

Renderização Volumétrica e Realidade Virtual Aplicada a

# PROBLEMAS DE PERCOLAÇÃO NA INDÚSTRIA DO PETRÓLEO

Carlos Magno de Lima

Renderização Volumétrica e Realidade Virtual Aplicada a

# PROBLEMAS DE PERCOLAÇÃO NA INDÚSTRIA DO PETRÓLEO

Carlos Magno de Lima

Atena  
Editora  
Ano 2021

**Editora chefe**

Profª Drª Antonella Carvalho de Oliveira

**Editora executiva**

Natalia Oliveira

**Assistente editorial**

Flávia Roberta Barão

**Bibliotecária**

Janaina Ramos

**Projeto gráfico**

Camila Alves de Cremo

Daphynny Pamplona

Gabriel Motomu Teshima

Luiza Alves Batista

Natália Sandrini de Azevedo

**Imagens da capa**

iStock

**Edição de arte**

Luiza Alves Batista

2021 by Atena Editora

Copyright © Atena Editora

Copyright do texto © 2021 Os autores

Copyright da edição © 2021 Atena Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.

Open access publication by Atena Editora



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição Creative Commons. Atribuição-Não-Comercial-Não-Derivativos 4.0 Internacional (CC BY-NC-ND 4.0).

O conteúdo do texto e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva do autor, inclusive não representam necessariamente a posição oficial da Atena Editora. Permitido o *download* da obra e o compartilhamento desde que sejam atribuídos créditos ao autor, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

**Conselho Editorial****Ciências Exatas e da Terra e Engenharias**

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto

Profª Drª Ana Grasielle Dionísio Corrêa – Universidade Presbiteriana Mackenzie

Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás

Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná

Prof. Dr. Cleiseano Emanuel da Silva Paniagua – Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia

Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná



Profª Drª Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro  
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará  
Profª Dra. Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho  
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande  
Profª Drª Luciana do Nascimento Mendes – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte  
Prof. Dr. Marcelo Marques – Universidade Estadual de Maringá  
Prof. Dr. Marco Aurélio Kistemann Junior – Universidade Federal de Juiz de Fora  
Profª Drª Neiva Maria de Almeida – Universidade Federal da Paraíba  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Profª Drª Priscila Tessmer Scaglioni – Universidade Federal de Pelotas  
Prof. Dr. Sidney Gonçalo de Lima – Universidade Federal do Piauí  
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista



# Renderização volumétrica e realidade virtual aplicada a problemas de percolação na indústria do petróleo

**Diagramação:** Natália Sandrini de Azevedo  
**Correção:** Bruno Oliveira  
**Indexação:** Amanda Kelly da Costa Veiga  
**Revisão:** O autor  
**Autor:** Carlos Magno de Lima

## Dados Internacionais de Catalogação na Publicação (CIP)

L732 Lima, Carlos Magno de  
Renderização volumétrica e realidade virtual aplicada a problemas de percolação na indústria do petróleo / Carlos Magno de Lima. – Ponta Grossa - PR: Atena, 2021.

Formato: PDF  
Requisitos de sistema: Adobe Acrobat Reader  
Modo de acesso: World Wide Web  
Inclui bibliografia  
ISBN 978-65-5983-737-3  
DOI: <https://doi.org/10.22533/at.ed.373211712>

1. Realidade virtual. 2. Visualização científica. 3. Renderização volumétrica. 4. Percolação. 5. Indústria do Petróleo. 6. ZSweep. I. Lima, Carlos Magno de. II. Título.  
CDD 066.8

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

**Atena Editora**  
Ponta Grossa – Paraná – Brasil  
Telefone: +55 (42) 3323-5493  
[www.atenaeditora.com.br](http://www.atenaeditora.com.br)  
contato@atenaeditora.com.br



## DECLARAÇÃO DO AUTOR

O autor desta obra: 1. Atesta não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declara que participou ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certifica que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos; 4. Confirma a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhece ter informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autoriza a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.



## DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access*, *desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce*, ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.



## PREFÁCIO

No processo da recuperação do petróleo, a heterogeneidade das rochas exerce um impacto enorme na forma como os líquidos se movem no reservatório, definindo quanto petróleo pode ser recuperado. Com a finalidade de estudar esta variabilidade, a teoria da percolação, que descreve fenômenos envolvendo geometria e conectividade, é um modelo muito útil, especialmente a percolação dinâmica, que trabalha com um limiar de percolação não prefixado, que pode evoluir com o tempo. Ela viabiliza o estudo da conectividade e dos caminhos preferenciais auxiliando a caracterização e interpretação de fluxos em meios porosos, em particular, em fluxos de modelos geológicos. A técnica matemática utilizada quando aplicada a diversos modelos geológicos permite selecionar o modelo para estudo da simulação do fluxo. Os resultados da simulação de percolação são de caráter tridimensional e não têm nenhum significado físico até que sejam visualizados em imagens ou animações.

O Professor Carlos Magno apresenta, de forma inovadora, em sua tese o uso da Renderização Volumétrica associada a técnicas de Realidade Virtual baseadas em estereoscopia como solução a compreensão dos fenômenos ocorridos durante o processo de injeção, trazendo com isto valiosa contribuição para a interpretação e análise da formação dos aglomerados na percolação, o que certamente pode levar a decisões melhores sobre a exploração e recuperação de petróleo.

Seu trabalho nos apresenta de forma simples e pedagógica a metodologia e teoria empregada no desenvolvimento da aplicação, tornando-se assim uma referência para o ensino das técnicas de Percolação Dinâmica. Os Ambientes Virtuais, Técnicas de Estereoscopia, assim como a Renderização Volumétrica foram introduzidos de forma singular, permitindo a compreensão do seu papel fundamental como ferramenta tecnológica a ser utilizada no estudo de fenômenos complexos modelados matematicamente e impossíveis de serem visualizados e compreendidos sem a aplicação desta técnica em realidade virtual.

A aplicação ZSweepVR foi especialmente desenvolvida para o uso da Renderização Volumétrica Direta em ambientes de Realidade Virtual aplicados a Percolação Dinâmica. Trata-se de situações que envolvem uma quarta dimensão, que é o tempo. Assim, permite-se que qualquer parte dos dados seja vista, incluindo as estruturas e os detalhes internos, permitindo que o usuário siga a evolução da formação dos aglomerados, e as interações entre eles. Estas interações ocorrem dentro do conjunto de dados volumétricos, sendo só possível sua visualização a partir de técnicas de Renderização Volumétrica, adicionando com isto mais realismo, com os detalhes do interior dos dados.

Este trabalho merecidamente recebeu o Prêmio Petrobras de Tecnologia pela sua contribuição para o desenvolvimento tecnológico da Indústria do Petróleo.

Prof. Joaquim Elias de Freitas  
Departamento de matemática da UFRN

## SUMÁRIO

<b>RESUMO</b> .....	<b>1</b>
<b>ABSTRACT</b> .....	<b>2</b>
<b>CAPÍTULO 1</b> .....	<b>3</b>
Introdução .....	3
Organização do Texto .....	5
<b>CAPÍTULO 2</b> .....	<b>6</b>
Trabalhos Relacionados .....	6
Visualização Imersiva .....	6
Visualização de Simulações de Percolação .....	7
<b>CAPÍTULO 3</b> .....	<b>9</b>
Teoria de Percolação .....	9
Percolação dinâmica .....	12
Topologia .....	16
Fronteiras .....	17
Funcionamento da Percolação Dinâmica .....	18
Aplicação de Percolação na Indústria do Petróleo .....	20
<b>CAPÍTULO 4</b> .....	<b>22</b>
Ambientes Virtuais Imersivos .....	22
Realidade Virtual .....	22
Ambientes Virtuais Imersivos Colaborativos .....	23
Sensação de Profundidade .....	25
Gerando Imagens Estéreo .....	27
Observando Imagens Estéreo .....	32
Visualização Volumétrica .....	35
O Algoritmo ZSweep .....	38
Dados Volumétricos .....	40
<b>CAPÍTULO 5</b> .....	<b>42</b>
ZSweep em Ambiente Imersivo .....	42

O Processo de Renderização Volumétrica.....	42
O Processo de Renderização Volumétrica: Geometria.....	43
O Processo de Renderização Volumétrica: Renderização.....	46
Geração do Par Estereoscópico.....	47
Detalhes de Implementação.....	49
<b>CAPÍTULO 6.....</b>	<b>52</b>
ZSweepVR.....	52
Renderização Volumétrica e Realidade Virtual aplicados a percolação dinâmica.....	52
O Sistema ZSweepVR.....	53
Simulador de Percolação.....	56
Servidor de Renderização.....	57
<b>CAPÍTULO 7.....</b>	<b>61</b>
Experimento e Resultados.....	61
Metodologia.....	61
Experimento e Análise de Resultados.....	63
Renderização de Superfície.....	65
Renderização Volumétrica.....	69
Imagens Estéreo.....	77
Conclusão e Perspectivas.....	88
<b>APÊNDICES.....</b>	<b>91</b>
Apêndice A - Modelo de Iluminação em Renderização Volumétrica.....	91
Apêndice B - Trabalhos Publicados e Prêmio Petrobrás de Tecnologia.....	94
<b>REFERÊNCIAS.....</b>	<b>95</b>
<b>SOBRE O AUTOR.....</b>	<b>99</b>

## RESUMO

No processo da recuperação do petróleo, a heterogeneidade das rochas exerce um impacto enorme na forma como os líquidos se movem no reservatório, definindo quanto petróleo pode ser recuperado. A fim de estudar esta variabilidade, a teoria da percolação, que descreve fenômenos envolvendo geometria e conectividade é um modelo muito útil. Os resultados da simulação de percolação são de caráter tridimensional e não têm nenhum significado físico até que sejam visualizados em imagens ou animações. Embora ferramentas poderosas e sofisticadas de visualização tenham sido desenvolvidas, estas transformam grandes volumes de dados em imagens 2D. A fim de interpretarmos os dados como eles aparecem no mundo real, técnicas de realidade virtual baseadas principalmente em estereoscopia podem ser usadas. Neste trabalho, propomos uma ferramenta interativa, denominada ZSweepVR, baseada em técnicas de realidade virtual, que permite uma melhor compreensão dos dados volumétricos gerados por simulações de percolação dinâmica. O sistema desenvolvido tem a capacidade de renderizar imagens utilizando duas técnicas diferentes: renderização de superfícies e renderização de volumes. A renderização de superfície é realizada utilizando diretivas OpenGL, enquanto que a renderização volumétrica é realizada pelo algoritmo de renderização volumétrica direta ZSweep. No caso da renderização volumétrica, implementamos algoritmos para gerar imagens estéreo. Nós também propomos melhorias no algoritmo original de percolação dinâmica visando melhorar sua eficiência. Aplicamos as ferramentas desenvolvidas a dados de campos maduros, obtendo resultados satisfatórios. O uso de imagens estereoscópicas e volumétricas trouxe contribuições valiosas para a interpretação e análise da formação dos aglomerados na percolação, o que certamente pode levar a decisões melhores sobre a exploração e recuperação de petróleo.

## ABSTRACT

In the recovering process of oil, rock heterogeneity has a huge impact on how fluids move in the field, defining how much oil can be recovered. In order to study this variability, percolation theory, which describes phenomena involving geometry and connectivity are the bases, is a very useful model. Result of percolation is tridimensional data and have no physical meaning until visualized in form of images or animations. Although a lot of powerful and sophisticated visualization tools have been developed, they focus on generation of planar 2D images. In order to interpret data as they would be in the real world, virtual reality techniques using stereo images could be used. In this work we propose an interactive and helpful tool, named ZSweepVR, based on virtual reality techniques that allows a better comprehension of volumetric data generated by simulation of dynamic percolation. The developed system has the ability to render images using two different techniques: surface rendering and volume rendering. Surface rendering is accomplished by OpenGL directives and volume rendering is accomplished by the ZSweep direct volume rendering engine. In the case of volumetric rendering, we implemented an algorithm to generate stereo images. We also propose enhancements in the original percolation algorithm in order to get a better performance. We applied our developed tools to a mature field database, obtaining satisfactory results. The use of stereoscopic and volumetric images brought valuable contributions for the interpretation and clustering formation analysis in percolation, what certainly could lead to better decisions about the exploration and recovery process in oil fields.

## INTRODUÇÃO

As reservas de petróleo mundiais a cada dia se tornam mais escassas. O desafio então é aumentar a taxa de sucesso de encontrar e recuperar o petróleo remanescente nos campos maduros, isto é, naqueles que já pararam de produzir espontaneamente.

Na fase inicial de produção, os poços podem produzir naturalmente, sendo o petróleo empurrado pela pressão dos gases ou retirado por um simples bombeamento. Mesmo com toda tecnologia a serviço do homem, aproximadamente 60% do total do petróleo contido em uma área permanece no solo. Um dos métodos utilizados na recuperação desse petróleo é o de injeção de fluido em poços injetores que invade os poros, expulsando o óleo do seu interior para as células vizinhas, formando uma frente que vai progressivamente ocupando o reservatório até a expulsão do óleo em poços produtores.

Os campos de produção de petróleo são formações extremamente complexas que contêm estruturas geológicas em todas as escalas de tamanho, de pequenos poros até aglomerados com extensão de quilômetros, na maioria dos reservatórios. Neste contexto, os espaços porosos da rocha sedimentar poderão ser preenchidos ou não com o petróleo. As rochas ali contidas são estruturas heterogêneas, com propriedades tais como a permeabilidade, porosidade entre outras que podem variar espacialmente ao longo de todo o reservatório. A heterogeneidade de um campo tem um impacto fundamental em como os líquidos se movem e em definir quanto óleo pode ser recuperado do reservatório.

A fim de estudar esta variabilidade, pode-se usar a teoria da percolação [55], que é um modelo bem sucedido para descrever fenômenos nos quais a geometria e a conectividade são fundamentais. Com o uso desta teoria, somos capazes de modelar a distribuição espacial dos tipos da rocha e a conectividade entre estes meios, possibilitando a descrição do fluxo de fluidos em meios porosos.

Diversas técnicas de simulação utilizando a teoria da percolação têm sido propostas, dentre elas exploraremos o modelo de percolação dinâmica. Os resultados da simulação do modelo dinâmico de reservatórios de petróleo são de formato tridimensional e dependente do tempo, mas não têm nenhum significado físico até que sejam visualizados em imagens ou em animações. Daí a justificativa do uso de técnicas de visualização científica que possibilitem a análise espacial e temporal de um número maior de variáveis, visando auxiliar a tomada de decisão.

As técnicas de visualização científica são extensivamente reconhecidas como uma maneira eficaz de compreender e de interpretar conjuntos de dados volumétricos de

grande tamanho resultantes de simulações científicas. Devido a vasta quantidade de dados gerada por estes processos, torna-se mais eficaz a visualização e interpretação dos mesmos a partir de uma imagem. Embora ferramentas poderosas e sofisticadas de visualização tenham sido desenvolvidas, estas transformam grandes volumes de dados em imagens 2D ou imagens resultantes de projeções dos dados tridimensionais. A fim de interpretarmos os dados de forma mais realística, técnicas de Realidade Virtual são utilizadas, baseadas principalmente na visualização de pares estereoscópicos.

Estas tecnologias vêm mudando radicalmente a forma como as empresas e universidades conduzem seus trabalhos de pesquisa na área de simulação e visualização. Podemos contar hoje com os ambientes colaborativos baseados em sistemas de realidade virtual imersiva, tais como: Cavernas (“CAVES”), ambientes virtuais colaborativos com telas planas (“*Large Flat Screens*”) e telas curvas (“*Curve Screens Theaters*”) [58].

Os ambientes virtuais proporcionam a criação de mundos tridimensionais imersivos e interativos que são utilizados no estudo de problemas complexos envolvendo equipes multidisciplinares.

Neste trabalho, nós propomos uma ferramenta interativa, denominada ZSweepVR, baseada em técnicas de Renderização Volumétrica e no uso de realidade virtual, que permite uma compreensão melhor dos dados volumétricos gerados por simulações de percolação dinâmica. O sistema ZSweepVR é baseado em técnicas de Renderização Volumétrica direta que permite que qualquer parte dos dados seja vista, incluindo as estruturas e os detalhes internos. Em um exemplo particular de percolação dinâmica, a simulação permite que o usuário siga a evolução da formação dos aglomerados, e as interações entre eles. Estas interações ocorrem dentro do conjunto de dados volumétricos, sendo possível sua visualização somente a partir do uso de técnicas de Renderização Volumétrica, adicionando com isto mais realismo, com detalhes do interior da base de dados.

O sistema ZsweepVR contém dois módulos principais: um servidor de simulação e um servidor de visualização. Além disso, o ZSweepVR pode renderizar imagens com duas técnicas diferentes: Renderização de superfície e Renderização de volume. A renderização de superfície é feita pelo módulo principal a partir de diretivas OpenGL [6], enquanto que o rendering volumétrico é realizado pelo algoritmo de renderização Volumétrica ZSweep [14]. Esta característica do ZSweepVR permite ao usuário maior flexibilidade, com uma visualização mais pobre em detalhes, porém mais rápida, e uma visualização mais precisa e rica em detalhes, porém mais lenta, dos aglomerados de percolação.

Nossa contribuição principal é desenvolver uma ferramenta para o estudo do fluxo do óleo em meios porosos, que integra a análise por percolação em um ambiente de visualização imersiva. Esta integração foi feita unindo a saída da simulação à entrada do sistema de visualização na memória principal. Esta união melhora o desempenho do ZSweepVR, porque evita a conversão de dados e as operações de E/S entre a simulação e os usuários da visualização, ao fornecer uma maneira melhor de representar e interagir com os elementos do modelo de percolação dinâmica.

Aplicamos nossa ferramenta a uma base de dados de campos maduros obtida a partir de estudos geostatísticos que incluem a interpolação dos dados de perfil dos poços e da interpretação geofísica de parâmetros petrofísicos, tais como: a permeabilidade, a porosidade, e a saturação de água e de óleo. Nossa ferramenta gera imagens em tempo real, detalhando o interior da base de dados Volumétrica e fornecendo uma melhor visualização da evolução e conexões entre aglomerados de percolação. ZSweepVR traz contribuições valiosas para a interpretação do processo de formação dos aglomerados, melhorando o processo de tomada de decisão e auxiliando na escolha da estratégia a ser adotada, relativa aos poços injetores e produtores.

## ORGANIZAÇÃO DO TEXTO

O texto desta tese está organizado da seguinte forma. No Capítulo 2, relacionamos nosso trabalho com trabalhos precedentes em duas áreas distintas: visualização imersiva e visualização de simulações de percolação. Descrevemos no Capítulo 3, a teoria da percolação e o modelo de percolação dinâmica usado nesta tese.

No Capítulo 4, abordamos dois temas que formam a base teórica para o desenvolvimento de nosso trabalho: *Realidade Virtual* e *Visualização Volumétrica*. Utilizando esta base conceitual, descrevemos no Capítulo 5, como foram realizadas as implementações das técnicas de projeção em perspectiva e Geração de imagens estereoscópicas na API ZSweep.

Nos Capítulos 6 e 7 abordamos a Implementação do sistema ZSweepVR e apresentamos experimentos e resultados validando, com isso, a metodologia adotada para Visualização de dados oriundos de simulações de percolação utilizada na exploração e recuperação do petróleo.

Finalmente, no Capítulo 8, concluímos, apresentando as principais contribuições, bem como, direcionando a execução de possíveis trabalhos futuros.

## TRABALHOS RELACIONADOS

O uso de técnicas de Realidade Virtual e Visualização Volumétrica é de grande interesse para pesquisadores e pode ser usada em diversas aplicações tais como: na indústria, medicina, diversão, comércio, etc. [58]. Uma das razões para este interesse é que a cada dia mais informação fica disponível para ser interpretada e com menos tempo hábil para fazê-la. A tradução destes dados e informações para imagens é essencial para que o ser humano consiga processá-los e interpretá-los mais rapidamente, com ganho de produtividade para pesquisa e atividades industriais.

O estudo de técnicas de realidade virtual na visualização de resultados de simulação utilizando percolação dinâmica é um esforço multidisciplinar. Visa o desenvolvimento de uma abordagem para melhor visualizar e analisar o fluxo no processo da recuperação de petróleo. Poucos foram os trabalhos desenvolvidos utilizando técnicas de Realidade Virtual para visualizar dados de simulação e na área de visualização de modelos de percolação para investigar diversos fenômenos, entre eles a penetração de fluidos em meios porosos. Entretanto, tanto quanto nós sabemos, nenhum destes trabalhos integra técnicas de Renderização Volumétrica direta e Realidade Virtual para analisar resultados de simulação utilizando a teoria da percolação. Neste Capítulo relacionamos nosso trabalho com trabalhos precedentes em Visualização imersiva e Visualização de simulações de Percolação.

## VISUALIZAÇÃO IMERSIVA

Embora diversos sistemas comerciais e de resultado de pesquisas em visualização volumétrica tenham sido desenvolvidos [2, 27, 44, 49, 51], o uso da Realidade Virtual com Visualização Volumétrica ainda não foi explorado inteiramente.

O software Crumbs [4] é baseado em um ambiente de realidade virtual CAVE para visualização, exploração e identificação de estruturas biológicas complexas. O sistema CAVE5D [25] é uma implementação baseada na biblioteca de Visualização Vis5D [27] voltado para ambientes imersivos CAVE. A biblioteca Vis5d provê técnicas poderosas de visualização de dados multidimensionais originados de bases de dados atmosféricas, oceanográficas e de outras fontes. O software CAVE6D [37] é uma extensão do sistema CAVE5D desenvolvido para ambientes colaborativos de visualização imersiva interativa.

O sistema 3DIVE [3] é uma ferramenta de visualização e manipulação de dados volumétricos. Ela combina os conceitos da manipulação baseada em regiões

e filtragem de dados em um ambiente imersivo, muito útil para a visualização de dados médicos.

O projeto VR<sup>2</sup> [18] integra técnicas de Renderização Volumétrica com realidade virtual usando aplicativo Java para visualização, chamado RTVR [45]. O RTVR pre-filtra os dados para rapidamente renderizá-los, porém não executa Renderização de alta qualidade.

O trabalho de Gudukbay e de Yilmaz [24] propõe uma estrutura ( framework ) para a visualização estereoscópica de modelos do terreno em grande escala. Consegue taxas elevadas de visualização, mas usa somente renderização de superfície para gerar as imagens estereoscópicas. Há também alguns ambientes virtuais sofisticados onde a visualização é realizada usando hardware gráfico tridimensional avançado [31, 56].

Todos estes sistemas, entretanto, usam grades estruturadas para representar os dados volumétricos. As grades estruturadas são amplamente usadas em aplicações médicas, onde as estruturas anatômicas que aparecem na varredura feita com MRI e CT normalmente são internas e representadas por grades estruturadas e regulares. Observamos também que avanços consideráveis têm sido feitos em busca do realismo em ambientes imersivos para aplicações médicas.

Diferentemente dos dados oriundos de aplicações médicas, as fontes de dados geológicos, são baseadas em interpretação sísmica, horizontes, falhas, mapas de batimetria, propriedades de rocha, análise de perfis de poços, etc. Estes geralmente compõem bases de dados volumétricos com milhões de elementos, e são melhor representados por grades irregulares. A exemplo disto, temos os resultados da simulação por percolação, que recebe dados oriundos de análise de perfis de poços, análises geoestatísticas, entre outros e gera dados que são melhor representados por grades irregulares.

O trabalho de Dorn *et al.* [13], propõe como plataforma um ambiente imersivo para planejamento de poços, a fim reduzir tempos de ciclo do planejamento na Indústria de petróleo e gás. Este é provavelmente o trabalho que mais se relaciona com o nosso, entretanto, sua Visualização é baseado na ferramenta gOcad [57], um software comercialmente disponível que não fornece técnicas de Renderização Volumétrica direta, apenas Renderização de superfícies.

## **VISUALIZAÇÃO DE SIMULAÇÕES DE PERCOLAÇÃO**

Grande número de modelos relacionados com a teoria da percolação foram desenvolvidos para atender a vários domínios de aplicação [34, 48, 47, 64]. Entretanto, a visualização dos resultados destes modelos recebeu pouca atenção. Adler e Lev [1] desenvolveram um modelo de percolação chamado *Bootstrap* e utilizou o software de

Visualização AViz, projetado originalmente para simulações atomísticas. As imagens obtidas por Adler e Lev, entretanto, mostram apenas a natureza compacta dos aglomerados. Não fornecem imagens de renderização volumétrica direta do meio poroso.

Na área do petróleo, o trabalho de George e Kovsky [19] investiga escalas de porosidade em fenômenos de fluxo em micro modelos, variando a pressão e a temperatura. Sagar e Castanier [54] apresentam resultados de visualização dos níveis de porosidade da estabilidade de espuma na presença do óleo. Sua investigação visual, entretanto, encontra-se em imagens de alta resolução dos microscópios, não em imagens tridimensionais, resultados de simulação. O trabalho de King et al. [36] usa a teoria da percolação para estimar as taxas de fluxo em reservatórios de hidrocarbonetos, faz estimativas muito rápidas, mas não fornece nenhum resultado visual. Dorn et al. [11] aplicou Visualização Volumétrica com atributos de horizonte aos dados sísmicos interpretados, seu ambiente, entretanto, não é imersivo.

Assim, acreditamos que nossa proposta é inovadora e será de grande valia na análise de dados da Indústria do petróleo. Como será abordado, realizamos implementações diversas propondo novas funções, visando adaptar o ZSweep para análise de Percolação, bem como propomos mudanças fundamentais no algoritmo de percolação dinâmica, visando sua otimização e integração com o sistema de Visualização.

## TEORIA DE PERCOLAÇÃO

A teoria da percolação usa modelos matemáticos bem definidos para descrever fenômenos nos quais a geometria e a conectividade são fundamentais [62]. Ela é oriunda do estudo do fenômeno de transporte de fluídos em meios porosos. De uma forma simples, diz-se que um fluído percola através de um meio físico quando existe um fluxo entre os seus extremos. Entre vários dos fenômenos estudados que utilizam a teoria da percolação, estão: a transição de fase condutor/isolante, a propagação de fogo em florestas, o estudo de fraturas e corrosão, fluxos em meios porosos e a recuperação de petróleo pela injeção de água e/ou gás.

Há várias formas de percolação. Destacaremos a percolação por sítios, Percolação por ligações e a percolação por ligações e sítios. Define-se como sítios os pontos da rede e como ligações os segmentos unitários que ligam dois sítios (Figura 3.1).

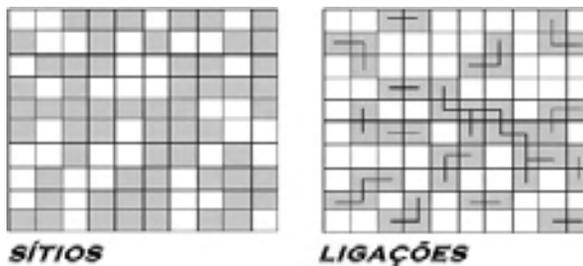


Figura 3.1: Formas de percolação

Na percolação por sítios, consideramos um modelo similar a um tabuleiro de xadrez. Cada célula ou sítio tem uma probabilidade de estar ocupado igual a  $p$ , e de estar vazia igual a  $1-p$ . Esta regra é aplicada a todos os sítios. Para valores pequenos de  $p$ , como podemos observar na Figura 3.2a, a maioria dos sítios ocupados se encontram isolados. A medida que  $p$  aumenta, os sítios ocupados formam agrupamentos (Figura 3.2b).

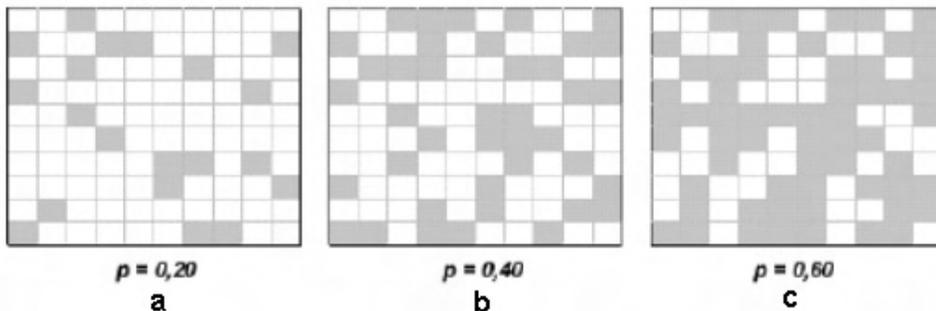


Figura 3.2: Modelo percolação por sítios

Por definição, dois sítios ocupados estão conectados se eles possuem um lado em comum. Ao conjunto de todos os sítios conectados a um mesmo sítio ocupado denomina-se aglomerado. Um aglomerado percolante é aquele que contém sítios nas duas faces opostas do tabuleiro. Quando isto ocorre dizemos que a rede percola (Figura 3.2c). A probabilidade média para a qual o sistema comece a percolar numa rede infinita é chamado limiar de Percolação e é representado por  $p_c$ . A Figura 3.3 mostra os valores de limiar de percolação para diversos tipos de rede. No caso de redes quadradas de sítios, é demonstrado que  $p_c = 0,5927$  [62, 22].

Percolation		Lattice
bond - $p_c$	site - $p_c$	
$2 \sin \frac{\pi}{18}$	$\frac{1}{2}$	Triangle
$\frac{1}{2}$	0.5927	Square
0.2488	0.3116	Cubic

Figura 3.3: Limiar de percolação

Na Percolação por ligações, de forma análoga, define-se a probabilidade de uma ligação estar ativa igual a  $p$  e de estar inativa igual a  $1-p$ . Uma ligação é uma conexão entre sítios primeiros vizinhos (Figura 3.4).

Duas ligações estão conectadas quando existe pelo menos um caminho de ligações ativas entre elas. Ao conjunto de todas as ligações conectadas a uma mesma ligação ativa, chama-se de aglomerado. O sistema percola quando existe um aglomerado com ligações ativas conectando faces opostas (Figura 3.4).

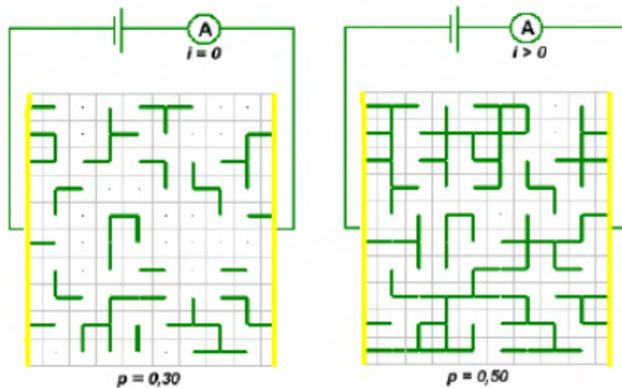


Figura 3.4: Modelo percolação por ligações

O modelo matemático para percolação 3D é composto de uma grade espacial, onde os sítios ou células são ocupadas aleatoriamente com uma probabilidade de ocupação  $p$ . Os aglomerados neste caso são conjuntos de sítios ocupados que são conectados entre si através de suas fronteiras, duas células estão conectadas se existe uma face em comum nas direções  $x$ ,  $y$  ou  $z$ . (Ver Figura 3.5).

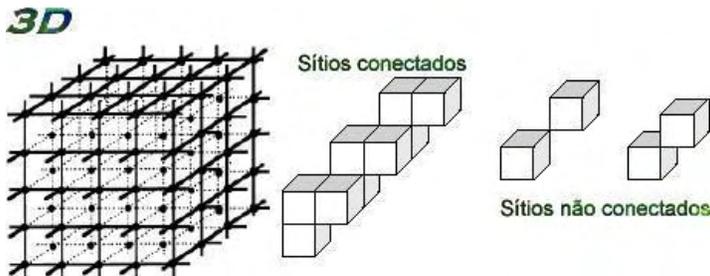


Figura 3.5: Modelo percolação 3d

Um aglomerado percolante é aquele que conecta um sítio pertencente a uma face da grade espacial 3D a outro sítio na face oposta. Para que possa existir uma conexão entre duas faces opostas da grade espacial, necessariamente deverá existir uma probabilidade  $p$ , tal que  $p$  seja maior ou aproximadamente igual a  $p_c$ . No caso de uma rede cúbica de ligações, é demonstrado que  $p_c = 0,2488$  [62, 22].

O modelo de percolação padrão de sítios e/ou ligações é considerado estático, ou seja, os sítios da rede são ocupados e os aglomerados gerados com uma probabilidade  $p$  fixa. Isto apenas possibilita o estudo do aglomerado formado com a probabilidade  $p$  definida. Para determinar quando o sistema percola, diversas simulações

são necessárias, tornando o sistema inadequado para a análise de eventos temporais. Diversos algoritmos têm sido desenvolvidos, buscando solucionar este problema. Dentre eles se encontram a percolação por espalhamento [38], percolação por invasão [64], *stirred percolation* [23], entre outros.

Estes modelos são extensões da percolação padrão e são dependentes do tempo, mas a sua conexão com a percolação padrão se restringe a alguns casos ou situações particulares [9]. Nosso trabalho explora um modelo que preserva sua identidade com a percolação padrão em cada instante, permitindo assim estudar as flutuações de grandezas envolvidas no sistema.

## PERCOLAÇÃO DINÂMICA

Freitas *et al* [10, 9] propuseram o modelo chamado de percolação dinâmica, onde o processo de percolação é considerado dependente do tempo, ou seja a probabilidade  $p$  é interpretada como parâmetro temporal. A probabilidade  $p$  de ocupação aumenta linearmente com o tempo. Desta maneira, é possível investigar a evolução de propriedades físicas e como o tamanho dos aglomerados evoluem no tempo. Diferentemente da percolação padrão, onde a formação dos aglomerados ocorre de forma estática, a percolação dinâmica gera mais cenários, possibilitando a avaliação de eventos temporais.

Diversas abordagens numéricas têm sido desenvolvidas com a finalidade de gerar os aglomerados e outras estruturas inerentes ao processo de percolação. Dentre as principais, podemos citar os algoritmos de Leath [38], de Hoshen-Kolpeman [30] e o método de Newman e Ziff [46]. Este último é particularmente importante pois representa, juntamente com o algoritmo implementado por Freitas *et al*, índices de melhor eficiência na determinação do limiar de percolação  $p_c$  [8].

A abordagem da percolação dinâmica descreve redes finitas e discretas e que são classificadas como hiper cúbicas, ou seja redes que são uma generalização do hipercubo, que por sua vez é uma generalização de um cubo para um espaço de dimensão  $d$ . Quando a dimensão da rede  $d$  é igual a 2, o hiperparalelepípedo reto é um retângulo, quando a dimensão  $d$  é igual a 3 é um paralelepípedo reto, e quando todas as arestas são iguais, um cubo.

Para facilitar as interpretações geométricas pode-se considerar quadrados para  $d = 2$  e cubos para  $d = 3$ . As redes finitas correspondem a um conjunto bem definido de sítios e/ou ligações com uma certa regularidade geométrica.

## Estrutura da Rede

Considere-se agora uma rede finita definida por um hiperparalelepípedo de dimensão  $d$  (Ver Figura 3.6), determinado pela sua diagonal principal  $\overline{AB}$ , onde:

- $A = (a_1, a_2, \dots, a_d)$  e  $B = (b_1, b_2, \dots, b_d)$
- com,  $a_i$  e  $b_i$ , números inteiros para  $a_i \leq b_i$  com  $i = 1, 2, \dots, d$ , através dos pontos do produto cartesiano  $\{a_1, a_1 + 1, \dots, b_1\} \times \{a_2, a_2 + 1, \dots, b_2\} \times \dots \times \{a_d, a_d + 1, \dots, b_d\}$

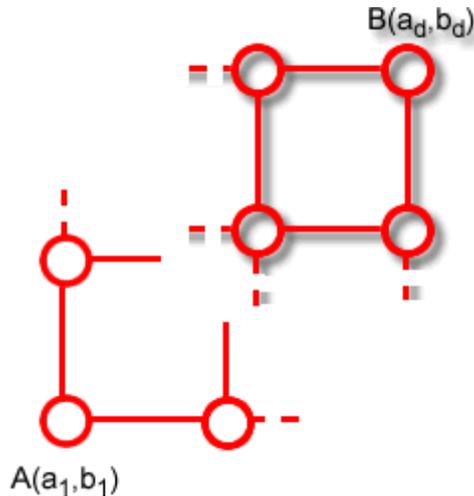


Figura 3.6: Estrutura de uma rede hipercúbica

Com a finalidade de tornar o algoritmo mais genérico e mais eficiente nas redes  $d$ -dimensionais foi estabelecido uma correspondência entre uma rede  $d$ -dimensional e a linear, associando a cada elemento da rede, um número que o identifique, denominado sequencial e que facilite a determinação de seus vizinhos.

A identificação de cada elemento da rede linear, assim como o cálculo da quantidade de elementos da rede, é realizada ao fornecer a dimensão  $d$  do hiperparalelepípedo e suas arestas. As Figuras 3.7, 3.8 e 3.9 demonstram o esquema de representação, para rede de sítios, rede de ligações e redes de sítios e ligações .

As formulações a seguir permitem a descrição das propriedades das redes relacionadas com a topologia, conectividade e quantificação dos aglomerados.

## Rede de sítios

- Aqui, a letra maiúscula  $X$  é usada para representar os pontos ( sítios da rede ); usaremos  $x_i$  para representar sua coordenada de ordem  $i$ .

- O cálculo da quantidade de elementos da rede linear é dado por:

$$T_S(A, B) = \prod_{j=1}^d (b_j - a_j + 1). \quad (3.1)$$

- A identificação dos Sequenciais da rede linear é dada por:

$$S(A, B, X) = \sum_{i=1}^d (x_i - a_i) T_S(A_{i-1}, B_{i-1}) \quad (3.2)$$

onde

$$T_S(A_0, B_0) = 1. \quad (3.3)$$

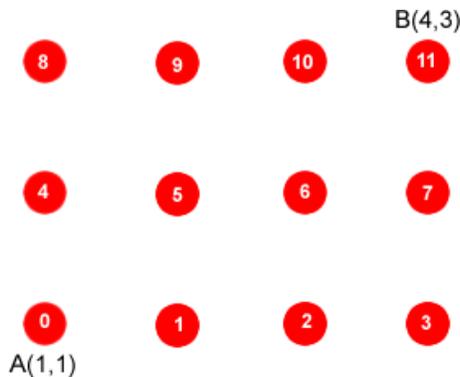


Figura 3.7: Estrutura de identificação Sequencial em uma rede de Percolação por sítios

### Rede de ligações

- O cálculo da quantidade de elementos da rede linear é dado por:

$$T_L(A, B) = T_S(A, B) \sum_{j=1}^d \frac{b_j - a_j}{b_j - a_j + 1}. \quad (3.4)$$

- Em uma rede de ligações, sendo  $L = \overline{XY}$  uma ligação na direção  $j$ , com  $y_j = x_j + 1$ , associa-se um número  $L(A, B, X, j)$  que identifica o Sequencial das ligações, que é dado por:

$$L(A, B, X, j) = dS(A, B, X) + j \quad (3.5)$$

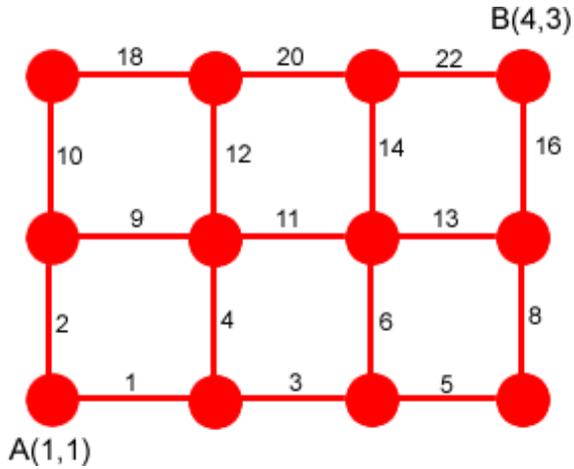


Figura 3.8: Estrutura de identificação Sequencial em uma rede de Percolação por ligações

#### Rede de sítios e ligações

- O cálculo da quantidade de elementos da rede linear é dado por:

$$T_{SL}(A, B) = T_S(A, B) + T_L(A, B) \quad (3.6)$$

- A identificação dos Sequenciais da rede linear é dada por:

$$SL_j(A, B, X, j) = (d + 1)S(A, B, X) + j \quad (3.7)$$

Para  $j = 0$ , aplica-se a fórmula para rede de sítios, em caso contrário aplica-se a de ligações.

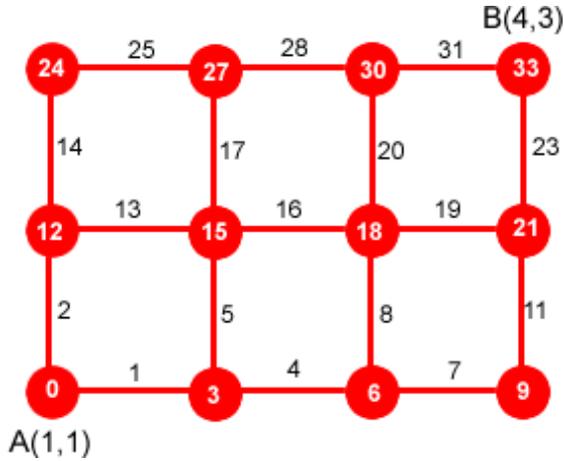


Figura 3.9: Estrutura de identificação Sequencial em uma rede de Percolação por sítios e ligações

### Topologia

A topologia da rede procura definir a partir de um sítio ou ligação quais os incrementos que determinam seus vizinhos. A identificação destes elementos estabelece a relação de conectividade na rede. Observe na Figura 3.10 abaixo, o caso de uma rede de sítios com dimensão  $d = 2$ , onde  $A=(1,1)$  e  $B=(5,4)$ .

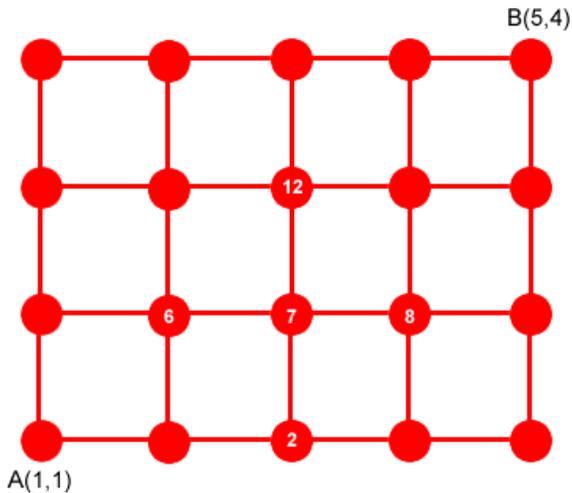


Figura 3.10: Conectividade em uma rede de percolação por sítios

Para se determinar os vizinhos de um sítio no interior da rede, basta somar o identificador de um sítio a cada ordenada do vetor de incrementos (1, -1, 5, -5). Como podemos observar na Figura 3.10, para o sítio de Sequência igual a 7, temos os vizinhos ( 8, 6, 12, 2). Mas esta regra não se aplica aos pontos de fronteira.

Para que se possa identificar quais os vizinhos de sítios e/ou ligações estão fora da fronteira da rede, é necessário alterar o esquema de representação da rede, passando-se a utilizar uma rede expandida, como mostrado na Figura 3.11. Observamos agora que é possível identificar os sítios que estão na fronteira.

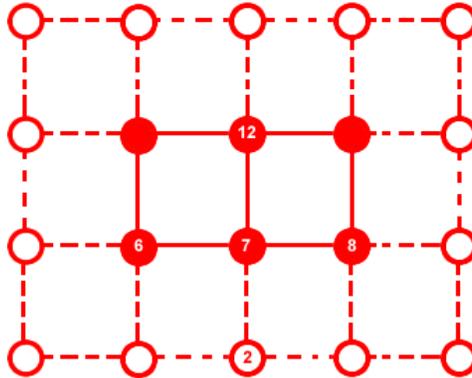


Figura 3.11: Conectividade em uma rede de percolação por sítios expandida

## Fronteiras

Um sítio ou ligação está na fronteira de uma rede, se existir algum dos seus possíveis vizinhos fora da rede de diagonal principal  $\overline{AB}$ . A função abaixo indica se o sítio está em uma face da rede na direção  $j$ .

$$Fr(X, j) = \begin{cases} 2^{2j-2} & \text{se } x_j = a_j \\ 2^{2j-1} & \text{se } x_j = b_j \\ 0 & \text{caso contrário.} \end{cases} \quad (3.8)$$

Definimos fronteira de um sítio  $X$  na rede e o denotamos por  $Fr(X)$  como sendo:

$$Fr(X) = \bigvee_{j=1}^i Fr(X, j). \quad (3.9)$$

O símbolo  $\vee$  denota a operação booleana “ou” entre bits, sendo necessário que os números estejam representados internamente na forma binária.

Na Figura 3.12 temos:  $Fr(5, 1) = 1$ ,  $Fr(5, 2) = 0$ ,  $Fr(6, 1) = Fr(6, 2) = 0$  e  $Fr(5) = 1$ .

As direções estão apresentadas dentro dos quadrados.

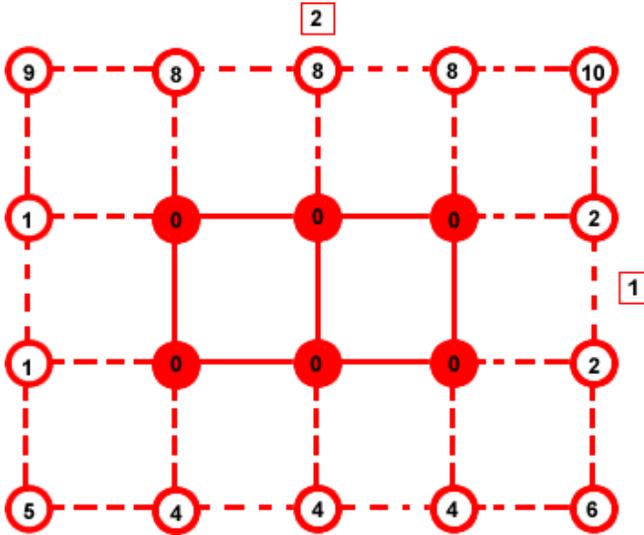


Figura 3.12: Fronteiras de uma rede de sítios

Definimos fronteira de uma ligação  $L = \overline{XY}$  e denotamos por  $Fr(X, Y)$  como sendo:

$$Fr(X, Y) = Fr(X) \vee Fr(Y).$$

### Funcionamento da Percolação Dinâmica

A Percolação dinâmica usa uma grade  $d$ -dimensional como a entrada. Esta grade pode conter dados de propriedades físicas do meio que está sendo explorado. O método pode ser aplicado a percolação em sítios e/ou ligações. Os sítios (ou as ligações) são classificados por seus valores e, para cada etapa do tempo  $t$ , os sítios não ocupados com menor valor associado são marcados como ocupados. Quando um sítio está ocupado e alguns de seus vizinhos também estiverem ocupados, estes passam a formar um único aglomerado. Cada aglomerado em formação recebe a marca do maior aglomerado participante da formação.

Vejamos o exemplo a seguir onde temos um rede de sítios com  $A(1,1)$  e  $B(7,5)$ . A quantidade de elementos é dada pela equação 3.1 e a identificação das Sequências

pela equação 3.2.

A Figura 3.13 mostra a distribuição espacial dos Sequenciais da rede e a Figura 3.14 apresenta os oito passos seguidos pelo algoritmo na formação do aglomerado percolante. Observe que durante a formação do aglomerado percolante número 19, os aglomerados de números 11, 9, 31 e 6 se juntam ao mesmo. O aglomerado de número 15 se junta ao aglomerado de número 29 e assim como o aglomerado 21, não fazem parte do aglomerado percolante. Ver Sequência de formação do aglomerado percolante na Figura 3.15.

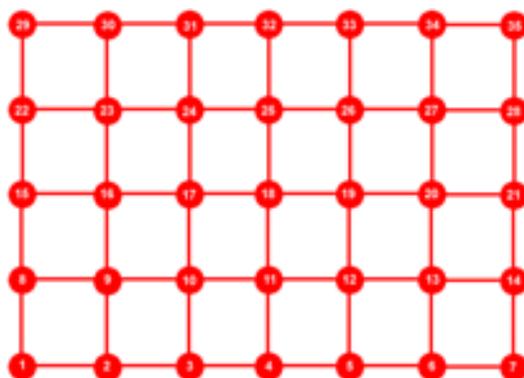


Figura 3.13: Distribuição espacial dos Sequenciais de uma rede de sítios de dimensão  $d=2$

PASSO	AG							
1	11	19	29	9	15	31		
2		26				32		
3		12						
		11						
4		10					6	
		9						
5		18						
6		17	22					21
			15					
7		24						
		31						
		32						
8		16						
		6						

Figura 3.14: Evolução da formação de aglomerados

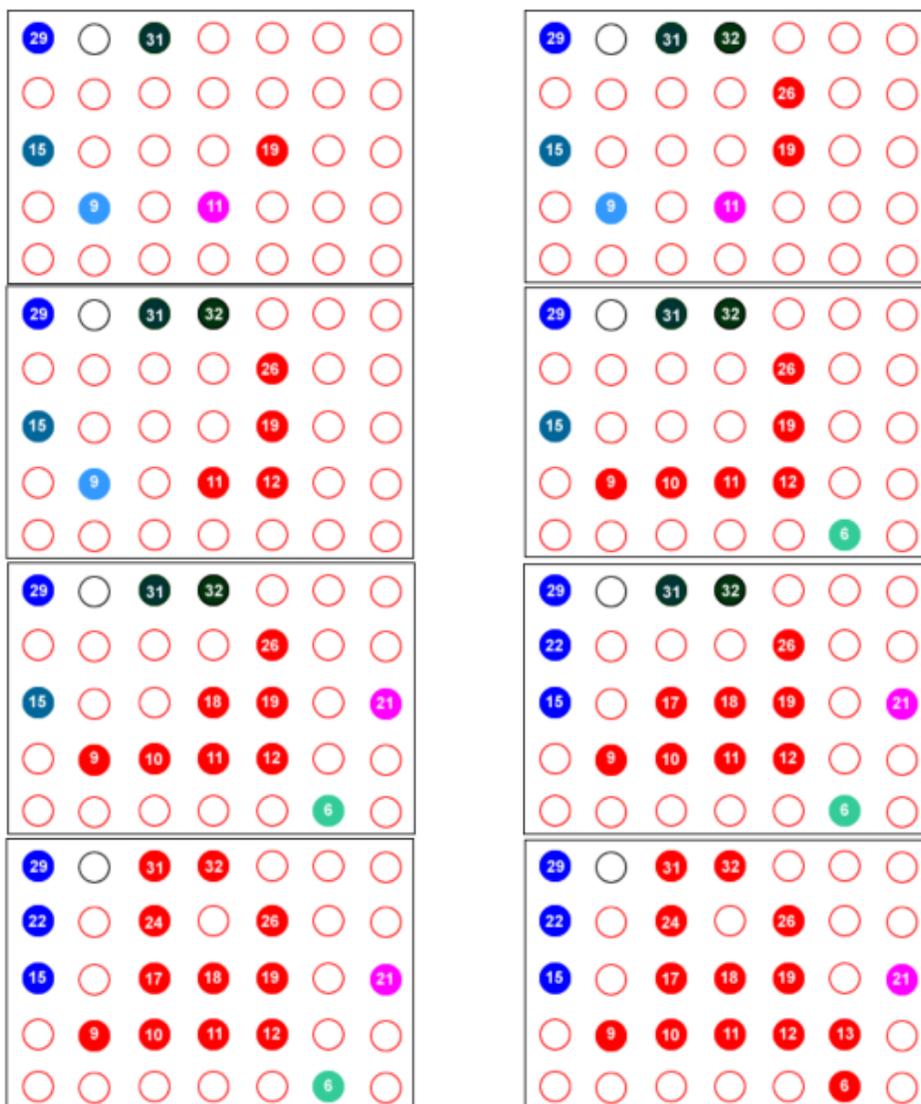


Figura 3.15: Evolução dos aglomerados em 8 passos

## APLICAÇÃO DE PERCOLAÇÃO NA INDÚSTRIA DO PETRÓLEO

Como foi visto, a recuperação de petróleo em reservatórios naturais é uma tarefa complexa, na qual, mesmo com toda a tecnologia a serviço do homem, aproximadamente 60% do total de petróleo contido em uma área permanece no solo. O desafio é aumentar a taxa de sucesso de encontrar e recuperar o petróleo remanescente. Um dos métodos usados para recuperação é o de injeção alternada de água e gás no reservatório

com a finalidade de empurrar o petróleo, que escolhe caminhos menos resistivos ao deslocamento (Figura 3.16).

Este problema pode ser modelado através da utilização de percolação dinâmica. Para isto, ao invés da probabilidade  $p$  usaremos as propriedades petrofísicas do reservatório que melhor determinam a capacidade do fluido de escoar no meio (i.e permeabilidade). Sabendo que a variação da permeabilidade presente no modelo definirá a geração dos aglomerados durante o processo da evolução da percolação dinâmica, podemos assim definir melhores estratégias de recuperação de petróleo. Ainda a partir desta simulação podemos definir quais os possíveis candidatos a poços injetores e receptores. Uma das vantagens de usar a percolação dinâmica é a redução dos custos computacionais comparados aos métodos tradicionais.

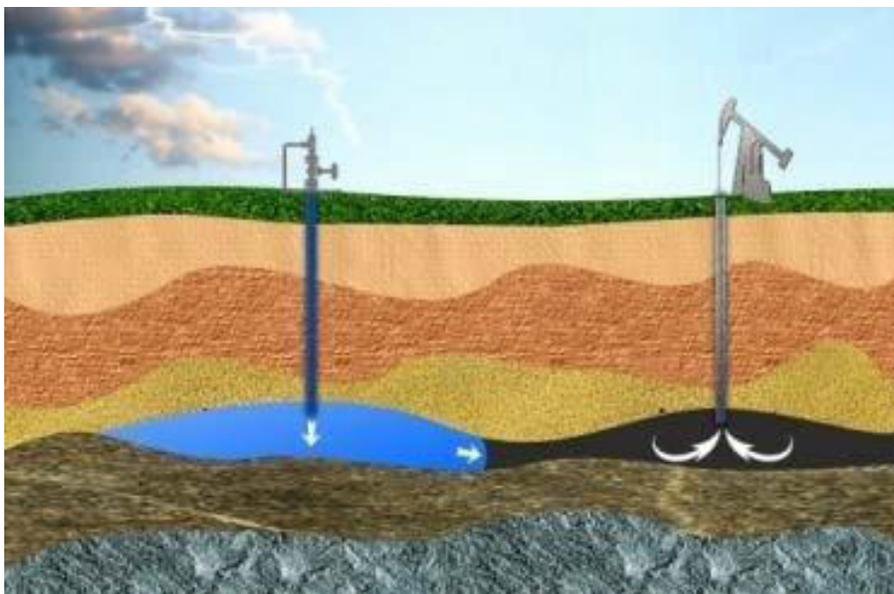


Figura 3.16: Esquema de injeção de  $CO_2$  para recuperação de petróleo

## AMBIENTES VIRTUAIS IMERSIVOS

Ambientes Virtuais Imersivos vem provocando um grande impacto como ferramenta de visualização Volumétrica, permitindo aos cientistas melhorar sua habilidade de estabelecer níveis de abstração necessários a interpretação de grandes conjuntos de dados volumétricos. Sua construção é uma das tarefas mais importantes da tecnologia da Realidade Virtual (RV). Neste Capítulo abordaremos conceitos básicos sobre os temas: *Realidade Virtual* e *Visualização Volumétrica*, base teórica para o desenvolvimento de nosso trabalho.

## REALIDADE VIRTUAL

A Realidade Virtual (RV) é atualmente a mais avançada interface entre o homem e o computador. Ela habilita o usuário a navegar através dos modelos 3D matemáticos, e vê-los como seriam fisicamente no espaço. O uso da RV na Visualização Científica aumenta potencialmente a habilidade de cientistas de criar níveis de abstrações necessárias na manipulação de grandes volumes de dados. Isto se dá pelo fato de envolvermos com maior eficiência nossas capacidades sensoriais e intuitivas nestes ambientes.

A RV pode ser caracterizada como sendo a junção de três idéias básicas: imersão, navegação e interação (Figura 4.1). Ela permite experiências sensoriais, habilitando os usuários a navegar e interagir de maneira significativa em ambiente sintético tridimensional gerado pelo computador [39]. A imersão é considerada por muitos autores como o elemento principal de um ambiente virtual, essencial na interpretação de dados oriundos de diversos problemas científicos. As vantagens principais do uso de ambientes imersivos na visualização de dados, são:

- melhoria na habilidade dos usuários de compreender rapidamente o volume dos dados;
- fornecer uma ilustração mais exata de algumas características e processos;
- permitir um exame dos dados a pequenas e a grandes distâncias do objeto; e
- fornecer uma apreciação melhor da geometria dos dados;



Figura 4.1: Características da Realidade Virtual [39]

Idealmente, o usuário deve sentir-se completamente imerso na cena, interagindo com o volume simulado. Enquanto em uma tela plana comum estimula aproximadamente 45 graus do nosso campo de visão, as exposições a ambientes de RV imersiva podem estimular totalmente o campo de visão. Isto é extremamente importante, já que mais de 50 por cento de nossos neurônios são dedicados à visão [28].

### **Ambientes Virtuais Imersivos Colaborativos**

Os sistemas tradicionais de Visualização tendem a focalizar a solução de problemas voltados a um domínio específico. Neles a Visualização fica limitada pela quantidade de dados apresentada na tela comum de computador e ao número de pessoas que pode observá-lo simultaneamente, limitando assim a integração entre os dados e o trabalho colaborativo entre mais de duas pessoas.

Diferentemente destes, os sistemas de visualização imersiva colaborativos permitem integração entre diferentes bases de dados e a participação de um número maior de pessoas que podem simultaneamente visualizar e trabalhar em conjunto em uma mesma base de dados volumétrica.

Entre os equipamentos utilizados para Visualização nestes sistemas podemos citar: Large Flat Screens (Figura 4.2), Curve Screen Theaters (Figura 4.3) e os ambientes CAVES (Figura 4.4)



Figura 4.2: Ambiente de Visualização Imersiva Reality Center - Retroprojeção - tela plana



Figura 4.3: Ambiente de Visualização Imersiva - Projeção frontal - tela curva

A exploração e a recuperação de petróleo, é um exemplo de aplicação que pode se beneficiar de tais ambientes. Os resultados da simulação do processo da recuperação do petróleo são dependentes do tempo e requerem a integração de uma variedade extensa de dados, em escalas diferentes de profundidade e comprimento. A potencialidade da observação em três dimensões facilita, por exemplo, a percepção da profundidade.



Figura 4.4: Sistemas de Visualização Imersiva - Caverna (CAVE)

### Sensação de Profundidade

A noção de profundidade a partir de imagens 2D pode ser percebida utilizando nossa capacidade de distinguir as formas, as cores, a textura e a relação espacial existente entre os objetos do mundo real (Figura 4.5). Basicamente, existem três categorias de estímulos visuais usados pelo cérebro para formar uma imagem 3D [12], são elas:

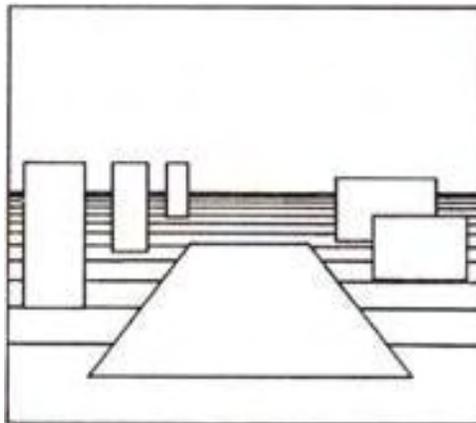


Figura 4.5: Sensação de profundidade

**Informações monoculares** - também chamadas de informações estáticas de profundidade ( *static depth cues* ). Como exemplo, podemos citar a noção de perspectiva linear, oclusão, densidade de texturas, variação da reflexão da luz e as cores.

**Informações óculo-motoras** - são informações fornecidas a partir da movimentação dos olhos.

**Informações visuais estereoscópicas** - A estereoscopia está relacionada à capacidade de se enxergar em três dimensões. O princípio de funcionamento da maioria dos dispositivos estereoscópicos é o fornecimento de imagens distintas aos olhos esquerdo e direito do observador, proporcionando a sensação de profundidade, da mesma forma que se observa os objetos no mundo real.

Basicamente, as imagens captadas por nossos olhos são diferentes, por eles estarem posicionados em locais diferentes (Figura 4.6). Se olharmos para um objeto contínuo no espaço, ocorrerá um "arrasto" da posição de um ponto em sua superfície projetado em uma imagem, com relação a outra. Se considerarmos um sistema de coordenadas 2D (arbitrário, por exemplo, com origem no centro da imagem), haverá uma diferença de coordenadas entre um ponto em uma imagem e o seu correspondente na outra imagem. A esta diferença de coordenadas, chamamos de *disparidade binocular*.

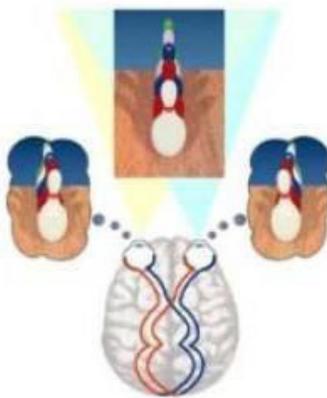


Figura 4.6: Percepção estereoscópica [63]

O efeito da tridimensionalidade é uma combinação do que o olho humano vê e o que o cérebro processa. Quando um observador olha uma cena 3D, cada um de seus olhos vê uma imagem de uma perspectiva ligeiramente diferente. O cérebro funde a informação das duas imagens 2D diferentes para criar a percepção tridimensional- a isto denominamos estereoscopia [29]. Um único par estéreo gerado corretamente

pode fornecer mais informação do que uma pequena animação e pode ajudar em uma melhor compreensão de representações abstratas complexas.

## Gerando Imagens Estéreo

Em um sistema de visualização estereoscópica, duas projeções da cena são geradas e apresentadas de forma que o olho esquerdo vê somente a imagem esquerda e o olho direito vê somente a imagem direita. Como resultado, o cérebro do observador percebe uma cena que tem verdadeiramente uma aparência 3D.

Projeções são representações de modelos 3D em um espaço 2D. De acordo com Hearn e Baker [26] ou Foley [17], as projeções em perspectiva (Figura 4.7) tem como fundamento o equacionamento descrito a seguir, onde  $(x_p, y_p)$  são as coordenadas do ponto projetado, considerando-se o sistema de coordenadas da imagem no centro da mesma.

$$x_p = \frac{x * d}{d + z} \quad y_p = \frac{y * d}{d + z} \quad z_p = 0 \quad (4.1)$$

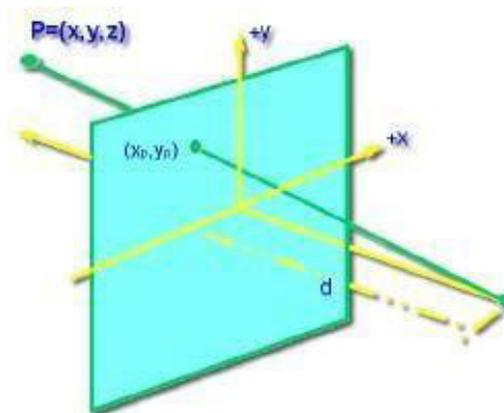


Figura 4.7: projeção em perspectiva

Com relação ao centro de projeção, duas abordagens são utilizadas na construção das visões em perspectiva que correspondem ao olho direito e esquerdo e que compõem uma imagem Estéreo, a projeção off-axis e a projeção on-axis, descritas a seguir.

### Projeção off-axis

Para a projeção off-axis, dois centros de projeção serão estabelecidos, assumindo-se que cada um deles está disposto um ao lado do outro, horizontalmente, como pode ser visto na Figura 4.8. Para o centro de projeção da direita (RCOP), as coordenadas no eixo z são  $(e/2, 0, -d)$  e, para o centro de projeção da esquerda (LCOP) as coordenadas no eixo z são  $(-e/2, 0, -d)$ , onde e representa a distância entre os centros

de projeção.

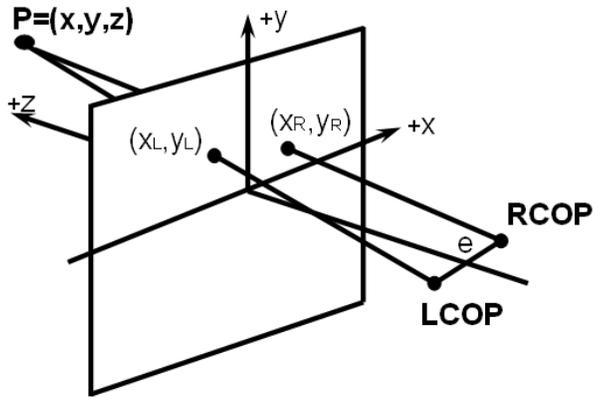


Figura 4.8: projeção off-axis

A projeção de um ponto  $P=(x,y,z)$  qualquer correspondente ao lado esquerdo será  $PL = (x_L, y_L, z_L)$ , com coordenadas no plano de projeção dadas por:

$$x_L = \frac{x * d - \frac{z * e}{2}}{d + z} \quad y_L = \frac{y * d}{d + z} \quad z_L = 0 \quad (4.2)$$

As coordenadas correspondentes ao lado direito serão  $P_R = (x_R, y_R, z_R)$ , com coordenadas no plano de projeção dadas por:

$$x_R = \frac{x * d + \frac{z * e}{2}}{d + z} \quad y_R = \frac{y * d}{d + z} \quad z_R = 0 \quad (4.3)$$

Das equações acima, observamos que os valores de  $y_L$  e  $y_R$  são idênticos, calculando-se uma única vez seu valor para as duas projeções. Assumindo-se que  $x_L$  tenha sido calculado podemos obter  $x_R$  a partir da equação:

$$x_R = x_L + e \left( \frac{z}{d + z} \right) \quad (4.4)$$

Este movimento horizontal entre as projeções depende da profundidade  $z$  do ponto a ser projetado, da distância  $d$  do centro de projeção ao plano de projeção e da distância  $e$  entre as duas projeções geradas ( esquerda e direita ).

### Projeções on-axis

Para projeção on-axis assumimos a existência de um único centro de projeção (Figura 4.7) em conjunto com translações horizontais do conjunto de dados. As equações 4.1, usadas para calcular  $x_R$  e  $x_L$ , podem ser re escritas algebricamente de modo diferente, com vemos abaixo:

$$x_L = \frac{d(x + \frac{e}{2})}{d + z} - \frac{e}{2} \quad x_R = \frac{d(x - \frac{e}{2})}{d + z} + \frac{e}{2} \quad (4.5)$$

Este equacionamento permite adaptar os algoritmos de Volume Rendering sem muita complicação, já que não existe deslocamento do centro de projeção. Vejamos abaixo como seria o algoritmo para visão do olho esquerdo:

1. Para cada ponto  $P=(x,y,z)$  a ser projetado

Translada  $x$  de  $( e/2 )$

Projeta  $P$  usando a projeção em perspectiva padrão.

2. Desloque a imagem inteira de  $( - e/2 )$ .

Para a visão do olho direito pode proceder-se de forma similar. Translada-se  $x$  de  $(-e/2)$  e desloca-se a imagem de  $( e/2 )$

### Calculando a disparidade $e$

Nas imagens estereoscópicas geradas pelo computador a distância entre as imagens esquerda e direita geradas é chamada de *paralaxe* e a distância horizontal entre dois pontos homólogos nas imagens esquerda e direita de uma imagem sobreposta na retina é chamada de *disparidade* [7]. Assim, a paralaxe existente entre as imagens do par estereoscópico produz disparidade nos olhos, o que permite ao cérebro ter a sensação de profundidade. A quantidade de paralaxe, ou seja, distância maior ou menor no par Estéreo determina a qualidade da visualização estereoscópica.

Existem quatro tipos de paralaxe [7] obtidas pelos sistemas, são elas: zero,

positiva, negativa e divergente. Ver Figura 4.9.

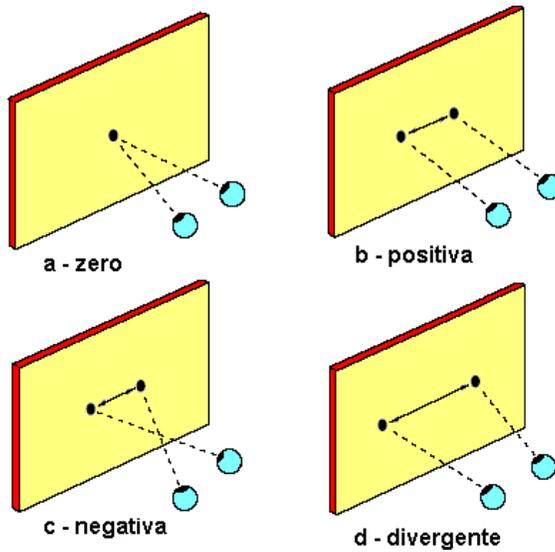


Figura 4.9: Tipos de Paralaxe

No primeiro caso (Figura 4.9a), paralaxe zero, os eixos oculares cruzam-se no plano de visualização. Na paralaxe positiva (Figura 4.9b) os eixos oculares cruzam-se atrás do plano de visualização. Este tipo de paralaxe ocorre no mundo real quando se olha objetos a uma grande distância. No caso da paralaxe negativa (Figura 4.9c), os eixos oculares se cruzam na frente do plano de visualização, ou seja entre o observador e o plano de visualização. No caso da paralaxe divergente (Figura 4.9d) os eixos oculares nunca se cruzam. Isto não acontece geralmente em observações do mundo real, mas sim quando se força os músculos dos olhos para juntar as imagens, o que causa desconforto.

O cálculo da disparidade  $e$  é crítico para a determinação de como o observador junta as duas imagens para ter a sensação de profundidade. Um valor apropriado para  $e$  depende de uma série de fatores, tais como: a localização do objeto na tela, distância do observador da tela, tamanho da tela e da distância linear entre o objeto e o plano de visualização.

O cálculo da paralaxe  $p$  pode ser obtido a partir das equações 4.2 e 4.3 com a seguinte operação:  $p = x_R - x_L$ , resultando em:

$$p = e \left( 1 - \frac{d}{d+z} \right) \quad (4.6)$$

A quantidade total de paralaxe  $p$  calculada para um determinado  $e$  depende da distância do objeto ao plano de projeção. Por exemplo, para um objeto no plano  $z$  afastado de  $z = d$  no lado oposto ao plano de projeção, para dois centros de projeção tem o valor de  $p$  igual a  $e/2$ . Em geral o valor de disparidade  $e$  é calculado a partir de:

$$e = \frac{p}{\left(1 - \frac{d}{d+z}\right)} = \frac{p(i+1)}{i} \quad (4.7)$$

Onde  $i * d$  é a máxima profundidade da área que contém os objetos da cena.

As telas de projeção variam de tamanho, de monitores em computadores pessoais, até grandes telas de projeção presentes em ambientes imersivos colaborativos. Daí a necessidade de calcular o valor da disparidade  $e$  em função de outras variáveis. Neste caso, observando a Figura 4.10 abaixo, extraímos de relações geométricas o valor de  $e$  em função da máxima abertura angular horizontal permitida  $\beta$ .

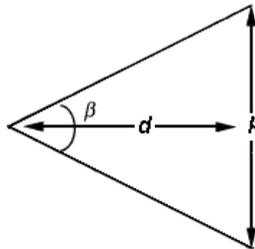


Figura 4.10: Máxima abertura angular horizontal -  $\beta$

$$p = 2d \tan \frac{\beta}{2} \quad (4.8)$$

Utilizando-se a equação 4.7, temos:

$$e = \frac{2d \tan \left(\frac{\beta}{2}\right) (i+1)}{i} \quad (4.9)$$

A obtenção da máxima abertura angular horizontal permitida  $\beta$ , não é uma tarefa fácil, já que esta pode variar. Se  $\beta$  for muito grande, fica difícil a fusão das imagens, e se for muito pequeno o efeito de estereoscopia se perde. O valor deste ângulo deve ser baseado em fatores, tais como: a distância interocular do usuário e sua distância até a tela de projeção.

Vallyus [15] recomenda um limite máximo para  $\beta$  de 1.6 graus. A fim de reduzir a fadiga visual Levoian [15] argumenta em favor de usar um valor igual a um terço do tamanho do paralaxe máximo. Uma série de experimentos controlados foi publicada por Yeh e Silverstein [67]. Sendo que Hodges [28] recomenda como boa aplicação um limite para  $\beta$  de 1.5 graus, obtido aplicando-se a relação de  $e = 0.028d$ . Devido a fatores tecnológicos e de características do sistema visual humano que influenciam na qualidade da apresentação de imagens estereoscópicas, Yeh e Silverstein fazem recomendações mais conservadoras [29].

## Observando Imagens Estéreo

Após a produção das imagens esquerda e direita correspondentes a visão do olho esquerdo e direito respectivamente, dois outros passos são necessários para a visualização do par estereoscópico [33].

- Alinhamento entre imagens
- Codificação da imagem Estéreo

No alinhamento entre imagens, as duas imagens são deslocadas uma acima da outra até possibilitar o efeito de estereoscopia. É importante citar que, mesmo para imagens não muito bem alinhadas, à visão humana é capaz de juntar as duas imagens e criar a impressão tridimensional.

Estando as imagens alinhadas, faz-se necessário a codificação da imagem Estéreo. A codificação e visualização pode ser feita por softwares especiais [61, 50] que salvam a imagem em padrões conhecidos ( JPEG, PNG).

Estas imagens podem ser observadas por dois tipos de processo. O processo de estereoscopia passiva e estereoscopia ativa. A estereoscopia passiva envolve a visualização de imagens codificadas em anáglifo e de imagens geradas utilizando técnicas de polarização de luz.

O anáglifo é o nome dado a figuras planas cuja sensação de profundidade se obtém pelo uso de cores complementares, normalmente vermelho e verde, vermelho e azul. Neste caso, cada um dos olhos utiliza um filtro diferente para visualizar as imagens do par estereoscópico (Figura 4.11).

Devido a filtragem das cores, o efeito de tridimensionalidade é melhor observado quando a Geração das imagens esquerda e direita é feita em tons

de cinza. Essa técnica pode ser utilizada, tanto na visualização em ambientes de projeção, como em imagens impressas.

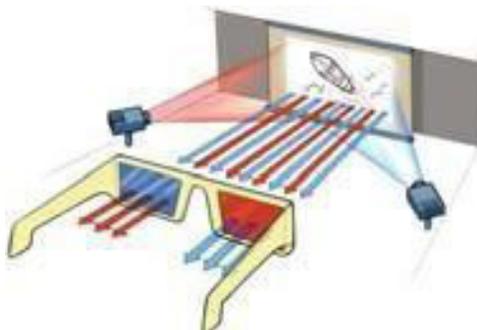


Figura 4.11: Estereoscopia passiva - Anaglifo

No processo de geração de imagens utilizando técnicas de polarização de luz utiliza-se filtros polarizadores, os quais fazem com que as imagens projetadas do par Estereoscópico sejam polarizadas em planos ortogonais (por exemplo: um plano vertical e um horizontal). Dessa forma, o observador, utiliza filtro polaroides ortogonais correspondentes aos planos de projeção e vê com cada olho apenas uma das imagens projetadas. A fusão das duas imagens vistas por cada olho provoca a sensação de profundidade (Figura 4.12).

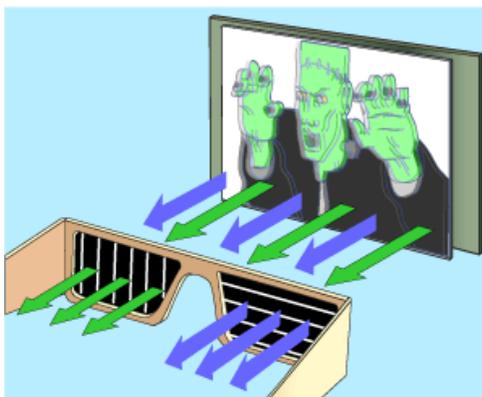


Figura 4.12: Estereoscopia passiva - Uso de filtros polarizadores

Esta técnica apresenta boa qualidade e pode ser utilizada na apresentação de imagens em ambientes imersivos colaborativos.

A estereoscopia ativa envolve o uso de codificação de imagens entrelaçadas que são observadas em óculos especiais. Os óculos obturadores sincronizados (*goggles*) tem lentes de cristal líquido e neles as duas imagens são apresentadas de forma

sincronizada. As lentes podem ficar instantaneamente transparentes ou opacas de acordo com um controle eletrônico sincronizado com o sinal de vídeo. Este sinal deixa em um instante opaca a lente da esquerda e transparente a da direita quando estiver exibindo a imagem correspondente ao olho direito na tela e vice-versa (Figura 4.13).

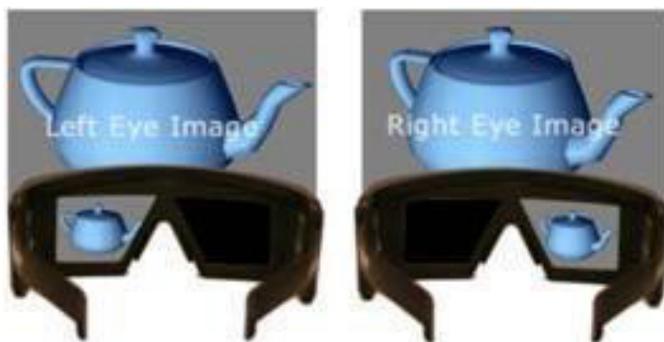


Figura 4.13: Estereoscopia ativa - óculos obturadores sincronizados (*goggles*)

É comum o uso de óculos obturadores sincronizados em ambientes imersivos colaborativos, porém dentre os dispositivos apresentados, este é o que apresenta maior custo.

Além destas técnicas, ainda podemos observar o efeito de profundidade utilizando nossa capacidade visual de juntar as duas imagens para formar o efeito de profundidade. Nesta técnica duas imagens são apresentadas, lado a lado, geradas de forma que cada imagem seja posicionada considerando-se a distância entre os olhos do observador e, ainda, ligeiramente deslocadas na horizontal. Para visualização de um par Estéreo, o observador deve vergar os olhos até ver três imagens. A imagem central aparece com profundidade (Figura 4.14).

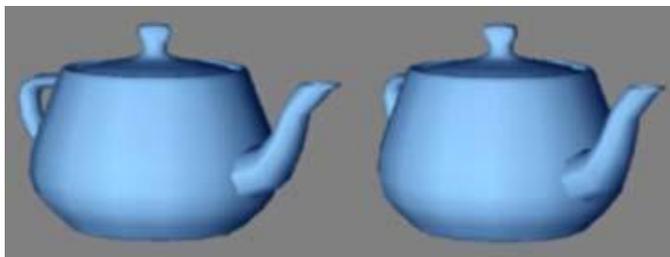


Figura 4.14: Par de imagens Estéreo

## VISUALIZAÇÃO VOLUMÉTRICA

A visualização volumétrica é uma técnica usada para extrair significado das informações oriundas de dados volumétricos usando gráficos interativos e imagens. Isto está relacionado a representação do volume de dados, modelagem, manipulação e renderização [66, 35].

Alguns dos sistemas de RV envolvem em seu núcleo central técnicas de visualização volumétrica. Há três abordagens principais para a visualização de bases de dados volumétricos. Vejamos quais, observando as imagens geradas a partir de análise de dados atmosféricos no software OpenDX [32].

- “*Slicing*” - remove uma fatia do volume 3D (geralmente ortogonal a um dos eixos), reduzindo o problema 3D para 2D (Figura 4.15);

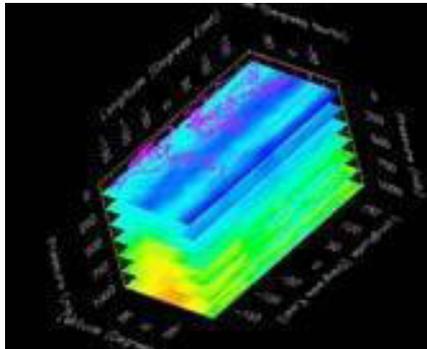


Figura 4.15: Slicing

- “*Isosurfacing*” - extrai uma superfície a partir de um volume (Figura 4.16);

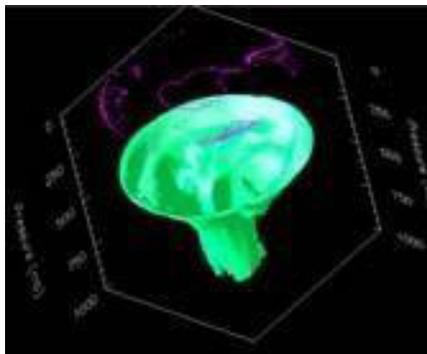


Figura 4.16: Isosurfacing

- “*Volume rendering*” Renderização Volumétrica - apresenta o interior do volume de dados, capacitando o usuário a visualizar regiões internas do volume (Figura 4.17).

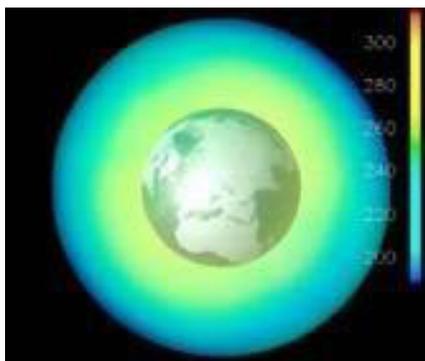


Figura 4.17: renderização de Volumes

Os dois primeiros métodos são geralmente denominados renderização Volumétrica Indireta (*Indirect Volume Rendering*). A renderização Volumétrica indireta envolve a extração de um subconjunto do volume de dados, convertendo-os em primitivas geométricas ( malha de polígonos ) que são renderizadas para formar a imagem a ser visualizada. Ao utilizar estas técnicas, algumas informações sobre os dados volumétricos são perdidas. Além disso, fenômenos como nevoeiro, fumaça, fogo, entre outros, com características amorfas não são adequadamente representados utilizando superfícies.

O terceiro método, denominado renderização Volumétrica Direta (*Direct Volume Rendering*) permite a visualização de todo o conjunto de dados. Os dados volumétricos são diretamente renderizados sem o uso de primitivas geométricas intermediárias. Daqui por diante, o termo *renderização Volumétrica* será usado para referenciar a técnica de renderização Volumétrica direta.

A Renderização Volumétrica é uma técnica em que o objeto de interesse pode ser representado por blocos cúbicos ( hexaedros ) ou tetraedros chamados voxels (Figura 4.18). A cada voxel, está associado um ou mais valores de propriedades medidas ou calculadas do objeto original, tal como a transparência, luminosidade, a densidade, ou velocidade de fluxo.

Nestes métodos, modela-se o volume como se o mesmo fosse composto por células de material semi-transparente que emite luz própria, transmite parcialmente esta luz para outras células e absorve luz incidente [43]. A cor final de um pixel na imagem vem da soma das contribuições de cada voxel. No volume renderizado,

cada face é considerada como semi-transparente, permitindo que o raio de luz passe de voxel a voxel, em vez de parar em sua superfície.

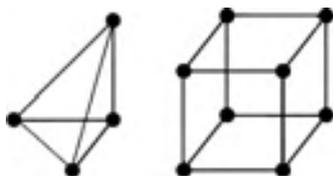


Figura 4.18: Tetraedros e Hexaedros - Voxels

Comparado com as outras técnicas, a Renderização Volumétrica manipula muito mais dados e requer maior poder computacional. No entanto, ela tem a capacidade de preservar a integridade dos dados originais durante todo o processo de visualização. Algumas destas características importantes poderiam ser perdidas na renderização de superfícies, ficando no interior dos dados volumétricos, ou escondido por superfícies opacas.

Diversas pesquisas têm sido realizadas em prol do desenvolvimento de algoritmos de Renderização Volumétrica. As duas abordagens mais comuns são:

- Ray-casting - Método baseado no espaço da imagem ( *image-space* )
- Projeção - Método baseado no espaço da objeto ( *object-space* )

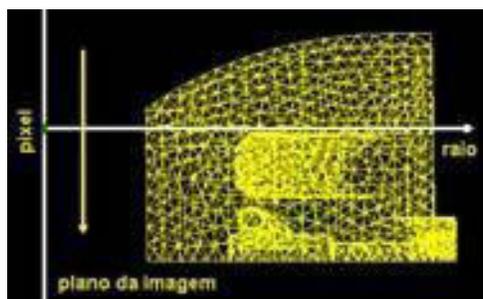


Figura 4.19: Método baseado no espaço da imagem

Os algoritmos de ray-casting determinam para cada pixel no plano da imagem quais são as amostras que contribuem para o cálculo da sua intensidade. Os raios são lançados a partir do ponto de Visualização ( câmara ) através de cada pixel da imagem, determinando que células do volume o raio cruza (Figura 4.19). Para cada voxel interceptado pelo raio são acumulados cor e opacidade associadas, até a opacidade atingir o valor máximo, ou seja igual a 1, ou quando o raio deixar o volume.

Os algoritmos de projeção requerem que as células sejam primeiramente classificadas na ordem de visibilidade. A contribuição de cada voxel é calculada e combinada com a contribuição dos outros voxels que projetam no mesmo pixel cor e opacidade (Figura 4.20). A grande vantagem dos métodos projetivos é que estes podem ser eficientemente implementados em hardware gráfico padrão. Diversos algoritmos de projeção de células foram implementados utilizando placas gráficas [53, 59, 65].

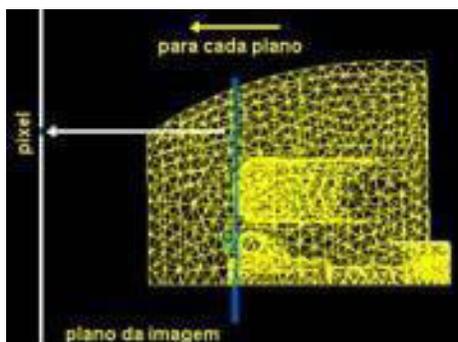


Figura 4.20: Método baseado no espaço do objeto

Existem alguns algoritmos de projeção de células implementados em software [14, 42]. Esta abordagem é flexível e permite com facilidade sua paralelização. O algoritmo ZSweep, desenvolvido por Farias *et al.* [14] é o mais recente algoritmo baseado em projeção de células e será usado em nosso sistema.

## O Algoritmo ZSweep

O algoritmo ZSweep proposto por Farias *et. al* [14], é um algoritmo de renderização Volumétrica direta baseado no paradigma de varredura (*sweep*).

O paradigma de varredura (*sweep*) está relacionado com o processamento de entidades geométricas em uma determinada ordem, passando uma linha de varredura por um plano ou um plano de varredura através de um espaço 3D [52]. Este paradigma tem sido muito usado na construção de algoritmos de renderização volumétrica.

Giertsen [20] foi o primeiro a utilizar este conceito no desenvolvimento de algoritmos de Renderização de volume. Seu algoritmo de varredura é baseado em um plano de varredura que é ortogonal ao plano de visualização. Seguindo a mesma ideia de varrer os dados, Yagel [66] apresenta uma abordagem diferente para renderizar grades irregulares, utilizando suporte de hardware. Seu algoritmo de varredura é baseado em um plano de varredura paralelo ao plano de visualização. Estes dois algoritmos apresentam um aumento na eficiência pois não precisam manter informações

explícitas sobre a adjacência das células. A principal deficiência no algoritmo de Yagel é a necessidade de arquivamento dos polígonos resultantes de cada interação do plano de varredura.

Silva *et al* [60] desenvolveram o algoritmo Lazy Sweep, baseado na técnica de ray casting e no uso de um plano de varredura como em Giertsen [20]. Este algoritmo apresenta-se bem mais rápido e melhor em termos de uso de memória que seus predecessores. Além de um vetor de vértices e de células, a única informação adicional usada é uma lista, onde para cada vértice são guardados os índices das células que são incidentes nestes vértices.

O algoritmo ZSweep foi projetado com a finalidade de combinar simplicidade, velocidade e bom desempenho em memória, sendo facilmente adaptável a dados volumétricos cujo formato de células não sejam apenas tetraedros.

O algoritmo Sweep é baseado na varredura dos dados com um plano paralelo ao plano de visualização, na ordem crescente da coordenada  $z$ . Cada vez que o plano encontra um vértice  $v$ , as faces das células incidentes em  $v$ , que estão a frente do plano de varredura são projetadas. Este plano de varredura segue na ordem crescente do eixo  $z$ , através de todo o conjunto de dados, projetando as faces das células que são incidentes em cada vértice encontrado pelo plano de varredura (ver Figura 4.21).

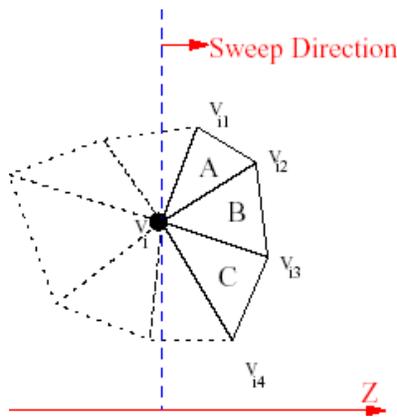


Figura 4.21: ZSweep [14]

Utilizando-se de estruturas de dados eficientes, o algoritmo ZSweep economiza memória e tempo de processamento. Primeiramente, os dados são lidos para a memória, onde é processada a lista de células que incidem sobre cada vértice (*vertex use set*). Utilizando uma estrutura de dados do tipo *heap*, os dados são ordenados pela coordenada  $z$  dos vértices, que determina a ordem dos eventos. Com o uso do *heap*, o algoritmo se tornou mais eficiente, e, o mais importante, permitindo que o

mesmo seja adaptado a situações dinâmicas, onde novos vértices podem ser inseridos no heap.

A partir do ordenamento, é acionado o principal procedimento do algoritmo, ou seja, uma varredura na direção  $z$ , que é efetuada passo a passo através da lista de eventos, quando o  $i$ -ésimo vértice,  $v_z$ , da lista é encontrado, *projeta-se* cada face  $f$  que é incidente em  $v_z$  para o qual  $v_z$  é o vértice que tem a coordenada  $z$  mínima, necessariamente, tais faces  $f$  não tenham sido varridas. As faces a serem projetadas são determinadas pelo exame do conjunto de vértices (*vertex use set*) de  $v_z$  (ver Figura 4.21).

Para que se possa executar a projeção das faces, é calculada a caixa envolvente ("*bounding box*") e percorridas as coordenadas  $X$  e  $Y$ , calculando para cada pixel dentro desta, o valor da coordenada  $z$  (profundidade) e os valores interpolados (interpolação bilinear) para o campo escalar.

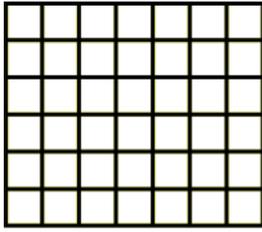
Com os vértices em ordem crescente da coordenada  $z$ , a qual determina a ordem dos eventos, o algoritmo inicia a varredura na direção de  $z$  e a projeção das faces. O algoritmo ZSweep, utiliza uma técnica chamada composição adiantada, o que o diferencia dos demais algoritmos de projeção. Para cada pixel  $p$  a ser projetado, é criado uma lista com as projeções das faces, ordenada por ordem crescente de  $z$ . Os cálculos de iluminação em  $p$ , entretanto, são efetivamente realizados quando o plano de varredura atinge um determinado valor de  $z$ , chamado *target-z*. O modelo de Iluminação utilizado pelo ZSweep pode ser consultado no Apêndice A.

A composição da cor é feita a partir da lista de intersecções, todas as vezes que o plano de varredura atingir o seu *target-z*. O *target-z* de um plano que interceptou o vértice  $v$  é um plano paralelo que passa pelo vértice com maior coordenada  $z$ , de primeira ordem de  $v$ , ou seja, vértices pertencentes às células incidentes em  $v$ .

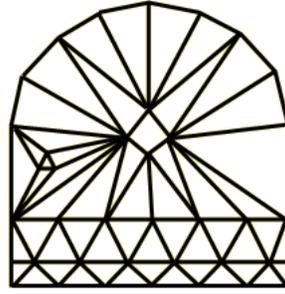
Versões do algoritmo ZSweep foram implementadas para ambientes de computação paralela [15] e técnicas de renderização de grandes volumes de dados em máquinas com limite de memória também foram implementadas [16].

## Dados Volumétricos

Os dados volumétricos usados pelos algoritmos de renderização Volumétrica são obtidos geralmente a partir de simulações ou técnicas modelagem, e representados geralmente na forma de grades estruturadas ou não estruturadas (Figura 4.22). No caso mais simples, de uma grade estruturada, os pontos se apresentam na forma de uma matriz com uma relação dos mesmos e seus vizinhos bem definida. As grades estruturadas são construídas a partir de uma estrutura topológica rígida e a relação de conectividade entre as células é implícita. Na prática, estas grades são encontradas na maior parte em processos de aquisição indiretos como MRI e CT na aplicação médica.



**Grids estruturados**



**Grids não estruturados**

Figura 4.22: Forma de representação dos dados Volumétricos

Por outro lado, nas grades não estruturadas, as células não necessariamente são cubos uniformes, a informação de conectividade é fornecida explicitamente. Estas grades foram propostas como um meio efetivo de representar bases de dados não uniformes, onde a distribuição dos pontos não seguem um padrão regular. As grades não estruturadas têm a vantagem de serem genéricas e de se conformar a quase toda a geometria desejada, sendo assim, úteis para representar geometrias complexas. Esta generalização faz das grades não estruturadas apropriadas para a representação visual de muitos fenômenos científicos.

## ZSWEEP EM AMBIENTE IMERSIVO

O algoritmo ZSweep [14] foi projetado somente para renderização volumétrica não suportando operações de Geração de imagens estereoscópicas. Ele trabalha apenas com projeção paralela, onde toda a projeção é feita considerando que os raios projetantes sejam paralelos e perpendiculares ao plano de projeção. No entanto, as pessoas observam o mundo em perspectiva. Para que possamos criar ambientes virtuais imersivos, onde se possa ter a sensação de estar no mundo real, necessitamos incluir rotinas de cálculo de projeção em perspectiva na API ZSweep.

Com o compromisso de manter a desempenho e a simplicidade na Geração de imagens que é inerente ao algoritmo de volume rendering ZSweep, apresentaremos neste Capítulo o trabalho desenvolvido na Implementação do código para Geração de projeção em perspectiva e de pares estereoscópicos necessários em ambientes de realidade virtual imersiva.

## O PROCESSO DE RENDERIZAÇÃO VOLUMÉTRICA

A Renderização Volumétrica é um processo complexo e exige um esforço computacional intenso. Adotamos neste caso uma abordagem onde este esforço é dividido em fases, onde cada processo é otimizado, como mostrado na Figura 5.1.

Cada passo do processo de renderização volumétrica agrupa um conjunto de operações necessárias para concluir uma determinada parte do processo completo. Sua subdivisão depende de cada implementação. Por conveniência o dividiremos em duas partes: Geometria e Renderização.

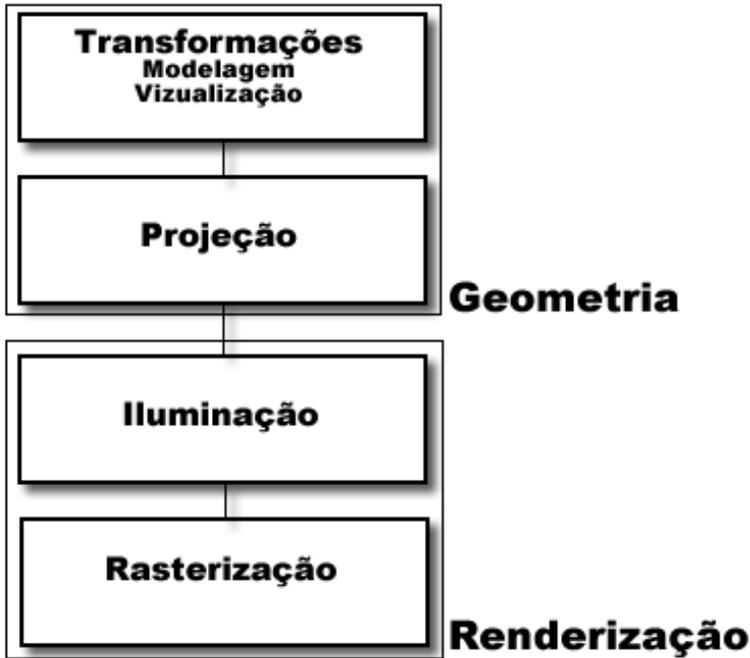


Figura 5.1: Processo de renderização Volumétrica

## O Processo de Renderização Volumétrica: Geometria

Inicialmente, os dados volumétricos são lidos para a memória em Coordenadas de Objeto. Um primeiro conjunto de transformações geométricas, vistas a seguir, são aplicados aos mesmos de forma a posicioná-los em um único sistema de coordenadas - Coordenadas de Mundo, permitindo que sejam tratados como um só bloco durante todo o processo.

### Transformações de Modelagem e visualização

O processo envolve transformações de visualização, que posiciona a câmera na origem do sistema de coordenadas, olhando para uma determinada direção, usualmente na direção do eixo  $z$ . Estas operações fazem parte do conjunto de métodos implementados nos objetos da API de Renderização Volumétrica do ZSweep.

### Projeção

Projeções são representações de modelos 3D em um espaço 2D. Estas podem ser obtidas pela aplicação de transformações geométricas. A API ZSweep, por *default*, apresenta suas imagens em projeção paralela. Ver Figura 5.2.

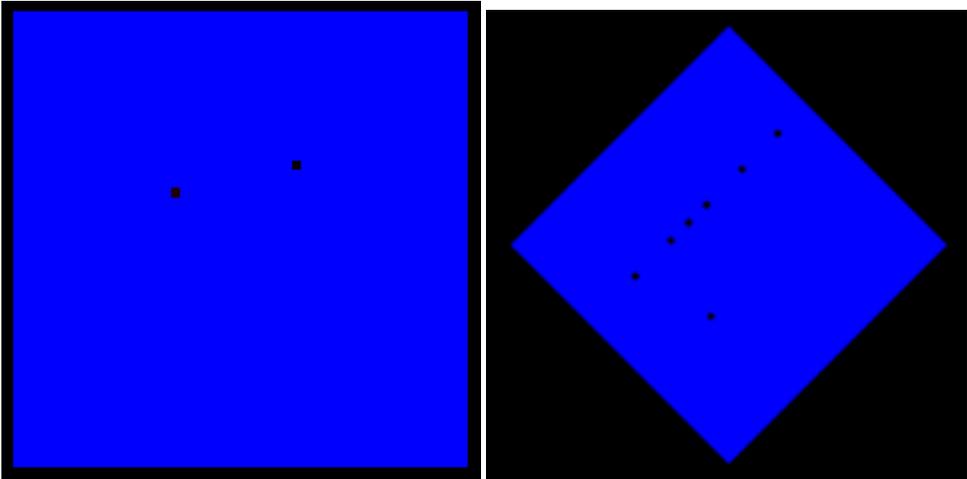


Figura 5.2: Imagens em projeção Paralela em duas posições - padrão ZSweep

De acordo com Hearn e Baker[26] ou Foley [17], as projeções em perspectiva (Figura 5.3) têm como fundamento o equacionamento descrito a seguir.

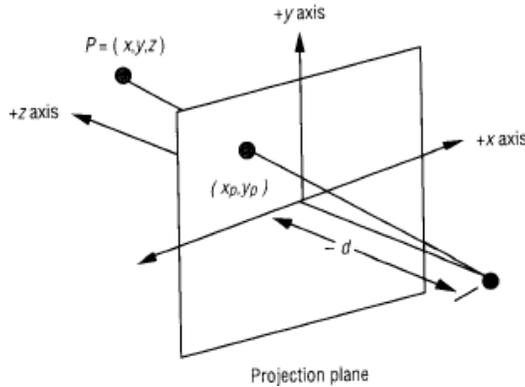


Figura 5.3: projeção em perspectiva

Utilizando um sistema de coordenadas onde se encontram o plano de projeção e um centro de projeção no eixo z localizado em  $(0,0,-d)$ , podemos aplicar as transformações abaixo e encontrar a projeção resultante do ponto  $P(x,y,z)$  no plano de projeção equivalente a  $(x_p, y_p, z_p)$ .

$$\begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (5.1)$$

$$x_p = \frac{x * d}{d + z} \quad y_p = \frac{y * d}{d + z} \quad z_p = 0 \quad (5.2)$$

A matriz acima denomina-se matriz de transformações homogêneas. Para permitir que a API ZSweep projete imagens em perspectiva foi necessário implementarmos a classe `MatrixH` e incorporá-la ao seu conjunto de classes. Veja na Figura 5.4 abaixo o seu diagrama da classe.

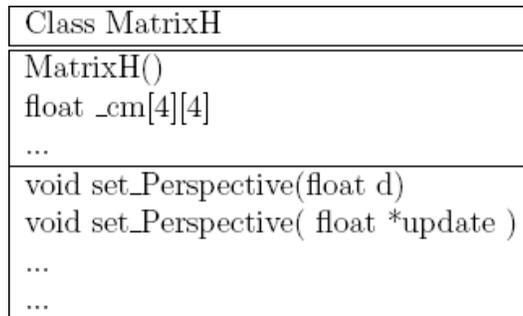


Figura 5.4: Classe `MatrixH` - transformações homogêneas

O par de imagens apresentado na Figura 5.5, representa a projeção em perspectiva do mesmo objeto apresentado na Figura 5.2, renderizado utilizando projeção paralela. Podemos observar nas imagens abaixo a característica de tridimensionalidade inerente das projeções em perspectiva.

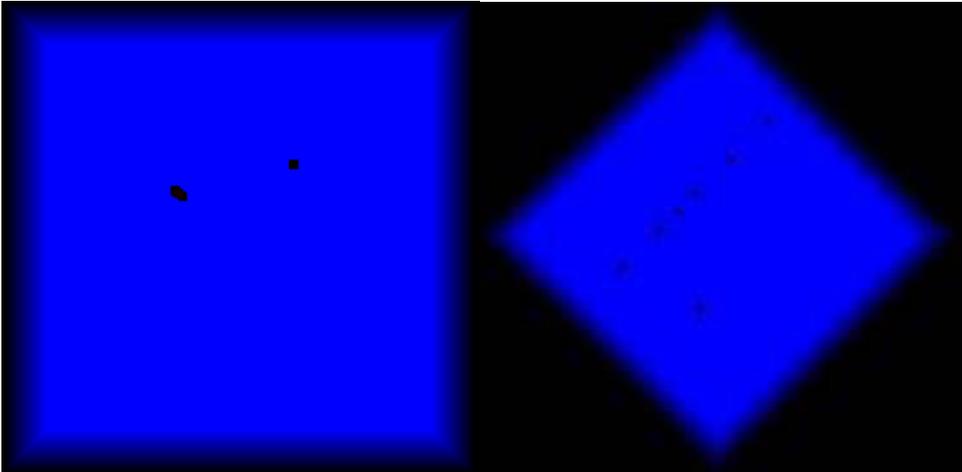


Figura 5.5: Imagens em projeção Perspectiva em duas posições - ZSweepVR

### O Processo de Renderização Volumétrica: Renderização

Neste passo, todas as transformações geométricas já foram realizadas, e os dados estão prontos para Geração da imagem 2D que será apresentada na tela do computador. O processo de renderização envolve as etapas de Iluminação e rasterização. Estas operações fazem parte do conjunto de métodos implementados nos objetos da API de Renderização Volumétrica ZSweep.

A melhor maneira de se compreender o processo de rasterização é se considerar um conjunto de dados que representam uma face triangular. A partir destes dados, calcula-se o sua caixa envolvente ("*bounding box*") e percorre-se as coordenadas X e Y, calculando para cada pixel dentro do "*bounding box*", o seu valor de z. Se existirem valores escalares para cada vértice, estes também são levados em conta e calculados a partir de uma interpolação bilinear dos valores definidos para cada um dos três vértices. Para cada pixel, ter-se-á o valor correspondente do escalar que será utilizado no cálculo da contribuição das componentes (r, g, b) de cor. O formato em que estes dados são apresentados é (r,g,b,o) onde o representa a opacidade. Esses valores são mantidos no buffer de cores.

É usual manter-se dois buffers ao mesmo tempo. Desta forma, o desenho é efetuado em um buffer e somente enviado para a tela quando todas as primitivas tiverem sido rasterizadas. Neste momento, começa-se a desenhar no segundo buffer, que quando completo, é também enviado, trocando-se de buffer, e assim por diante.

Esta técnica, conhecida como *double buffering*, evita efeitos indesejáveis, tais como efeitos de tela piscando ("*flickering*"), enquanto o programa desenha as primitivas.

## GERAÇÃO DO PAR ESTEREOSCÓPICO

Os sistemas de visualização estereoscópica requerem o uso de projeções em perspectiva. Duas projeções da cena são geradas e apresentadas de forma que o olho esquerdo vê somente a imagem esquerda e o olho direito vê somente a imagem direita. Como resultado, o cérebro do observador percebe uma cena que tem verdadeiramente uma aparência 3D.

O efeito da tridimensionalidade é uma combinação de o que o olho humano vê e o que o cérebro processa. Quando um observador olha uma cena 3D, cada um de seus olhos vê uma imagem de uma perspectiva ligeiramente diferente. O cérebro funde a informação das duas imagens diferentes para criar a percepção tridimensional.

Com relação ao centro de projeção, duas abordagens são utilizadas na construção das visões em perspectiva que correspondem ao olho direito e esquerdo e que compõem uma imagem Estéreo, a projeção off-axis e a projeção on-axis, como descritas no Capítulo 4. Neste trabalho exploraremos a Geração de imagens estereoscópicas utilizando a abordagem de projeção off-axis.

As imagens 5.6 e 5.7 demonstram a aplicação do código desenvolvido utilizando projeção off-axis incluídos no renderizador volumétrico ZSweep para Geração das imagens estereoscópicas, olho esquerdo e direito respectivamente.

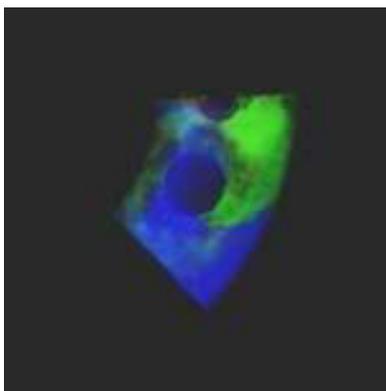


Figura 5.6: Imagem olho direito

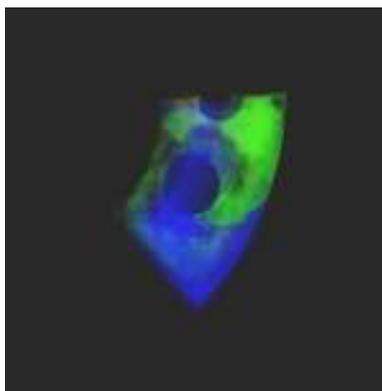


Figura 5.7: Imagem olho esquerdo

A Figura 5.8 abaixo apresenta o par Estéreo de um conjunto de dados volumétricos da NASA, conhecido como SPX, que pode ser observado vergindo-se os olhos até formar uma imagem central que pode ser vista em profundidade.

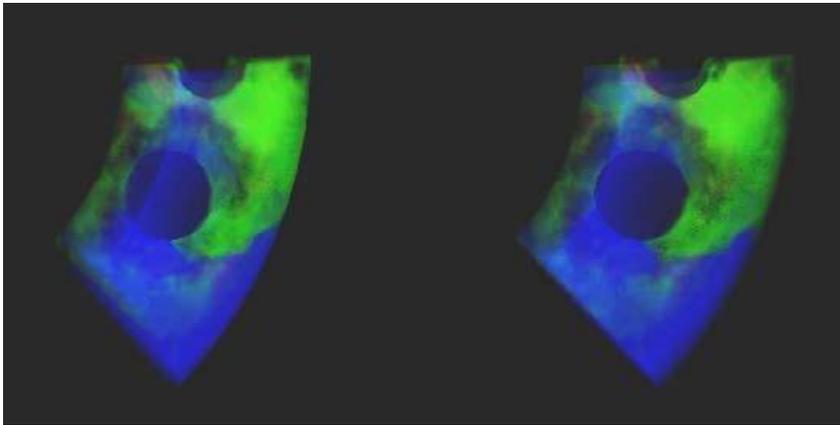


Figura 5.8: Par Estéreo SPX

Já as Figuras 5.9 e 5.10 necessitam para visualização do efeito de profundidade de óculos especiais que utilizem a técnica de complementação de cores, chamada *anaglifo*, conforme apresentado no Capítulo 4.

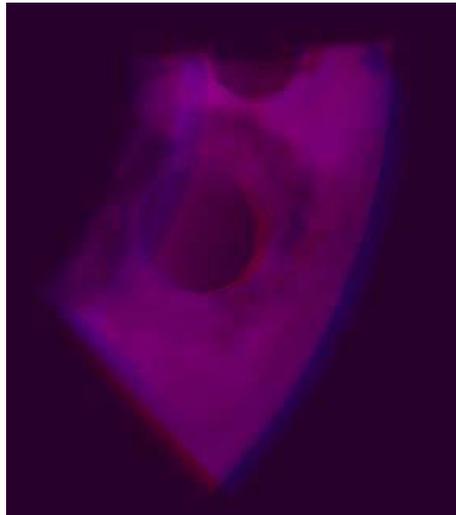


Figura 5.9: "Red-Blue" anaglifo

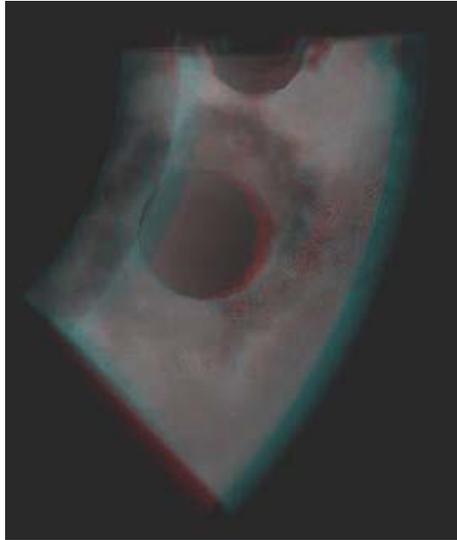


Figura 5.10: “Gray” anaglifo

### Detalhes de Implementação

Durante a Implementação do código para geração de projeções em perspectiva e de imagens em Estéreo procuramos estabelecer funções de interface com o usuário da API ZSweep, a fim de possibilitar o nível de abstração necessária, escondendo detalhes de Implementação e tornando o código mais simples.

Para gerar imagens em perspectiva o usuário só precisa definir a distância  $d$  do plano de Visualização e utilizar a função de interface **ZSweep setPerspective()**, ficando o código encarregado de montar a matriz de transformações homogêneas que será aplicada aos pontos da base de dados Volumétrica. A Figura 5.11 apresenta detalhes desta operação.

## Projeção em Perspectiva

Interface	Operações
<b>ZSweep_setPerspective</b> ViewPlane VPlane..... VPlane.UpdatePerspectiveMatrix( float d ).....	Cria objeto responsável por operações de visualização Informa o valor da distância do observador ao plano de visualização
Núcleo	Operações
<b>Viewpoint</b> MatrixH _pMatrix ..... UpdatePerspectiveMatrix( float d ) ..... _pMatrix.set_Perspective(float d)..... Project(vector<Point> &tf_points_VEC)..... perspec_Point( Point& p) .....	<b>Transformações de visualização</b> Cria objeto responsável por operações de transformações homogêneas Atribui o valor da distância do observador ao plano de visualização Informa o valor da distância para geração da matriz de transformações homogêneas Informa o conjunto de pontos a serem projetados em perspectiva Aplica a matriz de transformação homogênea a cada ponto em particular
<b>MatrixH</b> set_Perspective(float d) .....	Constroi a matriz de transformação homogênea
<b>Scene</b> Scene *cena..... cena.render(vector<Point> &tf_points_VEC).. ..ViewPlane *vp..... ) .....	<b>Renderização</b> Cria objeto responsável pela geração da cena renderizada Informa os atributos necessários para renderização da cena ( Pontos, Células, Plano de visualização, parâmetros de Iluminação )

Figura 5.11: Gerando imagens em perspectiva na API ZSweep

Para Geração de imagens estereoscópicas utilizamos a mesma abordagem, o usuário só precisa informar a distância ao plano de Visualização  $d$ , utilizando a função de interface **ZSweep setVR()** e qual o lado  $s$  que a imagem vai ser gerada (direita ou esquerda), através da função de interface **ZSweep setVRside()**.

O equacionamento necessário a Geração das matrizes de transformação homogênea aplicada ao conjunto de dados volumétricos foi discutido e apresentado no Capítulo 4. Podemos observar na Figura 5.12 detalhes da implementação desta operação.

Como discutido anteriormente, a geração das imagens correspondentes ao centro de projeção da direita (RCOP) e ao centro de projeção da esquerda (LCOP) envolve uma série de variáveis que podem afetar a sensação de profundidade requerida na imagem final estereoscópica. Neste trabalho, procuramos adotar recomendações já estabelecidas como de boa prática. Hodges em [29] recomenda a utilização da seguinte relação:  $e = 0.028 d$ , onde a disparidade  $e$  necessária a montagem da matriz de transformação homogênea é função da distância  $d$  entre o plano de visualização e o conjunto de dados volumétricos.

Visando verificar a relação, realizamos alguns testes em computador pessoal de tela plana 17", variando-se o fator de correlação entre estas duas variáveis. Para isto adotamos um valor de  $d$  arbitrário. Um outro ensaio realizado, foi para a correlação recomendada de 0.028, onde variamos a distância  $d$  arbitrariamente. Com isto podemos aferir a relação, que passou a ser utilizada no nosso algoritmo.

## Esterescopia - Realidade Virtual

Interface	Operações
<b>ZSweep_setVRside</b> ViewPlane VPlane..... VPlane.vr_setSide(int s )..... <b>ZSweep_setVR</b> VPlane.UpdateVR_Matrix( float d ).....	Cria objeto responsável por operações de visualização Informa o lado em que a imagem deve ser renderizada Informa o valor da distância do observador ao plano de visualização
Núcleo	Operações
<b>Viewpoint</b> MatrixH _pMatrix ..... vr_setSide(int s )..... UpdatePerspectiveMatrix( float d ) ..... _pMatrix.set_Perspective(float d)..... vr_Project(vector<Point> &tf_points_VEC)..... vr_Point( Point& p ) .....	<b>Transformações de visualização</b> Cria objeto responsável por operações de transformações homogêneas Determina o lado em que a imagem deve ser renderizada Atribui o valor da distância do observador ao plano de visualização Informa o valor da distância para geração da matriz de transformações homogêneas Informa o conjunto de pontos a serem projetados em perspectiva <b>Projeção off-axis.</b> Aplica a matriz de transformação homogênea a cada ponto em particular, levando em conta a distância entre os centros de projeção - e.
<b>MatrixH</b> set_Perspective(float d) .....	Constroi a matriz de transformação homogênea
<b>Scene</b> Scene *cena ..... cena.render(vector<Point> &tf_points_VEC)... ..ViewPlane *vp.... ) .....	<b>Renderização</b> Cria objeto responsável pela geração da cena renderizada Informa os atributos necessários para renderização da cena ( Pontos, Células, Plano de visualização, parâmetros de Iluminação )

Figura 5.12: Gerando imagens estereoscópicas na API ZSweep.

## ZSWEEPVR

A visualização volumétrica imersiva tem se destacado como uma importante ferramenta para compreensão de bases de dados volumétricos oriundas de simulações de fenômenos físicos. Este tipo de simulação trata de situações que envolvem uma quarta dimensão que é o tempo e que é usualmente visualizado com o uso de técnicas de animação. Infelizmente, quando os objetos estão em movimento, o sistema visual humano não pode captar detalhes da imagem.

Neste trabalho, nós propomos uma ferramenta interativa, denominada ZSweepVR, baseada em técnicas de Renderização Volumétrica e realidade virtual que permitem uma melhor compreensão dos dados volumétricos gerados por simulações de percolação dinâmica.

## RENDERIZAÇÃO VOLUMÉTRICA E REALIDADE VIRTUAL APLICADOS A PERCOLAÇÃO DINÂMICA

ZSweepVR é baseado em técnicas de Renderização Volumétrica direta que permite que qualquer parte dos dados seja vista, incluindo as estruturas e os detalhes internos. A percolação dinâmica [9], permite que o usuário siga a evolução da formação dos aglomerados, e as interações entre eles. Estas interações ocorrem dentro do conjunto de dados volumétrico, sendo só possível sua visualização a partir de técnicas de Renderização volumétrica, adicionando com isto mas realismo, com os detalhes do interior dos dados. Os softwares tradicionais de visualização e modelagem de objetos geológicos, como por exemplo, o gOcad [40], não pode fornecer este tipo de informação (Figura 6.1). Além disso, a Visualização destas simulações utilizando a teoria do percolação na exploração e na recuperação de óleo, podem eficazmente ajudar na estratégia de definição dos possíveis candidatos a poços injetores e produtores.

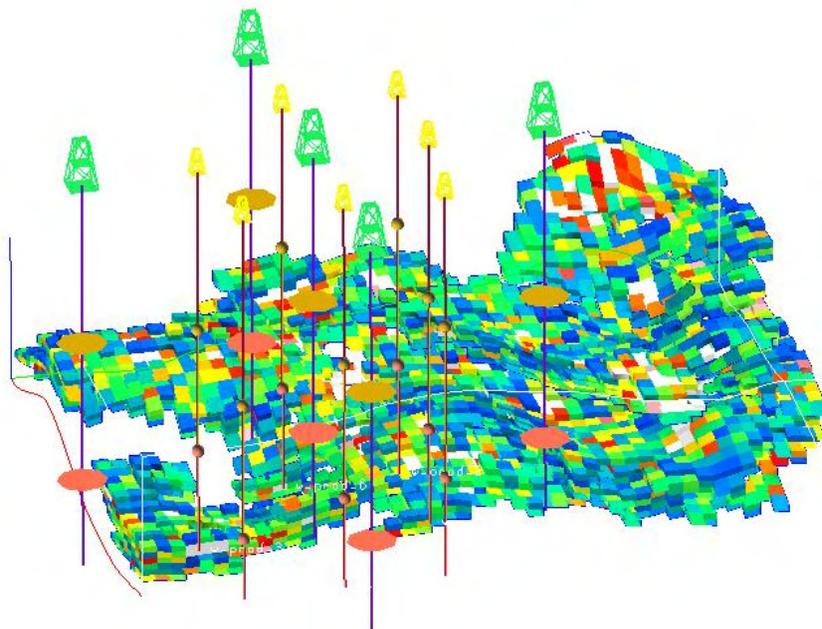


Figura 6.1: Visualização Volumétrica de aglomerados de percolação utilizando o gOcad.

A visualização destes dados em Ambientes Virtuais Imersivos Colaborativos, fornece ao usuário a sensação de perfeita interação com a base de dados Volumétrica. Esta técnica e ambiente quando utilizados por equipes multidisciplinares pode se tornar imprescindível na tomada de decisões, afetando todo o processo de produção e exploração de petróleo.

Esta é a razão principal para a integração da teoria da percolação com a renderização volumétrica em ambiente imersivo e colaborativo.

## O SISTEMA ZSWEEPVR

ZSweepVR foi desenvolvido em linguagem de programação C++ para plataformas Linux e IRIX, usando a API QT como base para o desenvolvimento de sua interface. A arquitetura básica do sistema ZSweepVR é apresentada na Figura 6.2. Há dois módulos principais: o servidor de simulação e o servidor de visualização. A característica principal de ZSweepVR é que os dois processos separados e de forma independente cuidam das operações de visualização e simulação, podendo trabalhar em um pipeline, quando máquinas paralelas forem utilizadas. Enquanto o servidor de Visualização gera uma imagem para o instante  $t_p$ , o servidor de simulação pode trabalhar o instante  $t_{p+1}$ . Além disso, desacoplando o módulo da simulação do de Visualização nós podemos:

- Visualizar diferentes pontos de vista sem requerer novas simulações;
- Replicar o servidor de renderização a fim de tirar vantagem de arquiteturas paralelas.

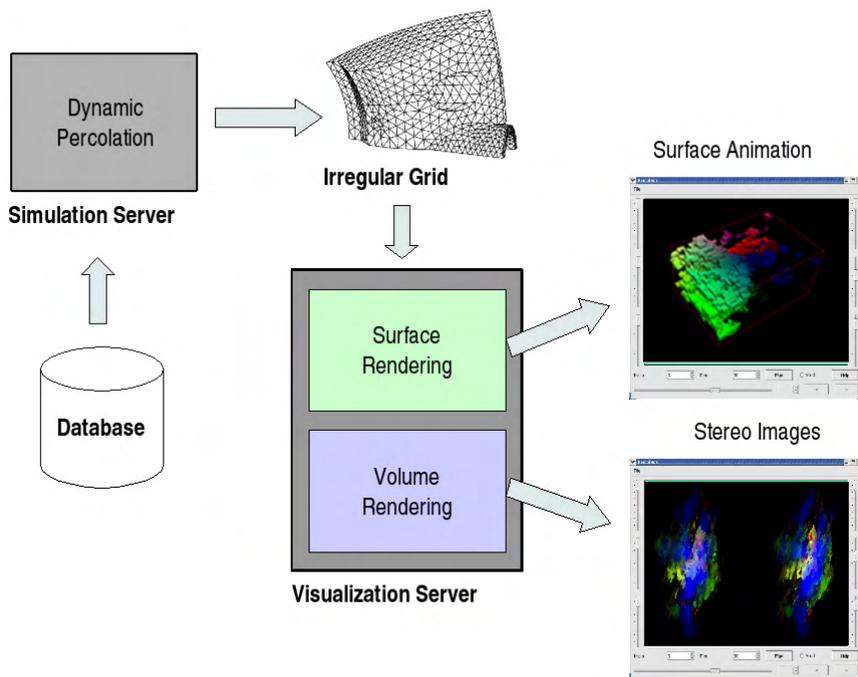


Figura 6.2: O Sistema ZSweepVR.

Uma outra importante característica do ZSweepVR é a possibilidade de gerar as imagens com duas técnicas diferentes: Renderização de Superfície e Renderização Volumétrica. A Renderização de superfície é executada a partir de diretrizes do OpenGL, enquanto a renderização Volumétrica é executada pelo algoritmo de renderização Volumétrica ZSweep. Esta característica do ZSweepVR permite que o usuário tenha mais flexibilidade, com uma visualização pobre de detalhes, porém mais rápida (OpenGL) e uma visualização mais rica em detalhes, porém mais lenta (ZSweep).

O módulo renderização de superfície fornece uma animação de todos os instantes  $t_i$  de Percolação, onde o usuário pode observar a evolução da formação dos aglomerados. O módulo de renderização Volumétrica, por outro lado, gera um volume mais detalhado, em modo Estéreo ou não, quando solicitado pelo usuário. A ideia é ter o usuário decidindo se quer observar apenas a evolução do processo ou visualizar a correlação entre as propriedades internas na formação dos aglomerados.

Estes dois módulos são executados em diferentes processos, enquanto o usuário está observando a animação no renderizador de superfície, e o processador central é quase inativo, o processo de Geração da imagem de um instante  $t_i$  é disparado e executado pelo renderizador volumétrico. Esta geração da imagem pode ser feita antes do pedido explícito do usuário, nós usamos a seguinte heurística: gerar as imagens dos instantes  $t_0$  a  $t_k$  durante o tempo em que o usuário navega na animação.

A interface ZSweepVR para a visualização da evolução dos aglomerados percolantes é mostrada na Figura 6.3. Como pode se ver, a interface permite que o usuário observe em momentos diferentes resultados da percolação, e veja-os como uma animação. Os *widgets* executam operações para diante e para trás, com a opção de controlar faixas de evolução no tempo, assim como de explorar um instante  $t_i$  qualquer. Esta flexibilidade permite ao usuário visitar aglomerados particulares ou faixas de formação dos mesmos. Uma outra opção que pode ser explorada são os saltos, ou seja toda vez que existe uma ou mais conexões entre aglomerados.

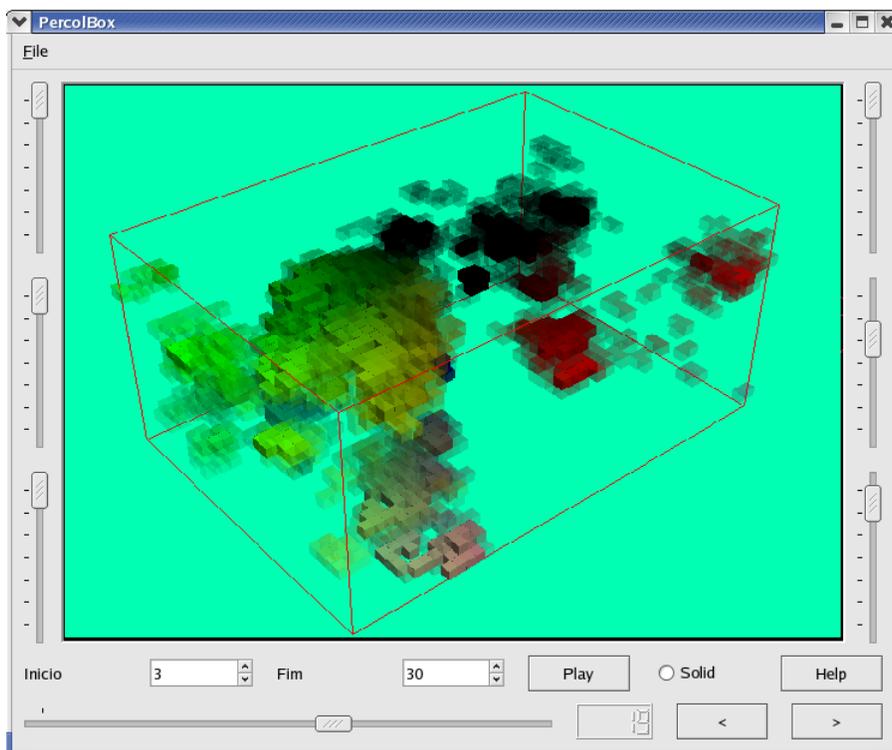


Figura 6.3: Interface do ZSweepVR

A integração entre o módulo da simulação e o módulo de visualização é feita por uma grade armazenada na memória principal. Este grade une a saída da simulação à entrada da visualização volumétrica, evitando qualquer tipo de conversão de dados e de operações de E/S. A ideia é usar esta grade na passagem da estrutura espacial representada de forma linear que é inerente ao módulo de percolação para estrutura tridimensional necessária ao módulo de visualização. Isto permite a geração de informações sobre a geometria e conectividade necessárias utilizadas pelo renderizador de superfície e o renderizador volumétrico. O modelo da percolação dinâmica é um problema dependente do tempo, e gera aglomerados diferentes para cada instante do tempo  $t_i$ . Os sítios de percolação, entretanto, são os mesmos. Daí, podemos usar a mesma grade para resultados diferentes de percolação no tempo. Esta característica provoca um aumento na quantidade de dados gerados no tempo a serem renderizados, tornando a integração entre a simulação e o módulo de visualização crucial para o desempenho da renderização.

### Simulador de Percolação

O módulo da simulação de percolação é baseado no modelo de percolação dinâmica explicado no Capítulo 3. Nós escolhemos este modelo porque é eficiente e permite o estudo da evolução, distribuição e formação dos aglomerados em função do tempo. Uma outra característica importante da percolação dinâmica é que tem um custo computacional baixo. Enquanto a percolação dinâmica varia  $p$  com  $t$ , pode produzir muito mais cenários do que métodos de percolação padrão, sem porém requerer maior poder computacional.

A Percolação dinâmica usa uma grade  $d$ -dimensional como entrada. Em uma aplicação prática, este grade pode conter dados de propriedades do meio que está sendo explorado, como por exemplo: propriedades petrofísicas dos reservatórios de petróleo.

Com a finalidade de tornar o algoritmo mais genérico e mais eficiente nas redes  $d$ - dimensionais foi estabelecido uma correspondência entre uma rede  $d$ -dimensional e a linear, associando a cada elemento da rede, um número que o identifique, e facilite a determinação de seus vizinhos.

A Percolação dinâmica foi implementada no ZSweepVR como um conjunto de classes que definem a rede e a conectividade entre sítios e a formação dos aglomerados. Neste trabalho nos restringimos as redes hiper cúbicas de dimensão  $d = 3$ , ou seja redes tridimensionais. A identificação de cada elemento da rede linear, assim como o cálculo da quantidade de elementos da rede, é realizada ao fornecer a dimensão do hiperparalelepípedo e suas arestas.

Nós usamos o método proposto por Freitas *et al* [9] com duas melhorias. A primeira melhoria está na entrada de dados com a ordenação que os classifica.

Introduzimos uma estrutura *heap* a fim melhorar o desempenho da ordenação. O *heap* reduz também o número das visitas nos sítios, quando sítios diferentes têm o mesmo valor da propriedade estudada. A segunda está na formatação da estrutura de dados de saída para o módulo de visualização, que inclui a representação linear de localização espacial dos sítios.

A estrutura de dados linear utilizada pelo módulo de percolação permite uma rápida e eficiente identificação de cada sítio e/ou ligação na rede, assim como sua topologia. Além disto, reduz a sobrecarga no pipeline de comunicação entre o módulo de simulação e o de visualização, sendo necessário apenas um número para representar as coordenadas espaciais  $x$ ,  $y$  e  $z$  de localização de cada sítio. Escalares que representam propriedades assim como os momentos de percolação também são enviados pelo pipeline.

Os métodos responsáveis pela implantação da grade residente na memória identificam a representação linear que chega do módulo de simulação e especifica a geometria e conectividade necessária ao módulo de Visualização.

Os elementos geométricos utilizados pelo visualizador de superfície e pelo renderizador volumétrico são hexaedros gerados pela relação de conectividade estabelecida anteriormente. Os hexaedros podem facilmente serem inseridos no heap pertencente ao renderizador volumétrico ZSweep, gerando com isto maior flexibilidade na geração de imagens.

## **Servidor de Renderização**

O servidor de renderização é responsável pela apresentação das imagens resultantes dos resultados da Percolação. Este é composto de dois diferentes módulos:

- Renderização de Superfície;
- Renderizador Volumétrico.

O módulo de renderização de superfície gera as imagens 3D usando a biblioteca de diretrizes OpenGL. O módulo de renderização Volumétrica gera imagens do conjunto de dados, usando o algoritmo de renderização Volumétrica ZSweep. Utilizando técnicas de estereoscopia, a fim fornecer mais informação e mais introspecção aos dados.

Nós escolhemos ZSweep para a renderização Volumétrica devido a sua simplicidade, eficiência de memória e por ser um algoritmo rápido de projeção das células. ZSweep permite a renderização de grades estruturadas e não estruturadas, composto de células formadas por tetraedros e/ou hexaedros.

## Renderização de Superfícies

O módulo renderização de superfície foi executado com o objetivo de gerar imagens a taxas de Visualização que permitam a interação em tempo real e fornece ao usuário informações através de imagens da evolução e formação dos aglomerados. Nossa ideia é que o usuário tenha acesso primeiramente a uma animação não detalhada da evolução e que em algum instante  $t_i$  possa gerar uma imagem detalhada utilizando o renderizador volumétrico, sejam elas Estéreo ou não.

Para cada instante  $t_i$  de percolação, o módulo de renderização de superfície usa a estrutura da grade tridimensional fornecida pelo usuário da simulação. Consideramos que cada sítio da percolação é um voxel e que os sítios ocupados contém os valores escalares que correspondem a uma propriedade física específica, normalizada pela probabilidade de ocupação. Estes valores são usados para gerar cores diferentes para cada conjunto, usando a tabela de cores (*lookup table*) da biblioteca de OpenGL.

Como opção disponibilizamos a visualização utilizando a função de transparência do OpenGL, que permite observar a evolução da formação dos aglomerados sem perder a visão do todo o processo (Figura 6.4).

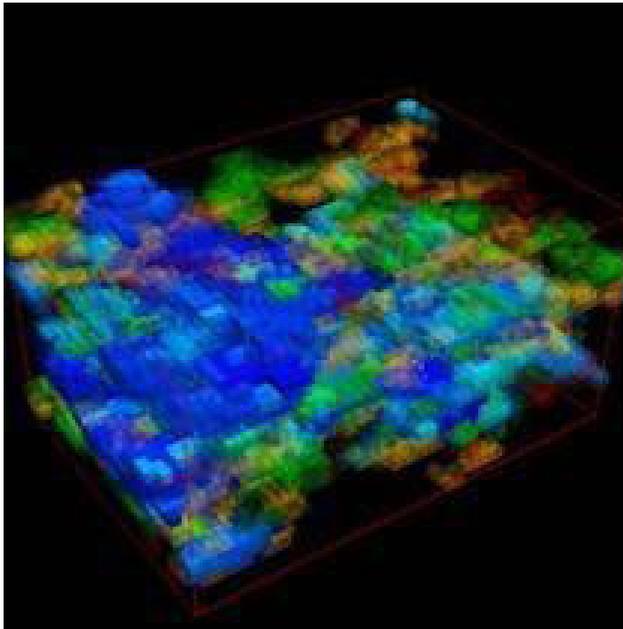


Figura 6.4: renderização de superfícies - Função de transparência.

## Renderização Volumétrica - ZSweep

O renderizador volumétrico ZSweep é o responsável pela Geração de imagens dos

aglomerados de percolação que revelam informações escondidas no interior do volume estudado. Ele disponibiliza a partir da interface do renderizador de superfície, as seguintes opções para geração de imagens renderizadas volumetricamente:

1. Geração de um conjunto de imagens correspondentes ao processo de evolução dos aglomerados;
2. Geração de um conjunto de imagens relativas a um instante  $t_i$  escolhido correspondente a Rotação em 360 graus em torno de um ou mais eixos (  $x$ ,  $y$  e  $z$ ), a um determinado passo angular definido pelo usuário.
3. Geração de imagem que corresponde ao ponto de Visualização definido pelo usuário em um instante  $t$ . Ver Figura 6.5.
4. Geração de imagens correspondentes a uma cenário de visualização explorado pelo usuário a partir de opções disponibilizadas no renderizador de superfície.

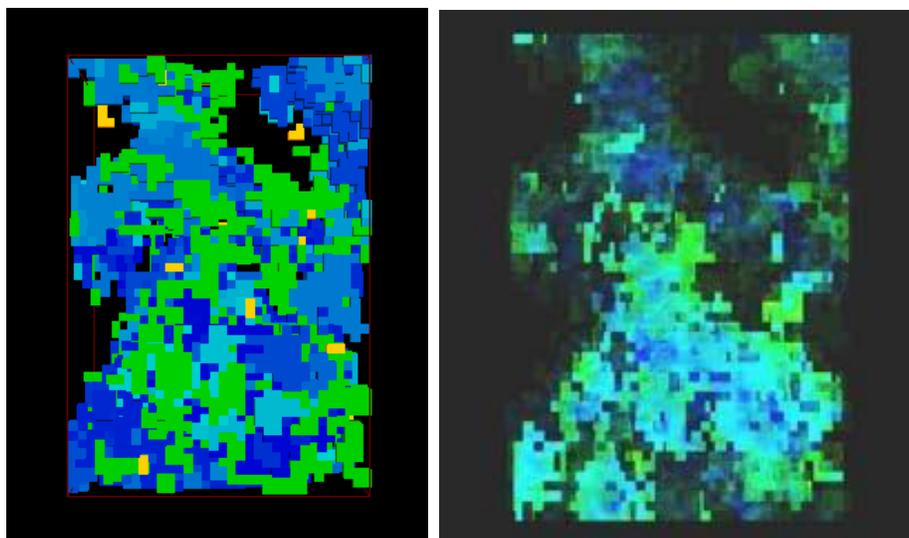


Figura 6.5: Imagem do renderizador de superfície e sua correspondente no renderizador volumétrico

Os itens um e dois podem ser apresentados em forma de filme, resultando em uma boa abordagem para exploração de eventos temporais (item 1) e espaciais (item 1 e 2). O item três e quatro disponibiliza também a Geração de imagem Estéreo que pode ser observada em ambientes imersivos utilizando óculos especiais (Figura 6.6).

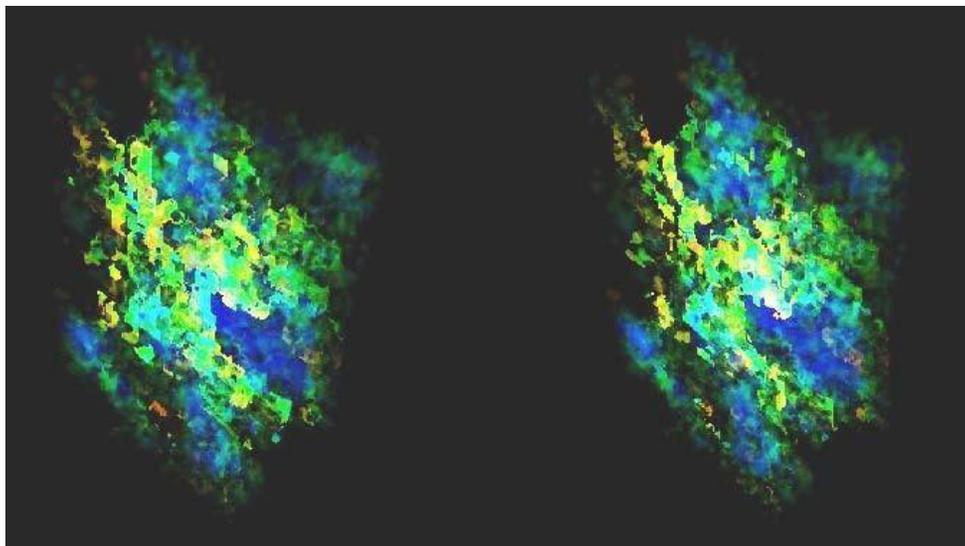


Figura 6.6: Par Estéreo - Renderizador Volumétrico.

A geração de imagens Estéreo é uma opção que pode ser voltada ao uso em ambientes imersivos colaborativos, com a função de adicionar realismo e facilitar a compreensão da base de dados volumétrica. Isto permite que equipes multidisciplinares possam discutir e emitir opiniões necessárias a uma tomada de decisão segura.

## EXPERIMENTO E RESULTADOS

Neste Capítulo apresentamos os resultados de experimentos realizados com o sistema ZSweepVR, utilizando um conjunto de dados reais oriundo de um campo de produção maduro, ou seja de reservatórios de petróleo que já deixaram de produzir espontaneamente.

Avaliaremos inicialmente o servidor de percolação e em seguida o servidor de visualização comparando as imagens e o desempenho dos módulos de renderização de superfície e Renderização Volumétrica, assim como a Geração de imagens Estéreo.

## METODOLOGIA

Para realizar os experimentos utilizamos um computador Intel XEON 2.00 GHz e 1Gb de memória. Como entrada para o servidor de simulação utilizamos uma base de dados Volumétrica oriunda de informações de análise geoestatística feita no simulador de fluxo IMEX da CMG [5] e interpretação de parâmetros petrofísicos de um campo de produção maduro. Estes dados e parâmetros são obtidos a partir de análise de testemunhos de perfis de poços e de ensaios onde se faz medidas de velocidade de propagação do som.

Para análise do desempenho do sistema computacional, associamos duas características particulares do modelo de percolação dinâmica, são elas:

- geração de aglomerados ( número de sítios adicionados ao sistema no tempo )
- número de sítios acumulados no tempo.

Estes dois itens estão diretamente ligados com a comunicação entre os servidores e com o volume de voxels a serem renderizados pelo sistema de Visualização.

O servidor de simulação é o responsável pela Geração da simulação de percolação dinâmica com a montagem de cenários a serem renderizados no servidor de Visualização.

A base de dados utilizada por este simulador gera uma grade de  $36 \times 51 \times 27$ , onde cada célula corresponde a uma região de  $25\text{m} \times 25\text{m} \times 0.5\text{m}$ , representada por uma estrutura linear. Estes dados trafegam entre o sistema de percolação e o de visualização, de forma que apenas um valor numérico representa as coordenadas espaciais x, y e z de cada sítio da rede de percolação. Além das informações sobre localização espacial, são também enviados os atributos relacionados à codificação das informações

em imagens. No sistema de Visualização, estes dados são tratados geométrica e topologicamente, gerando os hexaedros necessários à renderização.

O gráfico da Figura 7.1 mostra o fluxo de informações, ou seja o número de sítios que trafegam entre o simulador de percolação e o módulo de Visualização, sendo este um bom indicativo do desempenho no pipeline de comunicação.

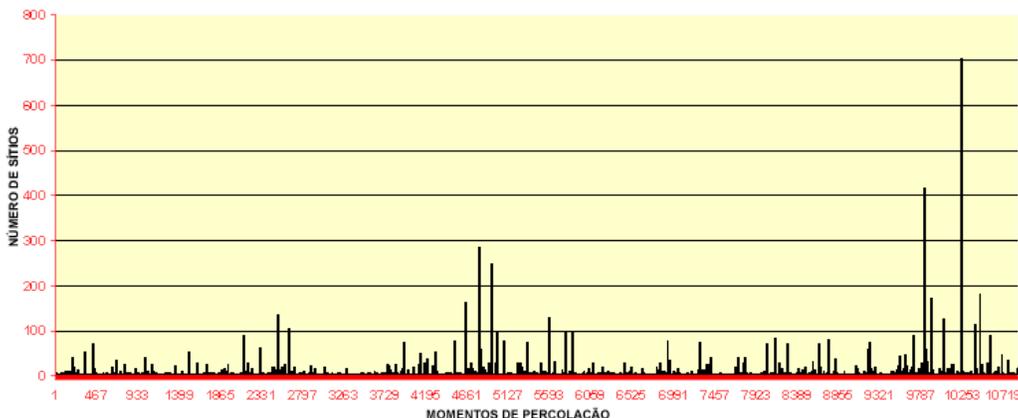


Figura 7.1: Fluxo de informações que trafega no pipeline de comunicação entre o servidor de simulação e o servidor de Visualização

O sistema de visualização recebe as listas de sítios por momentos de percolação e gera uma grade espacial envolvendo a definição geométrica e topológica necessária ao renderizador.

O fluxo de informações que chega ao pipeline de renderização é crescente e variável com o número de sítios que são adicionados para formação dos aglomerados até que se tenha alcançado o limiar de percolação, sendo este um fator determinante do desempenho de Renderização de superfície e de Renderização Volumétrica. Podemos assumir que esta varia conforme o gráfico da Figura 7.2.

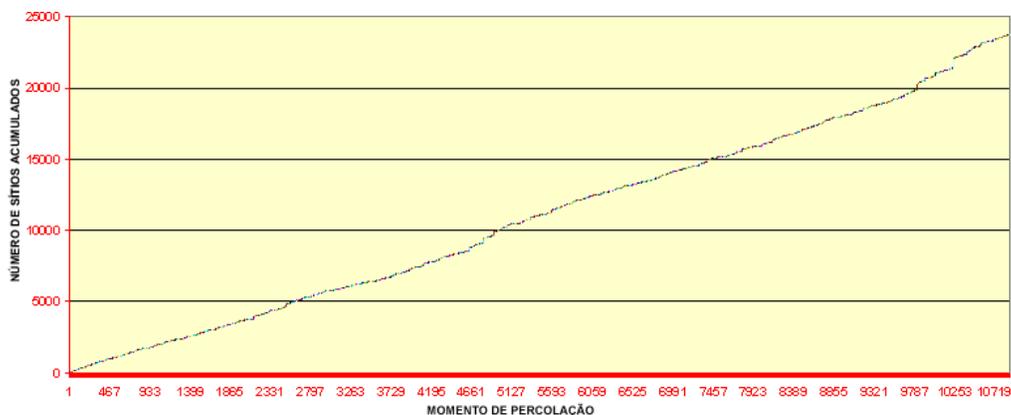


Figura 7.2: Desempenho do sistema de renderização diretamente ligada ao número de voxels adicionados ao pipeline de renderização no tempo

Para análise e comparação das imagens geradas pelos módulos de renderização de superfície e de renderização Volumétrica, separamos uma amostra que se relaciona com a quantidade de sítios adicionados ao volume de percolação no tempo. A seguir exploraremos este experimento e analisaremos os resultados.

## EXPERIMENTO E ANÁLISE DE RESULTADOS

Tomando como base a tabela apresentada na Figura 7.3 que fornece o fluxo de informações geradas pelo servidor de simulação para o servidor de Visualização com um número de sítios adicionados maior que 60. Analisaremos as imagens geradas pelos módulos de renderização de superfície e Renderização Volumétrica, assim como as imagens Estéreo.

A coluna **MOMENTO** refere-se ao passo em que o simulador libera um número de sítios a serem adicionados, informado na coluna **sítios ADICIONADOS**. A coluna **NÚMERO DE sítios** informa o número de sítios acumulados na base de dados Volumétrica a serem renderizados.

PASSO	MOMENTO	NR. SÍTIOS	SÍTIOS ADICIONADOS
1	464	974	70
2	2168	3933	88
3	2351	4317	62
4	2551	4809	134
5	2673	5141	104
6	3975	7286	75
7	4550	8451	77
8	4667	8778	162
9	4836	9374	284
10	4963	9925	247
11	5017	10115	97
12	5111	10344	77
13	5371	10866	76
14	5618	11430	128
15	5808	11842	97
16	5878	12043	100
17	<b>6956</b>	13960	78
18	7329	14643	76
19	8081	16089	70
20	8171	16313	83
21	8309	16636	69
22	8670	17382	70
23	8777	17637	81
24	9238	18557	75
25	<b>9730</b>	19604	88
26	<b>9856</b>	20256	417
27	<b>9945</b>	20650	173
28	10075	20996	127
29	<b>10278</b>	22105	703
30	10435	22502	114
31	10490	22764	180
32	10596	23089	89

Figura 7.3: Fluxo de informações gerados pelo servidor de simulação para número de sítios adicionados maior que 60

Os gráficos das Figura 7.4 e 7.5 estabelecem o desempenho dos dois servidores para os dados em análise distribuídos em 32 passos ( momentos ) da simulação de percolação. Este desempenho é função dos momentos de percolação e nestes caracterizamos o número de sítios que estão sendo adicionados na formação dos aglomerados, com consequente contribuição na formação do aglomerado percolante (passo de número 10278).

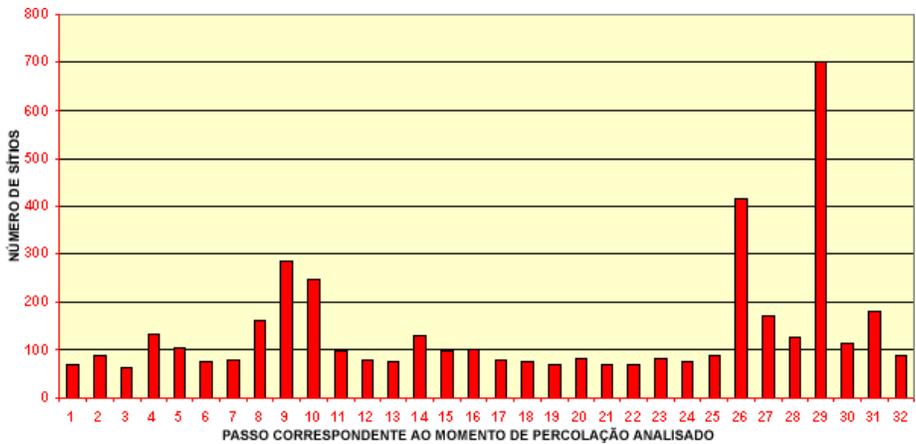


Figura 7.4: sítios adicionados ao sistema no tempo

Os gráficos aqui apresentados caracterizam o processo evolutivo de Geração de aglomerados, sendo importante ressaltar que quando um número grande de sítios é adicionado ao pipeline de comunicação é provável a junção entre aglomerados.

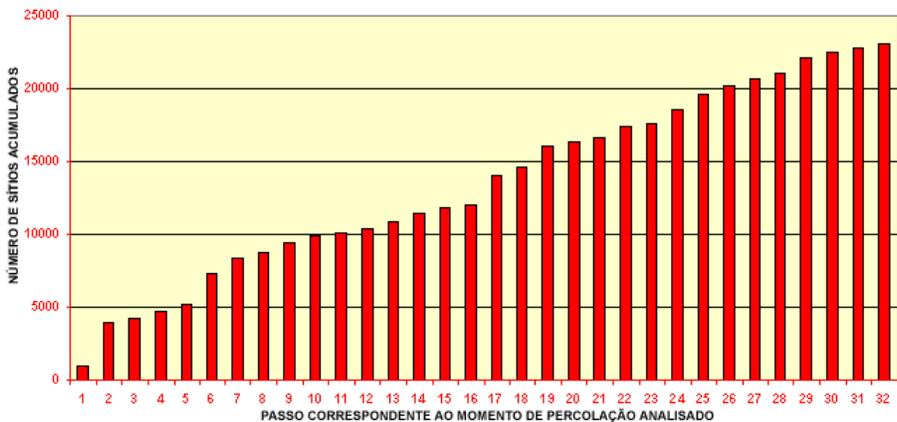


Figura 7.5: número de sítios acumulados no tempo

## Renderização de Superfície

A Renderização de superfície é executada a partir de diretrizes do OpenGL e não permite o acesso direto a detalhes da estrutura interna do volume. A Figura 7.6 mostra o resultado da renderização de superfícies para o momento final de percolação (10950).

Nela podemos diretamente observar uma característica presente nos reservatórios de petróleo, que são espaços tridimensionais complexos e que apresentam heterogeneidades geológicas que variam espacialmente.

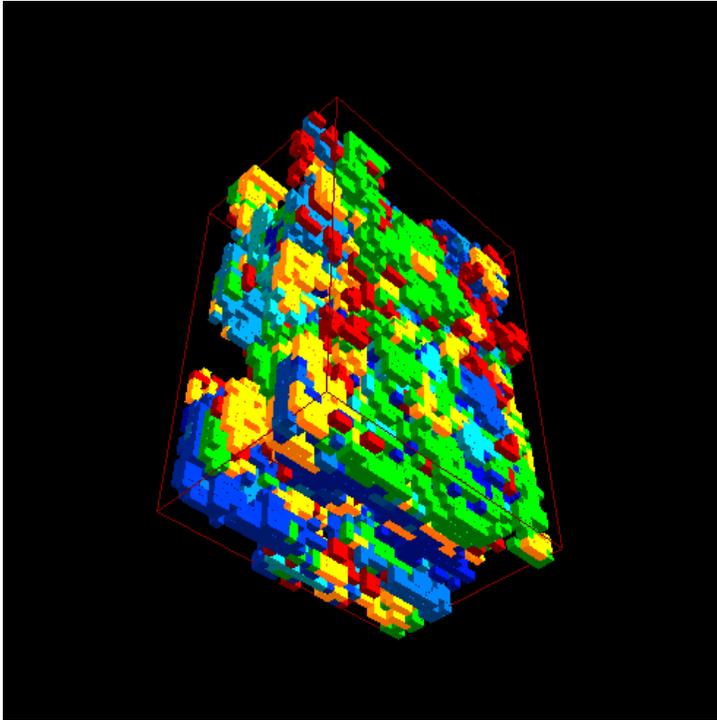


Figura 7.6: Momento final de percolação - renderização de superfície

No contexto do reservatório de petróleo, os espaços preenchidos estão associados a espaços porosos da rocha sedimentar que poderão estar preenchidos ou não com o petróleo. Neste experimento, podemos caracterizar a distribuição espacial da heterogeneidade presente no reservatório, e como estas influenciam o processo de exploração e recuperação de petróleo. Informações locais e individuais, assim como distribuídas e em aglomerados, podem ser observadas e analisadas através de todo o processo de evolução da percolação.

As Figuras 7.32 e 7.34, apresentadas no final deste Capítulo demonstram o processo de evolução da simulação de Percolação, com imagens correspondentes aos momentos de percolação, 464, 2168, 2351, 2673, 5371, 5618, 5808, 5878, 9730, 9856, 9945, 10075, 10278 e 10596 relacionados na tabela da Figura 7.3 e o instante final 10950, dispostos da esquerda para direita de cima para baixo.

É importante observar que estes são momentos onde um volume considerável de sítios foram adicionados, com provável junção de aglomerados. Observe no gráfico da Figura 7.7 o comportamento de formação de aglomerados no tempo. Nem todos os aglomerados formados farão parte do aglomerado percolante.

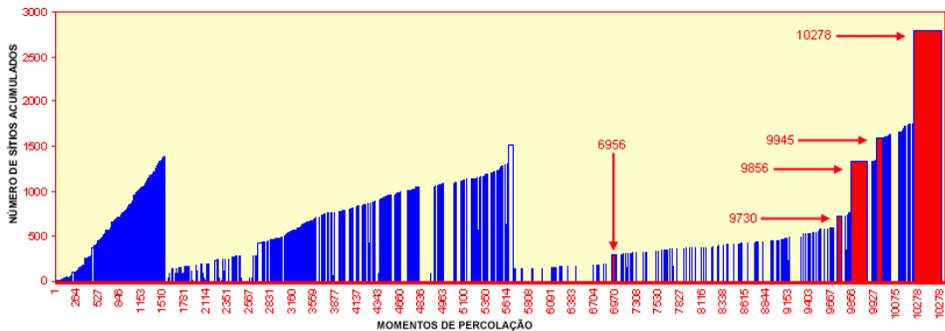


Figura 7.7: Comportamento de formação dos aglomerados no tempo

Os momentos 6956, 9730, 9856, 9945 e 10278 realçados na Figura 7.3 e indicados no gráfico da Figura 7.7 mostra as principais junções de aglomerados que fazem parte da construção do aglomerado percolante. O gráfico da Figura 7.8 apresenta o comportamento da evolução do aglomerado percolante de forma isolada dos demais aglomerados.

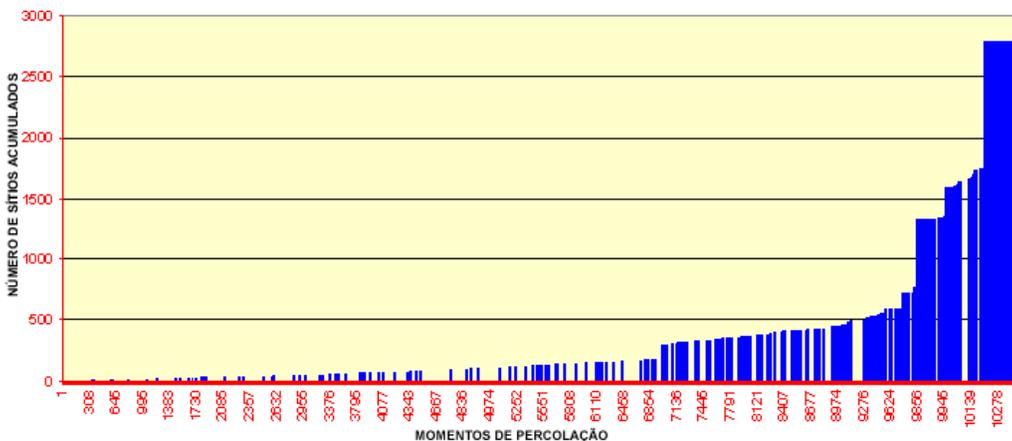


Figura 7.8: Momento final de percolação

A fim de visualizar os aglomerados formados através do processo evolutivo sem perder a visão global da formação dos mesmos, usamos funções de transparência disponíveis na API OpenGL. Usando transparência, o usuário pode visualizar a formação dos aglomerados durante o processo evolutivo correlacionando-os com os demais, como pode ser observado no conjunto de imagens na Figura 7.9.

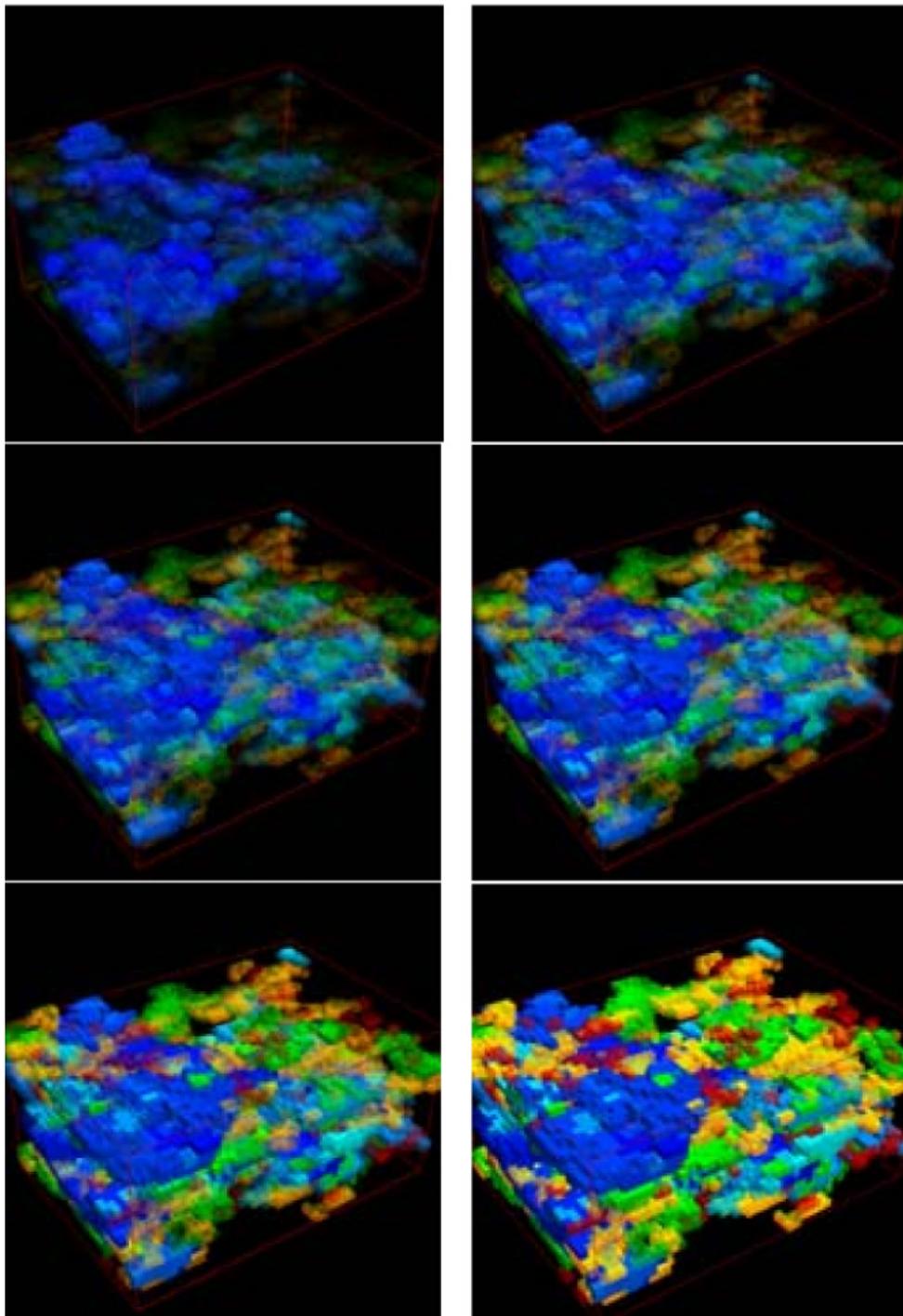


Figura 7.9: Usando a função de transparência para observar o interior do volume

O aspecto mais importante sobre o renderizador de superfície é a obtenção de taxas de visualização que podem chegar facilmente até 30 frames por segundo (*frame-rate = 30 fps*). Com esta taxa, o ZSweepVR pode ser considerado uma ferramenta interativa de tempo real quando se usa o módulo Renderização de superfície (OpenGL).

A Renderização Volumétrica é um processo muito caro e demorado. Neste caso adotamos no ZSweepVR a seguinte abordagem: As imagens são volumetricamente renderizadas em *background*, ou seja, enquanto o usuário navega e interage utilizando o renderizador de superfície, o renderizador volumétrico pode estar processando imagens que foram solicitadas pelo usuário. Estas imagens podem ser solicitadas a partir do renderizador de superfície em um determinado instante pelo usuário que define que opção quer explorar.

Em nossos experimentos, observamos que se o usuário despende 30s observando uma animação, em média o sistema pode gerar até aproximadamente 13 imagens renderizadas volumetricamente ou 6.5 imagens estereoscópicas em *background*. Imagens Estéreo podem ser renderizadas e analisadas em ambiente imersivo colaborativo.

### Renderização Volumétrica

O módulo de renderização Volumétrica ZSweepVR gera imagens muito mais detalhadas do que as criadas pelo módulo de renderização de superfície. Observe na Figura 7.10b a influência de estruturas internas do volume na formação da imagem, acentuando características não visualizadas na imagem do renderizador de superfície (Figura 7.10a).

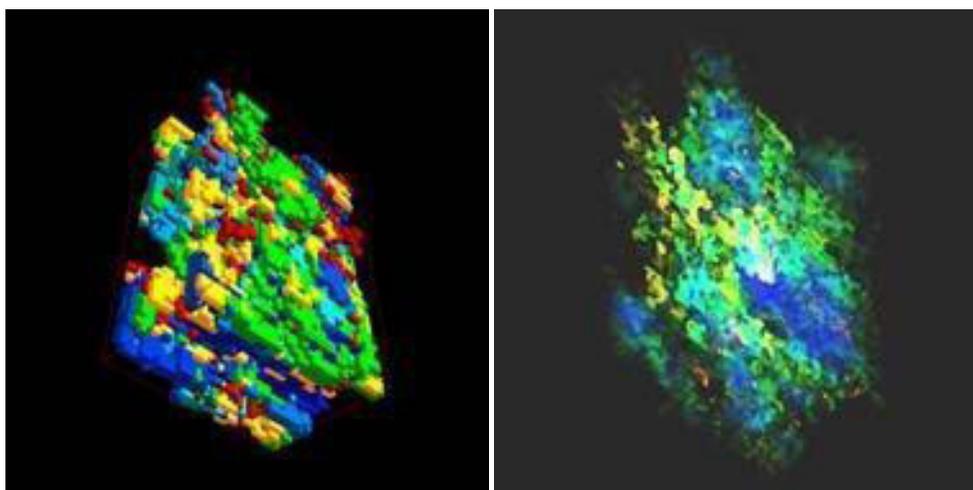


Figura 7.10: Imagens geradas pelo renderizador de superfície e renderizador volumétrico - Momento 10950

As Figuras 7.11a e b são resultantes do momento de percolação 4667. Utilizamos o renderizador de superfície para gerar a imagem da Figura 7.11a, enquanto que para Figura 7.11b utilizamos o renderizado volumétrico. Duas características importantes podem ser associadas a imagens renderizadas volumetricamente oriundas do processo de percolação dinâmica, são elas a densidade e a localização espacial dos aglomerados dentro do volume.

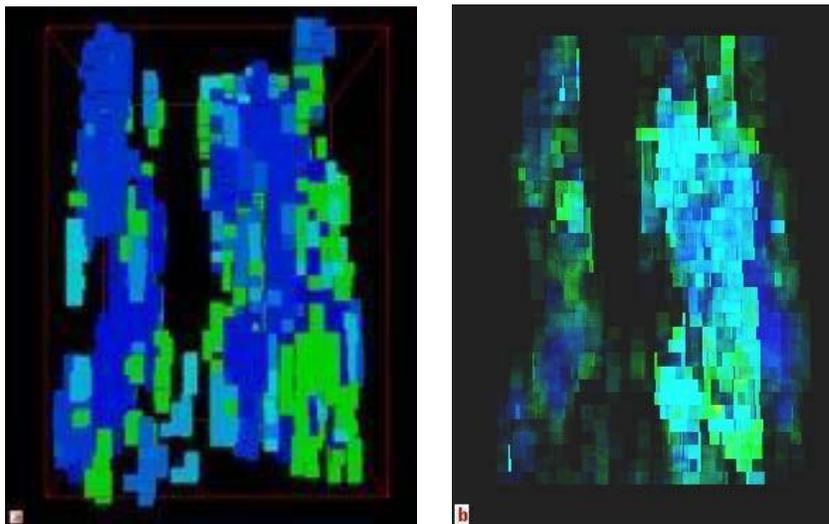


Figura 7.11: Imagens geradas pelo renderizador de superfície e renderizador volumétrico - Momento 4667

Na imagem da Figura 7.11b observa-se com clareza detalhes do interior do volume explorado, que não são possíveis de visualizar utilizando apenas as imagens geradas pelo renderizador de superfície. Regiões onde a densidade de aglomerados se torna maior, são reveladas pela imagem da direita.

A Figura 7.11a, neste ponto de Visualização, não fornece informações relevantes sobre o processo de percolação. Apesar de está sendo projetada em perspectiva, muito pouco do que está por trás dos sítios frontais fica visível. Sem o uso do renderizador volumétrico duas alternativas nos restariam para fazer esta análise. A primeira seria buscar outros pontos de visualização para interpretar melhor a imagem e a segunda utilizar-se de técnicas de transparência implementadas no ZSweepVR, no módulo em OpenGL.

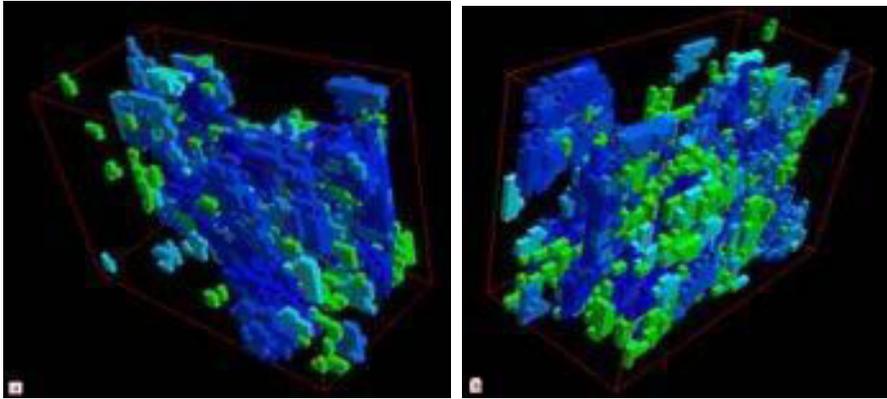


Figura 7.12: Rotação lateral na base de dados Volumétrica - Momento 4667

As Figuras 7.12a e b demonstram o processo de Rotação no sentido horário e anti-horário do volume, a fim de permitir a caracterização da distribuição dos sítios nas laterais do volume e avaliar sua contribuição para interpretação do problema. Mesmo assim os sítios presentes na lateral esquerda e direita, parte superior e inferior não contribuem para formação da imagem mental do que existe no interior do volume.

Na busca de solução para este problema, adotamos a abordagem de aumentar a transparência das imagens no renderizador de superfície em função do processo de formação dos aglomerados, tornando assim possível visualizar o interior do volume explorado.

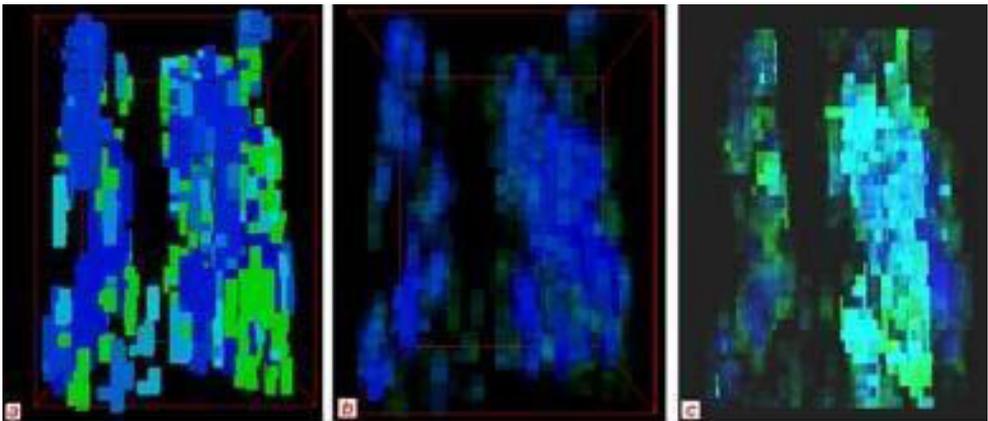


Figura 7.13: Opções de visualização - Vista lateral - Momento 4667

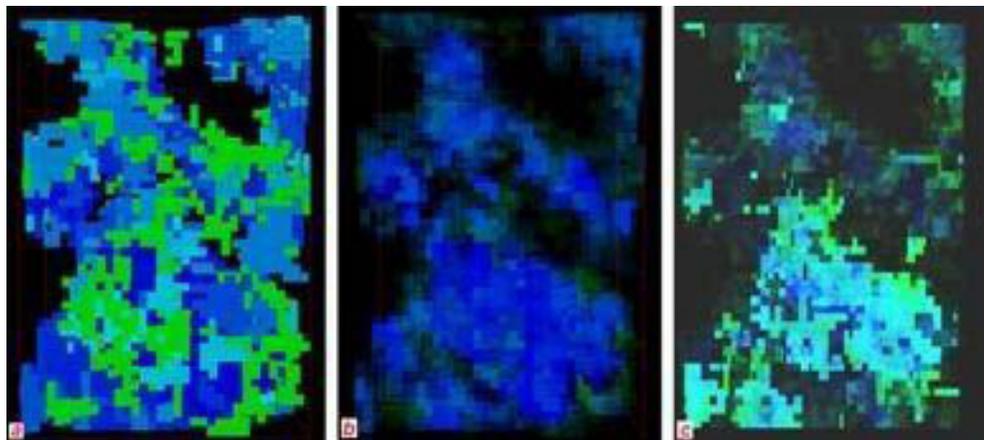


Figura 7.14: Opções de visualização - Vista de topo - Momento 4667

As Figuras 7.13a, b e c apresentam respectivamente, a imagem gerada pelo renderizador de superfície (a), a mesma imagem com aplicação de transparência, revelando a formação do maior aglomerado (b) e a imagem gerada pelo renderizador volumétrico (c).

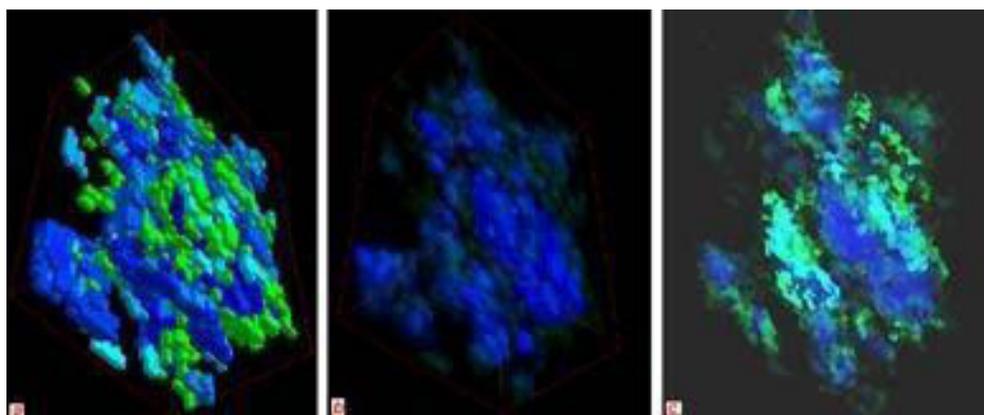


Figura 7.15: Opções de visualização - Vista 45° em torno dos eixos x, y e z - Momento 4667.

Na mesma sequência de formação encontram-se as Figuras 7.14 e 7.15, porém correspondentes a diferentes posições de visualização.

Apesar de ser esta uma boa estratégia, perde-se toda a correlação existente entre possíveis propriedades espalhadas pelos aglomerados. Esta correlação pode ser melhor explorada nas imagens que usam renderização volumétrica.

Um outro sentimento que o usuário pode ter é o de profundidade, que

está ligado diretamente a localização espacial dos aglomerados dentro do volume explorado. Em especial nas imagens renderizadas volumetricamente das Figuras 7.16, 7.17 e 7.18 podemos capturar este tipo de informação.

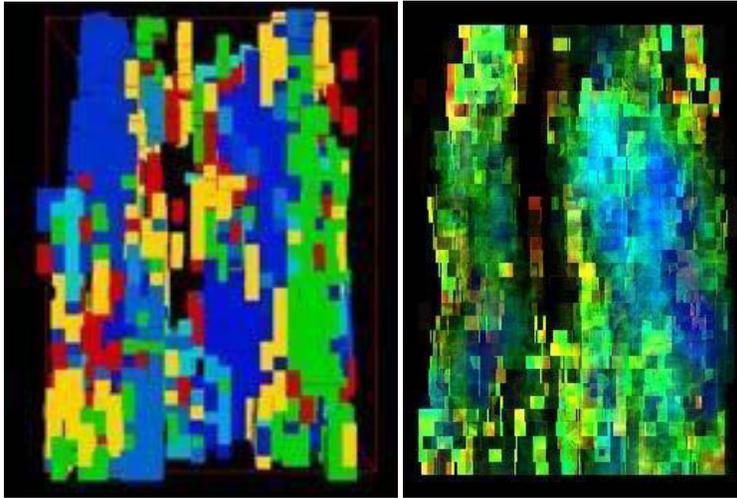


Figura 7.16: Imagens geradas pelo renderizador de superfície e renderizador volumétrico - Momento 10950

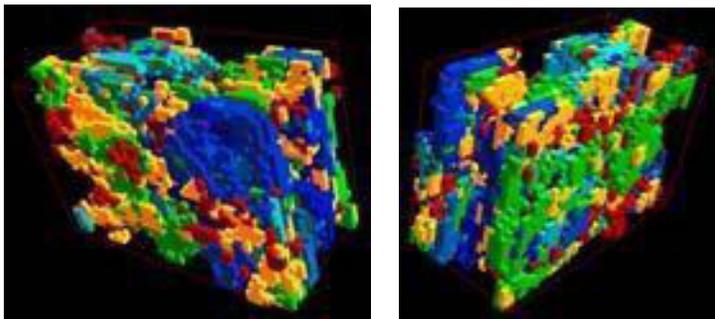


Figura 7.17: Rotação lateral na base de dados Volumétrica - Momento 10950

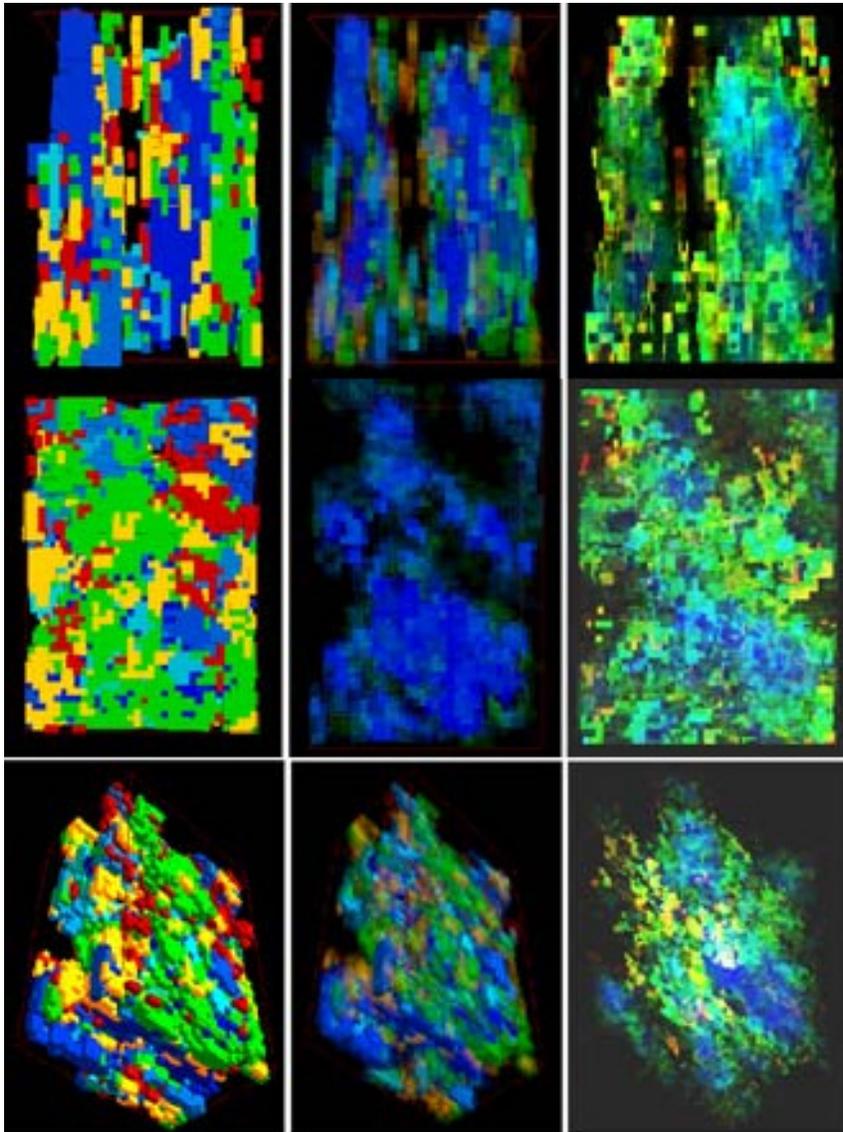


Figura 7.18: Mesmas posições das vistas nas Figuras 7.13, 7.14 e 7.15 - Momento 10950.

Uma boa forma de explorar as diferenças entre o uso de renderizador de superfície e o renderizador volumétrico na Visualização da evolução da simulação de percolação é comparar imagens geradas a partir de um mesmo ponto de Visualização. As imagens das Figuras 7.32 e 7.34, oriundas do renderizador de superfícies, apresentadas no final deste Capítulo estão diretamente correlacionadas com as imagens das Figuras 7.33 e 7.35 renderizadas volumetricamente. Estas demonstram o processo de evolução da simulação de Percolação correspondentes aos momentos

de percolação, 464, 2168, 2351, 2673, 5371, 5618, 5808, 5878, 9730, 9856, 9945, 10075, 10278 e 10596 relacionados na tabela da Figura 7.3 e ao instante final 10950 dispostos da esquerda para direita de cima para baixo.

Para efeito de validação, isolamos o aglomerado principal e o visualizamos no renderizador de superfície e no renderizador volumétrico (Ver Figuras 7.19 e 7.20) .

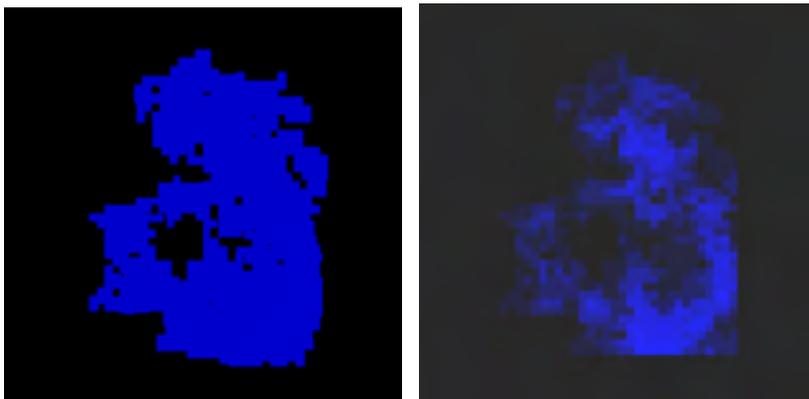


Figura 7.19: Visualização do aglomerado principal

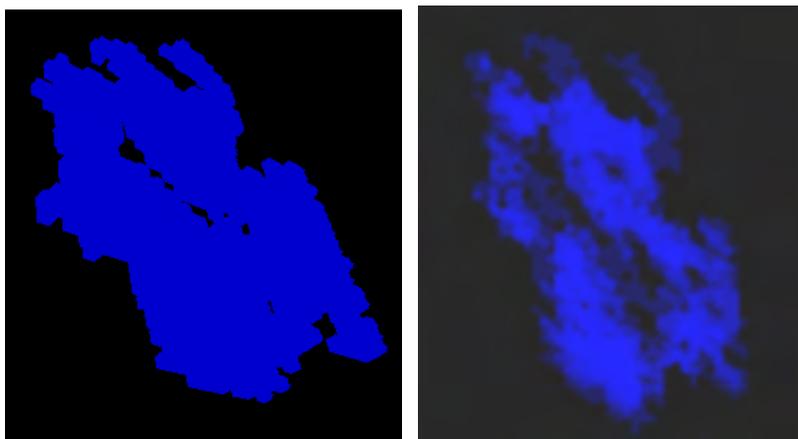


Figura 7.20: Visualização do aglomerado principal. Rotação de 45° em torno dos eixos x,y e z

Se compararmos estas imagens com o grupo de imagens exploradas neste Capítulo, podemos concluir que só a renderização Volumétrica é capaz de revelar em todo o processo de evolução da simulação a forma e a distribuição das heterogeneidades dos aglomerados presentes no interior do volume, evidenciando a formação do aglomerado percolante, melhorando substancialmente o processo.

Até então, as cores presentes nas imagens exploradas indicavam o processo de formação dos aglomerados. Passamos agora a explorar imagens onde as cores presentes estão relacionadas com uma propriedade petrofísica do reservatório, seguindo o mesmo processo de evolução.

Adotamos a mesma metodologia anteriormente empregada para análise da importância do uso da renderização volumétrica na melhor caracterização do volume explorado. A Figura 7.21a, oriunda do renderizador de superfície, não revela informações capazes de fornecer subsídios à tomada de decisão.

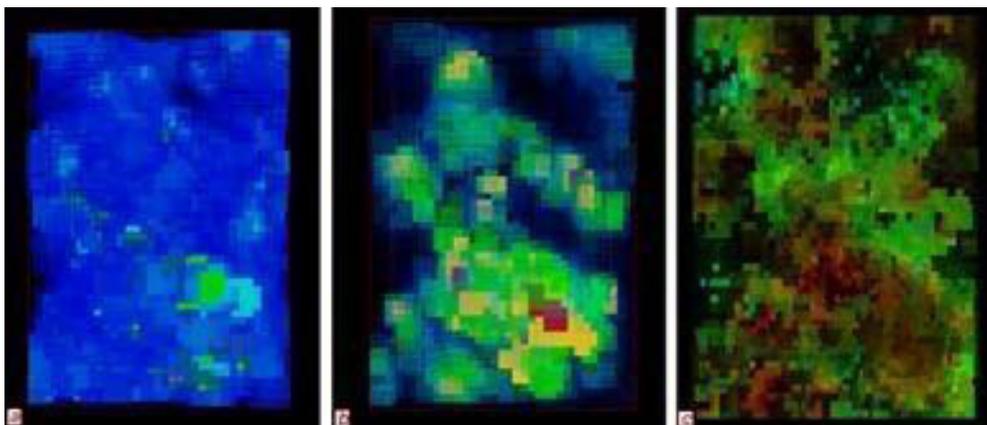


Figura 7.21: Comparação entre as abordagens utilizadas para visualização volumétrica.

O uso da técnica de transparência ( Ver Figura 7.21b) não é suficiente para estabelecer uma imagem precisa das informações contidas no interior do volume. Já a Figura 7.21c mostra com clareza a formação dos aglomerados e a correlação das propriedades presentes no volume.

Dentro desta mesma metodologia, exploramos uma característica especial das imagens renderizadas volumetricamente e que não estão presentes em imagens oriundas do renderizador de superfícies. Variando brilho e contraste podemos capturar mais informações sobre a formação do volume explorado. As Figuras 7.22a, 7.22b e 7.22c apresentam variação de brilho e contraste de 50, 70 e 100 por cento respectivamente.

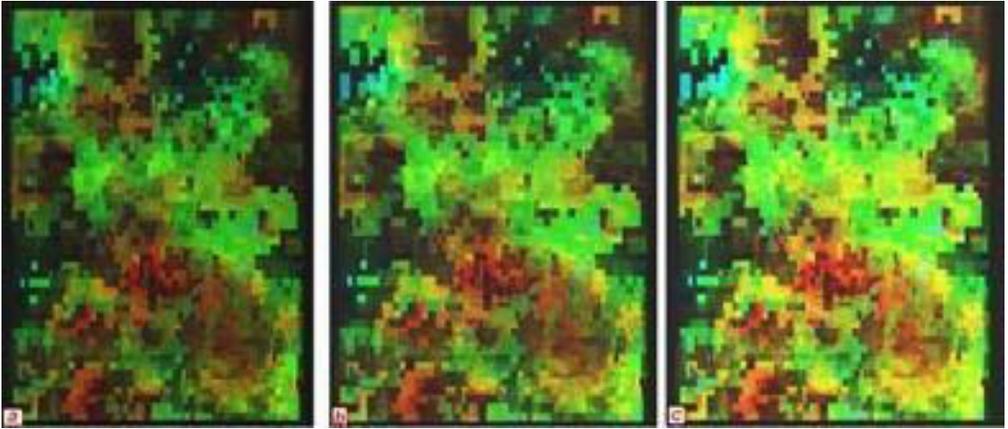


Figura 7.22: Imagens do renderizador volumétrico com variação de brilho e contraste

A qualidade das imagens renderizadas volumetricamente está relacionada diretamente com a função de transferência usada na Iluminação dos dados. Estamos estudando novas funções de transferência e variações de opacidade que revelem melhor as informações contidas na base de dados Volumétrica oriunda da simulação de Percolação.

### Imagens Estéreo

Com a finalidade de aumentar a qualidade do processo de avaliação da simulação de percolação e buscando explorar as vantagens do uso dos ambientes imersivos colaborativos, discutidas no Capítulo 4, aplicamos, as imagens geradas pelo renderizador volumétrico técnicas de estereoscopia discutidas anteriormente.

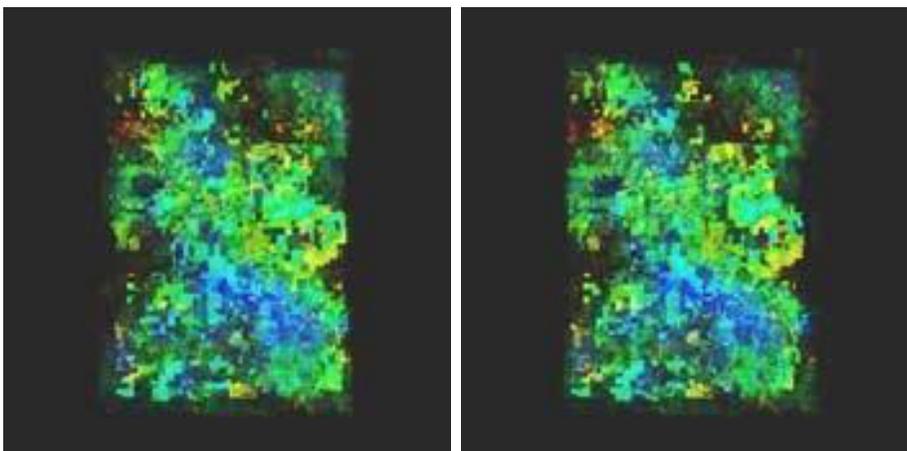


Figura 7.23: Imagens correspondentes ao olho direito e esquerdo - Momento de percolação 10950

A Figura 7.23 mostra as imagens correspondentes ao centro de projeção da direita (RCOP) e ao centro de projeção da esquerda (LCOP), ou seja ao olho direito e ao olho esquerdo do usuário, geradas utilizando o método de projeção off-axis.

Utilizando o ZSweepVR, geramos uma série de imagens Estéreo correspondentes aos momentos de percolação citados na tabela da Figura 7.3. A seguir exploramos os momentos de Percolação 4667 (Figuras 7.24 e 7.25), 9730 (Figuras 7.26 e 7.27) e 10278 (Figuras 7.28 e 7.29). Estas imagens representam o processo de evolução da simulação de percolação para os pontos de Visualização a 45 e 135 graus de Rotação em torno dos eixos x, y e z, aplicadas a posição inicial adotada pelo sistema ZSweepVR e que podemos ver na Figura 7.23.

As Figuras 7.24 a 7.29 mostram pares Estéreo, enquanto as Figuras 7.30 e 7.31 mostram imagens em anaglifo. As imagens geradas pelo ZSweepVR que correspondem a visão pelo olho esquerdo e direito podem também ser visualizadas utilizando estereoscopia ativa. Para isto, necessitamos codificar as imagens para o uso de entrelaçamento (como vimos no Capítulo 4) e observá-las com óculos especiais (*goggles*) em ambientes imersivos.

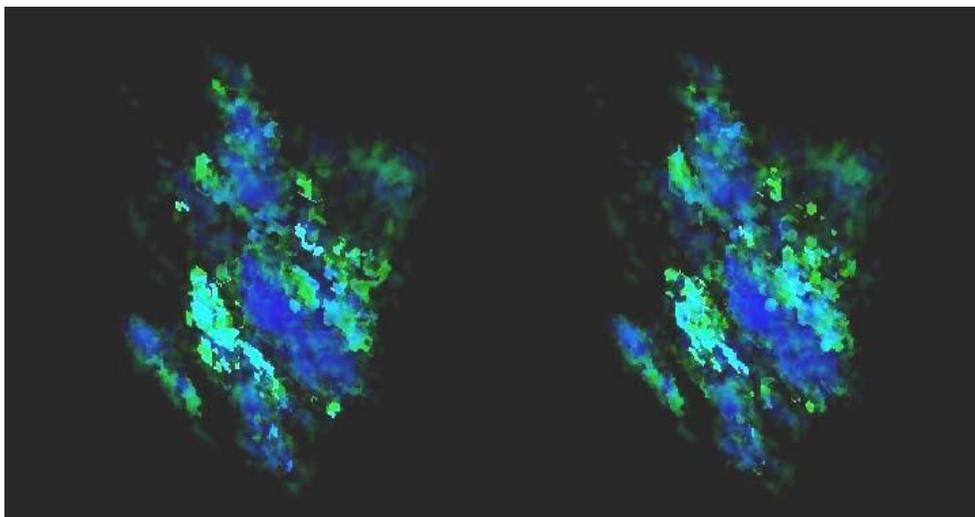


Figura 7.24: Par Estéreo - visualização momento percolação 4667. Rotação de 45o em torno dos eixos x, y e z

Em nossos experimentos, cada imagem com resolução  $512 \times 512$  levou em torno de 2.3 segundos para ser renderizada volumetricamente. Um resultado que é bom considerando-se a complexidade de sua geração. Para Geração das imagens Estéreo, gastou-se aproximadamente 4.6 segundos. Estes resultados mostram que as técnicas de estereoscopia implementadas na API ZSweepVR não provoca sobrecarga, pois estas são geradas com o mesmo espaço de tempo que as imagens mono.

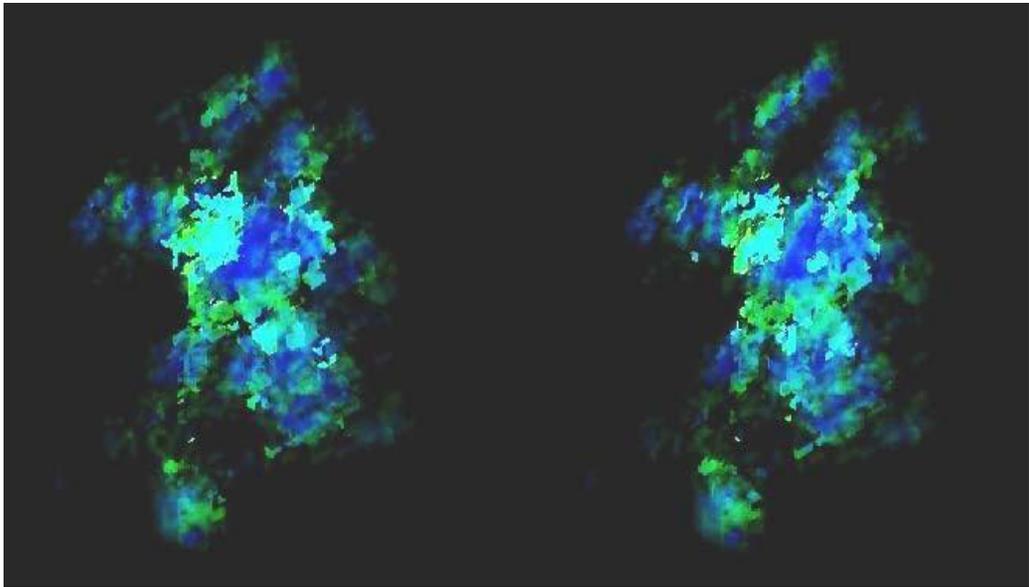


Figura 7.25: Par Estéreo - Visualização momento percolação 4667. Rotação de 135o em torno dos eixos x, y e z

Com isto permite-se uma melhor exploração dos dados oriundos de simulação de percolação, sendo de grande interesse da comunidade científica para aplicações na investigação de fenômenos que envolvem o transporte de fluidos em meios porosos [41]. Além de estimular a pesquisa na área de exploração e recuperação de petróleo, podemos associá-lo às diversas aplicações que podem ser tratadas pela teoria da percolação.

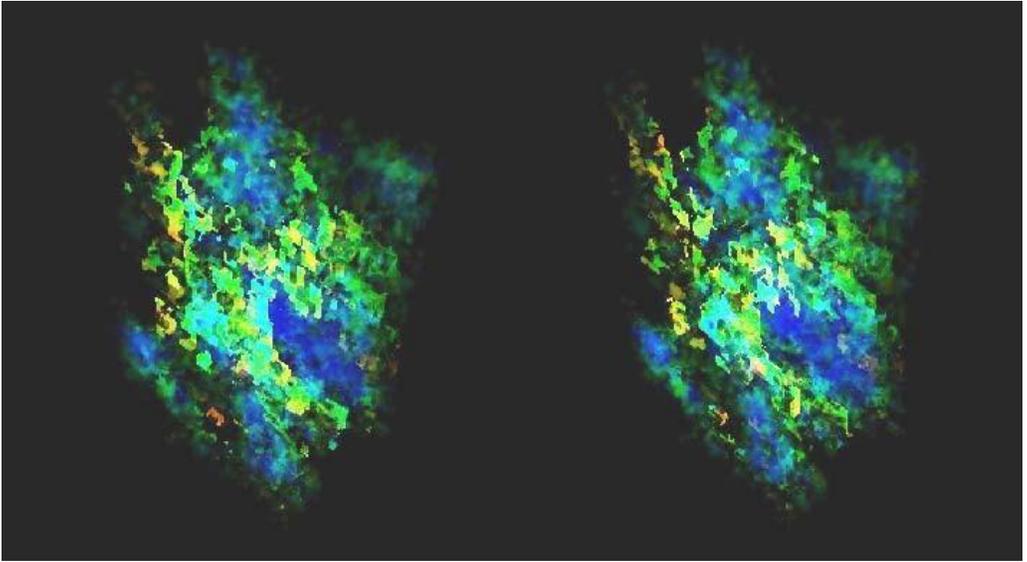


Figura 7.26: Par Estéreo - visualização momento percolação 9730. Rotação de  $45^\circ$  em torno dos eixos x, y e z

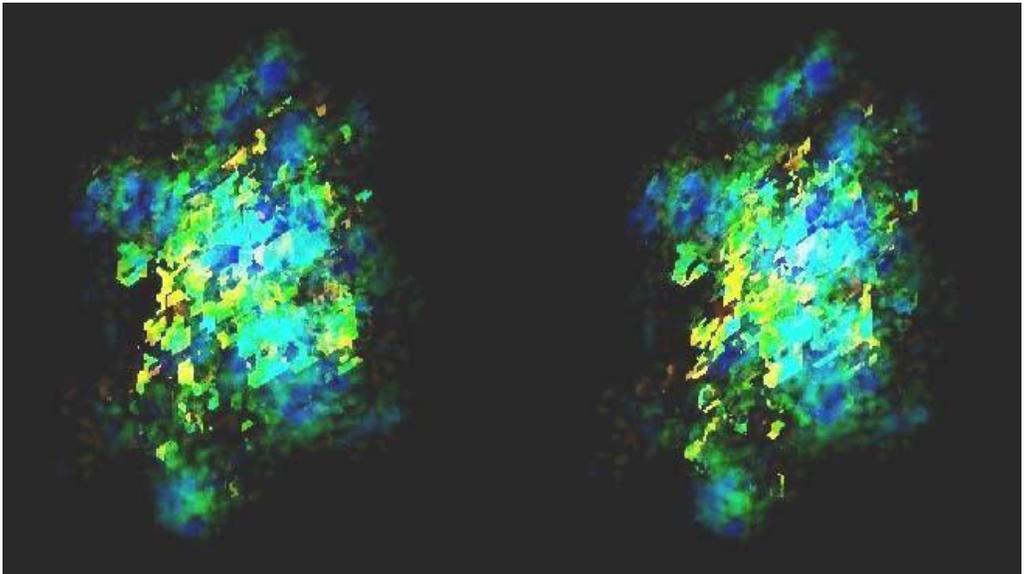


Figura 7.27: Par Estéreo - Visualização momento percolação 9730. Rotação de  $135^\circ$  em torno dos eixos x, y e z

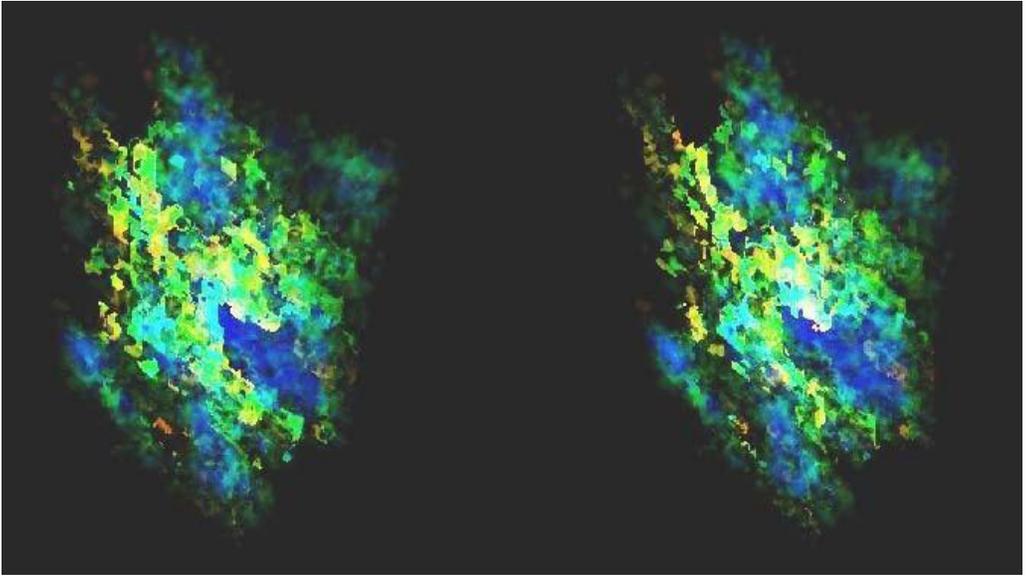


Figura 7.28: Par Estéreo - Visualização momento percolação 10278. Rotação de  $45^\circ$  em torno dos eixos x, y e z

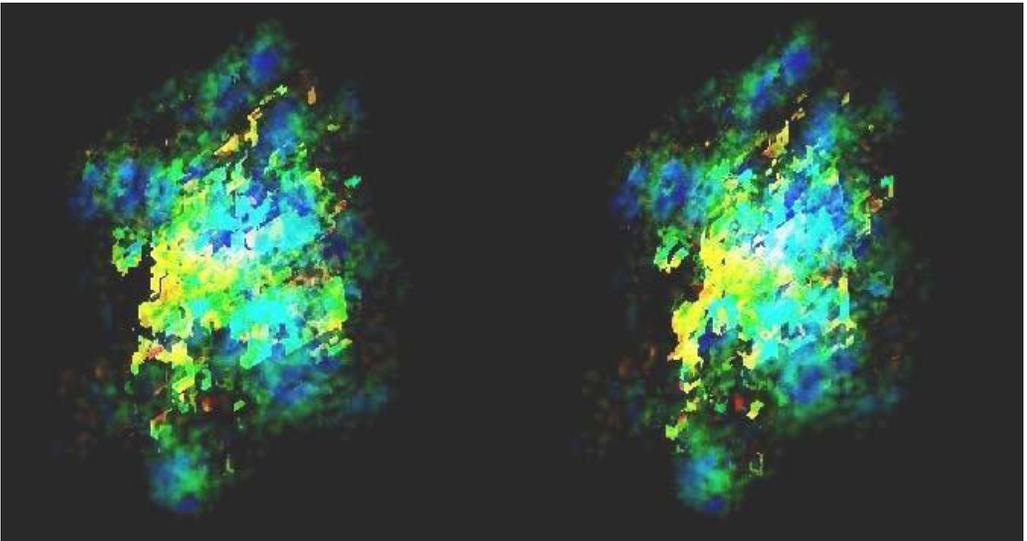


Figura 7.29: Par Estéreo - visualização momento percolação 10278. Rotação de  $135^\circ$  em torno dos eixos x, y e z

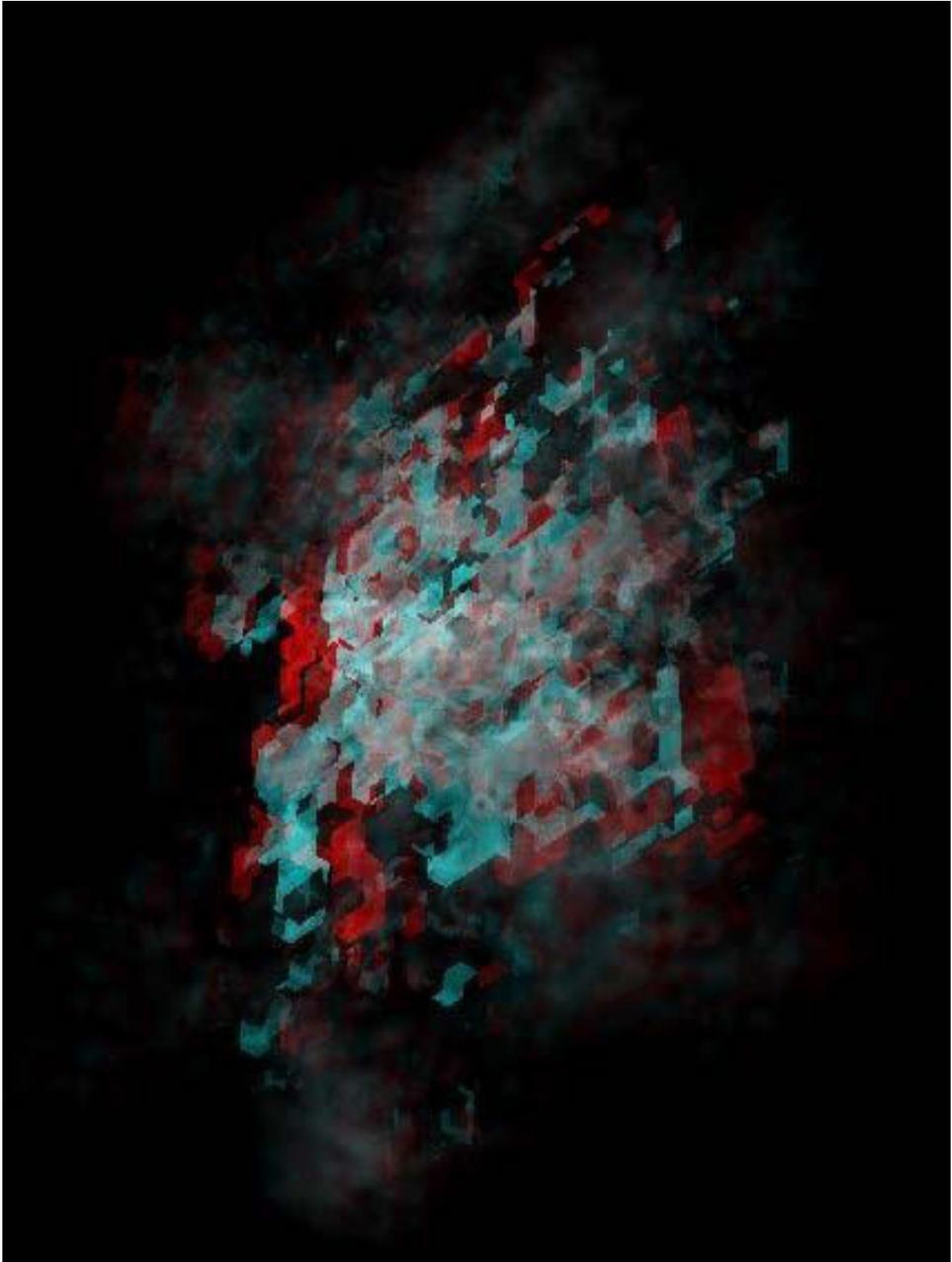


Figura 7.30: Anaglifo - Visualização momento percolação 10278. Rotação de  $135^\circ$  em torno dos eixos x, y e z

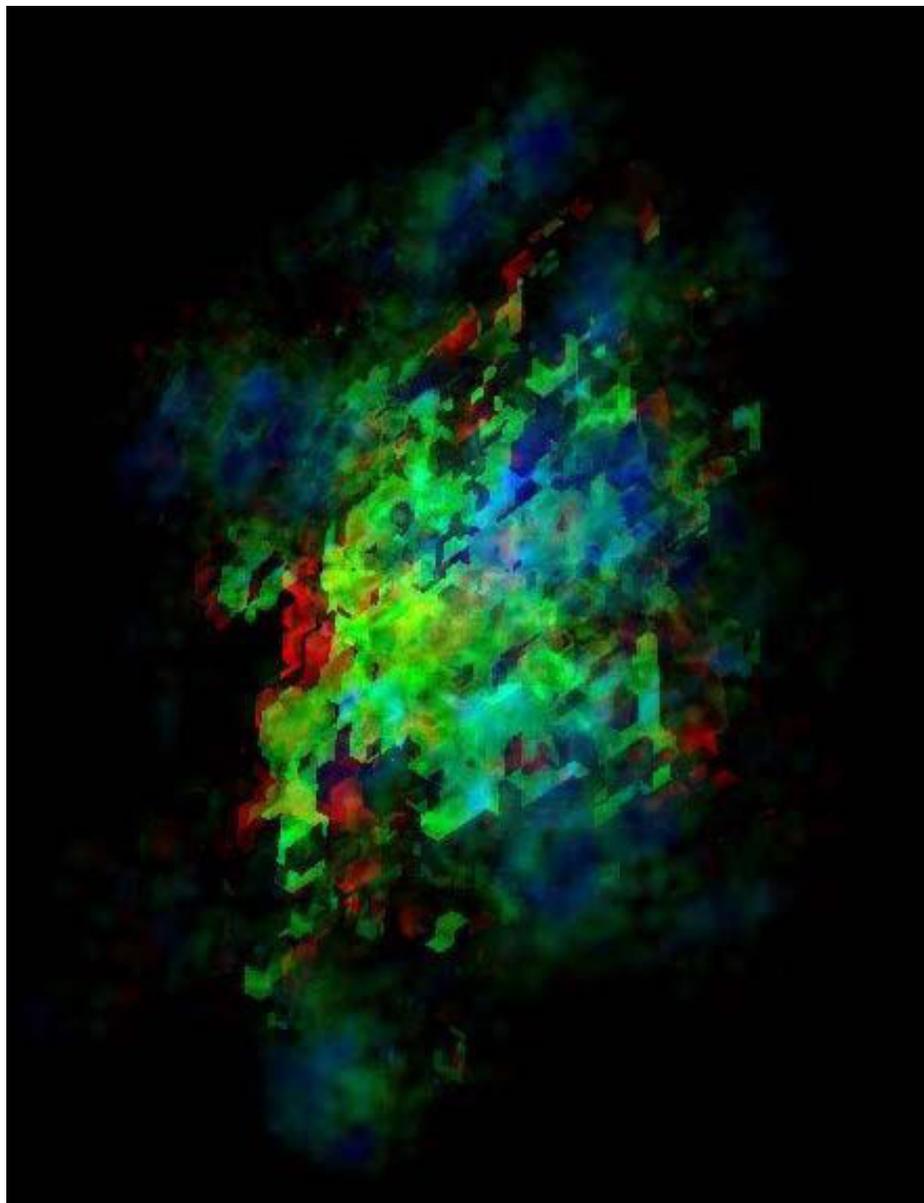


Figura 7.31: Anaglifo - Visualização momento percolação 10278. Rotação de  $135^\circ$  em torno dos eixos  $x$ ,  $y$  e  $z$

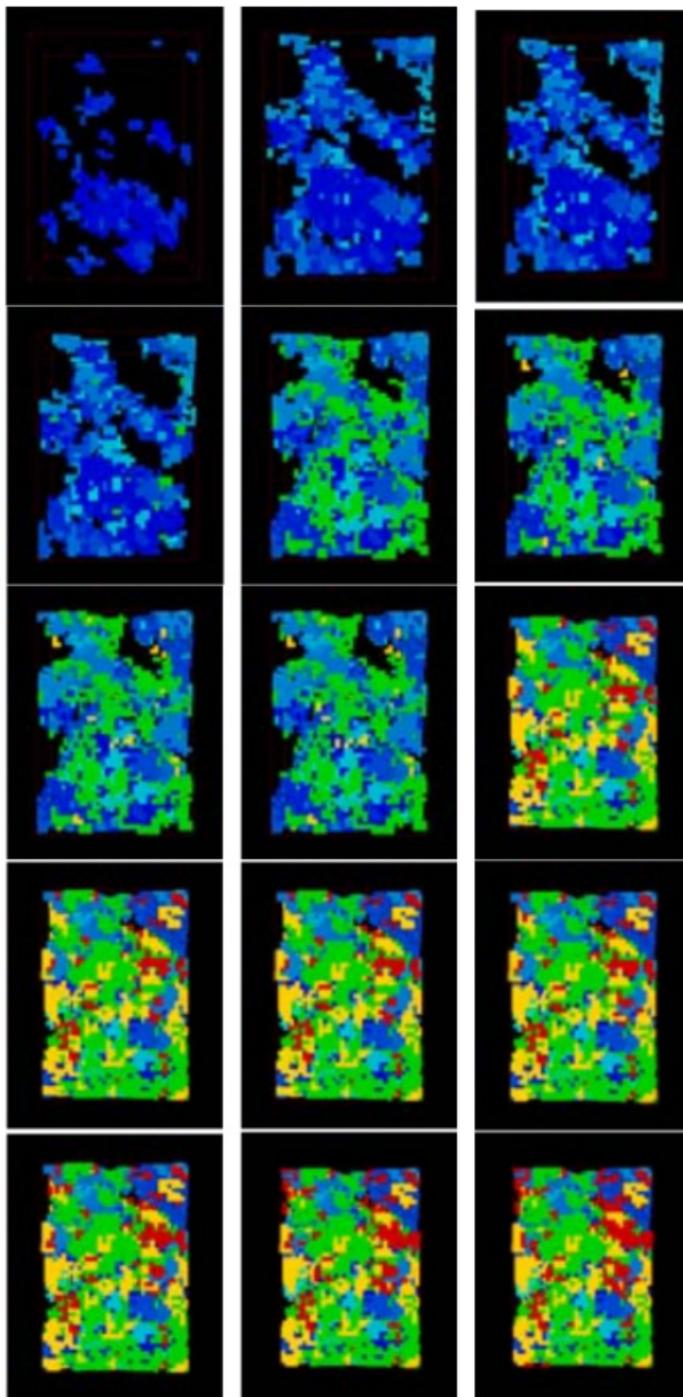


Figura 7.32: Sequência de imagens que representa a formação dos aglomerados - Rendrição de superfície - Vista de topo

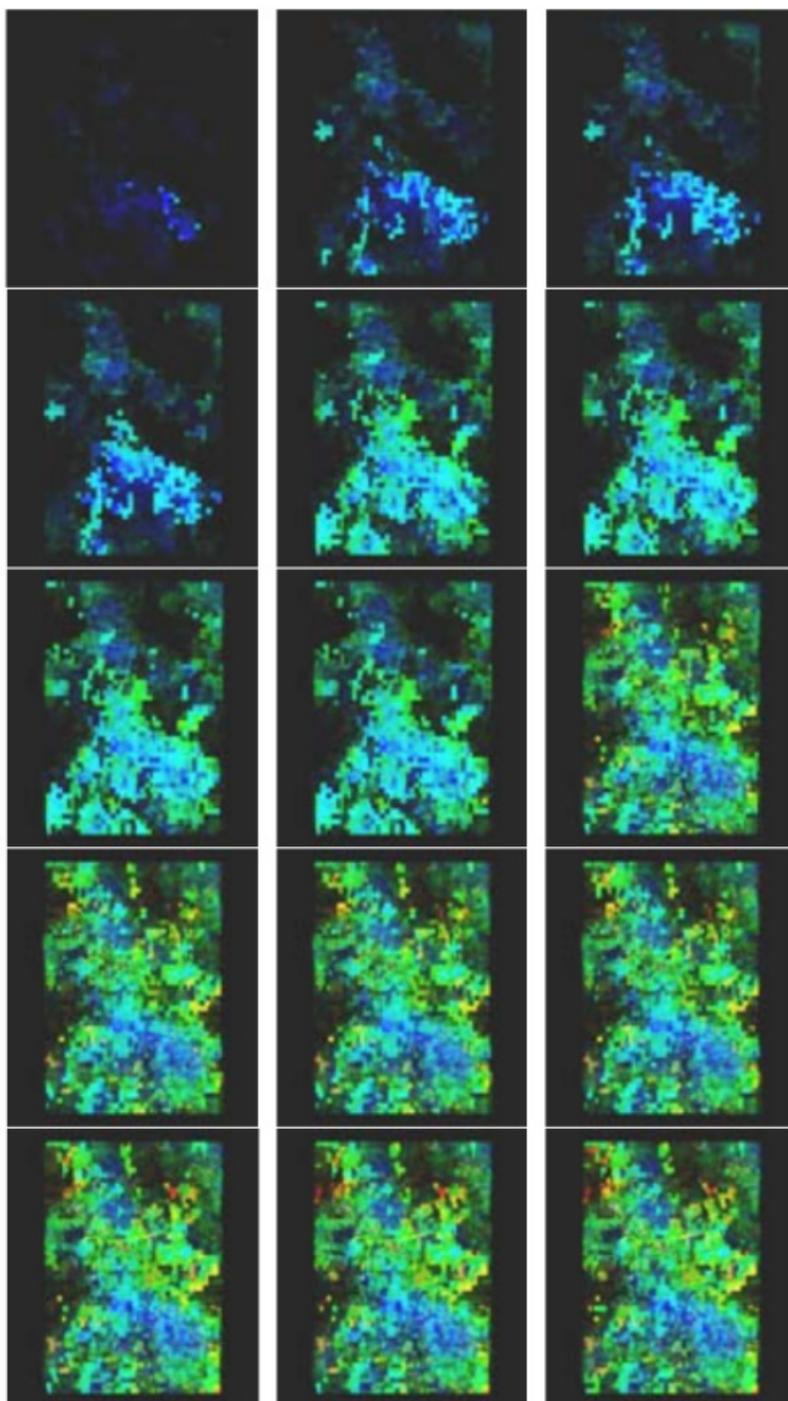


Figura 7.33: Sequência de imagens que representa a formação dos aglomerados -  
Renderização Volumétrica - Vista de topo

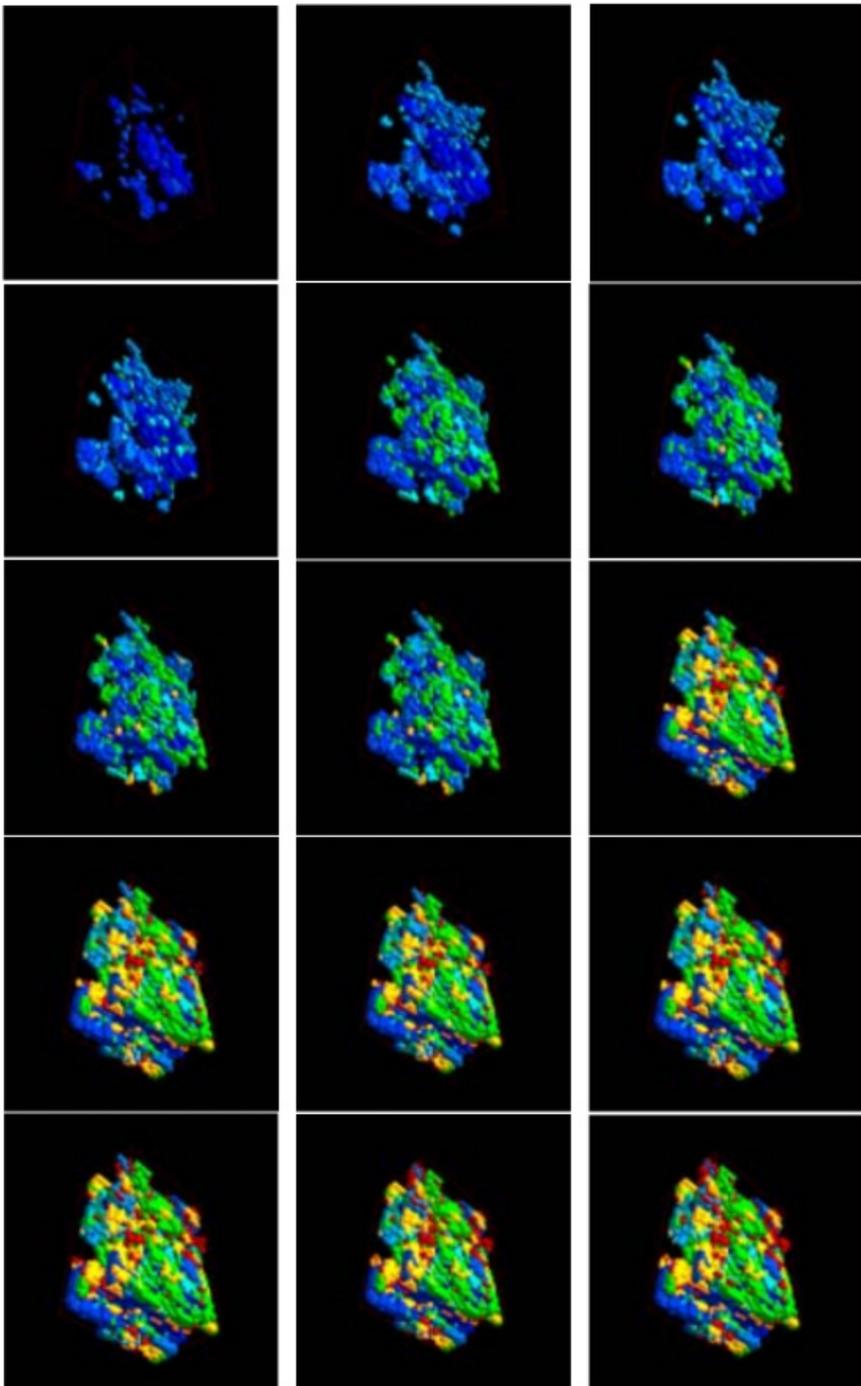


Figura 7.34: Sequência de imagens que representa a formação dos aglomerados - Renderização de superfície - Vista com Rotação de 45° em torno dos eixos x,y e z.

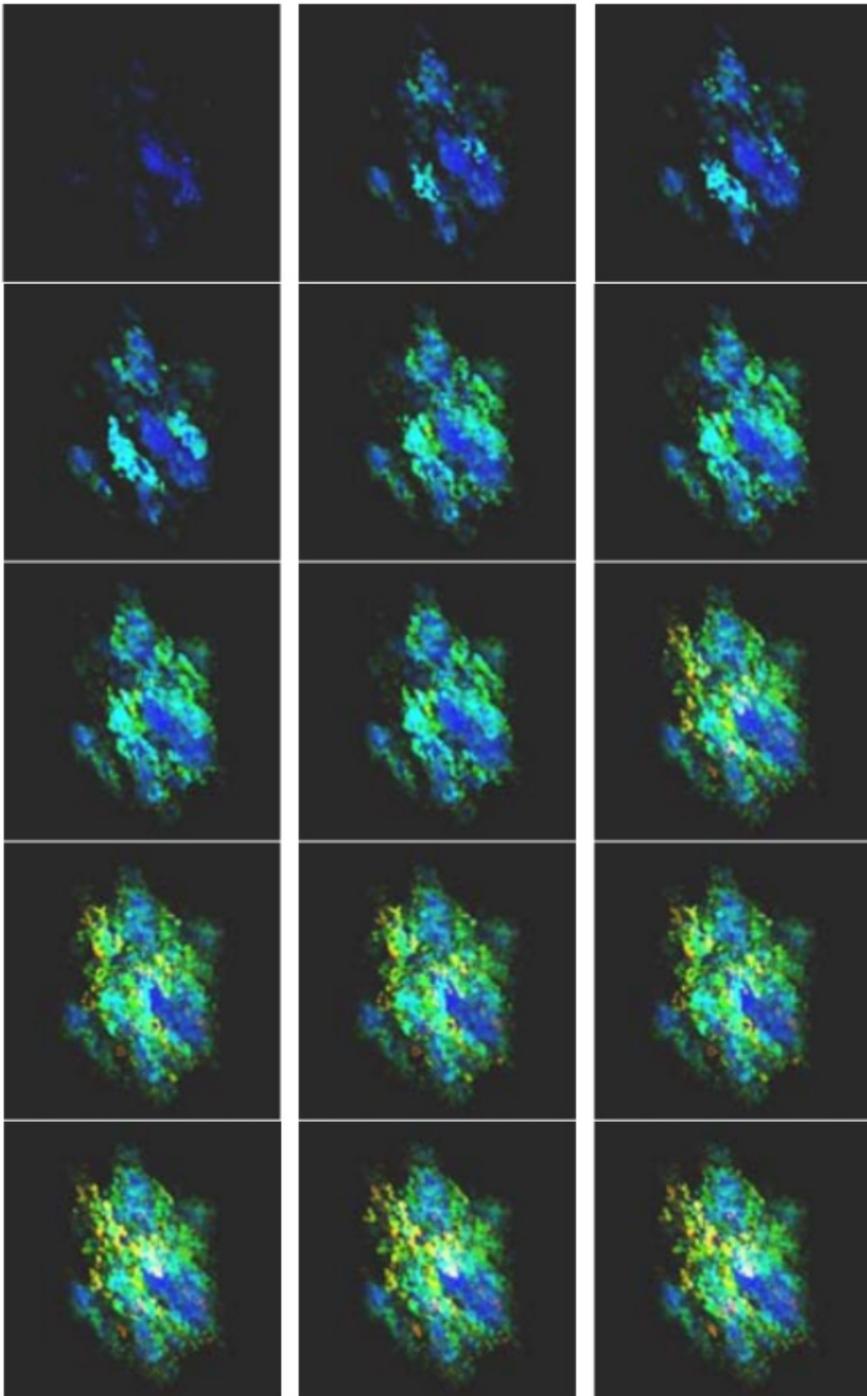


Figura 7.35: Sequência de imagens que representa a formação dos aglomerados - Renderização Volumétrica - Vista com Rotação de  $45^\circ$  em torno dos eixos x,y e z.

## CONCLUSÃO E PERSPECTIVAS

Neste trabalho, desenvolvemos uma metodologia inovadora para visualização de dados oriundos de simulações de percolação utilizada na exploração e recuperação do petróleo. Propomos uma ferramenta interativa, denominada ZSweepVR, baseada em técnicas de Renderização Volumétrica e Realidade Virtual. A renderização volumétrica permite preservar a integridade dos dados originais durante todo o processo de visualização, permitindo a captura de informações contidas no interior do volume. O uso de Realidade Virtual na Visualização científica aumenta potencialmente a habilidade de criar níveis de abstrações necessárias a manipulação de grandes volumes de dados. Isto se dá pelo fato de envolvermos com maior eficiência nossas capacidades sensoriais e intuitivas nestes ambientes.

Com o compromisso de manter o desempenho e simplicidade na geração de imagens que é inerente ao algoritmo de renderização Volumétrica ZSweep, inicialmente desenvolvemos e implementamos o sistema para geração de imagens em projeção perspectiva e para geração de pares estereoscópicos, contribuindo para o aperfeiçoamento dessa API. Isto possibilitou o desenvolvimento de uma ferramenta para uso em ambiente imersivo colaborativo.

O sistema ZsweepVR foi desenvolvido com dois módulos principais: um servidor de simulação e um servidor de visualização. O servidor de simulação gera os dados que são processados pelo servidor de Visualização que por sua vez renderiza as imagens, utilizando duas técnicas diferentes: Renderização de superfície e Renderização de volume.

O servidor de simulação é baseado no modelo de Percolação Dinâmica, implementada no ZSweepVR como um conjunto de classes que definem a rede, a conectividade entre sítios e a formação dos aglomerados. Propomos nesta fase mudanças no algoritmo original, com a finalidade de melhorar seu desempenho e adaptá-lo ao módulo de Visualização. A integração entre o módulo da simulação e o módulo de visualização é feita usando uma estrutura do tipo grade armazenada na memória principal. Esta grade une a saída da simulação a entrada Visualização volumétrica, evitando qualquer tipo de conversão de dados e de operações de E/S. Isto permite a geração de informações sobre a geometria e conectividade utilizadas pelo renderizador de superfície e o renderizador volumétrico.

O módulo de renderização de superfície permite com facilidade a navegação em tempo real no conjunto de dados Volumétricos, disponibilizando uma ferramenta

de exploração baseada em funções de transparência da API OpenGL, que permite observar a evolução da formação dos aglomerados sem perder a visão de todo o processo.

O módulo de renderização Volumétrica do ZSweepVR permite que qualquer parte dos dados seja vista, incluindo as estruturas e os detalhes internos. No caso particular da percolação dinâmica, a simulação permite que o usuário siga a evolução da formação dos aglomerados, e as interações entre eles. Estas interações ocorrem dentro do conjunto de dados volumétricos, sendo só possível sua visualização no módulo de renderização Volumétrica do ZSweepVR.

Além disto, disponibilizamos imagens renderizadas volumetricamente em estéreo, possibilitando a exploração dos dados volumétricos em ambiente imersivo colaborativo.

Testamos o sistema ZSweepVR utilizando uma base de dados volumétrica oriunda de campos de produção maduros, ou seja, que já deixaram de produzir espontaneamente. Obtivemos resultados satisfatórios relativos à exploração das heterogeneidades e distribuição das propriedades espalhadas pelo volume de dados simulado. O uso de imagens estéreo contribui substancialmente para a interpretação e análise do processo de formação dos aglomerados, conduzindo equipes multidisciplinares a formação de conceitos e abstrações que melhor conduzem a tomada de decisão segura sobre as atividades de exploração e a recuperação de petróleo.

ZSweepVR se estabelece então como uma ferramenta original para exploração de dados oriundos de simulação de percolação, sendo de grande interesse para a comunidade científica na investigação de fenômenos que envolvem a penetração de fluidos em meios porosos. Além de estimular a pesquisa na área de exploração e recuperação de petróleo, podemos associá-lo a diversas aplicações que podem ser tratadas pela teoria da percolação.

Diferentemente da abordagem utilizada em alguns trabalhos citados no Apêndice B, onde usamos o GoCAD para visualizar o processo de simulação da percolação, o sistema ZSweepVR é uma iniciativa ímpar na área de desenvolvimento de software dedicado ao uso desta tecnologia na exploração e recuperação de petróleo. Geramos, com isto, diferentes perspectivas de atuação em diferentes áreas de interesse científico e industrial, entre as quais podemos citar: Teoria da percolação, renderização Volumétrica e Realidade Virtual.

Com o uso da percolação dinâmica como base para o servidor de simulação, atingimos dentro deste estudo níveis de análise nunca antes alcançados. Devido ao fator dinâmico envolvido nesta aplicação de Percolação, podemos gerar muito mais cenários, possibilitando a investigação de eventos temporais e espaciais. Esta característica temporal preserva sua identidade com a percolação padrão em cada instante, permitindo assim estudar as flutuações de grandezas envolvidas no sistema. Como

trabalho futuro nesta área (Teoria da percolação), devemos desenvolver ferramentas gráficas de análise de propriedades que envolvem a teoria da percolação e que podem ser associadas com o objeto de estudo.

Na área de renderização volumétrica contribuimos para o aperfeiçoamento da API ZSweep, com a implementação de código para geração de projeções em perspectiva e de imagens estéreo. Porém, sabemos que a qualidade das imagens geradas podem ser melhoradas. Para isso, planejamos a título de trabalhos futuros estudar a Implementação de novas funções de transferência capazes de modificar este cenário. Outra contribuição importante que pode ser implementada, é o desenvolvimento de técnicas de Renderização Volumétrica que usem placas gráficas, com a finalidade de melhorar a desempenho de Visualização em renderização Volumétrica no sistema ZSweepVR.

A iniciativa de usar ambientes imersivos colaborativos na Visualização de base de dados Volumétricas oriundas de simulação de Percolação, aumenta substancialmente a capacidade de envolvimento de equipes multi e interdisciplinares no processo de avaliação e tomada de decisão. Além do uso de estereoscopia, seguramente podemos investir no desenvolvimento de novas técnicas de Realidade Virtual no sistema ZSweepVR. Entre elas, a Implementação de estereoscopia no módulo de renderização de superfície, e no desenvolvimento de interfaces de comunicação para ambientes imersivos, com o uso de periféricos especiais ( luvas, trackers, etc).

Ainda a título de melhorias está sendo planejada a Implementação do módulo de Visualização em ambientes paralelos, com uso de clusters de PCs. Isto pode ser facilmente implementado para o módulo de renderização Volumétrica, já que contamos com versões do ZSweep para ambientes paralelos.

# APÊNDICES

## APÊNDICE A

### Modelo de Iluminação em Renderização Volumétrica

Os modelos ópticos usados na renderização volumétrica são baseados em modelos físicos de interação entre a luz e a matéria. O modelo de iluminação utilizado pelo renderizador volumétrico ZSweep descrito abaixo é baseado em modelos físicos de absorção e emissão de luz [21]. A equação matemática que a descreve denomina-se “*integral de volume rendering*” (VRI).

Para que possamos compreender melhor a obtenção da expressão referente ao modelo óptico, observemos a Figura 8.1a. Consideremos a quantidade de energia absorvida e emitida por uma seção transversal cilíndrica de dados com volume igual a  $V = A \Delta s$ , onde  $A$  é área da base do cilindro e  $\Delta s$  a sua altura.

Simplificando, assumimos que as partículas que compõem o objeto são consideradas esferas, com raio  $r$ , resultando em uma área projetada  $S = \pi r^2$ . Ver Figura 8.1b. Sendo  $\rho$  a densidade de partículas no volume, então, existem  $N = \rho A \Delta s$  partículas na seção transversal. Para um pequeno  $\Delta s$ , a área total projetada pode ser aproximada por  $N \pi r^2 = \rho A \pi r^2 \Delta s$  ou  $\rho A \pi r^2 \Delta s$  por unidade de área da seção transversal.

Com o limite  $\Delta s \rightarrow 0$  podemos escrever a seguinte equação diferencial:

$$\frac{dI}{ds} = \rho(s) \pi r^2 I_e(s) - \rho(s) \pi r^2 I_a(s) \quad (8.1)$$

Os termos  $I_e(s)$  e  $I_a(s)$  são respectivamente as intensidades de luz absorvida e emitida por elemento diferencial da seção transversal. Usando os termos  $\tau(s) = \tau(s) \pi I_e(s)$  e  $E(s) = \rho(s) \pi r^2$  que respectivamente são chamados de coeficiente de absorção e termo de emissão, tem-se:

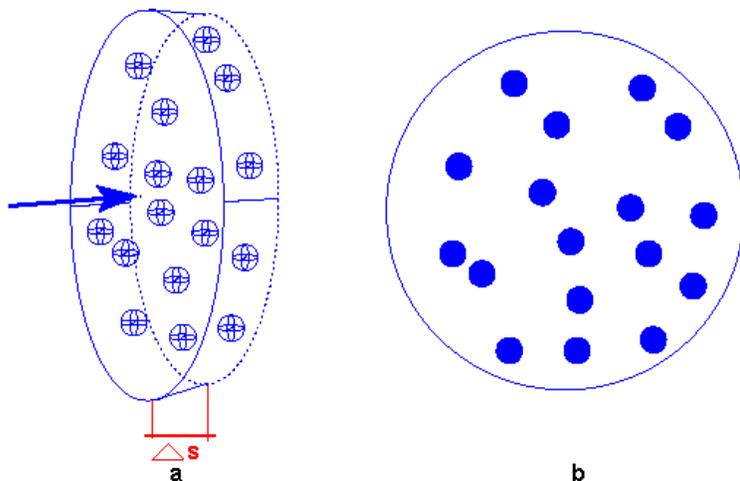


Figura 8.1: Modelo de Iluminação

$$\frac{dI}{ds} = E(s) - \tau(s)\pi I_a(s) \quad (8.2)$$

Esta equação pode ser resolvida passando o segundo termo da direita para a esquerda e multiplicando toda a equação pelo termo  $\exp(\int_0^s \tau(s) ds)$  e calcula a integral com relação a  $s$ , de 0 a  $z$ , em ambos os lados da equação. O resultado é o seguinte:

$$I = I_0 T(z) + \int_0^z E(z) dT(z) dz \quad (8.3)$$

O termo  $T(z)$  pode ser entendido como a transparência do objeto, a transparência e a opacidade são relacionados como segue  $T(z) = 1 - O(z)$ . Considerando a intensidade  $I$  como a intensidade em termos de cor absorvida ou emitida pelas partículas no objeto, podemos reescrever a equação acima como:

$$C(z) = C_0 + \int_0^z c(z)(1 - O(z)) dz \quad (8.4)$$

A opacidade  $O(z)$  é obtida de

$$O(z) = O_0 + \int_0^z o(z)dz \quad (8.5)$$

Isto pode ser aproximado como

$$O_n = O_c + \frac{1}{2}(o_c + o_n)\Delta z \quad (8.6)$$

Finalmente, obtemos a expressão analítica para a solução

$$C_n = C_c - \frac{1}{2}(c_c + c_n)(O_c - 1)\Delta z - \frac{1}{24}(3c_c o_c + 5c_n o_c + c_c o_n + 3c_n o_n)\Delta z^2 \quad (8.7)$$

Observar que os limites de integração foram mudados de  $(0, z)$  para os limites  $(z_c, z_n)$ , de forma a corresponder com a integração do *valor atual* de  $z$  ao *próximo valor* de  $z$ . Ver Figura 8.2.

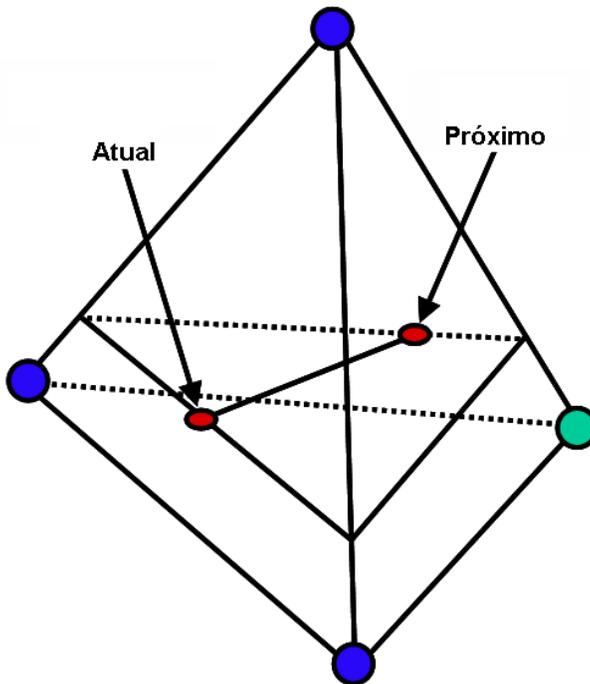


Figura 8.2: Limites de Integração para composição da cor e opacidade

## APÊNDICE B

### Trabalhos Publicados e Prêmio Petrobrás de Tecnologia

#### VisFluid: An Analysis Tool for Fluid Flow in Porous Media

In: VII Brazilian Symposium on Geoinformatics, 2005, São Paulo - SP

#### Visualização 3D utilizando o gOcad em Ambiente Imersivo Reality Center

In: II Workshop da Rede Cooperativa de Pesquisa em Risco Exploratório - 2005 - Belém - PA

#### Hiperpresence: An Application Environment for Control of Multi-User Agents in Mixed Reality Spaces.

In: Proceeding of 35th Simulation Symposium, 2003, Orlando, Florida, USA. Proceedings of 35th Simulation Symposium. Los Los Alamitos, CA: IEEE CS Press, 2003.

#### Caracterização de Reservatórios Utilizando Teoria de percolação: Modelagem e Visualização 3D

In: Revista Técnica de Energia Petróleo e Gás, 2002, Rio de Janeiro

#### Caracterização de Reservatórios Utilizando Teoria de percolação: Modelagem e Visualização 3D

In: 1o. Congresso Brasileiro de P & D em petróleo e gás, 2001, Natal. Anais do Congresso Brasileiro de P & D em petróleo e Gás.

#### Modelagem e Visualização 3D Aplicada a Reservatórios de Petróleo

In: ERMAC 2002, Segundo Encontro Regional de Matemática Aplicada e Computacional. Natal:2002.



## REFERÊNCIAS

- J. Adler and U. Lev. Bootstrap percolation: visualizations and applications. *Braz. J. Phys.*, 33(3), 2003.
- R. Avila, L. Hong T. He, A. Kaufman, H. Pfister, C. Silva, L. Sobierajski, and S. Wang. Volvis: A diversified volume visualization system. In *Proc. of IEEE Visualization'94*, pages 31–38, October 1994.
- M. Boyles and S. Fang. 3dive: an immersive environment for interactive volume data exploration. *J. Comput. Sci. Technol.*, 18(1):41–47, 2003.
- R. Brady, J. Pixton, G. Baxter, P. Moran, C. Potter, B. Carragher, and A. Bel-monts. Crumbs: A virtual environment tracking tool for biological imaging. In *IEEE Symposium on Frontiers in Biomedical Visualization*, pages 18–25, October 1995.
- CMG. Computer modelling group ltd. <http://www.cmgl.ca/>, 2005.
- SGI Corporation. Opengl tutorial. <http://www.opengl.org>, 2005.
- StereoGraphics Corporation. Stereographic developers handbook: Background on creating imagens for cristaleyes and simuleyes, 1997.
- J. E. de Freitas. Equivalence between the flr time dependent percolation model and the newman-ziff algorithm. *International Journal of Modern Physics*, 11:1581–1584, 2000.
- J. E. de Freitas. *Estudo de Alguns Sistemas Complexos: percolação dependente do tempo, aplicação a problemas de Petróleo, percolação de longo alcance e modelo de reação difusão*. PhD thesis, Universidade Federal do Rio Grande do Norte, 2002.
- J. E. de Freitas, L. S. Lucena, and S. Roux. Percolation as a dynamical phenomenon. *Physica A*, 1(266):81–85, 1999.
- G. A. Dorn, M. J. Cole, and K.M. Tubman. Predicting oil recovery using percolation. *The Leading Edge*, 1995.
- Eduardo Azevedo e Aura Conci. *Computação Gráfica - Teoria e Prática*. Editora Elsevier, 2003.
- Dorn et al. Immersive 3-d visualization applied to drilling planning. *The Leading Edge*, (20):1389–1392, 2001.
- R. Farias, J. Mitchell, and C. Silva. Zsweep: An efficient and exact projection algorithm for unstructured volume rendering. In *2000 Volume Visualization Symposium*, pages 91 – 99, October 2000.
- Ricardo Farias and Claudio T. Silva. Parallelizing the zsweep algorithm for distributed-shared memory architectures. In *VG01 - International Workshop on Volume Graphics 2001*. IEEE Press, 2001.
- Ricardo Farias and Claudio T. Silva. Out-of-core rendering of large, unstructured grids. *IEEE Comput. Graph. Appl.*, 21(4):42–50, 2001.
- James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics Principles and Practice*. AW, 1992.
- A. L. Fuhrmann, B. Ozer, L. Mroz, and H. Hauser. Vr2, interactive volume rendering using pc-based virtual reality. <http://www.vrvis.at/vr/voxelstube>, 2005.

D. S. George and A. R. Kavscek. Visualization of solution gas drive in viscous oil. Technical report, 2001. Stanford University, CA, USA.

Christopher Giertsen. Volume visualization of sparse irregular meshes. *IEEE Comput. Graph. Appl.*, 12(2):40–48, 1992.

Andrew S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, 1995.

H. Gould and J. Tobochnik. *An Introduction to Computer Simulation Methods*. Addison-Wesley, 1996.

G. S. Grest, I. Webman, S. A. Safran, and A. L. R. Bug. Dynamic percolation in microemulsions. *Phys. Rev. A*, 4(33):2842–2845, 1986.

U. Gudukbay and T. Yilmaz. Stereoscopic view-dependent visualization of terrain height fields. *IEEE Transactions on Visualization and Computer Graphics*, 8(4):330–345, 2002.

CAVE5D User's Guide. [www.ccpo.odu.edu/cave5d/homepage.html](http://www.ccpo.odu.edu/cave5d/homepage.html), 1998.

Donald Hearn and M. Pauline Baker. *Computer graphics*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986.

W. Hibbard and D. Santek. The Vis5D system for easy interactive visualization. In *Proc. of the 2nd IEEE Visualization Conference (Vis1990)*, pages 28 – 35, 1990.

L. F. Hodges. Basic principles of stereographic software development. *Stereoscopic and Applications II, Proc SPIE 1457*, 12(2):9–17, 1991.

L. F. Hodges. Tutorial: Time-multiplexed stereoscopic computer graphics. *IEEE Comput. Graph. Appl.*, 12(2):20–30, 1992.

J. Hoshen and R. Kopelman. *Phys. Rev. B*, 14:3438, 1976.

R. Hudson and A. Malagoli. Networked virtual reality for real-time 4d navigation of astrophysical turbulence data simulation. In *Multiconference of the Society for Computer Simulation*, 1996.

IBM. Open visualization data explorer - opendx, 1999.

Axel jacobs and John Solomon. *Howto3d*, 2004.

G. R. Jerauld and S. J. Latter. The effect of pore-structure on hysteresis in relative permeability and capillary pressure: pore-level modelling. *Transp. Porous Media*, 1(5):103–151, 1990.

Arie Kaufman. *Volume Visualization - Tutorial*. IEEE Computer Society Press, 1991.

P. R. King, J. S. Andrade, S. V. Buldyrev, N. V. Dokholyan, Y. Lee, S. Havlin, and H. E. Stanley. Predicting oil recovery using percolation. *Physica A*, (266):107–114, 1999.

C. Lascara, G. Wheless, D. Cox, R. Patterson, S. Levy, A. E. Johnson, and J. Leigh. Tele Immersive virtual environments for collaborative knowledge discovery. In *Proceedings of Advanced Simulation Technologies Conference*, pages 11–15, April 1999.

- P. L. Leath. Cluster size and boundary distribution near percolation threshold. *Phys. Rev. B*, 1(14):5046–5055, 1976.
- C. M. Lima. Desenvolvimento de conteúdo para o ensino da tecnologia do plasma utilizando técnicas de multimídia e realidade virtual. Master 's thesis, Universidade Federal do Rio Grande do Norte, 2001.
- Carlos Magno Lima, Silvio José Bezerra, and Joaquim Elias Freitas. Caracterização de reservatórios utilizando a teoria da percolação: Modelagem e Visualização 3d. *Petróleo e Gás*, 1:86, 2002.
- Carlos Magno Lima and Ricardo Farias Luiz M. G. Gonçalves, Cristiana Bentes. Visfluid: An analysis toll for fluid flow in porous media. *VII Brazilian Symposium on Geoinformatics*, 2005.
- N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99 – 108, June 1995.
- Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2), 1995.
- A. McPherson and M. Maltrud. Poptex: Interactive ocean model visualization using texture mapping hardware. In *Proc. of the 9th IEEE Visualization Conference (Vis1998)*, pages 471 – 474, October 1998.
- L. Mroz and H. Hauser. Rtvr - a flexible java library for interactive volume rendering. In *Proceedings IEEE Visualization 2001*, pages 279–286, 2001.
- M. Newman and R. M. Ziff. *Phys. Rev. Lett.*, 85:4104, 2000.
- D. Or and M. Tuller. Hydraulic conductivity of partially saturated fractured porous media: flow in a cross-section. *Advances in Water Resources*, 1(26):883–898, 2003.
- S. P. Friedman P and N.A. Seaton. Critical path analysis of the relation-ship between permeability and electrical conductivity of three dimensional pore networks. *Water Resour. Res.*, 7(34):1703–1710, 1998.
- P. Peggy, S. Whitman, R. Mendonza, and J. Tsiao. ParVox – A parallel splatting volume rendering system for distributed visualization. In *James Painter, Gordon Stoll and Kwan-Liu Ma, editors, IEEE Parallel Rendering Symposium*, pages 7 – 14, November 1997.
- Andreas Petersik. Adreas petersik's stereoscopic applet, 2005.
- H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The volume pro real-time ray-casting system. In *SIGGRAPH'99*, August 1999.
- F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- S. Rottger, M. Kraus, and T. Ertl. Hardware-accelerated volume and isosurface rendering based on cell-projection. In *VIS '00: Proceedings of the conference on Visualization '00*, pages 109–116, 2000.
- N. S. Sagar and L. M. D. Castanier. Oil-foam interactions in a micromodel. Technical report, 1997. Dept. of Petroleum Engineering, Stanford University, CA, USA.
- M. Sahimi. *Applications of Percolation and Theory*. London: Taylor & Francis, 1994.

J. P. Schulze, U. Wossner, S. P. Walz, and U. Lang. Volume rendering in a virtual environment. In *Proc. of 5th IPTW and Eurographics Virtual Environments*, pages 187–198, 2001.

Earth Decision Sciences. Gocad. <http://www.arthdecision.com>, 2000.

William R. Shermann and Allan B. Craig. *Understand Virtual Reality: Interface, Application and Design*. Morgan Kaufmann Publishers, 2003.

C. T. Silva, J. S. B. Mitchell, and P. L. Williams. An exact interactive time visibility ordering algorithm for polyhedral cell complexes. In *VVS '98: Proceedings of the 1998 IEEE symposium on Volume visualization*, pages 87–94, 1998.

Claudio T. Silva and Joseph S. B. Mitchell. The lazy sweep ray casting algorithm for rendering irregular grids. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):142–157, 1997.

Muttan's Stereographic Software. Stereo photo maker, 2005.

D. Stauffer and A. Aharony. *Introduction to Percolation Theory*. London: Taylor & Francis, 1992.

Kirk Watson, Bharti Temkin, Wm. LeRoy Heinrichs, Thomas M. Krummel, and Paul J. Gorman. Development of stereoscopic-haptic virtual environments. In *CBMS*, 1999.

D. Wilkinson and J. F. Willemsen. Invasion percolation: a new form of percolation theory. *J. Phys. A*, 1:3365–3376, 1983.

B. Wylie, K. Moreland, L. A. Fisk, and P. Crossno. Tetrahedral projection using vertex shaders. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 7–12, 2002.

Roni Yagel. *Efficient methods for volumetric graphics*. PhD thesis, 1991. Adviser-Arie E. Kaufman.

Yei-Yu Yeh and Louis D. Silverstein. Limits of fusion and depth judgment in stereoscopic color displays. *Hum. Factors*, 32(1):45–60, 1990.

## **SOBRE O AUTOR**

**CARLOS MAGNO DE LIMA** - Graduado em Engenharia Mecânica pela Universidade Federal do Rio Grande do Norte (1985), com ampla experiência profissional na Indústria Química e Petroquímica nas áreas de Inspeção de Equipamentos e Manutenção Industrial. Possui Mestrado em Engenharia Mecânica pela Universidade Federal do Rio Grande do Norte (Tecnologia CAD/CAM, Multimídia e Realidade Virtual aplicada ao Ensino à Distância - Ano 2001) e Doutorado em Engenharia Elétrica pela Universidade Federal do Rio Grande do Norte (Renderização Volumétrica e Realidade Virtual aplicada a Indústria do Petróleo, Gás e Energia - Ano 2005). Atualmente é Professor Associado II da Universidade Federal do Rio Grande do Norte no Departamento de Engenharia Mecânica. Tem experiência na área de Engenharia Mecânica e Ciências da Computação, com ênfase em Projetos Mecânicos e Realidade Virtual, atuando principalmente nos seguintes temas: Tecnologia CAx, Automação de Projetos, BIM, Realidade Virtual, Computação Gráfica, Modelagem e Visualização 3d.

Renderização Volumétrica e Realidade Virtual Aplicada a

# PROBLEMAS DE PERCOLAÇÃO NA INDÚSTRIA DO PETRÓLEO

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

[@atenaeditora](https://www.instagram.com/atenaeditora) 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

 **Atena**  
Editora

Ano 2021

Renderização Volumétrica e Realidade Virtual Aplicada a

# PROBLEMAS DE PERCOLAÇÃO NA INDÚSTRIA DO PETRÓLEO

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

[@atenaeditora](https://www.instagram.com/atenaeditora) 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

 **Atena**  
Editora

Ano 2021