

UMA ABORDAGEM PARA SIMULAÇÃO DA DISPERSÃO DE EFEITOS  
SENSORIAIS COM DINÂMICA DE FLUIDOS COMPUTACIONAL

Renato de Oliveira Rodrigues

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:

Joel André Ferreira dos Santos

# Uma Abordagem para Simulação da Dispersão de Efeitos Sensoriais com Dinâmica de Fluidos Computacional

Dissertação de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Renato de Oliveira Rodrigues

Aprovada por:

---

Presidente, Professor D.Sc. Joel André Ferreira dos Santos (CEFET/RJ)

---

Professor D.Sc. Diego Nunes Brandão (CEFET/RJ)

---

Professor D.Sc. Glauco Fiorott Amorim (CEFET/RJ)

---

Professor D.Sc. Celso Alberto Saibel Santos (UFES)

---

Professor D.Sc. Georgita Ghinea (Brunel University)

Rio de Janeiro,  
Outubro de 2023

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

R696 Rodrigues, Renato de Oliveira

Uma abordagem para simulação da dispersão ao de efeitos sensoriais com dinâmica de fluidos computacional / Renato de Oliveira Rodrigues. — 2023.

53f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2023.

Bibliografia : f. 50-53

Orientador: Joel André Ferreira dos Santos

1. Fluidodinâmica computacional. 2. Hidrodinâmica de Partículas Suavizadas. 3. Multímídia interativa. I. Santos, Joel André Ferreira dos (Orient.). II. Título.

CDD 530.150285

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

## RESUMO

### Uma Abordagem para Simulação da Dispersão de Efeitos Sensoriais com Dinâmica de Fluidos Computacional

Renato de Oliveira Rodrigues

Orientadores:

Joel André Ferreira dos Santos

Resumo da Dissertação submetida ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

A inclusão de efeitos sensoriais em aplicações multimídia tem o potencial de aumentar a Qualidade da Experiência (QoE) e melhorar a imersão do usuário. No entanto, criar tais aplicações apresenta desafios decorrentes da necessidade de controlar a renderização de efeitos sensoriais no ambiente físico juntamente com a apresentação de efeitos sensoriais no ambiente virtual e a constante mudança de estado do efeito sensorial de acordo com a interação do usuário. Técnicas de dinâmica dos fluidos computacional podem ser usadas para simular os efeitos sensoriais em um ambiente virtual e usar os dados gerados para criar aplicações mais imersivas. Neste trabalho propomos uma abordagem usando o método Smoothed Particle Hydrodynamics (SPH) que permite renderizar uma variedade de efeitos sensoriais, como borrifo de água, vento, neblina, cheiro, etc., em uma aplicação interativa onde os usuários podem explorar os efeitos sensoriais em diferentes cenários em tempo real.

Palavras-chave:

Dinâmica de fluidos computacional, SPH, Mulsemídia.

Rio de Janeiro,

Outubro de 2023

## ABSTRACT

Uma Abordagem para Simulação da Dispersão de Efeitos Sensoriais com Dinâmica de Fluidos Computacional

Renato de Oliveira Rodrigues

Advisors:

Joel André Ferreira dos Santos

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

The inclusion of sensory effects in multimedia application has the potential to increase the Quality of Experience (QoE) and improve user's immersion. However, authoring such applications presents challenges arising from the need to control the rendering of sensory effects in the physical environment along with the presentation of sensory effects in the virtual environment and the constantly changing sensory effect's state according to user interaction. Computational fluid dynamics techniques can be used to simulate the sensory effects in a virtual environment and use the generated data to create more immersive applications. In this work we propose an approach using the SPH method allowing to render a variety of sensory effects, such as water spray, wind, fog, scent, etc., in an interactive application where the users can explore the sensory effects in different scenarios in real-time.

Key-words:

Computational Fluid Dynamics, SPH, Mulsemmedia.

Rio de Janeiro,

Outubro de 2023

## Sumário

<b>I</b>	<b>Introdução</b>	<b>11</b>
I.1	Questão de Pesquisa	12
I.2	Contribuições	13
I.3	Estrutura	13
<b>II</b>	<b>Simulação de Fluidos</b>	<b>14</b>
II.1	Métodos Baseados em Malha e Baseados em Partículas	14
II.2	Simulação de Fluidos Incompressíveis	16
II.3	Simulação de Fluidos Gasosos	18
<b>III</b>	<b>Trabalhos Relacionados</b>	<b>21</b>
III.1	Simulação de fluidos computacional	21
III.2	Controle de Efeitos Sensoriais em Mulsemídia	22
<b>IV</b>	<b>Simulação de Efeitos Sensoriais</b>	<b>25</b>
IV.1	Arquitetura	25
IV.2	Simulação	26
IV.3	Parâmetros	28
IV.4	Sensores e Atuadores	29
IV.5	Configuração da Simulação	30
IV.6	Módulo de visualização	34
IV.7	Modo <i>headless</i>	35
<b>V</b>	<b>Casos de Uso de Simulação</b>	<b>36</b>
V.1	Descrição do ambiente	36
V.2	Descrição dos dispositivos	37
V.3	Descrição da simulação	39
V.4	Cenário de simulação	40
<b>VI</b>	<b>Considerações Finais</b>	<b>49</b>



## Lista de Figuras

II.1	Representação de uma malha bidimensional	15
II.2	Representação de partícula. Fonte: [Giorgi, 2018]	15
II.3	As três funções de <i>kernel</i> $W_{poly6}$ , $W_{spiky}$ and $W_{viscosity}$ (da esquerda para direita) usadas nas simulações. As linhas grossas apresentam as funções, as linhas finas seus gradientes na direção do centro e as linhas tracejadas seu Laplaciano. Fonte:[Müller et al., 2003]	17
II.4	(a) Abordagem padrão de preencher o ambiente com partículas de ar (azul). (b) Abordagem de Ren et al. [2016] remove as partículas de ar e adiciona forças de compensação atuando sobre as partículas de gás (vermelho). Fonte: [Ren et al., 2016].	18
III.1	Arquitetura do padrão MPEG-V e cenários de transmissão de dados[Yoon et al., 2015]	23
IV.1	Arquitetura do FluidSense.	26
V.1	Visão superior do ambiente descrito na configuração.	37
V.2	Visão das partículas emitidas pelo atuador.	42
V.3	Perspectiva aérea da simulação.	43
V.4	Visão das partículas atrás do obstáculo.	44
V.5	Perspectiva aérea da simulação com obstáculo.	44
V.6	Visão das partículas emitidas pelos atuadores.	46
V.7	Perspectiva aérea da simulação.	47



## Lista de Tabelas

IV.1 Parâmetros comuns a ambos os métodos SPH.	28
IV.2 Parâmetros específicos à simulação de fluidos gasosos.	29
V.1 Parâmetros usados para os testes de simulação.	36
V.2 Fragmento do arquivo de saída da simulação.	48

## Lista de Abreviações

3DOF	Three Degrees Of Freedom	12
6DOF	Six Degrees Of Freedom	11
CFD	Computational Fluid Dynamics	12, 13, 21, 26, 49
GUI	Graphical User Interface	23
HMD	Head-Mounted Display	21
NCL	Nested Context Language	24
QOE	Quality Of Experience	11, 49
SEMP	Sensory Effect Media Player	23
SESIM	Sensory Effect Simulator	23
SEVINO	Sensory Effect Video Annotation	23
SPH	Smoothed Particle Hydrodynamics	15, 16, 17, 18, 26, 27, 28, 29, 30, 49, 50
VR	Virtual Reality	21
YAML	YAML Ain't Markup Language	30

## Capítulo I Introdução

Aplicações multimídia que permitem a apresentação de efeitos sensoriais (por exemplo, aroma, temperatura, etc.) sincronizados com outros tipos tradicionais de objetos de mídia (por exemplo, áudio, vídeo, imagens) são denominadas aplicações mulsemídia [Ghinea et al., 2014]. Tais aplicações têm o potencial de melhorar a qualidade da experiência - do inglês Quality of Experience (QoE) - dos usuários ao consumir o conteúdo multimídia. Visando demonstrar esse potencial de melhoria da QoE, estudos como [Kahol et al., 2006; Apostolopoulos et al., 2012; Covaci et al., 2019; Comşa et al., 2019] apresentam aplicações interativas enriquecidas com efeitos sensoriais resultando em uma melhor QoE. Outros como [Tortell et al., 2007] indicam um aumento na retenção de informações pelos usuários em aplicações com efeitos sensoriais. Por fim, estudos como [Rodrigues et al., 2019; Josué et al., 2018] focam em prover suporte para renderização de efeitos sensoriais em aplicações multimídia estendendo motores de apresentação tradicionais, ou provendo suporte para comunicação de características de efeitos entre dispositivos [Saleme and Santos, 2015].

A inclusão de efeitos sensoriais em aplicações multimídia, no entanto, não é uma tarefa simples. Atuadores, isto é, dispositivos que renderizam os efeitos sensoriais no ambiente onde a aplicação é executada, devem ser controlados pelo dispositivo de apresentação multimídia de forma que a percepção dos efeitos esteja em sincronia com o conteúdo audiovisual [Rodrigues et al., 2021]. Para lidar com este desafio, garantindo que um efeito de cheiro atinja o usuário no momento correto, Murray et al. [2017] garantem uma distância fixa entre usuário e atuador pedindo que os mesmos coloquem sua cabeça em uma posição fixa (galão d'água ou outro suporte) enquanto experimentam com uma aplicação mulsemídia olfativa. Neste trabalho, comandos para o atuador são enviados antes do momento em que o efeito é renderizado para compensar o atraso na rede, tempo de processamento do atuador, e a dispersão do efeito. Uma abordagem similar é apresentada por Josué et al. [2018] que propõem estender motores de apresentação multimídia com a funcionalidade de preparar atuadores antes do momento de execução de um efeito.

Além dos problemas de atraso mencionados anteriormente, motores de apresentação mulsemídia devem renderizar os efeitos sensoriais de forma que sua intensidade esteja de acordo com o que é apresentado para o usuário. Suponha, por exemplo, uma aplicação Six Degrees of Freedom (6DOF) em que existe um aroma sendo emitido em uma determinada posição no ambiente virtual. Ao caminhar pelo ambiente virtual, a intensidade do aroma aumenta quando o usuário se aproxima

da fonte do aroma. Além disso, a composição do ambiente virtual influencia como o aroma se dispersa, apresentando intensidades variadas em diferentes regiões. Para aplicações Three Degrees of Freedom (3DOF), a intensidade do efeito varia de acordo com a direção da cabeça do usuário.

Uma primeira abordagem possível para resolver as questões acima seria medir o tempo gasto entre o momento em que um atuador é ativado até o momento em que o usuário percebe o efeito sensorial. De forma similar, autores de uma aplicação mulsemídia podem testar diferentes intensidades de efeito (no atuador) para definir a que melhor se adequa ao conteúdo apresentado para o usuário. Por fim, dependendo do efeito sensorial sendo renderizado, por exemplo, aroma e vento, desambiguação também é necessária para entender como um efeito influencia na dispersão do outro. Entretanto, entender a interação entre efeitos não é tão direto quanto medir seu atraso e intensidade na posição do usuário.

## I.1 Questão de Pesquisa

Conforme discutido em [Rodrigues et al., 2021], é importante entender como se propaga um efeito sensorial pelo ambiente onde a aplicação está sendo executada e isso ajuda na resolução dos problemas apresentados. Alguns efeitos sensoriais táteis e olfativos incluídos em aplicações se comportam como fluidos, tais como vento, aroma, borrifo d'água e névoa. Assim, de forma a entender como esses efeitos se dispersam em um determinado ambiente, como o local físico onde a aplicação será executada, é possível simulá-los computacionalmente como fluidos. A Dinâmica de Fluidos Computacional - do inglês Computational Fluid Dynamics (CFD) - implementa as equações de Navier-Stokes, que modelam matematicamente o movimento de boa parte dos fluidos que ocorrem na natureza [Wesseling, 2009].

Este trabalho, portanto, tem o objetivo de responder à seguinte pergunta de pesquisa:

- *Dado que um modelo de CFD permite a simulação de efeitos sensoriais, como podemos utilizar os dados dessa simulação para entender aspectos importantes sobre o comportamento dos efeitos sensoriais para que sejam reproduzidos em aplicações mulsemídia?*

Para auxiliar na resposta a essa pergunta, é proposta uma arquitetura construída sobre métodos de CFD para simular a propagação de efeitos sensoriais como vento, cheiro, borrifo d'água e névoa. Autores de aplicações mulsemídia podem utilizar a simulação para explorar diferentes cenários e explorar como os diferentes efeitos sensoriais se comportam em cada situação. Os dados da simulação podem ser usados para definir os parâmetros apropriados para um motor de apresentação mulsemídia controlar a intensidade do efeito de forma a corresponder ao que é apresentado na aplicação. Adicionalmente, desenvolvedores podem utilizar a simulação para construir ferramentas

para controlar os tempos de ativação dos efeitos sensoriais de forma que a apresentação dos efeitos seja percebida em sincronia com o conteúdo audiovisual.

A arquitetura proposta é validada usando um método de CFD baseado em partículas para simular diferentes configurações de ambientes e atuadores.

## **I.2 Contribuições**

As contribuições deste trabalho apresentadas nessa dissertação estão relacionadas às lacunas identificadas de simulação da propagação de efeitos sensoriais num ambiente de execução de aplicações multemídia. São esperadas as seguintes contribuições:

- A proposta de uma arquitetura para simulação de efeitos sensoriais que se comportam como fluidos;
- A instanciação da arquitetura com dois métodos de CFD baseados em partículas;
- A disponibilização de uma ferramenta para simulação de efeitos sensoriais em diferentes configurações de ambientes.

## **I.3 Estrutura**

Esta dissertação está estruturada conforme apresentado nos parágrafos a seguir.

Capítulo II apresenta os principais conceitos sobre simulação de fluidos que serão utilizados para simular os efeitos sensoriais.

Capítulo III apresenta os trabalhos relacionados focados no tempo de ativação de efeitos sensoriais e na intensidade de acordo com o conteúdo da aplicação.

Capítulo IV apresenta a simulação proposta neste trabalho.

Capítulo V apresenta alguns casos de uso para validar a simulação proposta.

Capítulo VI apresenta os resultados preliminares.

## Capítulo II Simulação de Fluidos

Conforme discutido anteriormente, alguns dos efeitos sensoriais táteis e olfativos se comportam como fluidos. É o caso de efeitos como vento, cheiro, borrifo d'água e névoa. Portanto, para entender como estes efeitos se dispersam em um determinado ambiente (seja o mundo virtual ou o local físico onde a aplicação está sendo renderizada), é possível simular computacionalmente esses fluidos. Os dados de simulação podem então ser usados para definir os parâmetros apropriados para renderizar os efeitos sensoriais de acordo com o que está sendo apresentado ao usuário. Em outras palavras, o dispositivo de apresentação será capaz de controlar os atuadores de acordo com o comportamento do fluido durante a execução da aplicação.

As seções a seguir apresentam uma breve introdução às técnicas computacionais de dinâmica dos fluidos usadas aqui para simular efeitos sensoriais.

### II.1 Métodos Baseados em Malha e Baseados em Partículas

As equações de Navier-Stokes modelam matematicamente o movimento da maioria dos fluidos que ocorrem na natureza [Wesseling, 2009]. A simplificação das equações para fluidos incompressíveis podem ser definidas como nas Equações II.1 e II.2.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (\text{II.1})$$

$$\rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v} \quad (\text{II.2})$$

onde  $\mathbf{v}$  é o campo da velocidade,  $\rho$  é o campo da densidade,  $p$  é o campo da pressão,  $\mathbf{g}$  é a gravidade e  $\mu$  é viscosidade do fluido.

A Equação II.1 é chamada de equação de continuidade. Esta equação define que a massa do sistema deve ser conservada. A Equação II.2 é chamada de equação de movimento. Esta equação define a aceleração do fluido, dadas as forças de pressão, gravidade e viscosidade aplicadas sobre o fluido. Dois métodos podem ser utilizados para resolver estas equações, o método baseado em malhas, e o método baseado em partículas.

Nos métodos baseados em malhas, a superfície do fluido é representada por uma malha. Cada célula da malha contém um campo vetorial das forças aplicadas sobre o fluido dentro de sua área. A

Figura II.1 representa uma visão bidimensional de uma malha, onde  $p$  é a pressão e  $u$  é a velocidade do fluido em cada parte da superfície.

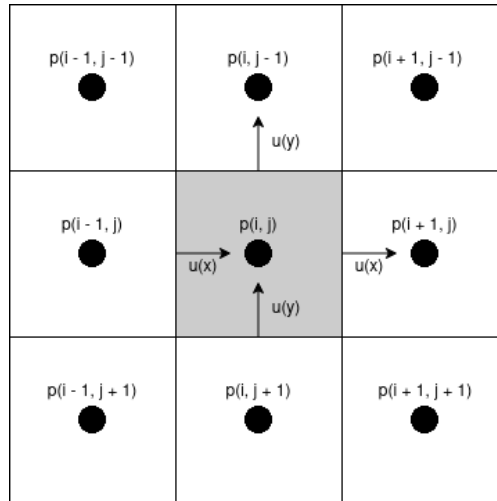


Figura II.1: Representação de uma malha bidimensional

Nos métodos baseados em partículas, o fluido é representado por um conjunto de partículas. O método SPH é um método baseado em partículas desenvolvido originalmente para aplicações astrofísicas [Lucy, 1977; Gingold and Monaghan, 1977]. No entanto, este método tem sido utilizado para simulações computacionais [Müller et al., 2003]. Cada partícula no sistema armazena os dados para sua representação, ou seja, as forças aplicadas sobre as partículas e as forças que causam a integração com outras partículas dada um alcance representado por uma função de *kernel*. A Figura II.2 apresenta um conjunto de partículas dispersas no espaço e a representação da função de *kernel* de vizinhança.

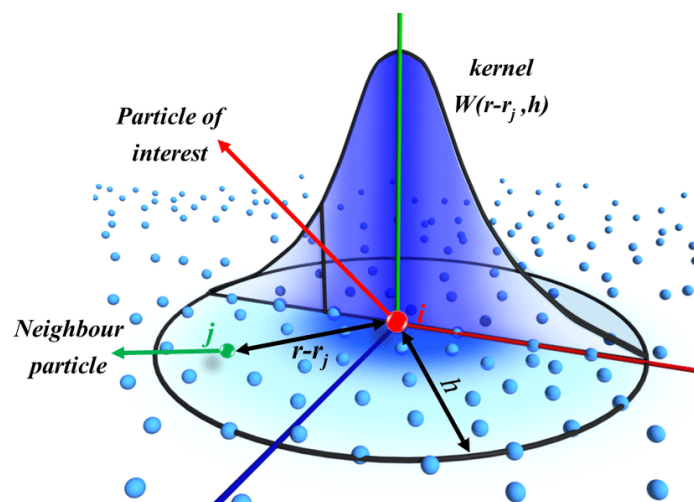


Figura II.2: Representação de partícula. Fonte: [Giorgi, 2018]

Ambos os métodos mencionados apresentam vantagens e desvantagens dados diferentes cenários [Hosain and Fdhila, 2015]. Em geral, os métodos de malhas são mais custosos computacionalmente,

pois a malha deve compreender todo o domínio da simulação para calcular a advecção<sup>1</sup> do fluido. Métodos baseados em partículas, por outro lado, tendem a ser mais eficientes, pois são focados no fluido e não no espaço. No entanto, métodos de malhas produzem simulações mais realistas do que métodos baseados em partículas. Além disso, a precisão e estabilidade dos métodos de partículas dependem da densidade das partículas em uma área.

No contexto de aplicações mulsemídia, para simulação de efeitos sensoriais, não estamos interessados em simular todo o domínio, por exemplo, toda uma sala ou ambiente virtual onde efeitos são emitidos. O interesse é nos locais em que o usuário pode ocupar durante a execução da aplicação. Além disso, os atuadores disponíveis para renderizar efeitos sensoriais em aplicações mulsemídia são normalmente *hardwares* experimentais [Saleme et al., 2019], que não proveem um controle preciso da renderização de efeitos, em oposição àqueles usados para renderizar conteúdos audiovisuais. Assim, uma simulação de alta fidelidade não é necessária. Portanto, entre velocidade e fidelidade da simulação, é preferível produzir simulações interativas em tempo real que apresentem resultados capazes de garantir as características físicas fundamentais da dispersão e advecção de um fluido para permitir o controle de atuadores. Por esse motivo, um método baseado em partículas foi escolhido para a simulação de efeitos sensoriais no presente trabalho.

A escolha por um método baseado em partículas, além das razões apresentadas acima, nos permite combinar diferentes efeitos sensoriais no mesmo ambiente. Bastando, para tanto, incluir informações sobre estes novos efeitos nas partículas. A proposta apresentada nesta dissertação pretende combinar efeitos que se comportam como líquidos, como um borrifo d'água, e efeitos que se comportam como gases, como névoa, aroma, vento, etc.

## II.2 Simulação de Fluidos Incompressíveis

Müller et al. [2003] apresentam um método baseado em partículas para simulação interativa de fluidos e renderização. O modelo físico é baseado no SPH e usa funções de *kernel* de propósito especial para aumentar a estabilidade e a velocidade do método. Para cada campo de força definido pelas equações de Navier-Stokes (Equações II.1 e II.2), isto é, forças externas, pressão e viscosidade, há uma função de *kernel* diferente. As funções de *kernel* são definidas em termos da posição da partícula  $\mathbf{r}$  e seu raio de vizinhança  $h$ . A Equação II.3 define a função de *kernel* usada para calcular a densidade das partículas, a Equação II.4 define a função de *kernel* para calcular a pressão devido às partículas vizinhas e a Equação II.5 define a função de *kernel* para viscosidade. A Figura II.3 apresenta as visualizações das funções de *kernel*.

---

<sup>1</sup>Convecção é o movimento ascendente ou descendente de matéria em um fluido (i.e. líquidos, gases e reids). Advecção é o termo empregado para o movimento horizontal, em particular para massas de ar.



$$W_{poly6}(r, h) = \frac{315}{64\pi h^9} \begin{cases} (h^2 - r^2)^3, & 0 \leq r \leq h \\ 0, & otherwise \end{cases} \quad (\text{II.3})$$

$$W_{spiky}(r, h) = \frac{15}{\pi h^6} \begin{cases} (h - r)^3, & 0 \leq r \leq h \\ 0, & otherwise \end{cases} \quad (\text{II.4})$$

$$W_{viscosity}(r, h) = \frac{15}{2\pi h^3} \begin{cases} -\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1, & 0 \leq r \leq h \\ 0, & otherwise \end{cases} \quad (\text{II.5})$$

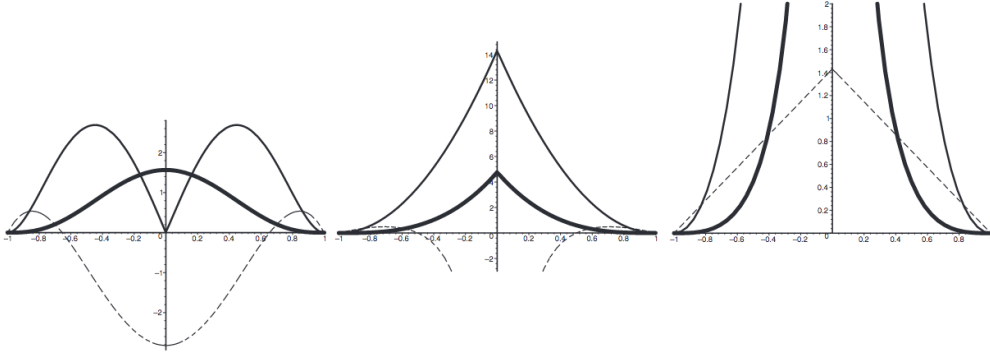


Figura II.3: As três funções de *kernel*  $W_{poly6}$ ,  $W_{spiky}$  and  $W_{viscosity}$  (da esquerda para direita) usadas nas simulações. As linhas grossas apresentam as funções, as linhas finas seus gradientes na direção do centro e as linhas tracejadas seu Laplaciano. Fonte:[Müller et al., 2003]

Dadas as três forças definidas na Equação II.2 para modelagem de pressão ( $-\nabla p$ ), forças externas ( $\rho g$ ) e viscosidade ( $\mu \nabla^2 \mathbf{v}$ ), durante a simulação no SPH, uma quantidade escalar é interpolada em uma localização  $\mathbf{r}$  por uma soma ponderada das contribuições de todas as partículas com massa  $m$ . Isso resulta nas Equações II.6 até II.8 definidas por Müller et al. [2003].

$$\rho_i = \sum_j m_j W_{poly6}(r_i - r_j, h) \quad (\text{II.6})$$

$$f_i^{pressure} = - \sum_j m_j \frac{\rho_i + \rho_j}{2\rho_j} \nabla W_{spiky}(r_i - r_j, h) \quad (\text{II.7})$$

$$f_i^{viscosity} = \mu \sum_j m_j \frac{\mathbf{v}_i + \mathbf{v}_j}{\rho_j} \nabla^2 W_{viscosity}(r_i - r_j, h) \quad (\text{II.8})$$

A Equação II.6 é usada para calcular a densidade  $\rho$  da partícula, que é usada para calcular as forças externas. A Equação II.7 é usada para calcular a pressão no local da partícula, dado que  $\rho = k(\rho - \rho_0)$  onde  $k$  é a constante do gás e  $\rho_0$  é a densidade em repouso. A Equação II.8 é usada para calcular a viscosidade no local da partícula, dada a viscosidade do fluido  $\mu$  e a velocidade  $\mathbf{v}$ .

O método proposto por Müller et al. [2003] é mais apropriado para a simulação de fluidos não compressíveis, ou seja, água ou outros líquidos. Quando se trata da simulação de efeitos sensoriais, alguns efeitos se comportam como fluidos gasosos, por exemplo, cheiro, névoa e vento. Nesses casos, uma abordagem específica para gases é mais adequada.

### II.3 Simulação de Fluidos Gasosos

A abordagem padrão para simular fluidos gasosos usando SPH envolve o preenchimento do ambiente com partículas de ar. Essas partículas de ar teriam parâmetros diferentes (massa, viscosidade, pressão de repouso, etc.) das partículas que representam o fluido de interesse. Durante a simulação, as partículas de ar aplicam suas forças às partículas do fluido gasoso de interesse simulando a força realizada pelo ambiente no gás. O problema com essa abordagem é que a adição de partículas extras torna a simulação mais custosa computacionalmente. Neste contexto, Ren et al. [2016] apresentam uma abordagem onde forças de compensação são usadas para que não haja necessidade de se preencher o ambiente com partículas de ar. Essa abordagem é apresentada na Figura II.4.

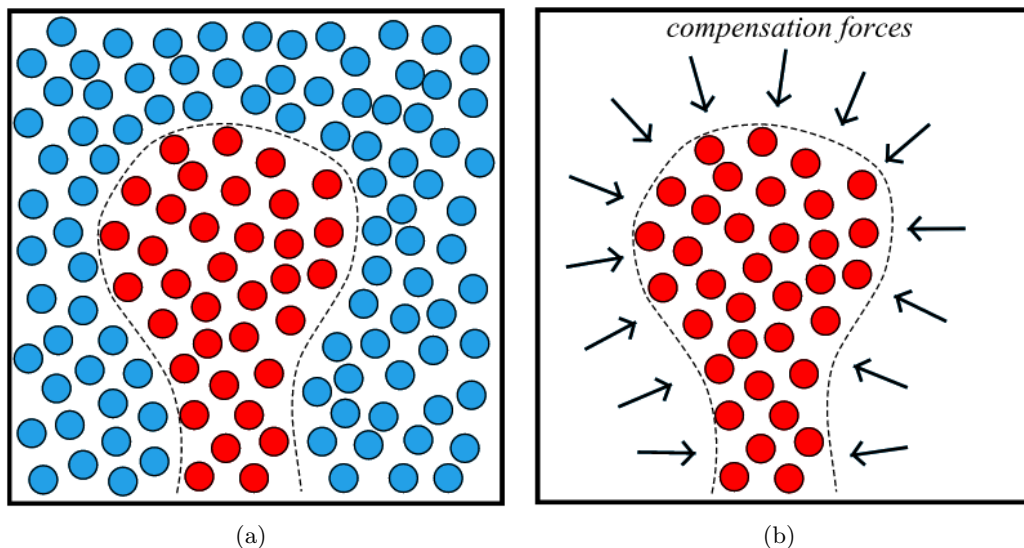


Figura II.4: (a) Abordagem padrão de preencher o ambiente com partículas de ar (azul). (b) Abordagem de Ren et al. [2016] remove as partículas de ar e adiciona forças de compensação atuando sobre as partículas de gás (vermelho). Fonte: [Ren et al., 2016].

Para contabilizar as forças de compensação, um termo extra é incluído nas Equações II.6 a II.8. Esse termo considera uma partícula virtual  $k$  adicionada à soma ponderada das contribuições de

todas as partículas, tal que:

$$r_i - r_k = \delta h \quad (\text{II.9})$$

$$\rho_k = \rho_i \quad (\text{II.10})$$

$$m_k = V_0 \bar{\rho}_i \quad (\text{II.11})$$

$$V_0 = \frac{|\sum \frac{m_j}{\bar{\rho}_j} \nabla W_{ij}|}{|\nabla W_{ik}|} \quad (\text{II.12})$$

A Equação II.9 define a distância da partícula virtual  $k$  à sua partícula emparelhada  $i$ . Essa distância é dada por um parâmetro  $\delta$  que em [Ren et al., 2016] foi definido heurísticamente como

$$\delta \in [0, 4; 0, 7].$$

A Equação II.10 define a densidade da partícula virtual para ser a mesma que a densidade de sua partícula emparelhada  $i$ . A Equação II.11 define a massa da partícula virtual dado o valor predeterminado  $V_0$  e a densidade não corrigida  $\bar{\rho}_i$  de sua partícula emparelhada  $i$ . A Equação II.13 corrige a densidade das partículas na ausência de partículas de ar ambiente ao redor.

$$\rho_i = \bar{\rho}_i (1 + V_0 W_{ik}) \quad (\text{II.13})$$

Usando as partículas virtuais mencionadas anteriormente, um termo extra é adicionado à Equação II.7 para compensar o equilíbrio de força local, resultando na Equação II.14 para calcular a pressão.

$$\tilde{F}_{p,i} = - \sum_j m_j \frac{\rho_i + \rho_j}{2\rho_j} \nabla W(r_i - r_j, h) + m_k \frac{\rho_i + \rho_k}{2\rho_k} \nabla W(r_i - r_k, h) \quad (\text{II.14})$$

A pressão atmosférica empurra o gás para sua direção interna. Para simular esse efeito, a Equação II.15 calcula o vetor normal local na partícula  $i$ . Então, a Equação II.16 define a força de pressão atmosférica, onde  $P_a$  é um coeficiente de força definido pelo usuário que reflete a pressão do ar ambiente.

$$n_i = \sum \frac{m_j}{\rho_j} \nabla W_{ij} \quad (\text{II.15})$$

$$\tilde{F}'_{p,i} = P_a n_i \quad (\text{II.16})$$

Para levar em consideração o efeito da viscosidade local do gás, a Equação II.17 define um efeito

de amortecimento proporcional à velocidade da partícula, onde  $C_d$  é o coeficiente de amortecimento e  $\mathbf{v}_i$  é a velocidade da partícula  $i$ . O efeito de amortecimento é aplicado apenas nos casos em que a norma da partícula excede um determinado valor limite, isto é,  $|n_i| > C_N$ .

$$\tilde{\mathbf{a}}_{damp,i} = -C_d \mathbf{v}_i \quad (\text{II.17})$$

Uma força de empuxo também é introduzida por este método. O empuxo está relacionado com a temperatura das partículas de gás. A Equação II.18 define como a temperatura de cada partícula de gás evolui, onde  $T_i$  é a temperatura da partícula  $i$ ,  $D_c$  é a condutividade térmica do gás, e  $\gamma^2 \ll 1$  é um pequeno número positivo para evitar a singularidade computacional.

$$\frac{\partial T_i}{\partial t} = \sum_j \frac{m_j}{\rho_i \rho_j} D_c (T_i - T_j) \frac{(r_i - r_j) \cdot \nabla W_{ij}}{(r_i - r_j)^2 + \gamma^2} \quad (\text{II.18})$$

Para as partículas com  $|n_i| > C_N$ , Equação II.19 define um efeito de radiação a ser aplicado para simular a transferência de calor para as partículas ausentes, onde  $D_r$  denota a meia vida da radiação.

$$\frac{\partial T_i}{\partial t} = \frac{-T_i}{D_r} \quad (\text{II.19})$$

Então, a Equação II.20 define a aceleração da partícula devido às forças de empuxo, onde  $C_b$  é o coeficiente de empuxo e  $\mathbf{b}$  é um vetor unitário apontando para a direção do empuxo.

$$\tilde{\mathbf{a}}_{b,i} = C_b T_i \mathbf{b} \quad (\text{II.20})$$

Para calcular a aceleração de cada partícula, a pressão atmosférica definida pela Equação II.16 é adicionada à pressão local definida pela Equação II.7 para formar o termo da pressão, as acelerações de amortecimento e empuxo também estão incluídas. A Equação II.21 define a equação completa deste método.

$$\mathbf{a}_i = \frac{1}{\rho_i} (\tilde{\mathbf{F}}_{p,i} + \tilde{\mathbf{F}}'_{p,i}) + \frac{1}{\rho_i} F_{v,i} + \mathbf{g} + \tilde{\mathbf{a}}_{damp,i} + \tilde{\mathbf{a}}_{b,i} \quad (\text{II.21})$$

Ambos os métodos [Müller et al., 2003] e [Ren et al., 2016] são usados para simular diferentes tipos de fluidos. Em aplicações mulsemmedia podem existir efeitos sensoriais que se comportam mais como líquidos (por exemplo, borrifo d'água) e outros que se comportam mais como fluidos gasosos (por exemplo, vento, névoa e cheiro). Dada essa característica, a proposta presente neste trabalho trás uma abordagem híbrida combinando ambos os métodos apresentados, com o propósito de permitir a simulação de efeitos sensoriais variados.

## Capítulo III Trabalhos Relacionados

Este capítulo discute estudos relacionados à nossa proposta. Eles são organizados naqueles focados na simulação de fluidos computacional (*i*) de efeitos sensoriais, mais especificamente a dispersão de cheiro em um ambiente virtual, e (*ii*) mulsemídia em geral.

### III.1 Simulação de fluidos computacional

Ishida et al. [2008]; Matsukura et al. [2009] propõem utilizar uma simulação de CFD juntamente com um emissor de odor para calcular o fluxo de ar em um ambiente e controlar a dispersão do odor. Os artigos acima apresentam duas demonstrações dessa integração. Em [Matsukura et al., 2009] o cheiro é emitido em sincronia com a reprodução de um vídeo. No [Ishida et al., 2008] a difusão de aromas é gerada de acordo com um fluxo de ar variável no ambiente virtual. O software usado para calcular a simulação de fluido foi o *CFD2000*<sup>1</sup>, e os dados da simulação foram armazenados e usados na simulação.

Matsukura and Ishida [2009] descrevem um conjunto de boas práticas para obter uma experiência mais realista do efeito sensorial de um perfume, dados os resultados das experiências apresentadas anteriormente. Uma das questões consideradas no artigo indica que a percepção do cheiro é diferente para cada pessoa. Portanto, a intensidade do efeito sensorial deve ser adaptada de acordo com a sensibilidade olfativa da pessoa. Outra questão importante é considerar a presença do corpo humano na simulação, a convecção causada pelo calor do corpo induz mudanças significativas na concentração do cheiro.

Nakamoto et al. [2020] desenvolveram uma aplicação Virtual Reality (VR) onde o usuário deve procurar uma fonte de odor no ambiente virtual. O ambiente virtual consiste em um prédio de dois andares com quatro salas em cada andar e uma fonte de odor localizada em uma dessas salas. O software utilizado para a simulação de CFD foi o *ANSYS18.1*<sup>2</sup>. A simulação foi executada para calcular a difusão do odor no ambiente. Em seguida, os dados da simulação foram armazenados para serem utilizados posteriormente durante a execução da aplicação. O usuário, equipado de um Head-Mounted Display (HMD) e de um dispositivo emissor de odor, percorre o ambiente virtual em busca da origem do odor. A intensidade do cheiro emitido pelo dispositivo muda de acordo com

---

<sup>1</sup><http://www.adaptive-research.com/>

<sup>2</sup><https://www.ansys.com/>

a posição do usuário no ambiente, os dados gerados pela simulação e a duração da execução.

Nos artigos apresentados nesta seção, a simulação de efeitos sensoriais é realizada antes da execução da aplicação. Os dados de simulação são armazenados para uso posterior na execução da aplicação. Essa abordagem apresenta algumas limitações quando há mudanças no ambiente virtual durante a execução da aplicação e essas mudanças afetam a simulação de efeitos sensoriais. Seria interessante se a simulação acontecesse em tempo real reagindo a estas mudanças. No entanto, a simulação de fluidos é uma tarefa complexa que exige muito processamento computacional [Nakamoto et al., 2020]. Ainda assim, conforme visto no Capítulo II, algumas abordagens permitem executar simulações em tempo real. Embora comprometam um pouco a fidelidade da física envolvida no comportamento do fluido, elas produzem resultados suficientes para o controle de atuadores de aplicações mulsemédia.

### III.2 Controle de Efeitos Sensoriais em Mulsemídia

MPEG-V é o padrão definido para aplicações em que o conteúdo audiovisual está associado a efeitos sensoriais. O padrão fornece uma arquitetura e especifica a representação da informação para possibilitar a interoperabilidade entre os mundos virtual e real, isto é, o estado da aplicação e os atuadores e sensores [Yoon et al., 2015]. O padrão consiste em 7 partes. A parte 1 [ISO/IEC, 2016a] fornece uma visão geral e define a arquitetura mostrada na Figura III.1. A parte 2 [ISO/IEC, 2016b] fornece ferramentas para a descrição das capacidades de atuadores e sensores e preferências do usuário. A parte 3 [ISO/IEC, 2016c] fornece as ferramentas para uma descrição do efeito sensorial em sincronia com o conteúdo da mídia. A parte 4 [ISO/IEC, 2016d] define as características de um objeto de mundo virtual para fornecer ferramentas que permitam a interoperabilidade entre mundos virtuais. A parte 5 [ISO/IEC, 2016e] especifica as interfaces ou formatos de dados para troca interoperável de informações de/para os sensores e atuadores. A parte 6 [ISO/IEC, 2016f] especifica a sintaxe e semântica dos tipos de dados e ferramentas que são comuns a mais de uma parte do padrão MPEG-V. Por fim, a parte 7 [ISO/IEC, 2016g] fornece o software de referência e especifica a conformidade.

O padrão MPEG-V define a arquitetura e a representação da informação associada para permitir a troca de informações entre os mundos virtuais (a própria aplicação) e o mundo real (sensores e atuadores no ambiente físico). No entanto, não define como a aplicação deve ser implementada. Os autores das aplicações são responsáveis por definir como o conteúdo audiovisual será apresentado, a sincronização do conteúdo e dos efeitos sensoriais e a captura da interação do usuário. Nesta seção, discutiremos alguns dos trabalhos que utilizam o padrão MPEG-V para construção de aplicações mulsemídia.

Waltl et al. [2013] apresentam uma visão geral de algumas ferramentas que usam o padrão

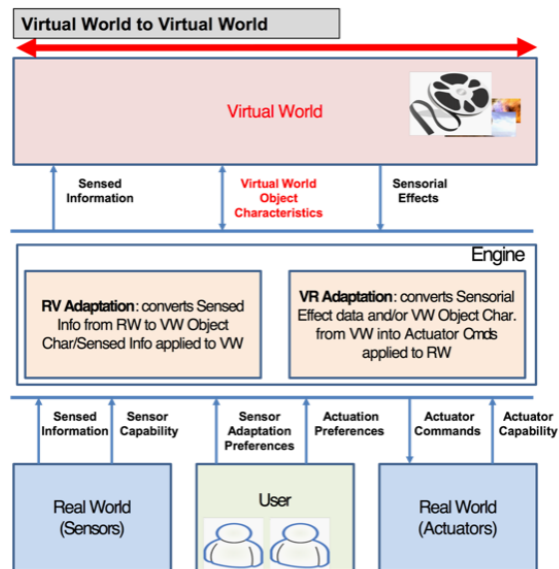
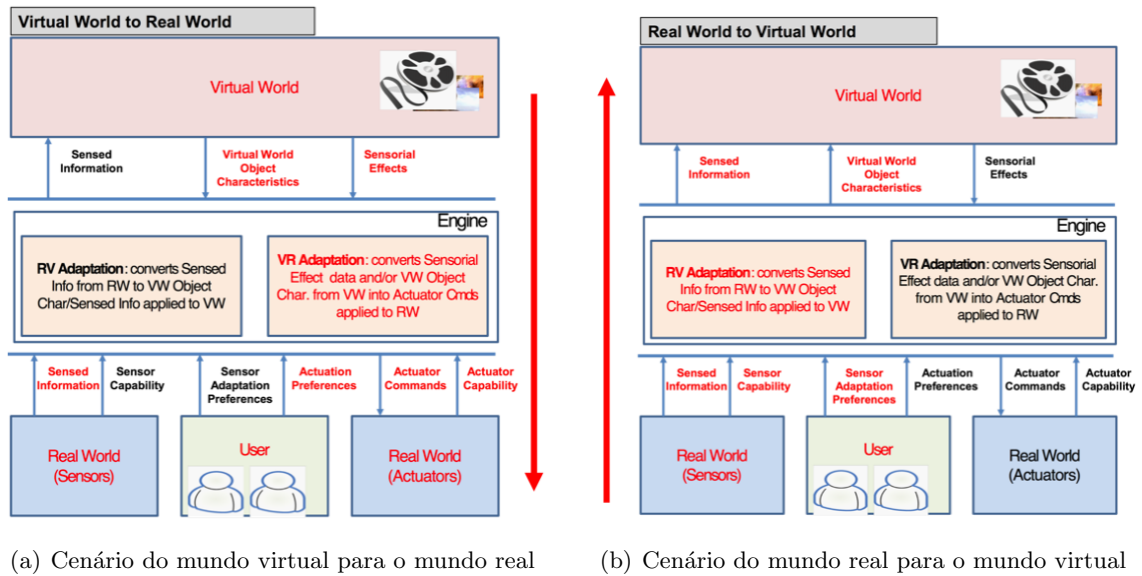


Figura III.1: Arquitetura do padrão MPEG-V e cenários de transmissão de dados[Yoon et al., 2015]

MPEG-V para autoria, simulação e renderização de efeitos sensoriais em aplicativos multimídia. Sensory Effect Video Annotation (SEVino) permite anotar seqüências de vídeo com informações de efeitos sensoriais e gera uma descrição de acordo com o padrão MPEG-V. Sensory Effect Simulator (SESim) simula os efeitos sensoriais dada a descrição gerada pelo SEVino, renderiza o conteúdo audiovisual e a representação dos efeitos sensoriais em uma Graphical User Interface (GUI). Sensory Effect Media Player (SEMP) é um reproduzidor de mídia baseado em DirectShow<sup>3</sup> que suporta os seguintes dispositivos para renderização de efeitos sensoriais: *amBX Gaming PC peripherals* (luz, vento e vibração), *Cyborg Gaming Lights for PC* (luz) e *Vortex Activ* (aroma).

<sup>3</sup>DirectShow – API provida pela Microsoft para *streaming* de mídia no Windows.

Saleme and Santos [2015] apresentam o *PlaySEM*, plataforma de renderização de efeitos sensoriais compatível com o padrão MPEG-V. A plataforma é escrita em Java e é composta por um reprodutor de vídeo (*SE Video Player*), um renderizador de efeitos sensoriais (*SE Renderer*) e um microcontrolador que recebe os comandos para ativar a renderização dos efeitos sensoriais dispositivos.

Para facilitar a autoria de aplicações mulsemídia e permitir uma melhor portabilidade, uma abordagem declarativa pode ser usada, fornecendo uma clara distinção entre a definição da aplicação e os detalhes da implementação. Josué et al. [2018] e Rodrigues et al. [2019] usam a linguagem Nested Context Language (NCL) [Soares and Barbosa, 2009] para definir efeitos sensoriais como objetos de mídia padrão. Esta abordagem permite o reaproveitamento dos recursos da linguagem de sincronização dos efeitos sensoriais com outros conteúdos midiáticos e definição da posição e animação dos efeitos sensoriais.

A arquitetura da proposta por este trabalho, apresentada com mais detalhes na Seção IV.1, é inspirada na arquitetura definida pelo padrão MPEG-V, mostrada na Figura III.1. A representação do mundo virtual é caracterizada pela descrição do ambiente para a simulação dos efeitos sensoriais, o motor da aplicação calcula a simulação com base na definição do ambiente e a interação do usuário. Os dados produzidos pela simulação são utilizados por um módulo de visualização permitindo explorar a simulação e disponibilizados em formato de arquivo para que possam ser analisados posteriormente ou utilizados em outras aplicações.

Dado que abordagens declarativas, como as que utilizam a linguagem NCL, podem facilitar a autoria das aplicações e permitir uma melhor portabilidade, optamos por possibilitar ao usuário que defina a simulação de maneira declarativa por meio de um arquivo de configuração. O arquivo de configuração, apresentado com mais detalhes na Seção IV.5, é constituído da representação do ambiente da simulação contendo os dispositivos sensoriais (atuadores e sensores), os atributos desses dispositivos, e os parâmetros gerais da simulação. Esta abordagem permite aos usuários que explorem facilmente diferentes cenários apenas alterando as informações no arquivo de configuração, sem a necessidade de se ater a detalhes da implementação da simulação.



## Capítulo IV Simulação de Efeitos Sensoriais

Este capítulo apresenta a proposta deste trabalho para simular a dispersão de efeitos sensoriais em um determinado ambiente. Por simplicidade, nos referimos à proposta como FluidSense - *Fluid Simulation for Sensory Effects*. Primeiramente, relembramos as principais características exigidas para a simulação de efeitos sensoriais apresentadas nos capítulos anteriores. Em seguida, apresentamos a arquitetura do FluidSense e seu uso pretendido em conjunto com aplicações mulsemídia.

Para a simulação de efeitos sensoriais, não estamos interessados na simulação de todo o domínio, ou seja, todo o mapa de um ambiente virtual ou de toda a sala, mas sim nos lugares que o usuário pode ocupar. Além disso, como os atuadores não fornecem muito controle na renderização de efeitos sensoriais, não se faz necessária uma simulação de alta fidelidade. Conforme discutido anteriormente, essa simplificação nos permite reduzir o custo computacional das simulações e, portanto, fornecer dados em tempo real para o controle de atuadores presentes no ambiente.

A escolha de um método baseado em partículas, além dos motivos acima, permite combinar diferentes efeitos em um mesmo ambiente, incluindo mais informações sobre eles nas partículas. FluidSense tem como objetivo combinar efeitos que se comportam como líquidos, como um spray de água, e aqueles que se comportam como um gás, como névoa, cheiro, vento e etc.

### IV.1 Arquitetura

A proposta apresentada neste trabalho para simulação de efeitos sensoriais conta com a estrutura capaz de lidar com a interação entre o mundo real e o virtual. FluidSense permite que os autores definam a especificação do mundo da simulação<sup>1</sup> e apliquem alterações no estado do mundo da simulação, dada a interação do usuário e os dados capturados por sensores. FluidSense, então, renderiza os efeitos sensoriais dado o estado do mundo de simulação e fornece informações para um motor de apresentação para controlar os atuadores presentes no ambiente de execução. A Figura IV.1 apresenta a arquitetura do FluidSense.

Para simular efeitos sensoriais automaticamente, o autor da aplicação mulsemídia define o mapa da simulação, isto é, a posição de paredes ou obstáculos. Além disso, é possível colocar atuadores

---

<sup>1</sup>Para evitar ambiguidade, usamos o termo “mundo da simulação” para a especificação do mundo virtual usado na simulação de dispersão de efeitos sensoriais. O mundo da simulação pode representar o mundo virtual de uma aplicação, bem como o local onde o usuário a executa. O termo “mundo virtual” é deixado para aquele apresentado pelo aplicativo mulsemídia.

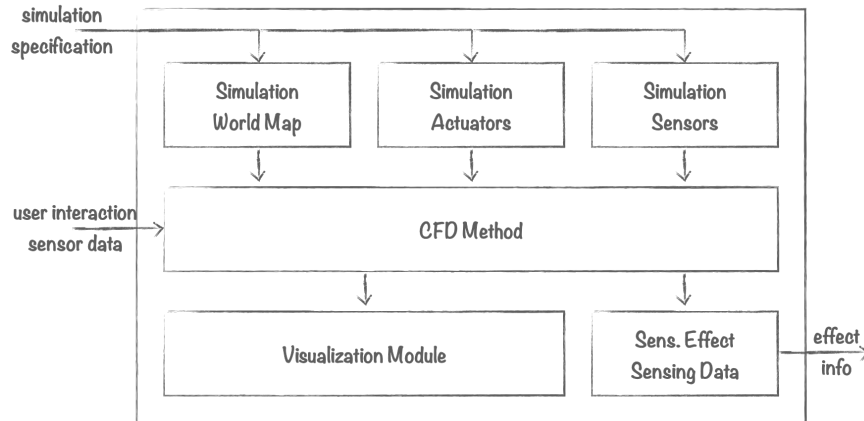


Figura IV.1: Arquitetura do FluidSense.

virtuais em diferentes locais no mundo da simulação e definir suas características, como o tipo de efeitos sensoriais, sua intensidade, etc. Dadas as definições do atuador de simulação, FluidSense controla o método de CFD que simula a dinâmica dos fluidos que representam os efeitos sensoriais especificados. Para poder consultar os dados de efeitos sensoriais, o autor da aplicação mulsemídia pode colocar sensores virtuais no mundo da simulação. Um sensor virtual pode definir uma área onde coletará dados e o tipo de informação de interesse. Esses sensores virtuais coletarão dados de todas as partículas naquela área e os combinarão para fornecer informações sobre o efeito sensorial. Finalmente, um módulo de visualização permite que o usuário visualize a simulação.

## IV.2 Simulação

Para validar a arquitetura proposta, usamos um método específico de CFD. No caso deste trabalho, o método usado é o SPH. Conforme descrito anteriormente, FluidSense usa uma implementação do SPH híbrida baseada nas ideias apresentadas por Müller et al. [2003] e Ren et al. [2016]. Esta abordagem permite simular tanto efeitos sensoriais que se comportam como fluidos não compressíveis, por exemplo, líquidos, quanto efeitos sensoriais que se comportam como gases. Assim, é possível simular efeitos de borrifo d'água, vento e dispersão de aromas. FluidSense, portanto, dado o tipo de efeito sensorial sendo simulado, decide se as forças de compensação (consulte a Seção II.3) devem ser aplicadas durante a simulação de fluido.

---

**Algorithm 1** FluidSense fluid simulation main loop using SPH.

---

**Require:** a set of particles  $P$  and a time step  $t_s$ .

$t \leftarrow 1$

loop

$P \leftarrow \text{ComputeDensities}(P)$

$P \leftarrow \text{ComputeForces}(P)$

$P \leftarrow \text{Integrate}(P, t_s)$

$t \leftarrow t + t_s$

end loop

---

O Algoritmo 1 descreve o *loop* principal para simulação de fluidos. Dado um conjunto de partículas  $P$  na simulação, primeiro calcula-se as densidades de cada partícula. As densidades das partículas são calculadas considerando suas partículas vizinhas conforme apresentado na Seção II.2 e a densidade das partículas é armazenada na partícula para uso posterior. Em seguida, calcula-se as forças externas que atuam sobre a partícula. As forças de pressão e viscosidade são calculadas levando em consideração a densidade da partícula e informações sobre suas partículas vizinhas. Também são consideradas forças adicionais, como a gravidade, ao definir a força resultante que atua sobre a partícula. Finalmente, uma etapa de integração é realizada para calcular a aceleração de cada partícula, obtendo assim sua velocidade e permitindo calcular a próxima posição das partículas. O método Leap-Frog [Pozrikidis et al., 1998] é usado para integração.

---

**Algorithm 2** *ComputeDensities*( $P$ )

---

```

for all  $p \in P$  do
   $N \leftarrow \text{GetNeighbours}(p, P)$ 
   $\rho \leftarrow \text{ComputeUncorrectedDensity}(p, N)$ 
  if  $p$  is gaseous then
     $\rho \leftarrow \text{CorrectDensity}(p)$ 
  end if
end for

```

---

O Algoritmo 2 descreve o procedimento para calcular densidades. Primeiramente, calcula-se a densidade não corrigida  $\bar{\rho}_i$  para cada partícula na simulação, independentemente do método SPH usado, de acordo com a Equação II.6. Este valor de densidade não considera as forças de compensação, mas é necessário tanto para o cálculo das forças externas sobre a partícula quanto para a correção do valor da densidade. Então, se a partícula for de um fluido gasoso, a densidade é corrigida com as forças de compensação, conforme Equação II.13.

---

**Algorithm 3** *ComputeForces*( $P$ )

---

```

for all  $p \in P$  do
   $N \leftarrow \text{GetNeighbours}(p, P)$ 
   $p \leftarrow \text{ComputePressure}(p, N)$ 
   $p \leftarrow \text{ComputeViscosity}(p, N)$ 
   $p \leftarrow \text{ComputeResultingForce}(p)$ 
  if  $p$  is gaseous then
     $p \leftarrow \text{ComputeAtmosphericPressure}(p, N)$ 
     $p \leftarrow \text{CorrectPressure}(p)$ 
     $p \leftarrow \text{ComputeTemperature}(p, N)$ 
     $p \leftarrow \text{ComputeDampingAndBuoyancy}(p, N)$ 
     $p \leftarrow \text{CorrectResultingForce}(p)$ 
  end if
end for

```

---

O Algoritmo 3 descreve o procedimento para calcular as forças que atuam nas partículas. Calcula-se as forças de pressão e viscosidade, conforme definido nas Equações II.7 e II.8, res-

pectivamente. Por fim, calcula-se a força resultante considerando as forças de pressão, viscosidade e gravidade.

No caso de um fluido gasoso, é necessário considerar também a pressão atmosférica atuando sobre a partícula. Portanto, o procedimento apresentado no Algoritmo 3 calcula a pressão atmosférica, depois corrige a pressão usando a força de compensação da partícula virtual. O procedimento também calcula a temperatura, o amortecimento obtido em função da pressão atmosférica e o empuxo. Por fim, corrige a força resultante que atua sobre a partícula com base nos cálculos acima.

Ambos os métodos compartilham passos em comum, portanto o algoritmo é facilmente adaptável para contemplar ambos os tipos de simulação dada a configuração do autor da aplicação. Além disso, os autores também podem configurar os parâmetros da simulação, permitindo a modelagem de fluidos com diferentes características. Da mesma forma, ambos os métodos compartilham parâmetros comuns e o método para simulação de fluidos gasosos possui alguns parâmetros adicionais usados no cálculo das forças de compensação.

### IV.3 Parâmetros

Dentre os parâmetros necessários para a simulação de efeitos sensoriais utilizando o método SPH, a Tabela IV.1 apresenta os que são comuns aos dois métodos. Esses parâmetros especificam as características das partículas a serem usadas para calcular a densidade da partícula e a força resultante agindo sobre ela.

Tabela IV.1: Parâmetros comuns a ambos os métodos SPH.

Parâmetro	Definição
Raio	Raio da partícula em metros usado para calcular a vizinhança da partícula.
Massa	Massa da partícula em gramas usada para calcular a densidade do fluido na posição da partícula.
Viscosidade	Constante de viscosidade do fluido.
Densidade em Repouso	Densidade do fluido em repouso.
Força da Gravidade	Vetor representando a força da gravidade.
Constante do Gás	Constante do gás que depende da temperatura.

Para a simulação de fluidos gasosos, também é necessário definir alguns parâmetros que serão utilizados durante a aplicação das forças de compensação. Esses parâmetros adicionais são apresentados na Tabela IV.2. Conforme descrito na Seção II.3, para calcular as forças de compensação, uma partícula virtual é considerada somando aos pesos das contribuições de todas as partículas. O valor da partícula virtual é representado por um vetor tridimensional a uma distância da partícula de interesse definida por um valor fixo  $\delta h$ , onde  $h$  é o raio da partícula e  $\delta$  é heurísticamente defi-

nido como  $\delta \in [0.4, 0.7]$ . Portanto, este vetor pode assumir qualquer direção desde que respeite a distância especificada. A direção deste vetor define a direção do movimento do fluido, então o autor do aplicativo pode especificar a direção do fluido por este vetor, bem como o grau no cálculo das forças de deslocamento ajustando o valor de  $\delta$  dentro do intervalo definido.

Tabela IV.2: Parâmetros específicos à simulação de fluidos gasosos.

Parâmetro	Valor
$\delta$	Distância da partícula virtual em termos do raio da partícula.
Coefficiente de Amortecimento	Impacto do efeito de amortecimento aplicado à partícula.
Limite de amortecimento	Limite de velocidade para disparar o efeito de amortecimento na partícula.
Condutividade Térmica	Taxa com a qual a partícula transfere calor.
Meia vida de Radiação	Taxa com a qual a partícula perde calor a grandes velocidades.
Direção de Empuxo	Direção de empuxo definida pelo usuário.
Coefficiente de Empuxo	Coefficiente definido pelo usuário para aplicação do empuxo.

Outros parâmetros apresentados na Tabela IV.2 representam constantes relacionadas aos gases, como o coeficiente de amortecimento, usado se a pressão atmosférica exceder um limite de amortecimento, para calcular o efeito de amortecimento; a condutividade térmica e a meia-vida da radiação, ambas utilizadas no cálculo da temperatura; o empuxo e o coeficiente de empuxo, para calcular o empuxo da partícula. Alguns valores base para essas constantes também foram utilizados a fim de obter uma simulação estável para os testes.

#### IV.4 Sensores e Atuadores

A Seção IV.2 apresentou o *loop* de simulação que calcula repetidamente a densidade de uma partícula e as forças que atuam sobre ela e atualiza a posição e a velocidade da partícula. O conjunto de partículas  $\mathcal{P}$  apresentado no Algoritmo 1, 2 e 3 para a simulação de fluidos são emitidos pelos atuadores virtuais que o autor da aplicação posiciona no mundo da simulação. Portanto, diferente das simulações SPH tradicionais que criam todas as partículas de uma vez e depois executam a simulação, em FluidSense, o conjunto de partículas na simulação muda ao longo do tempo conforme novas partículas são emitidas pelos atuadores virtuais.

Cada atuador  $(\mathbf{o}, \vec{\mathbf{e}}_d, \mathbf{v}, r, t_{SE})$  consiste em um ponto tridimensional  $\mathbf{o}$  que dá a posição do atuador no mundo da simulação, um vetor tridimensional  $\vec{\mathbf{e}}_d$  definindo a direção de emissão das partículas, a velocidade inicial  $\mathbf{v}$  das partículas, uma faixa de emissão  $r$  das partículas e o tipo  $t_{SE}$  do efeito sensorial que o atuador emite. A faixa de emissão define uma área ao redor da posição do atuador. Esse valor é usado em uma função de *jitter* para definir aleatoriamente a posição de novas partículas à medida que são emitidas. A faixa de emissão também é importante para evitar

instabilidade da simulação. Finalmente, de acordo com o tipo de efeito sensorial, um subconjunto dos parâmetros de simulação apresentados na Seção IV.3 são usados.

Além dos atuadores virtuais, os sensores virtuais permitem sondar dados de efeitos sensoriais da simulação. Cada sensor virtual  $(\mathbf{o}, r)$  consiste em um ponto tridimensional  $\mathbf{o}$  que dá a posição do sensor no mundo da simulação e um alcance  $r$  em torno desse ponto onde as partículas devem ser sondadas. A ideia-chave é que o sensor combine informações sobre um conjunto de partículas  $N$  que estão dentro de seu alcance para inferir uma determinada característica de um efeito sensorial.

O procedimento de inferência pode considerar, por exemplo, a velocidade das partículas em  $N$  para definir a velocidade de rotação de um ventilador, a densidade das partículas para definir a intensidade de um cheiro e a temperatura das partículas para ajustar um efeito de temperatura. Uma questão chave aqui é a forma como as informações sobre as partículas em  $N$  são combinadas. Atualmente, uma função de média simples é usada. No entanto, outra abordagem em investigação é o uso de uma função *kernel* para ponderar a influência das partículas em  $N$ , de acordo com sua distância da posição do sensor.

## IV.5 Configuração da Simulação

Nas seções anteriores foram descritos os parâmetros necessários para a simulação utilizando o método SPH, assim como a configuração dos dispositivos sensores e atuadores a serem considerados na simulação. Os parâmetros de simulação incluem as propriedades do fluido e as condições de contorno, e são utilizados como entrada para a execução do FluidSense. Com base nesses parâmetros, é possível construir o modelo matemático que descreve o comportamento do fluido e utilizá-lo para simular diferentes cenários. Uma vez configurada a aplicação, os parâmetros são aplicados de maneira global para a execução permitindo a análise do comportamento do fluido. Adicionalmente, também deve ser fornecida a configuração descrevendo a geometria do domínio da simulação, contendo o posicionamento dos dispositivos ao qual são associadas as demais informações específicas dos atuadores e sensores.

FluidSense permite que a configuração da simulação seja feita por um arquivo no formato YAML Ain't Markup Language (YAML), amplamente utilizado para estruturar dados de configuração de forma legível para humanos. O formato YAML permite definir as configurações da aplicação de simulação de fluidos de maneira concisa e organizada, utilizando espaçamento e hierarquia para indicar a estrutura dos dados. O arquivo é dividido em quatro seções: *environment*, *actuators*, *sensors* e *simulation*. Cada seção é identificada pelo seu nome seguido de dois pontos (:), e os atributos são definidos com um espaçamento específico. Esse formato facilita a edição e manutenção do arquivo de configuração, tornando mais simples ajustar os parâmetros de acordo com as necessidades da simulação de fluidos.

Na seção *environment* é apresentada uma representação em forma de grade do ambiente da simulação com obstáculos (chão e paredes) que limitam o domínio da simulação, o ponto de partida do usuário e o posicionamento dos dispositivos sensores e atuadores. As paredes são representadas pelo caractere “#” e complementarmente devem ser posicionados dentro do perímetro das paredes o caractere “.” para representar o chão do ambiente, espaços em branco podem ser utilizados para definir contornos mais complexos na grade do ambiente, as paredes e o chão delimitam o domínio da simulação. O ponto de partida do usuário é definido pelo caractere “@”, entende-se que a posição inicial do usuário deve estar dentro do domínio da simulação, portanto considera-se que o usuário está sobre o chão do ambiente.

Os caracteres “#”, “.” e “ ” (espaço) são reservados para representar a grade do ambiente, “@” para o ponto de partida do usuário, e outros caracteres podem ser utilizados dentro do domínio da simulação para indicar o posicionamento de dispositivos atuadores e sensores. Os mesmo caracteres são utilizados posteriormente nas seções *actuators* e *sensors* para definir configurações adicionais aos dispositivos, assim como o ponto de partida do usuário considera-se que os dispositivos estão sobre o chão dentro do domínio da simulação. A Listagem IV.1 apresenta um exemplo de definição do ambiente que limita o domínio da simulação.

Listing IV.1: Grade de representação do ambiente de simulação.

```
#####
# . @ . # . . a #
# . . . # . . . #####
# . . . c . . . . . #
# . . . . . . . . . b #
#####
```

Na seção *actuators* são definidas as configurações dos atuadores. Cada atuador é identificado por um carácter que o associa à posição em que este carácter possui na grade da representação do ambiente. No exemplo apresentando na Listagem IV.1 são definidos os identificadores “a”, “b” e “c” que podem ser usados para posicionar atuadores. Além de seu identificador, cada atuador possui os atributos:

- *height* com a altura do atuador, indicando sua posição no eixo *y* complementando os eixos *x* e *z* obtidos pela posição na grade do ambiente;
- *direction* indicando a direção de emissão das partículas;
- *initial\_velocity* indicando a velocidade inicial das partículas;
- *temperature* (opcional) indicando a temperatura das partículas;

- *range* utilizado para definir o raio de emissão das partículas;
- *fluid\_type* indicando o tipo de fluido (líquido ou gasoso);
- *interval* indicando o intervalo de emissão das partículas em milissegundos; e
- *particle* um objeto que contém os atributos *size* para o tamanho da partícula e *color* com a cor da partícula, estes atributos são utilizados pelo módulo de visualização para renderizar as partículas.

Na seção **sensors** são definidas as configurações dos sensores. Assim como os atuadores, cada sensor é identificado por um carácter que é associado ao carácter contido na posição na grade da representação do ambiente. Cada sensor possui os atributos:

- *height* assim como os atuadores complementa a posição na grade do ambiente;
- *range* define o alcance de onde as partículas são sondadas; e
- *output* (opcional) que indica o arquivo CSV onde são escritas as informações das partículas sondadas, caso este parâmetro não seja definido as informações são escritas na saída padrão da aplicação.

Atuadores e sensores podem possuir a mesma chave de identificação, indicando que os dispositivos estão na mesma posição no ambiente. Além disso, é possível que na grade do ambiente seja utilizado o mesmo símbolo em mais de uma posição, permitindo que dispositivos com as mesmas características estejam em posições diferentes sem que seja necessário especificar suas configurações mais de uma vez.

Por fim, na seção **simulation** são definidas as configurações globais da simulação. Essa configuração é aplicada para todas as partículas emitidas na simulação e são independentes das configurações individuais de cada atuador. Para a simulação são definidos os atributos *step* que define o avanço da simulação em milissegundos, bem como os parâmetros apresentados na Seção IV.3 e a direção da partícula virtual.

A Listagem IV.2 demonstra um exemplo de configuração completo, contendo um ambiente de 7 por 9 metros contendo dois atuadores nos cantos do lado direito, um sensor próximo ao centro, e o usuário do lado esquerdo ao centro. Os atuadores estão a uma altura de 1,5 metros e emitem partículas em direção ao centro do ambiente, ambos emitem partículas gasosas a uma velocidade inicial de 5,0 em um intervalo de 0,05 segundos, um dos atuadores emite partículas azuis e o outro vermelha. O sensor está a uma altura de 1 metro, sondando partículas que esteja distantes até 1 metro de sua posição. Na última seção são apresentadas as configurações gerais da simulação.



Listing IV.2: Exemplo de arquivo de configuração.

```

environment: |
    #####
    #.....a#
    #.....#
    #.@.....#
    #...c...#
    #.....b#
    #####
actuators:
  a:
    height: 1.5
    direction: [-1.0, 0.0, 1.0]
    initial_velocity: 5.0
    temperature: 25.0
    range: [0.2, 0.2, 0.2]
    fluid_type: Gaseous
    interval: 0.05
    particle:
      size: 0.02
      color: [0.0, 0.0, 1.0]
  b:
    height: 1.5
    direction: [-1.0, 0.0, -1.0]
    initial_velocity: 5.0
    range: [0.2, 0.2, 0.2]
    fluid_type: Gaseous
    interval: 0.05
    particle:
      size: 0.02
      color: [1.0, 0.0, 0.0]
sensors:
  c:
    height: 1.5
    range: [1.0, 1.0, 1.0]

```

```

    output: "sensor_c.csv"
simulation:
  step: 0.001
  radius: 0.0457
  mass: 0.02
  gas_constant: 3.0
  rest_density: 0.99829
  delta: 0.4
  thermal_conductivity: 0.598
  small_positive: 0.0001
  viscosity: 0.01
  damping_coefficient: 100.0
  damping_threshold: 1.0
  radiation_half_life: 1000.0
  buoyancy_coefficient: 0.8
  buoyancy_direction: [0.0, 1.0, 0.0]
  gravity: [0.0, -9.82, 0.0]
  virtual_particle_direction: [0.0, 1.0, 0.0]

```

## IV.6 Módulo de visualização

O módulo de visualização foi desenvolvido utilizando a linguagem de programação Rust<sup>2</sup> com a biblioteca wgpu<sup>3</sup> para a renderização dos gráficos, winit<sup>4</sup> para criação de janelas e gerenciamento de eventos na aplicação, e outras bibliotecas auxiliares para recursos como serialização e desserialização de dados, e carregamento de imagens para as texturas. As tecnologias utilizadas suportam as plataformas Linux, Windows e MacOS, de forma que seja possível que o FluidSense seja utilizado em quaisquer destas plataformas.

Dado o ambiente descrito no arquivo de configuração, o módulo de visualização processa a grade representando o ambiente e desenha paralelepípedos de base **1x1** e **3** de altura para as paredes, e planos de **1x1** para o chão. Cada um destes objetos é texturizado de acordo para simular a aparência de uma sala real. Um *shader* é utilizado para a renderização destes objetos do ambiente utilizando o modelo de reflexão Blinn-Phong [Blinn, 1977] com uma fonte de luz principal acompanhando a posição do usuário.

---

<sup>2</sup><https://www.rust-lang.org/>

<sup>3</sup><https://wgpu.rs/>

<sup>4</sup><https://github.com/rust-windowing/winit>

Para a renderização das partículas é utilizado um segundo *shader* que desenha as partículas emitidas pelos atuadores de acordo com o tamanho e a cor definidas no arquivo de configuração. As partículas são renderizadas como pequenos quadrados utilizando a técnica de *billboarding*, de forma que as partículas sejam posicionadas de frente para o usuário independente da posição da câmera facilitando sua visualização.

Para a interação com a simulação, é posicionada uma câmera em primeira pessoa na posição inicial do usuário configurada na grade do ambiente no arquivo de configuração. Dado o início da simulação, o usuário pode então utilizar as teclas “W”, “A”, “S” e “D” para se movimentar para frente, para a esquerda, para trás e para a direita, respectivamente. O usuário também pode utilizar o mouse para alterar a direção da câmera, e as teclas espaço e *shift* esquerdo para subir e descer. Estes controles permitem que o usuário navegue pela simulação para observar em diferentes posições o comportamento do fluido.

#### IV.7 Modo *headless*

Além do módulo de visualização, outra opção é utilizar o FluidSense no modo *headless*, nesse modo não é apresentada para o usuário a interface gráfica. No modo *headless*, o FluidSense apenas executa a simulação e os dados dos sensores são capturados. O modo *headless* pode ser utilizado para extrair e analisar os dados da simulação em cenários onde não é necessária a interação do usuário com a simulação para validar seu comportamento.

O modo *headless* é especialmente útil em situações em que a configuração da simulação já foi definida para contemplar o cenário especificado. Nesses casos, a interação visual com a simulação se torna dispensável, permitindo concentrar-se exclusivamente na análise dos dados capturados pelos sensores. Através dessa abordagem, é possível extrair informações valiosas sobre o comportamento da simulação e identificar padrões e tendências.

Além disso, o modo *headless* também pode ser utilizado quando o objetivo é otimizar os parâmetros de execução da simulação para obter resultados específicos. Ao executar a simulação repetidamente e ajustar os parâmetros relevantes, é possível explorar diferentes configurações e observar seu impacto nos dados capturados. Essa abordagem permite uma busca eficiente por soluções ótimas e aprimoramento dos resultados desejados. Assim, o modo *headless* do FluidSense oferece uma flexibilidade adicional aos usuário, permitindo extrair *insights* valiosos dos dados de simulação sem a necessidade de interação visual e explorar diferentes configurações para otimizar os resultados desejados.

## Capítulo V Casos de Uso de Simulação

Neste capítulo são apresentados os detalhes específicos sobre os testes de simulação de efeitos sensoriais. Visto que os métodos apresentados no Capítulo II não especificam os valores para os parâmetros para a reprodução das simulações, foi necessário a análise das fórmulas e experimentações com diferentes conjuntos de parâmetros com o objetivo de produzir uma simulação estável. Como resultado, foram definidos valores de referência para os parâmetros para serem utilizados nos cenários apresentados neste capítulo, estes valores são descritos na Tabela V.1. Experimentos subsequentes foram planejados para coletar dados de fluidos de um ambiente real para o refinamento dos valores destes parâmetros e estabelecer uma tolerância para que os autores possam configurar os parâmetros de acordo com suas necessidades sem afetar a estabilidade da simulação.

Tabela V.1: Parâmetros usados para os testes de simulação.

Parâmetros	Valores
Raio	0,0457
Massa	0,02
Viscosidade	0,01
Densidade em repouso	0,99829
Gravidade	(0,0; -9,8; 0,0)
Constante do gás	3,0
$\delta$	0,4
Coefficiente de amortecimento	1,0
Limite de amortecimento	100,0
Condutividade térmica	0,598
Meia-vida da radiação	1000,0
Coefficiente de empuxo	0,392

### V.1 Descrição do ambiente

O primeiro passo para elaborar uma simulação é a definição do ambiente. Este passo é importante para delimitar o domínio da simulação, apenas as partículas contidas dentro do perímetro do ambiente da simulação são consideradas para a simulação do fluidos, as partículas que atingem os limites da simulação são removidas automaticamente. Para fins de demonstração foi definido um ambiente simples de dimensões 7 por 9 metros, sendo destes 6 por 8 metros de área interna e 1 metro das paredes no perímetro. Estas dimensões foram definidas arbitrariamente para maior

facilidade na visualização do ambiente. A Listagem V.1 descreve a representação do ambiente no arquivo de configuração e a Figura V.1 ilustra uma visão superior do ambiente renderizado pelo módulo de visualização.

Listing V.1: Representação do ambiente no arquivo de configuração

```
environment: |
#####
# .....#
# .....#
# .....#
# .....#
# .....#
# .....#
#####
```

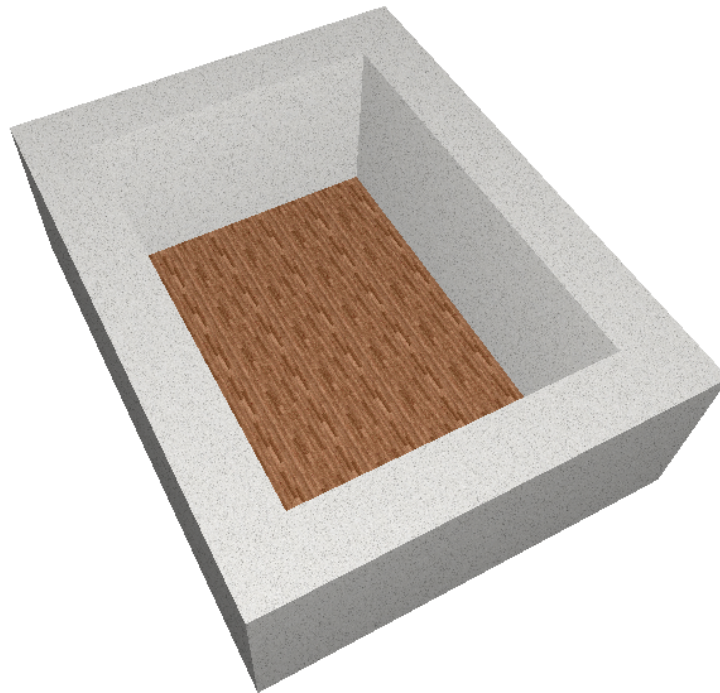


Figura V.1: Visão superior do ambiente descrito na configuração.

## V.2 Descrição dos dispositivos

Dado a descrição do ambiente apresentada na seção anterior, o próximo passo é posicionar os dispositivos sensores e atuadores no ambiente, e adicionar a respectiva configuração para cada dispositivo. Serão adicionados inicialmente dois dispositivos, um atuador e um sensor. O atuador será posicionado ao lado direito do ambiente no centro e o sensor será posicionado próximo ao centro

do ambiente. A Listagem V.2 demonstra a descrição do ambiente atualizada com os dispositivos, o caractere “a” será utilizado posteriormente para identificar a configuração dos atuador, e o caractere “c” será utilizado para configurar o sensor. Além disso, foi posicionado também o caractere “@” que indica a posição inicial do usuário.

Listing V.2: Representação do ambiente no arquivo de configuração

```
environment: |
#####
# .....#
# .....#
#.@.c...a#
# .....#
# .....#
#####
```

Com o posicionamento dos dispositivos, o próximo passo é definir as configurações específicas de cada dispositivo. A Listagem V.3 descreve as configurações dos dispositivos posicionados anteriormente. O atuador, representados pelos caracteres “a”, possui uma altura de **1,5** metros, a direção de emissão das partículas são para a esquerda, em direção a posição inicial do usuário, de forma que as partículas se aproxime do usuário no decorrer da simulação, velocidade inicial de **5,0m/s**, temperatura de **25°C**, distância para emissão das partículas em um raio de **0,2** metros, o fluido é do tipo gasoso, imitado em um intervalo de **0,05** milisegundos. As partículas emitidas pelo atuador são renderizadas com um tamanho de **0,2** metros, sendo as partículas da cor azul. O sensor representado pelo caractere “c” está posicionado a uma altura de **1** metro, sondando partículas em um raio de **1** metro de distância e irá escrever as informações obtidas das partículas em um arquivo chamado “sensor-c.csv” no formato CSV.

Listing V.3: Representação do ambiente no arquivo de configuração

```
actuators:
a:
  height: 1.5
  direction: [-1.0, 0.0, 1.0]
  initial_velocity: 5.0
  temperature: 25.0
  range: [0.2, 0.2, 0.2]
  fluid_type: Gaseous
  interval: 0.05
```

```

    particle:
      size: 0.02
      color: [0.0, 0.0, 1.0]
  b:
    height: 1.5
    direction: [-1.0, 0.0, -1.0]
    initial_velocity: 5.0
    temperature: 25.0
    range: [0.2, 0.2, 0.2]
    fluid_type: Gaseous
    interval: 0.05
    particle:
      size: 0.02
      color: [1.0, 0.0, 0.0]
sensors:
  c:
    height: 1.0
    range: [1.0, 1.0, 1.0]
    output: "sensor-c.csv"

```

### V.3 Descrição da simulação

Além da configuração dos dispositivos, para completarmos a simulação o último passo, é necessário definir os parâmetros gerais da simulação. Conforme definido no início deste capítulo, serão utilizados alguns valores de parâmetros pré-estabelecidos obtidos por meio de experimentação de forma que a estabilidade da simulação seja garantida. A Listagem V.4 define as configurações da simulação com base nos valores especificados na Tabela V.1. Além dos parâmetros definidos na tabela, é incluído um parâmetro adicional denominado “virtual\_particle\_direction”, utilizado para obter o vetor da partícula virtual utilizada para calcular as forças de compressão ao fazer a simulação de fluidos gasosos.

Listing V.4: Representação do ambiente no arquivo de configuração

```

simulation:
  step: 0.001
  radius: 0.0457
  mass: 0.02

```

```

gas_constant: 3.0
rest_density: 0.99829
delta: 0.4
thermal_conductivity: 0.598
small_positive: 0.0001
viscosity: 0.01
damping_coefficient: 100.0
damping_threshold: 1.0
radiation_half_life: 1000.0
buoyancy_coefficient: 0.8
buoyancy_direction: [0.0, 1.0, 0.0]
gravity: [0.0, -9.82, 0.0]
virtual_particle_direction: [0.0, 1.0, 0.0]

```

#### V.4 Cenário de simulação

Dada a configuração descrita nas seções anteriores, nesta seção será realizada a consolidação de todas as configurações apresentadas, formando assim a configuração completa que permitirá obter resultado desejado da simulação dado o cenário especificado. O cenário em questão é composto por dois dispositivos, mais especificamente um atuador e um sensor. O atuador encontra-se ao centro do lado direito, enquanto o sensor está posicionado próximo ao centro do ambiente. Durante a simulação, partículas serão emitidas diagonalmente pelo atuador, com a finalidade de se deslocarem em direção a posição inicial do usuário. É importante ressaltar que o usuário terá sua posição inicial ao lado esquerdo do ambiente. Porém, ao longo da simulação, essa posição será alterada de modo a explorar a simulação de forma mais abrangente. A Listagem V.5 apresenta a configuração consolidada que será utilizada como entrada para a demonstração da simulação.

Listing V.5: Representação do ambiente no arquivo de configuração

```

environment: |
#####
#.....#
#.....#
#.@.c..a#
#.....#
#.....#
#####

```



```
actuators:  
  a:  
    height: 1.5  
    direction: [-1.0, 0.0, 0.0]  
    initial_velocity: 5.0  
    temperature: 25.0  
    range: [0.2, 0.2, 0.2]  
    fluid_type: Gaseous  
    interval: 0.05  
    particle:  
      size: 0.02  
      color: [0.0, 0.0, 1.0]  
sensors:  
  c:  
    height: 1.0  
    range: [10.0, 1.0, 1.0]  
    output: "output.csv"  
simulation:  
  step: 0.001  
  radius: 0.0457  
  mass: 0.02  
  gas_constant: 3.0  
  rest_density: 0.99829  
  delta: 0.4  
  thermal_conductivity: 0.598  
  small_positive: 0.0001  
  viscosity: 0.01  
  damping_coefficient: 100.0  
  damping_threshold: 1.0  
  radiation_half_life: 1000.0  
  buoyancy_coefficient: 0.8  
  buoyancy_direction: [0.0, 1.0, 0.0]  
  gravity: [0.0, -9.82, 0.0]  
  virtual_particle_direction: [0.0, 1.0, 0.0]
```

A Figura V.2 apresenta uma imagem capturada em um determinado momento da simulação, na

qual é possível visualizar vários elementos e interações relevantes. No centro, encontra-se o atuador designado pelo símbolo “a”, emitindo partículas da cor azul. É possível observar o atuador emitir as partículas em uma trajetória linear em direção ao usuário no ambiente simulado. Conforme as partículas se movem, elas se dispersam e interagem com o ambiente. Cabe ressaltar que, no contexto da simulação, o sensor em si não é visualizado, sendo apenas uma entidade lógica utilizada pelo simulador para coletar os dados referentes as partículas e torná-los disponíveis para análise posterior.

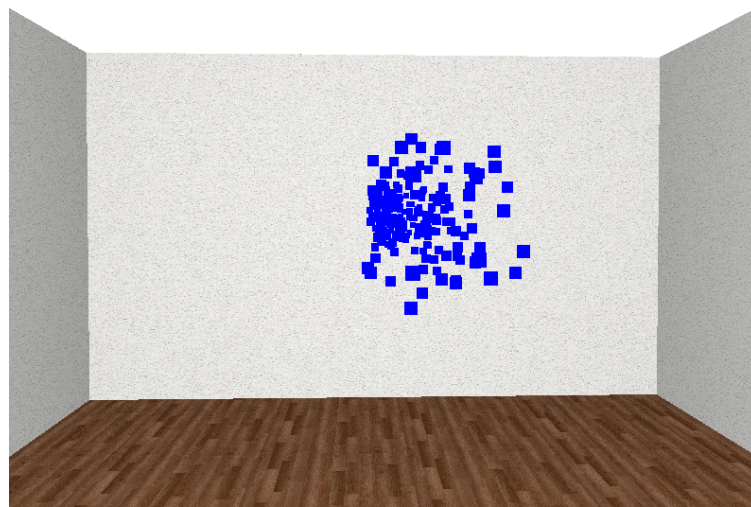


Figura V.2: Visão das partículas emitidas pelo atuador.

A Figura V.3 oferece uma perspectiva aérea abrangente da simulação, na qual se torna perceptível de forma mais evidente os contornos que demarcam os limites do domínio da simulação, sendo estes as paredes do ambiente. É importante destacar que a simulação é restrita apenas as partículas que se encontram dentro desse domínio delimitado, e qualquer partícula que ultrapasse esses limites é prontamente eliminada da simulação, pois não é considerada no contexto do modelo em questão.

Afim de observar o comportamento descrito anteriormente onde as partículas que ultrapassam os limites da simulação são eliminadas, foi posicionado um obstáculo no percurso das partículas. O obstáculo em questão é uma pilastra e é possível observar que as partículas que se chocam nesse obstáculo não podem ser mais observadas no percurso a sua frente, ou seja, não estão mais presentes na simulação. A largura do ambiente também foi ampliada para que haja espaço suficiente para que as partículas se disperssem e algumas delas não se choquem com o obstáculo e permaneçam por mais tempo na simulação. A Listagem V.6 apresenta a representação do ambiente após as alterações, as demais configuração permanecem as mesmas. A Figura V.4 ilustra a visão do usuário posicionado atrás do obstáculo de forma que é possível observar apenas as partículas que não colidiram com o obstáculo em questão, a Figura V.5 mostra uma perspectiva aérea da simulação onde é possível

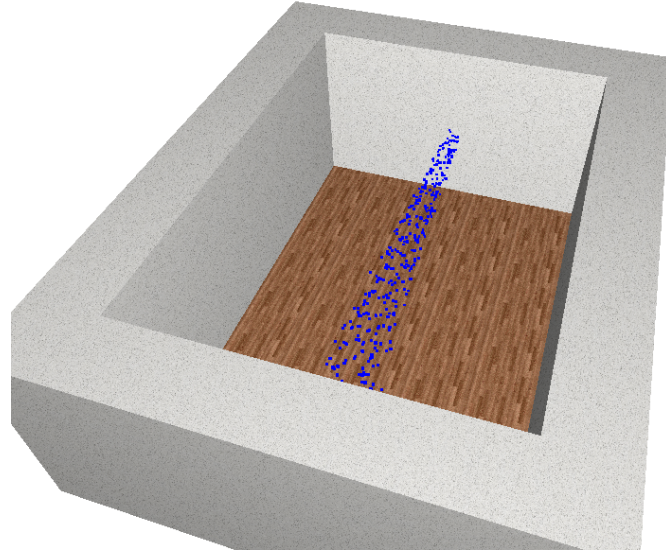


Figura V.3: Perspectiva aérea da simulação.

observar mais claramente o comportamento descrito.

Listing V.6: Representação do ambiente com obstáculo no arquivo de configuração

```
environment: |
#####
#.....#
#.....#
#.@.#c.....a#
#.....#
#.....#
#####
```

Para explorar cenários mais complexos onde existem múltiplos atuadores e as partículas emitidas por estes diferentes atuadores interagem umas com as outras o exemplo original descrito anteriormente foi modificado para incluir mais um atuador. O atuador original foi movido para uma das extremidades ao lado direito do ambiente. Um novo atuador representado pelo carácter “b” foi posicionado na extremidade oposta do mesmo lado do ambiente, emitindo partículas da cor vermelha e com o restante das configurações semelhantes ao primeiro atuador. A direção das partículas foi definida para que as partículas se desloquem na diagonal de forma que haja um ponto de convergência próximo ao centro do ambiente em que as partículas se encontram e interagem entre si. A Listagem V.7 define a nova configuração com o atuador adicional e as atualizações nas configurações do atuador original. As linhas marcadas com um “+” indicam as alterações realizadas na configuração original, evidenciando que com poucas alterações é possível explorar diferentes cenários e construir novas simulações.

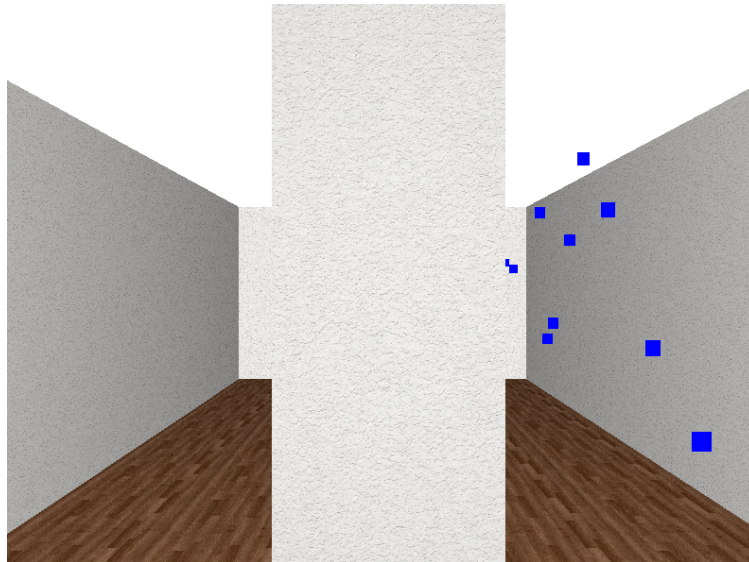


Figura V.4: Visão das partículas atrás do obstáculo.

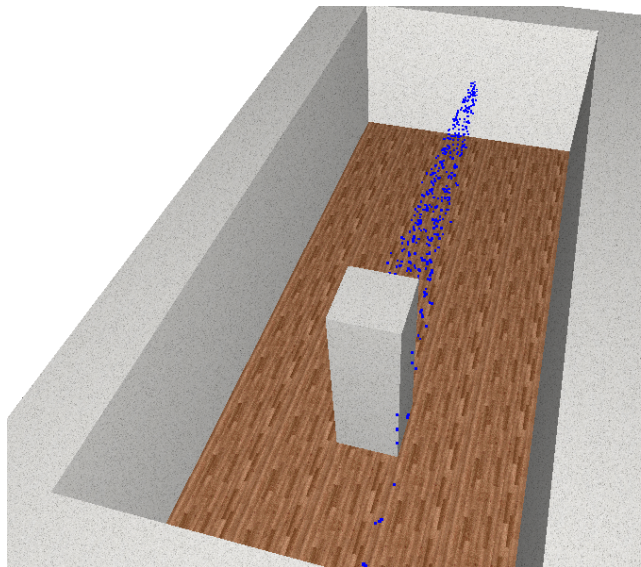


Figura V.5: Perspectiva aérea da simulação com obstáculo.

Listing V.7: Representação do ambiente no arquivo de configuração

```

environment: |
    #####
+   #.....a#
    #.....#
+   #.@.c...#
    #.....#
+   #.....b#
    #####
actuators:

```

```
a:
  height: 1.5
+  direction: [-1.0, 0.0, 1.0]
  initial_velocity: 5.0
  temperature: 25.0
  range: [0.2, 0.2, 0.2]
  fluid_type: Gaseous
  interval: 0.05
  particle:
    size: 0.02
    color: [0.0, 0.0, 1.0]
+ b:
+  height: 1.5
+  direction: [-1.0, 0.0, -1.0]
+  initial_velocity: 5.0
+  range: [0.2, 0.2, 0.2]
+  fluid_type: Gaseous
+  interval: 0.05
+  particle:
+    size: 0.02
+    color: [1.0, 0.0, 0.0]
sensors:
  c:
    height: 1.0
    range: [10.0, 1.0, 1.0]
    output: "output.csv"
simulation:
  step: 0.001
  radius: 0.0457
  mass: 0.02
  gas_constant: 3.0
  rest_density: 0.99829
  delta: 0.4
  thermal_conductivity: 0.598
  small_positive: 0.0001
```

```

viscosity: 0.01
damping_coefficient: 100.0
damping_threshold: 1.0
radiation_half_life: 1000.0
buoyancy_coefficient: 0.8
buoyancy_direction: [0.0, 1.0, 0.0]
gravity: [0.0, -9.82, 0.0]
virtual_particle_direction: [0.0, 1.0, 0.0]

```

A Figura V.6 apresenta uma imagem capturada em um determinado momento da simulação, na qual é possível visualizar as partículas de ambos os atuadores ultrapassando ponto de congência no centro do ambiente. Do lado esquerdo, encontra-se o atuador designado pelo símbolo “a”, emitindo partículas da cor azul, do lado direito, encontra-se o atuador designado pelo símbolo “b”, emitindo partículas da cor vermelha. É possível observar os atuadores emitirem as partículas em uma trajetória diagonal em direção ao centro do ambiente simulado. Conforme as partículas se movem, elas se dispersam, interagem com o ambiente e entre si em um ponto específico onde se encontram. Com base nos dados obtidos pelo sensor é possível observar como estas interações afetam os dados da partículas, por exemplo, a troca de temperatura, tanto com o ambiente como com as interações entre si, causando uma variação da temperatura inicial da emissão da partículas.

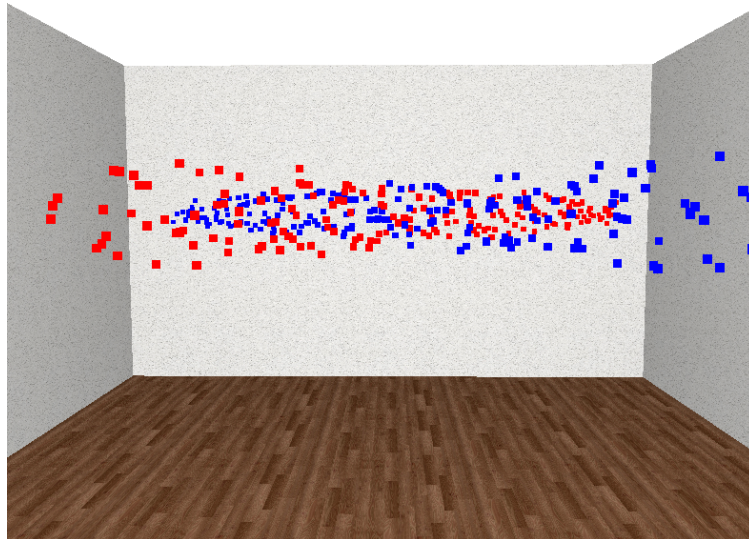


Figura V.6: Visão das partículas emitidas pelos atuadores.

A cada iteração da simulação, os sensores realizam uma verificação dentro do seu perímetro de sondagem para identificar a presença de alguma partícula. Caso alguma partícula seja detectada, os dados da partícula são inspecionados com o propósito de extrair informações precisas referente ao instante específico da simulação. Após a obtenção dos dados pertinentes da partícula, ocorre a

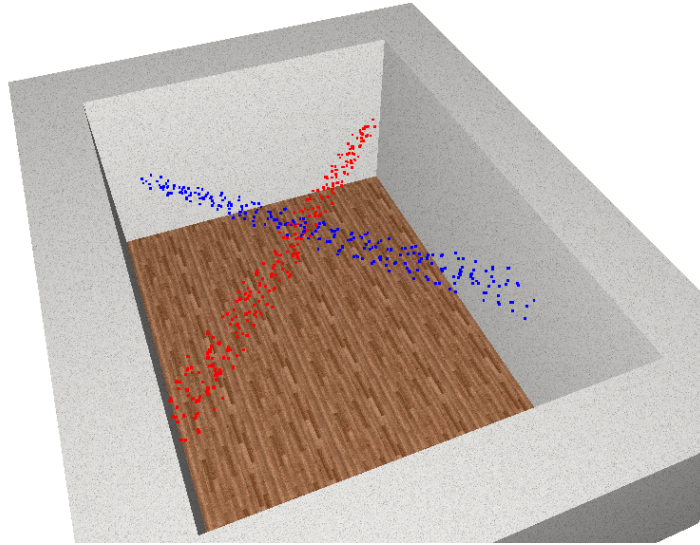


Figura V.7: Perspectiva aérea da simulação.

conversão para o formato CSV, sendo então registrado no arquivo de saída previamente definido nas configurações. Caso nenhum arquivo tenha sido especificado, as mesmas informações são apresentadas no registro de *log* da aplicação. A Tabela V.2 exemplifica um fragmento do arquivo de saída da simulação apresentada anteriormente, relativo ao sensor representado pelo símbolo “c” no exemplo acima. Cada linha do arquivo contém as principais informações da partícula naquele instante da simulação, tais como o símbolo do sensor que detectou a partícula, o tempo decorrido total da simulação, o tempo decorrido desde a emissão da partícula, sua posição, velocidade, aceleração, soma total das forças atuantes, densidade e temperatura.

Com base nos dados apresentados na Tabela V.2 podemos tirar algumas conclusões sobre o comportamento da simulação. Um usuário localizado na posição do sensor demoraria em torno de 5 segundos para perceber o efeito sensorial dado o tempo decorrido desde a emissão da partícula, portanto, para que a percepção do efeito sensorial esteja sincronizada com o conteúdo multimídia este tempo pode ser levado em consideração durante a elaboração da aplicação. Os valores das densidades do efeito sensorial podem ser utilizadas como referências para calcular a intensidade que um usuário iria perceber um efeito sensorial. Pode-se notar que houve uma perda de temperatura para o ambiente e com a interação entre as partículas, sendo esta diferente da temperatura de emissão das partículas, portanto, este parâmetro também pode ser ajustado para que o usuário perceba um efeito sensorial com a temperatura desejada pelos autores da aplicação.

Tabela V.2: Fragmento do arquivo de saída da simulação.

c	5.50	5.37	[5.93, 1.73, 3.00]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.52	5.38	[5.92, 1.73, 3.01]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.54	5.40	[5.92, 1.73, 3.01]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.55	5.47	[6.00, 1.48, 3.26]	[-4.87, -0.05, 4.90]	[463.76, -43.61, -351.76]	[7612531.50, -715918.38, -5773959.50]	16414.67	18.13
c	5.55	5.42	[5.91, 1.73, 3.02]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.57	5.48	[5.99, 1.48, 3.27]	[-4.87, -0.05, 4.90]	[463.70, -43.71, -351.31]	[7611477.50, -717441.12, -5766575.00]	16414.67	18.11
c	5.57	5.43	[5.91, 1.73, 3.02]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.59	5.50	[5.99, 1.48, 3.27]	[-4.87, -0.05, 4.90]	[463.64, -43.80, -350.85]	[7610423.00, -718964.50, -5759154.50]	16414.67	18.09
c	5.59	5.45	[5.90, 1.73, 3.03]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.60	5.52	[5.98, 1.48, 3.28]	[-4.87, -0.05, 4.90]	[400.98, -143.25, -242.31]	[6581924.00, -2351406.00, -3977518.00]	16414.67	18.08
c	5.60	5.47	[5.90, 1.73, 3.03]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.62	5.53	[5.98, 1.48, 3.28]	[-4.87, -0.05, 4.90]	[401.03, -143.38, -241.90]	[6582832.00, -2353488.00, -3970656.00]	16414.67	18.06
c	5.62	5.48	[5.89, 1.73, 3.04]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.64	5.55	[5.97, 1.48, 3.29]	[-4.87, -0.05, 4.90]	[463.42, -44.13, -349.38]	[7606826.50, -724411.75, -5735020.50]	16414.67	18.04
c	5.64	5.50	[5.89, 1.73, 3.04]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.65	5.57	[5.97, 1.48, 3.29]	[-4.87, -0.05, 4.90]	[463.38, -44.26, -348.89]	[7606228.00, -726462.12, -5726975.00]	16414.67	18.02
c	5.65	5.52	[5.88, 1.73, 3.05]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.67	5.58	[5.96, 1.48, 3.29]	[-4.87, -0.05, 4.90]	[463.38, -44.32, -348.51]	[7606160.50, -727483.50, -5720696.00]	16414.67	18.00
c	5.67	5.53	[5.88, 1.73, 3.05]	[-4.96, 0.05, 4.84]	[0.00, 5.38, 0.00]	[0.00, 88336.60, 0.00]	16414.67	19.00
c	5.67	5.47	[6.00, 1.55, 3.10]	[-4.88, -0.05, 4.83]	[427.40, -89.20, -740.29]	[7015597.50, -1464233.38, -12151666.00]	16414.67	18.11
c	5.69	5.60	[5.96, 1.48, 3.30]	[-4.87, -0.05, 4.90]	[463.31, -44.41, -348.04]	[7605109.00, -729025.38, -5713033.00]	16414.67	17.99



## Capítulo VI Considerações Finais

A inclusão de efeitos sensoriais em aplicações multimídia tem o potencial de aprimorar significativamente o QoE e a imersão do usuário. No entanto, compreender os desafios inerentes à criação de tais aplicações – como a sincronização da reprodução dos efeitos sensoriais no ambiente físico juntamente com a apresentação do conteúdo multimídia e a constante mudança do estado de um efeito sensorial de acordo com a interação do usuário – são fundamentais para garantir uma experiência rica e imersiva.

A sincronização dos efeitos sensoriais com o conteúdo multimídia é uma preocupação essencial, mas também é importante compreender como um efeito sensorial irá se comportar em diferentes ambientes físicos de forma que suas características sejam percebidas da maneira esperada pelos usuários, por exemplo, a intensidade de um aroma, a temperatura de um vento, ou a velocidade de um borrifo d'água.

Com este propósito, alguns trabalhos investigam a utilização de métodos de CFD para simular efeitos sensoriais de cheiro calculando a dissipação do fluido em um determinado ambiente virtual com a finalidade de usar os dados na aplicação para controlar a intensidade do aroma de acordo com a posição do usuário em determinado instante da simulação. Tais simulações demandam um alto poder computacional, portanto os cálculos devem ser realizados previamente e incorporados posteriormente nas aplicações, limitando a duração da execução aplicação ao número de iterações calculadas na simulação, e impossibilitando que possíveis alterações na simulação possam ocorrer durante a execução da aplicação, além da necessidade de calcular novamente a simulação para quaisquer modificações no ambiente simulado.

Uma evolução importante seria permitir simulações de efeitos sensoriais em tempo real para maior interatividade do usuário proporcionando uma melhor experiência, e permitindo aos autores das aplicações maior velocidade para experimentar e explorar diferentes cenários. Para alcançar simulações interativas em tempo real, o método SPH surge como uma solução eficiente. Apesar de ser menos realista do que os métodos tradicionais baseados em malha, o método requer menos recursos computacionais. Levando-se em conta que os atuadores que geram os efeitos sensoriais não são dispositivos muito precisos, dado o comprometimento entre velocidade e fidelidade visual, prefere-se produzir simulações interativas em tempo real mesmo com uma pequena perda de qualidade da solução.

Neste trabalho, propusemos uma abordagem híbrida para o método SPH que é capaz de simular fluidos não compressíveis, como líquidos, e fluidos gasosos, permitindo simular uma variedade de efeitos sensoriais, como borrifo d'água, névoa, cheiro, vento, etc. Dado este contexto, elaboramos um simulador que permite aos usuários definir o ambiente que se deseja simular, posicionar e especificar as características dos dispositivos sensores e atuadores, especificar os parâmetros da simulação, e explorar essa simulação em tempo real para verificar como os efeitos sensoriais irão se comportar em diferentes cenários. Além disso, os sensores capturam os dados relevantes dos efeitos sensoriais durante a execução permitindo que estes dados sejam analisados para tirar conclusões a respeito da simulação, assim como otimizar os parâmetros permitindo uma busca por soluções ótimas e aprimoramento dos resultados desejados.

Como prova de conceito da proposta, elaboramos um cenário para simular o comportamento de efeitos sensoriais de vento. Usamos alguns parâmetros como base para garantir que as simulações fossem estáveis e se comportassem conforme o esperado. A fim de aperfeiçoar a ferramenta desenvolvida, planejamos futuramente realizar experimentos para coletar dados sobre como os fluidos se comportam em um ambiente real para refinar esses parâmetros e estabelecer limites para que os autores possam configurar esses parâmetros de acordo com suas necessidades sem afetar a estabilidade da simulação.

Como resultado, esperamos que a ferramenta desenvolvida seja uma contribuição significativa para os autores de aplicações multimídia com efeitos sensoriais, e para os desenvolvedores que desejam incorporar efeitos sensoriais em suas aplicações e proporcionar uma experiência mais rica aos usuários. A compreensão aprofundada do comportamento de efeitos sensoriais em ambientes específicos e a obtenção de métricas para avaliar a percepção dos usuários são essenciais para a produção bem sucedida de aplicações imersivas. Com ferramentas que agilizem o desenvolvimento e permitam iterações mais rápidas, abre-se espaço para a experimentação e exploração de diferentes cenários, impulsionando a evolução contínua de aplicações multimídia com efeitos sensoriais.

## Referências Bibliográficas

- Apostolopoulos, J. G., Chou, P. A., Culbertson, B., Kalker, T., Trott, M. D. and Wee, S. [2012]. The road to immersive communication, *Proceedings of the IEEE* **100**(4): 974–990.
- Blinn, J. F. [1977]. Models of light reflection for computer synthesized pictures, *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pp. 192–198.
- Comşa, I.-S., Saleme, E. B., Covaci, A., Assres, G. M., Trestian, R., Santos, C. A. and Ghinea, G. [2019]. Do i smell coffee? the tale of a 360 mulsemmedia experience, *IEEE MultiMedia* **27**(1): 27–36.
- Covaci, A., Trestian, R., Saleme, E. B., Comsa, I.-S., Assres, G., Santos, C. A. and Ghinea, G. [2019]. 360 mulsemmedia: A way to improve subjective qoe in 360 videos, *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2378–2386.
- Ghinea, G., Timmerer, C., Lin, W. and Gulliver, S. R. [2014]. Mulsemmedia : State of the Art, Perspectives, and Challenges, *ACM Transactions on Multimedia Computing, Communications, and Applications* **11**(1s): 1–23.  
**URL:** <http://dl.acm.org/citation.cfm?doid=2675060.2617994>
- Gingold, R. A. and Monaghan, J. J. [1977]. Smoothed particle hydrodynamics: theory and application to non-spherical stars, *Monthly notices of the royal astronomical society* **181**(3): 375–389.
- Giorgi, G. [2018]. *Nonlinear hydrodynamic modelling of wave energy converters under controlled conditions*, PhD thesis, National University of Ireland Maynooth.
- Hosain, M. L. and Fdhila, R. B. [2015]. Literature review of accelerated cfd simulation methods towards online application, *Energy Procedia* **75**: 3307–3314. Clean, Efficient and Affordable Energy for a Sustainable Future: The 7th International Conference on Applied Energy (ICAE2015).  
**URL:** <https://www.sciencedirect.com/science/article/pii/S1876610215014824>
- Ishida, H., Matsukura, H., Yoshida, H. and Nakamoto, T. [2008]. Application of computational fluid dynamics simulation to olfactory display, *Proc. ICAT*, pp. 285–288.

- ISO/IEC [2016a]. Information technology – media context and control – part 1: Architecture, *ISO/IEC 23005-1:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016b]. Information technology – media context and control – part 2: Control information, *ISO/IEC 23005-2:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016c]. Information technology – media context and control – part 3: Sensory information, *ISO/IEC 23005-3:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016d]. Information technology – media context and control – part 4: Virtual world object characteristics, *ISO/IEC 23005-4:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016e]. Information technology – media context and control – part 5: Data format for interaction devices, *ISO/IEC 23005-5:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016f]. Information technology – media context and control – part 6: Common types and tools, *ISO/IEC 23005-6:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- ISO/IEC [2016g]. Information technology – media context and control – part 7: Reference software and conformance, *ISO/IEC 23005-7:2016*, ISO/IEC JTC 1/SC 29 Coding of audio, picture, multimedia and hypermedia information.
- Josué, M., Abreu, R., Barreto, F., Mattos, D., Amorim, G., dos Santos, J. and Muchaluat-Saade, D. [2018]. Modeling sensory effects as first-class entities in multimedia applications, *Proceedings of the 9th ACM Multimedia Systems Conference*, ACM, pp. 225–236.
- Kahol, K., Tripathi, P., Mcdaniel, T., Bratton, L. and Panchanathan, S. [2006]. Modeling context in haptic perception, rendering, and visualization, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **2**(3): 219–240.
- Lucy, L. B. [1977]. A numerical approach to the testing of the fission hypothesis, *The astronomical journal* **82**: 1013–1024.
- Matsukura, H. and Ishida, H. [2009]. Olfactory display: fluid dynamic considerations for realistic odor presentation, *Proceedings of the 15th Joint virtual reality Eurographics conference on Virtual Environments*, pp. 61–64.

- Matsukura, H., Yoshida, H., Ishida, H. and Nakamoto, T. [2009]. Interactive odor playback based on fluid dynamics simulation, *2009 IEEE Virtual Reality Conference*, IEEE, pp. 255–256.
- Müller, M., Charypar, D. and Gross, M. H. [2003]. Particle-based fluid simulation for interactive applications., *Symposium on Computer animation*, pp. 154–159.
- Murray, N., Ademoye, O. A., Ghinea, G. and Muntean, G.-M. [2017]. A tutorial for olfaction-based multisensorial media application design and evaluation, *ACM Comput. Surv.* **50**(5).  
**URL:** <https://doi.org/10.1145/3108243>
- Nakamoto, T., Hirasawa, T. and Hanyu, Y. [2020]. Virtual environment with smell using wearable olfactory display and computational fluid dynamics simulation, *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, pp. 713–720.
- Pozrikidis, C. et al. [1998]. *Numerical computation in science and engineering*, Vol. 6, Oxford university press New York.
- Ren, B., Yan, X., Yang, T., Li, C.-f., Lin, M. C. and Hu, S.-m. [2016]. Fast sph simulation for gaseous fluids, *The Visual Computer* **32**(4): 523–534.
- Rodrigues, E. C., Saleme, E. B. and Santos, C. A. S. [2021]. A haptic system for switching wind temperatures based on ultrasonic vibrations, peltier elements, and electrical resistances for multisensory applications, *Proceedings of the Brazilian Symposium on Multimedia and the Web, WebMedia '21*, Association for Computing Machinery, New York, NY, USA, p. 73–80.  
**URL:** <https://doi-org.ez108.periodicos.capes.gov.br/10.1145/3470482.3479638>
- Rodrigues, R. O., Josué, M. I., Abreu, R. S., Amorim, G. F., Muchaluat-Saade, D. C. and Santos, J. A. d. [2019]. A proposal for supporting sensory effect rendering in ginga-ncl, *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, pp. 273–280.
- Saleme, E. B., Covaci, A., Mesfin, G., Santos, C. A. and Ghinea, G. [2019]. Mulsemedia diy: A survey of devices and a tutorial for building your own mulsemedia environment, *ACM Computing Surveys (CSUR)* **52**(3): 1–29.
- Saleme, E. B. and Santos, C. A. S. [2015]. PlaySEM: A Platform for Rendering MulSeMedia Compatible with MPEG-V, *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web* pp. 145–148.  
**URL:** <http://doi.acm.org/10.1145/2820426.2820450>
- Soares, L. and Barbosa, S. [2009]. Programming in ncl 3.0: developing applications for the ginga middleware, digital tv and web.

- Tortell, R., Luigi, D., Dozois, A., Bouchard, S., Morie, J. F. and Ilan, D. [2007]. The effects of scent and game play experience on memory of a virtual environment, *Virtual Reality* **11**(1): 61–68.
- Waltl, M., Rainer, B., Timmerer, C. and Hellwagner, H. [2013]. An end-to-end tool chain for sensory experience based on mpeg-v, *Signal Processing: Image Communication* .
- Wesseling, P. [2009]. *Principles of computational fluid dynamics*, Vol. 29, Springer Science & Business Media.
- Yoon, K., Kim, S.-K., Han, J. J., Han, S. and Preda, M. [2015]. *MPEG-V: Bridging the virtual and real world*, Academic Press.