (n.d.). *Real-Time 3D Development Platform & Editor | Unity*. Retrieved November 16, 2022, from https://unity.com/products/unity-platform

Wahlqvist, J. (2018). *Visualization of fires in VR* (LUTVDG/TVBB—3222-SE 3222).

Wahlqvist, J., Ronchi, E., Gwynne, S. M. V., Kinateder, M., Rein, G., Mitchell, H., Bénichou, N., Ma, C., Kimball, A., & Kuligowski, E. (2021). The simulation of wildland-urban interface fire evacuation: The WUI-NITY platform. *Safety Science*, *136*, 105145. https://doi.org/10.1016/j.ssci.2020.105145

Weinert, D. W., Cleary, T., Mulholland, G. W., & Beever, P. F. (2003). Light Scattering Characteristics and Size Distribution of Smoke and Nuisance Aerosols. *NIST.*

https://www.nist.gov/publications/light-scattering-characteristics-and-size-distribution-smoke-and-nuisance-aerosols
Wetterberg, N., Ronchi, E., & Wahlqvist, J. (2021). Individual Driving Behaviour in Wildfire Smoke. *Fire Technology*, *57*(3), 1041–1061.
Widmann, J. F. (2003). Evaluation of the planck mean absorption coefficients for radiation transport through smoke. *Combustion Science and Technology*, *175*(12), 2299–2308. https://doi.org/10.1080/714923279
Wriedt, T. (2012). Mie theory: A review. *The Mie Theory: Basics and Applications*, 53–71.
Zhang, Q. (2010). *Image based analysis of visibility in smoke laden environments* [Department of Engineering, The University of Hull]. https://hydra.hull.ac.uk/resources/hull:5806
Zhang, Q., & Rubini, P. A. (2011). Modelling of light extinction by soot particles. *Fire Safety Journal*, *46*(3), 96–103. https://doi.org/10.1016/j.firesaf.2010.11.002