



Universidade Federal da Bahia  
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

**UMA PROPOSTA DE SIMULAÇÃO  
COMPUTACIONAL 3D PARA PINTURA  
HIDROGRÁFICA**

Leonardo Thomas Torres Santos

QUALIFICAÇÃO DE MESTRADO

Salvador  
10 de julho de 2018



LEONARDO THOMAS TORRES SANTOS

**UMA PROPOSTA DE SIMULAÇÃO COMPUTACIONAL 3D PARA  
PINTURA HIDROGRÁFICA**

Esta Qualificação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Karl Philips Apaza Agüero

Salvador  
10 de julho de 2018



## RESUMO

A Pintura Hidrográfica, *Water Transfer Printing* ou *Hydrographics* é uma técnica viável para colorir objetos criados em impressoras 3D. Porém, neste tipo de pintura há uma complexa interação entre um filme e um objeto impresso em 3D, na qual a imagem contida no filme deve aderir ao objeto. Este processo está sujeito a incertezas sobre como o filme se deformará para aderir ao objeto impresso em 3D e transferir a imagem para a superfície do objeto, conferindo-lhe um acabamento resultado de um mapeamento de uma imagem 2D em um modelo real 3D.

Diante destas dificuldades, será apresentada uma proposta de simulação computacional 3D para a técnica de pintura hidrográfica. Com base na interação física entre os corpos envolvidos neste fenômeno, será proposta uma simulação computacional 3D que mostre o que ocorre durante a pintura hidrográfica. Desta forma, será dado ao usuário um suporte ao usuário para antever o processo da pintura em formas mais complexas, que necessitem de alinhamentos com a imagem transferida ao objeto. Além disso, o trabalho também busca simular o fenômeno físico da pintura hidrográfica em tempo real, fazendo uma avaliação da utilização dos recursos computacionais da GPU (*Graphics Processing Unit*) empregados nesta simulação.

**Palavras-chave:** Impressão 3D, pintura hidrográfica, simulação baseada em física, renderização em tempo real, mapeamento de texturas



## ABSTRACT

Hydrographic printing, Water Transfer Printing or Hydrographics is a viable technique for coloring objects created with 3D printers. However, in this type of painting there is a complex interaction between a film and a 3D printed object, in which the image contained in the film must adhere over the object. This process is subjected to several uncertainties about how the film will deform to adhere over the 3D object and transfer the image to the surface of the object, giving it a finish resulting from a mapping of a 2D image into a real 3D model.

Faced with these difficulties, a 3D computational simulation proposal will be presented for the hydrographic printing technique. Based on the physical interaction between the bodies involved in this phenomenon, it will be proposed a 3D computer simulation that shows what happens during hydrographic printing. In this way, the user will be given a support to preview the painting process in more complex shapes, which require alignments with the image transferred to the object. Furthermore, the work also aims to simulate the physical phenomenon of hydrographic printing in real time, making an evaluation of the use of the computational resources of the *Graphics Processing Unit* (GPU) used in this simulation.

**Keywords:** 3D printing, hydrographic printing, physically-based animation, real-time rendering, texture mapping





# SUMÁRIO

<b>Capítulo 1—Introdução</b>	1
1.1 Contexto . . . . .	1
1.2 Problema . . . . .	2
1.2.1 Pintura Hidrográfica Computacional . . . . .	3
1.3 Hipóteses e objetivos da pesquisa . . . . .	4
<b>Capítulo 2—Fundamentação</b>	5
2.1 Simulação baseada em física . . . . .	5
2.1.1 Integração numérica de Euler . . . . .	6
2.1.2 Simulação de corpos deformáveis . . . . .	7
2.2 PBD ( <i>Position-based Dynamics</i> ) . . . . .	7
2.3 OpenGL . . . . .	10
2.3.1 Características . . . . .	10
2.3.2 Programação na GPU com <i>shaders</i> . . . . .	10
2.4 Mapeamento de Texturas . . . . .	11
<b>Capítulo 3—Definição da proposta</b>	13
3.1 Proposta . . . . .	13
3.1.1 Realização do experimento físico . . . . .	13
3.1.2 Implementação da simulação computacional . . . . .	14
3.1.2.1 Resultados iniciais . . . . .	17
3.1.3 Elaboração da dissertação e artigos . . . . .	18
3.2 Cronograma . . . . .	18
<b>Capítulo 4—Conclusões</b>	21
4.1 Conclusões preliminares . . . . .	21
<b>Apêndice A—Detalhamento do Experimento Preliminar</b>	25
A.1 Material utilizado . . . . .	25
A.2 Etapas do experimento . . . . .	25



## LISTA DE FIGURAS

2.1	Representação dos passos desde a simulação da pintura hidrográfica até a pintura física do objeto. <b>(a)</b> Textura (à esquerda) e objeto (à direita). <b>(b)</b> Simulação. <b>(c)</b> Filme gerado com base na simulação. <b>(e)</b> Pintura hidrográfica física. <b>(f)</b> Resultado final do objeto 3D colorido com pintura hidrográfica. (ZHANG et al., 2015). . . . .	6
2.2	Modelagem de um corpo deformável como um sistema de massas e molas. (GAO; GIBSON, 2006) . . . . .	7
2.3	Algoritmo <i>Position-based Dynamics</i> (PBD) (MÜLLER et al., 2007) . . .	8
2.4	Diagrama representativo dos estágios do <i>Open Graphics Library</i> (OpenGL) (SHREINER et al., 2013b) . . . . .	11
3.1	Experimento preliminar. . . . .	14
3.2	Representação das etapas do algoritmo implementado. . . . .	15
3.3	Representação das restrições de distância como sendo as linhas que ligam os vértices da malha 2D. As cores representam as partições do conjunto de vértices, que deve ser processado sequencialmente pelo <i>compute shader</i> . .	16
3.4	<b>(a)</b> Início da simulação. <b>(b)</b> Esfera tocando a película. <b>(c)</b> O objeto continua a se movimentar esticando a película. <b>(d)</b> A película segue o movimento aderindo ao objeto. . . . .	18



## **LISTA DE TABELAS**

3.1	Cronograma das atividades do mestrado. . . . .	19
-----	--	----



## LISTA DE SIGLAS

<b>API</b>	<i>Application Programming Interface</i>	
<b>CPU</b>	<i>Central Processing Unit</i>	
<b>FPS</b>	Frames por Segundo	
<b>GPU</b>	<i>Graphics Processing Unit</i> .....	v
<b>MAE</b>	Museu de Arqueologia e Etnologia da UFBA	
<b>MEF</b>	Método dos Elementos Finitos	
<b>OpenGL</b>	<i>Open Graphics Library</i> .....	11
<b>PBD</b>	<i>Position-based Dynamics</i> .....	8





## Capítulo

# 1

*Neste capítulo, será dada uma contextualização sobre o trabalho, a apresentação do problema, hipóteses de pesquisa e principais objetivos.*

## INTRODUÇÃO

### 1.1 CONTEXTO

O processo de fabricação com a impressão 3D vem se desenvolvendo rapidamente nos últimos anos, despertando o interesse de muitas áreas. Porém, apesar do desenvolvimento crescente na produção de formas complexas, criar objetos coloridos ainda é um desafio. Isto ocorre porque a maioria das impressoras 3D disponíveis geram objetos com somente uma cor. Há alguns dispositivos *high-end* que geram a forma e incorporam a coloração durante o processo, tornando a geração do objeto indissociável da pigmentação. Muitas vezes, esta opção é a menos desejável, pois restringe a escolha da impressora 3D, tirando-se a liberdade de se escolher um equipamento com o menor custo possível. Dentre as outras opções para se colorir um material em 3D, destacam-se as técnicas de pintura tradicional e as técnicas que utilizam decalques.

A pintura tradicional pode ser realizada com o aerógrafo (através de vaporização da tinta), rolo ou pincel. Todas estas técnicas requerem um bom domínio operacional e um bom conhecimento dos processos e materiais envolvidos (tipo de tinta adequado para o tipo de substrato, diluentes, preparação da superfície, aplicação de camadas, tempo de cura e acabamento). Parte do material usado na pintura é desperdiçado. Isto ocorre por aplicação da tinta em partes extras (provocando sujeira), por conta de dimensionamento inadequado do material, ou mesmo pelo desperdício do usuário. É um processo que depende bastante de se ter pessoas qualificadas, aumentando-se o custo de mão de obra. Técnicas tradicionais de pintura geram um resultado diferente para cada peça, dificultando a previsibilidade do trabalho. A natureza da técnica faz com que cada peça pintada à mão resulte em um trabalho artístico único.

A Pintura Hidrográfica é conhecida também pelos nomes *Water Transfer Printing*, *Hydrographic Printing* ou simplesmente *Hydrographics*. É considerada como uma das técnicas para coloração de formas 3D a partir de decalques. Pode ser aplicada em diversos tipos de materiais como cerâmica, fibra de vidro, madeira, plástico, metal. O objeto

trabalhado, portanto, pode ter sido gerado em qualquer equipamento. Nesta técnica, o desenho (ou padrão) é impresso em um filme especial de material polivinílico. Este filme é posto a flutuar em um tanque contendo água e é submetido a uma reação ativadora. O objeto é submergido lentamente no tanque, de forma que vai encostando na superfície da água, que contém a película. A transferência do padrão impresso ocorrerá a através do contato do filme com o substrato. O processo de pintura hidrográfica possui alguns passos bem definidos que uma vez seguidos, produzem um acabamento de qualidade para objetos com formas não tão complexas (como no caso de peças automotivas). Existem diversos tipos de filmes com estampas, cujos desenhos podem variar desde uma cor plana até padronagens complexas, como texturas que imitam madeira, rocha, camuflagem, pele, couro e uma grande variedade de estampas que imitam texturas do mundo real. Existem também filmes totalmente transparentes, comercializados no tamanho de um papel A4. Desta maneira, podem receber qualquer desenho impresso por uma impressora de pigmentos.

Outra técnica de pintura para objetos impressos em 3D é a pintura por Termoformagem. Apesar de ser uma técnica distinta da pintura hidrográfica, a termoformagem trata-se de um processo que possui algumas semelhanças com a pintura hidrográfica: ambos lidam com uma película na qual se dará a transferência de uma imagem para um objeto 3D.

No caso da termoformagem, uma lâmina plástica é aquecida em um dispositivo similar a uma estufa. A lâmina passa então a adquirir propriedades plásticas e elásticas. Após esta etapa, a lâmina plástica entrará em contato com o objeto pintado. Será então submetida a um processo de sucção, formando um vácuo no qual a pressão exercida terminará fazendo com que haja a adesão da lâmina plástica com o objeto. Neste caso, a termoformagem lida com outras questões, como por exemplo resolver as regiões da peça em que podem vir formar bolhas de ar.

## 1.2 PROBLEMA

Na pintura hidrográfica usual, frequentemente se transfere um padrão que é uma grafia repetitiva imitando algum tipo de material (fibra de carbono, camuflagem, madeira). Nestes casos, não há necessidade de precisão de alinhamento entre o padrão de cor contido no filme e o objeto que está sendo colorido. No entanto, caso o objetivo da pintura hidrográfica seja dar uma coloração específica para pontos específicos de uma peça impressa em 3D, seria necessário um mapeamento preciso das cores da imagem 2D, contida no filme, para os pontos da superfície do objeto 3D. Realizar esta operação com a pintura hidrográfica usual é uma tarefa difícil, pois não há um controle de como este mapeamento se realizará.

No contexto geral, a fabricação de objetos 3D coloridos é uma tarefa difícil que vem sendo pesquisada nos últimos anos. Os trabalhos (ZHANG et al., 2015), (PANOZZO et al., 2015) apresentaram contribuições importantes neste campo de estudo usando a pintura hidrográfica. Mesmo assim, devido às diversas dificuldades encontradas, ainda é um problema em aberto, com amplo espaço para desenvolvimento e surgimento de novas propostas.

### 1.2.1 Pintura Hidrográfica Computacional

No trabalho de (ZHANG et al., 2015), é apresentada uma abordagem de simulação computacional para hidropintura, que possibilita prever a distorção do filme e da cor durante o processo de pintura hidrográfica. O trabalho também propõe um método para automatizar parte do processo físico da pintura, através de um equipamento físico que controla a imersão do objeto na água. Ainda neste trabalho, é proposta uma solução de múltiplas imersões para dar um tratamento adequado que cubra completamente as formas mais complexas.

Outro trabalho, (PANOZZO et al., 2015) apresenta contribuições para o mesmo propósito. Neste trabalho é feito um estudo detalhado das condições em que se realiza a pintura hidrográfica. Com base neste levantamento descritivo, é proposto um conjunto de ferramentas físicas e computacionais que possibilite o usuário a obter resultados mais precisos e consistentes na realização da pintura hidrográfica. A solução algorítmica busca dar um tratamento ao problema de forma distinta do primeiro trabalho. É descrito um algoritmo no qual se pode simular o processo da pintura e a distorção sofrida pelo filme ao aderir na superfície de um corpo arbitrário.

Em comum, estes trabalhos propõem uma modelagem discreta do problema. Representam a película como sendo uma malha de triângulos 2D. Fazem uso do MEF (Método dos Elementos Finitos) (BATHE, 2007) em alguma etapa do processo para determinar os valores das velocidades em (ZHANG et al., 2015), ou para cálculo das forças no sistema elástico em (PANOZZO et al., 2015), quando em processo dinâmico de imersão do corpo. Em seguida, propõem uma simulação computacional para encontrar a função de mapeamento dos pontos do objeto 3D para a textura em 2D.

Os trabalhos citados deixam em aberto alguns problemas como:

- Dificuldade de controlar o processo quando o ângulo entre a superfície do corpo e a linha d'água é maior do que  $45^\circ$
- Dificuldade de determinar a distorção da cor
- Necessidade de sugerir uma posição de imersão que proporcione um melhor aproveitamento do filme, com uma melhor cobertura
- Necessidade de garantir a reprodução dos resultados
- Nem sempre se consegue trabalhar corretamente com objetos que possuem grandes concavidades ou auto-oclusões (mesmo com múltiplas imersões)
- Dificuldade em estimar a variação da espessura do filme, que se mostrou um processo não linear

Quanto à implementação da simulação computacional, em (ZHANG et al., 2015) não foi realizada uma simulação 3D na GPU. Optou-se então por realizar a simulação 3D na CPU, já que a simulação ficou suficientemente simples ao levar o problema para o domínio 2D (o cálculo de um campo de velocidades 2D para uma lâmina d'água). Assumiu-se que a taxa de atualização da animação ficou suficiente para as necessidades do usuário.

Porém, os exemplos usados levaram até 5 minutos para simular uma única imersão. Isto pode ser um tempo alto quando se deseja uma maior dinâmica na prototipação de um produto ou a implementação de uma simulação 3D em tempo real. Já em (PANOZZO et al., 2015), não é mencionado se houve a simulação 3D em GPU, ficando o trabalho omissso neste aspecto.

### 1.3 HIPÓTESES E OBJETIVOS DA PESQUISA

Diante do exposto, com base na exposição do problema e no conhecimento da possível combinação de técnicas da computação gráfica, a principal pergunta do estudo foi assim formulada:

É possível realizar uma simulação computacional eficiente que reproduza o fenômeno da pintura hidrográfica?

Segue abaixo uma relação de objetivos específicos do trabalho:

- Levantar detalhes de como ocorre processo físico da pintura hidrográfica
- Proporcionar uma ferramenta com base em simulação computacional que possibilite incorporar melhorias no processo da pintura. Entende-se que melhorias seriam quaisquer meios que ajudem a: evitar perda de material, dar uma previsibilidade do resultado, fazer mapeamento da imagem colorida do filme para o objeto imerso.
- Trabalhar os problemas relativos a distorções na cor
- Descobrir formas de se minimizar os problemas inerentes da técnica de pintura hidrográfica: minimização do esticamento do filme, maximizar o acesso a concavidades do objeto
- Investigar formas de executar a simulação eficiente da pintura hidrográfica em tempo real usando a GPU
- Comparar a técnica de simulação com a pintura real
- Comparar a técnica escolhida com outras técnicas de simulação existentes
- Disponibilizar a ferramenta implementada para prestar um suporte aos processos de impressão 3D colorida de objetos do MAE (Museu de Arqueologia e Etnologia da UFBA)

*Serão apresentadas as principais referências teóricas que reunidas constituem uma base para a elaboração do trabalho.*

## FUNDAMENTAÇÃO

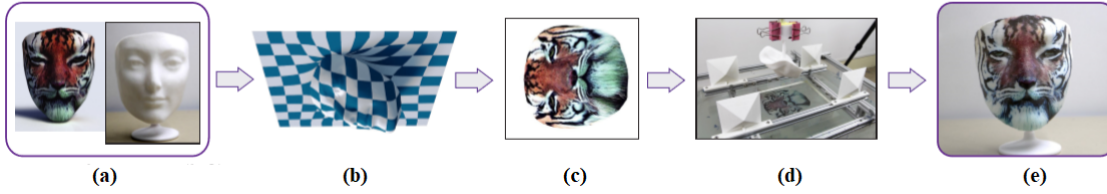
### 2.1 SIMULAÇÃO BASEADA EM FÍSICA

O processo da impressão hidrográfica é um fenômeno físico que envolve a interação entre dois objetos principais: um deles é a superfície 3D que está sendo pintada, o outro é o filme 2D que carrega a imagem impressa. O filme permanece flutuando na superfície da água, formando um composto imiscível. Comporta-se como uma lâmina viscosa que ao tocar na superfície do objeto, adere quase que imediatamente à superfície deste objeto, acompanhando o movimento de imersão. Ao fim da operação, os pontos da imagem impressa no filme ocuparão uma determinada posição no objeto.

A simulação baseada em física é a área de estudo na computação gráfica que busca, através do suporte da física, gerar simulações realistas de fenômenos naturais. Os principais avanços nesta área foram motivados pela necessidade de uma representação virtual consistente das interações complexas entre corpos. Estes estudos possibilitaram incrementar os efeitos especiais na indústria cinematográfica, assim como na indústria de entretenimento com jogos (REEVES, 1983) (MILLINGTON, 2007).

No design de produtos e sobretudo na engenharia, a simulação com computação gráfica tem sido particularmente útil no sentido de prever o comportamento de sistemas físicos complexos, sem a necessidade de um experimento real, economizando-se o tempo, a mão de obra e o material que seria empenhado. A figura 2.1 ilustra esta idéia, representando as etapas que vão desde a simulação da pintura hidrográfica até a realização da pintura física do objeto.

Simulações podem ser *offline*, quando o usuário não interage com os elementos da cena. Isto ocorre na indústria cinematográfica, pois os quadros da animação são produzidos anteriormente, durante a criação do filme. No caso de jogos, ou de aplicações que demandem uma interação com o usuário, será necessário realizar a simulação em tempo real. Para os padrões modernos, a apresentação de uma animação exibida em taxas acima de 25 FPS (Frames por Segundo) já pode ser considerada em tempo real.



**Figura 2.1** Representação dos passos desde a simulação da pintura hidrogáfica até a pintura física do objeto. (a) Textura (à esquerda) e objeto (à direita). (b) Simulação. (c) Filme gerado com base na simulação. (d) Pintura hidrogáfica física. (e) Resultado final do objeto 3D colorido com pintura hidrogáfica. (ZHANG et al., 2015).

O problema da simulação baseada em física traz encadeado o problema da detecção de colisão como uma etapa importante. Neste ponto, existem técnicas documentadas que dão o tratamento adequado para detectar a colisão entre os corpos, impondo ao sistema restrições importantes quanto a movimentação destes corpos. (MOORE; WILHELMS, 1988) (JIMÉNEZ; THOMAS; TORRAS, 2001)

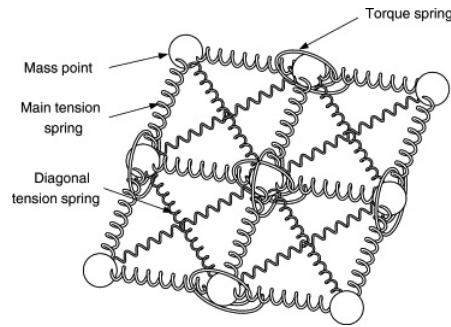
Dentre os estudos desenvolvidos, simulações baseadas em física podem ser classificadas em diversas categorias, cada uma com características próprias e um conjunto de problemas técnicos a serem resolvidos. Destacam-se então: a simulação de corpos rígidos, simulação de corpos deformáveis, simulação de partículas, simulação de multidões e a simulação de fluidos.

Para o desenvolvimento deste projeto, após uma análise do que ocorre no experimento físico conforme A.2, foi considerado que a modelagem simulação levaria principalmente à condição de um fenômeno de interação entre um corpo deformável (o filme com a película hidrogáfica) e um corpo rígido movimentado com velocidade constante (o objeto que será pintado). Apesar de existir a interação do objeto e do filme com um fluido (a água), observou-se que idealmente o sistema formado não deve sofrer interferência significativa da água, pois a película permanece na superfície sem se misturar com a água.

### 2.1.1 Integração numérica de Euler

Trata-se do método mais básico de solucionar numericamente equações diferenciais. A proposta de Euler é resolver as equações referentes ao fenômeno estudado com base em integração numérica, atribuindo valores às variáveis e discretizando os valores da função em intervalos conhecidos. Em (BARAFF; STUDIOS, 1999) pode ser encontrado um estudo prático no qual são apresentadas aplicações do método de Euler em simulações computacionais.

Por tratar-se de uma discretização de valores contínuos, no método de Euler será necessário determinar arbitrariamente o valor do intervalo utilizado, de forma que produza valores consistentes com o que se deseja implementar. Um intervalo muito grande, produzirá erros maiores. Um intervalo muito pequeno, produzirá aproximações indesejáveis. Para os fenômenos físicos estudados na computação gráfica, o objetivo final será o cálculo da posição dos objetos na cena em função do tempo.



**Figura 2.2** Modelagem de um corpo deformável como um sistema de massas e molas. (GAO; GIBSON, 2006)

### 2.1.2 Simulação de corpos deformáveis

Corpos deformáveis, diferentemente dos corpos rígidos, quando submetidos a forças externas, podem variar a sua forma em uma simulação. A simulação de corpos deformáveis pode representar visualmente o processo que ocorre em uma variedade de materiais como: borracha, plástico, cabelo, músculos, tecidos. Usualmente, neste tipo de simulação, a representação do material se dá através de uma malha contendo massas e molas. Nesta malha, podem ser atribuídas massas aos vértices, enquanto nas arestas são atribuídas molas. A simulação consiste então em calcular as forças de atração e repulsão exercidas entre os pontos do sistema ao longo do tempo. A resistência à deformação de um corpo, por exemplo, será dada pela resistência atribuída às molas que compõem a estrutura deste modelo. Representações 2D deste sistemas apresentarão topologias mais simples de molas. Este *design* está ilustrado na figura 2.2.

Em (BATTY et al., 2012), é apresentado um modelo discreto que representa a dinâmica de fluidos que se apresentam como uma lâmina viscosa, com a preservação do volume. Neste trabalho, a lâmina viscosa é apresentada como um corpo líquido sem forma delimitada, com volume constante, com um grau de elasticidade, uma tensão de superfície e que pode eventualmente interagir com outros corpos rígidos. A proposta modela as lâminas viscosas como malhas de triângulos, que representam o líquido em questão. Com este modelo é possível executar uma simulação discretizada, porém ainda consistente com a física, das forças envolvidas neste tipo de sistema. Uma variedade de líquidos que dobram ou que se enrolam quando derramados (mel, glace, lava) podem ser simulados com base neste trabalho.

## 2.2 PBD (**Position-based Dynamics**)

Muitos problemas de simulação física de corpos deformáveis têm sido resolvidos através de métodos matemáticos bem estabelecidos como MEF (Método dos Elementos Finitos) (JIMÉNEZ; THOMAS; TORRAS, 2001). Esta técnica possibilita um cálculo com maior exatidão dos valores, contudo leva a um tempo maior de processamento. Por este motivo, simulações em tempo real podem ser inviabilizadas se forem realizadas com MEF.

```

(1) forall vertices  $i$ 
(2)   initialize  $\mathbf{x}_i = \mathbf{x}_i^0, \mathbf{v}_i = \mathbf{v}_i^0, w_i = 1/m_i$ 
(3) endfor
(4) loop
(5)   forall vertices  $i$  do  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t w_i \mathbf{f}_{ext}(\mathbf{x}_i)$ 
(6)   dampVelocities( $\mathbf{v}_1, \dots, \mathbf{v}_N$ )
(7)   forall vertices  $i$  do  $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$ 
(8)   forall vertices  $i$  do generateCollisionConstraints( $\mathbf{x}_i \rightarrow \mathbf{p}_i$ )
(9)   loop solverIterations times
(10)    projectConstraints( $C_1, \dots, C_{M+M_{coll}}, \mathbf{p}_1, \dots, \mathbf{p}_N$ )
(11)  endloop
(12)  forall vertices  $i$ 
(13)     $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$ 
(14)     $\mathbf{x}_i \leftarrow \mathbf{p}_i$ 
(15)  endfor
(16)  velocityUpdate( $\mathbf{v}_1, \dots, \mathbf{v}_N$ )
(17) endloop

```

**Figura 2.3** Algoritmo *Position-based Dynamics* (PBD) (MÜLLER et al., 2007)

Simulações baseadas em física tradicionalmente tomam como ponto de partida as leis do movimento descritas por Newton, compondo um sistema de forças. O inconveniente deste tipo de simulação é a necessidade de se calcular primeiramente a aceleração das forças envolvidas, para derivar a equação das velocidades envolvidas (primeira ordem) e só então derivar as equações das posições envolvidas (segunda ordem).

No trabalho de (MÜLLER et al., 2007) é apresentada a técnica de PBD, um tratamento diferente para se modelar uma simulação física. O sistema de forças (tradicional na simulação com base das leis de Newton) fica em segundo plano nesta abordagem. O algoritmo que implementa esta técnica é apresentado na figura 2.3.

Usando PBD, os objetos simulados são representados em um conjunto de  $N$  vértices e  $M$  restrições. Cada vértice  $i \in [1..N]$  possui como atributos importantes a sua posição  $x_i$ , uma velocidade  $v_i$  e uma massa  $m_i$ . No algoritmo, as linhas (1) a (3) apenas iniciam os valores destes atributos.

Em cada iteração do laço principal do algoritmo linhas (4) até (17), é utilizado o intervalo de tempo  $\Delta t$  conforme o método de Euler. A parte que mais diferencia o algoritmo do método de integração de Euler está nas linhas (7), (9) até (11), (13) até (14).

Na linha (7), é calculada uma posição prevista  $p_i$  de cada vértice usando uma integração explícita de Euler. Na linha (8), a função *generateCollisionConstraints* realiza a verificação dos vértices que sofreram colisão com outro corpo na simulação. Como resultado desta verificação, um conjunto de restrições de colisão com cardinalidade  $M_{col}$  é gerado dinamicamente em cada  $\Delta t$ . As colisões encontradas nesta verificação são adicionadas ao conjunto original de restrições, ampliando este conjunto para



$j \in [1, 2, \dots, M, M + 1, M + 2, \dots, M + M_{col}]$ . A simulação processará então este novo conjunto de restrições.

Nas linhas (9) até (11), as restrições dos vértices são aplicadas através da solução iterativa para satisfazer o posicionamento de  $N$  vértices a  $M + M_{col}$  restrições. A quantidade de iterações determinada em *solverIteration* é arbitrária, e determinará a dinâmica de correção aplicada à posição prévia  $p_i$ . A linha (9) executa a correção propriamente dita da posição  $p_i$ . Cada restrição  $C_j$  está vinculada a um conjunto de vértices. Pode ocorrer de um mesmo vértice estar presente em mais de uma restrição. Como exemplo, uma restrição de distância  $d$  entre dois pontos  $x_1$  e  $x_2$  no espaço  $\mathbb{R}^3$  pode ser formulada como:  $C_j(x_1, x_2) = |x_1 - x_2|^2 - d^2$ . Satisfazer a esta restrição levará à solução de uma equação não linear. A resolução do conjunto de restrições  $C_1, C_2, \dots, C_{M_{col}}$  será feita através de uma aproximação para um conjunto de sistemas lineares, conforme as equações adotadas em (MÜLLER et al., 2007) que utiliza uma adaptação similar ao método iterativo de Gauss-Siedel. Nos passos (13) e (14), os valores finais das posições são finalmente definidos após a correção. As velocidades decorrentes destes ajustes também são atualizadas.

O algoritmo proposto possui uma estrutura similar com o algoritmo de integração numérica de Euler. Além disso, evita a instabilidade gerada no sistema por conta do uso das equações do movimento de Newton, que necessitam de derivações até a segunda ordem para finalmente determinar as posições dos corpos da cena. Isto porque os passos (13) até (14) não permitem extrapolações da posição prevista  $p_i$ . Tais extrapolações levariam o cálculo a uma condição impossível do sistema, visto que nos passos anteriores (9) a (11) as posições já foram computadas dentro das restrições permitidas. Por fim, oferece uma alternativa menos custosa que o MEF para simular corpos deformáveis, corpos rígidos e fluidos, dentre outros sistemas considerados complexos.

O trabalho de (MÜLLER et al., 2007) foi complementado posteriormente na proposta explicada em (FRATARCANGELI; PELLACINI, 2013), que oferece uma solução para paralelizar o algoritmo de PBD. A proposta leva a uma redução do uso da CPU (*Central Processing Unit*) e a distribuição do processamento para os vários núcleos disponíveis na GPU (*Graphics Processing Unit*). Isto deixa a execução da animação sem interrupções e favorece a implementação simulações complexas com PBD em tempo real.

Outra complementação importante para a técnica PBD foi feita mais tarde por (MAC-KLIN; MÜLLER; CHENTANEZ, 2016), que propuseram um método simples para tornar a rigidez dos corpos independente do intervalo de tempo determinado na simulação e independente da quantidade de iterações determinada para resolução das restrições. Com isto, o aspecto visual de elasticidade dos corpos em uma cena torna-se mais controlável através de um parâmetro independente, específico para este fim.

Por fim, a implementação da técnica PBD traz uma simplificação para o problema da simulação de corpos deformáveis. As restrições atuarão como uma parametrização do movimento do corpo. A implementação será a resolução numérica de um sistema não linear de equações através de  $N$  iterações.

## 2.3 OPENGL

### 2.3.1 Características

OpenGL (*Open Graphics Library*) é uma API (*Application Programming Interface*) padrão para a criação de aplicações 3D que precisem apresentar gráficos de qualidade, utilizando os recursos da placa gráfica com eficiência.

O desenvolvimento de software utilizando OpenGL permite o acesso aos recursos da placa gráfica, através de um *pipeline* que possui estágios pré-programados e estágios programáveis através da criação de programas denominados *shaders*. O uso do OpenGL dá ao usuário o acesso indireto aos recursos de processamento/entrada e saída do dispositivo gráfico através de uma máquina de estados, na qual se inicializam valores em áreas de memória (*buffers*) e submetem estes valores ao comando desejado.

A programação com OpenGL exige, portanto, que um passo a passo seja seguido rigorosamente, para que a máquina de estados se mantenha consistente. Cada comando executado levará a máquina a um novo estado e assim por diante. A renderização de uma imagem passa por passos específicos para a criação dos espaços de memória, movimentação dos dados para este espaço, montagem dos dados para desenho, desenho da figura na tela.

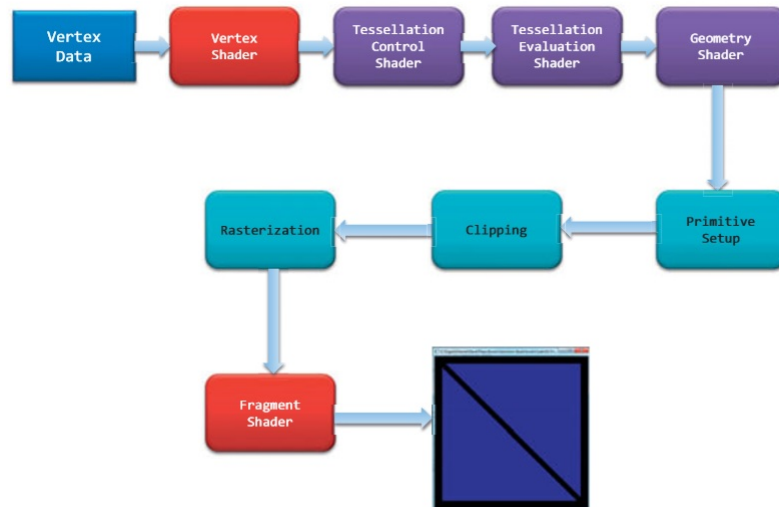
### 2.3.2 Programação na GPU com shaders

Na programação com OpenGL, a CPU tem como papel de gerenciar informações de entrada/saída, bem como controlar os dados de controle da aplicação principal. A biblioteca OpenGL oferece a possibilidade de executar programas na GPU, denominados *shaders*. Uma GPU é capaz de executar centenas de instruções em paralelo, o que é importante em aplicações gráficas, que frequentemente executam as mesmas instruções para um grande conjunto de dados (pontos) que serão desenhados na tela. A GPU poderá acionar diversas instâncias de um mesmo *shader* passando um conjunto de dados diferente. Desta forma, é possível realizar renderizações rapidamente, sem sobrecarregar a CPU.

A figura 2.4 representa as etapas programáveis do pipeline do OpenGL. O pipeline inicia com o fornecimento das estruturas de dados que representam as primitivas que serão desenhadas. O conjunto de dados dos vértices são transferidos do programa principal para os *buffers* da GPU, iniciando o processamento na etapa do *vertex shader*. Os três passos seguintes são opcionais. As etapas de tesselação cuidam basicamente de possíveis subdivisões das primitivas originais. Caso haja necessidade de modificar a geometria, portanto a primitiva de desenho, então será programado um *shader* específico para a geometria.

Os *shaders* são programas que operam sobre grandes conjuntos de dados. São especializados de acordo com a etapa do *pipeline* que foram designados. Dentre os *shaders* mais comuns na biblioteca OpenGL, há o *vertex shader* (destinado a operar sobre os vértices do modelo, sendo úteis para o processamento das transformações geométricas), o *fragment shader* (destinado a operar sobre os pixels da imagem gerada, sendo úteis no cálculo e estabelecimento final da cor).

A partir da versão 4.3 do OpenGL, é introduzido o *compute shader*. Este tipo de



**Figura 2.4** Diagrama representativo dos estágios do OpenGL (SHREINER et al., 2013b)

*shader* é particularmente útil quando se deseja realizar processamentos genéricos que não necessariamente irão resultar em dados que se converterão em imagens ou primitivas. Algoritmos paralelizáveis poderão tirar um melhor proveito dos recursos do hardware executando na GPU em um *compute shader*. Para simulações baseadas em física, o cálculo necessário para a atualização das velocidades e das posições dos corpos poderão ser paralelizados. Em outro passo, posteriormente, os valores calculados poderão ser transferidos para os *buffers* que serão utilizados pelos vértices que formam a imagem da cena.

Algoritmos de simulação de partículas podem ser paralelizáveis para fornecer uma apresentação fluida, sem interrupções perceptíveis pelo usuário. No caso da simulação de um corpo deformável, a paralelização deverá ser criteriosa, pois há uma interação mútua entre elementos vizinhos pertencentes a uma mesma malha. Neste caso, a atualização concorrente da posição de uma mesma partícula que seja um vizinho em comum causará condições de corrida, provocando uma inconsistência no sistema simulado.

Uma possível solução para este problema foi estudada em (FRATARCANGELI; PELLACINI, 2013). A proposta utiliza o conceito de coloração de grafos para formar grupos de partículas que não tenham um vizinho em comum. Cada partição será processada em paralelo. Como as partições não compartilham partículas, então o problema de inconsistência deixa de existir, e a malha é processada mais rapidamente.

## 2.4 MAPEAMENTO DE TEXTURAS

Mapeamento de texturas é um procedimento bastante utilizado na computação gráfica (HAEBERLI; SEGAL, 1993). O principal objetivo desta técnica é adicionar realismo a uma cena. A forma mais simples de mapeamento de textura consiste em aplicar uma imagem 2D a um objeto 3D da cena. (SHREINER et al., 2013a)

Para realização desta técnica, define-se uma função de mapeamento entre os domínios

2D (a textura) e 3D (a superfície do modelo) e, em seguida, são criadas coordenadas de textura associadas aos vértices do modelo 3D. Existem diversas maneiras de se definir a função de mapeamento conforme mostrado em (HORMANN; LÉVY; SHEFFER, 2007).

Na pintura hidrográfica, o mapeamento da textura será feito em três etapas.

- Primeiramente, a superfície textura será aplicada a um plano, modelado como uma malha regular de triângulos, na qual se conhece as posições iniciais de cada ponto da textura neste plano.
- Na segunda etapa, executa-se a simulação da impressão hidrográfica e verifica-se como o plano 2D se deformou em relação ao modelo 3D, levando cada ponto do plano 2D de uma posição inicial (antes da deformação do plano) para uma posição final (após a deformação do plano). O mapeamento ocorrerá automaticamente ao se relacionar cada ponto da posição inicial com a posição final.
- Por fim, o usuário do sistema, conhecendo como cada região da textura 2D se deformará para encontrar os pontos do objeto impresso em 3D, poderá estabelecer cores arbitrárias para cada região do objeto impresso em 3D. Em seguida, gerará um filme com uma impressão espelhada do desenho desejado, para finalmente executar a impressão hidrográfica física.

*Será feita uma descrição da proposta, detalhamento dos recursos necessários, primeiros resultados, cronograma*

## **DEFINIÇÃO DA PROPOSTA**

### **3.1 PROPOSTA**

A proposta para realização deste projeto de pesquisa divide-se em três grandes frentes de trabalho:

- Realização do experimento físico
- Implementação de uma simulação computacional
- Elaboração da dissertação e artigos

Estas três frentes são executadas em paralelo, incrementalmente, partindo de um estágio menos elaborado (com protótipos) e progressivamente chegando ao estágio mais elaborado (refinamentos do protótipo inicial), até o fim do trabalho.

#### **3.1.1 Realização do experimento físico**

Para realização do experimento físico, foram empregados os materiais básicos para a impressão hidrográfica descritos no A.1.

Um experimento preliminar foi realizado com uma peça de prova impressa em 3D e uma película hidrográfica com uma estampa de padronagem similar à fibra de carbono. A peça de prova escolhida foi um modelo 3D de um artefato cultural com formato de tartaruga, que encontra-se no acervo do Museu de Arqueologia e Etnologia da UFBA. Isto porque é possível que os resultados deste projeto sejam aproveitados para colorir outras peças do acervo, que atualmente está sendo submetido a um processo de reconstrução 3D.

A figura 3.1 é uma foto do aspecto final da peça de prova utilizada no experimento preliminar.



**Figura 3.1** Experimento preliminar.

A partir deste experimento preliminar, foi possível realizar anotações relevantes sobre o funcionamento da técnica. Estas anotações, junto às observações feitas em trabalhos anteriores foram levadas em consideração para a implementação da primeira simulação em software.

Posteriormente, para a criação de películas com estampas customizadas, será preciso realizar impressões em uma impressora com tinta pigmentada em um filme para pintura hidrográfica apropriado. A tinta pigmentada, por apresentar resistência à água, oferece melhor fixação para o propósito da pintura hidrográfica. Nesta etapa do trabalho, está sendo avaliado se será utilizado um serviço de impressão com tinta pigmento contratando uma gráfica ou se será adquirida uma impressora específica para este fim.

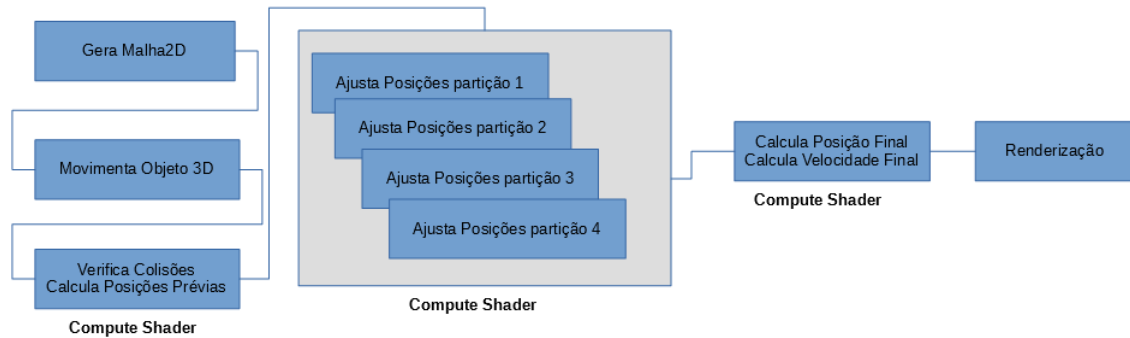
### **3.1.2 Implementação da simulação computacional**

Foi desenvolvido um software em C++ com OpenGL 4.3 e uso eficiente da GPU para renderização da simulação, que ocorre em tempo real.

A simulação computacional da pintura hidrográfica partiu da hipótese de que o comportamento do filme que flutua na superfície da água seria semelhante ao comportamento de uma malha deformável de pontos 2D. Quando colide com um corpo rígido em velocidade constante, a malha sofre uma deformação conforme a forma e o movimento do objeto que se desloca verticalmente, de cima para baixo, paralelo ao eixo Y.

Para efeito de simplificação dos cálculos para a verificação da colisão, o corpo rígido escolhido para a primeira simulação foi uma esfera.

O software de simulação recebe os parâmetros para criação da malha, que inicialmente são as quantidades de pontos na sua largura  $L$  e comprimento  $C$ . Inicialmente não é especificado um parâmetro para expressar a espessura da malha. No entanto, esta informação é necessária e encontra-se fixa na simulação inicial implementada. Também é passado para o programa principal um parâmetro referente à distância entre os pontos, o grau de rigidez (que determinará o quanto a malha poderá esticar), e a textura que a malha



**Figura 3.2** Representação das etapas do algoritmo implementado.

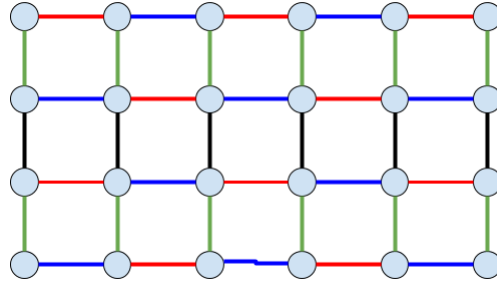
assumirá.

Nesta simulação preliminar, foram determinadas duas opções de textura: um padrão xadrez e um mapa-múndi. A função de mapeamento da textura para a malha é definida em através da fração da distância entre os vértices (dos triângulos que compõem a malha) e a dimensão total da malha. Os vértices da malha recebem, além das suas coordenadas e das suas normais, qual a coordenada de textura mapeada. A aplicação da textura ocorre ainda quando a malha está plana. Quando a forma da malha é alterada, como os pontos já estão vinculados às coordenadas de textura, o desenho é igualmente deformado, representando a distorção sofrida pela película da hidropintura ao aderir a um substrato.

Para simular a deformação da malha, foi assumido que os pontos do filme que colidiram com o objeto seguirão a mesma velocidade vertical deste corpo. Isto seria suficiente para representar a adesão da película ao corpo em movimento.

O processo de verificação da colisão da esfera com a malha 2D foi implementado através de um *compute shader*. Para este *compute shader*, foram passados: posição, raio e velocidade da esfera, espessura do filme, um vetor com as posições de todos os vértices compõem a malha 2D. Este vetor é indexado através de recursos próprios do OpenGL, que permitem sistematizar a execução das instâncias dos *compute shaders*, criando identificadores únicos para cada instância. Na implementação feita, *compute shaders* são disparados simultaneamente através de grupos de trabalho determinados pelas dimensões L e C da malha 2D. Cada grupo de trabalho, por sua vez, subdivide-se em dimensões X e Y expressas na propriedade *local size*, definida internamente no *compute shader*. O dimensionamento do *local size* foi afixado com o valor constante de 4 x 4, o que representou na prática um azulejamento da malha. Sendo assim, uma malha 2D de dimensões 64 x 32 pontos foi segmentada em azulejos 4 x 4, representando um total de 128 *compute shaders* processados simultaneamente. Cada instância deste *compute shader* verifica a distância entre um ponto do plano e o raio da esfera. Caso esta distância seja menor do que uma distância mínima estabelecida (considerando-se também a espessura da película), então houve a colisão. Neste caso, o ponto do plano seguirá o mesmo movimento no eixo Y imposto pela esfera.

Após verificação da colisão, os demais pontos do filme, que ainda não aderiram ao



**Figura 3.3** Representação das restrições de distância como sendo as linhas que ligam os vértices da malha 2D. As cores representam as partições do conjunto de vértices, que deve ser processado sequencialmente pelo *compute shader*.

objeto, são puxados pelo movimento do objeto. Estes pontos seguirão com uma velocidade que será uma composição da velocidade do objeto com a velocidade imposta pelas partículas vizinhas (através do cálculo das restrições). Na simulação preliminar, estes pontos da malha são mantidos equidistantes com a técnica PBD (*Position-based Dynamics*). As partículas da malha são associadas duas a duas a uma restrição de distância. A cada intervalo de tempo, as posições das partículas são recalculadas, mantendo a restrição de distância. O recálculo da velocidade é necessário pois dá o efeito realista de movimentação da malha, fazendo com que de maneira natural a película abrace o objeto, tal qual ocorre no experimento físico.

A paralelização do algoritmo com PBD para aplicação das restrições de distância se deu com base no estudo apresentado em (FRATARCANGELI; PELLACINI, 2013). Para o cálculo paralelizado seja possível, é necessário implementar uma forma de indexar os vértices da malha, agrupando os pares de vértices que podem ser calculados independente dos outros. Esta técnica pode ser comparada ao cálculo da coloração de um grafo. Com isto, é possível tratar a atualização de malhas que possuam partículas que tenham interferências entre si.

Na implementação, *design* das restrições seguiu uma topologia na qual cada ponto da malha ficou com uma vizinhança dos pontos situados acima, abaixo, à sua direita, ou à sua esquerda. Sendo assim, exceto pelos pontos que situam-se nas bordas da malha - que possuem 2 ou 3 vizinhos - cada ponto possui 4 vizinhos. O cálculo da coloração do grafo determinou que nesta topologia, podem ser formados 4 partições da malha (representadas pelas cores), de forma que cada partição deve ser processada sequencialmente, sem interferir nas demais. A figura 3.3 mostra a configuração das restrições e as cores representam as partições determinadas na implementação desta estrutura.

A figura 3.2 traz um diagrama que representa as etapas mais importantes da implementação feita.

No algoritmo implementado, recálculo das velocidades e posições finais através (após o cálculo particionado das restrições) também foi realizado com o uso de *compute shaders*, aproveitando a mesma lógica de indexação e azulejamento empregado no cálculo das colisões. Com isto, foi possível realizar uma simulação que proporcionasse a atualização



final das posições com mais fluidez.

**3.1.2.1 Resultados iniciais** A simulação foi executada em um ambiente compatível com:

- Processador Intel Core i5 7th Gen 7200U 2.5GHz
- Windows 10
- Opengl 4.3 ou superior
- Placa gráfica NVIDIA GeoForce 940MX com 2GB de memória RAM dedicada
- 8GB de memória RAM DDR4
- 1GB de disco rígido

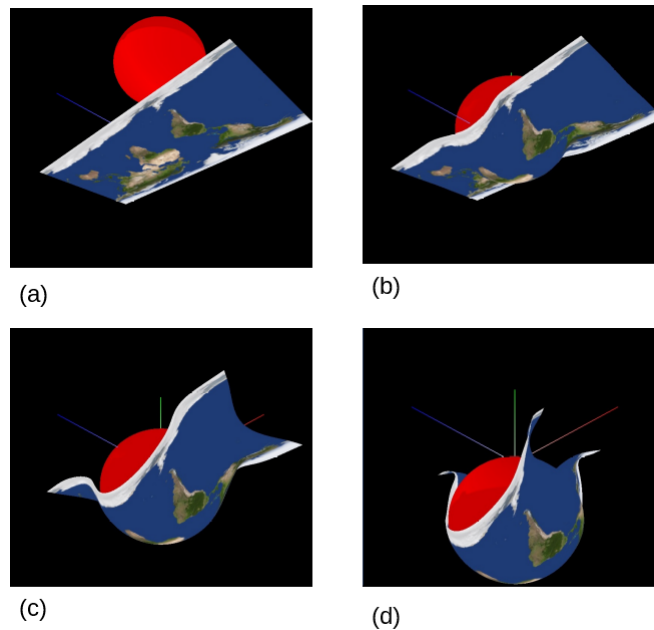
O software implementado possui as seguintes características:

- Simula através de uma animação computacional em tempo real a impressão hidrográfica de um dos padrões (xadrez ou mapa-múndi) em uma esfera.
- Permite verificar a distorção do filme em 3 câmeras diferentes.
- Permite iniciar e parar o movimento da impressão hidrográfica.
- Permite girar a cena.

A principal parte do algoritmo da simulação está implementada em um laço, muito semelhante a uma animação baseada em física com a estrutura da integração Euleriana. A principal distinção entre esta simulação e a integração Euleriana é que esta baseia-se no cálculo das posições. Como é sabido, pela técnica de PBD o cálculo de posições de uma malha que sofreu uma colisão é resolvido através de um sistema de equações formado pelo conjunto de restrições com cardinalidade conhecida e um conjunto de posições com cardinalidade desconhecida que é dado como o cálculo de um sistema não linear. A resolução deste sistema é resolvida com iterações, através aproximação que leva ao método de Gauss-Siedel.

A próxima etapa da simulação é calcular a posição dos pontos que colidiram. Isto é feito em duas situações distintas: no primeiro caso, o vértice que colidiu adere à superfície do objeto, assumindo a mesma velocidade e posição do objeto. Em uma segunda situação, o após a primeira colisão, haverá uma propagação do movimento para os pontos que ainda não colidiram. Esta propagação será imediatamente percebida nos elementos vizinhos aos que colidiram. Para os elementos mais distantes da colisão, esta propagação será mais lenta. Esta etapa também é executada por um *compute shader*, que dispara em paralelo o cálculo das restrições de toda a malha.

A sequência de imagens na figura 3.4 foi capturada da simulação da pintura obtida com esta implementação preliminar com a textura do mapa-múndi e uma malha de dimensões 32 x 64 pontos, totalizando 2048 pontos.



**Figura 3.4** (a) Início da simulação. (b) Esfera tocando a película. (c) O objeto continua a se movimentar esticando a película. (d) A película segue o movimento aderindo ao objeto.

A verificação da eficácia da técnica será feita comparando-se os resultados de pinturas com uma pintura guiada a partir da simulação hidrográfica computacional. A simulação será considerada bem-sucedida na medida em que conseguir reproduzir fidedignamente uma pintura física. Também se busca uma simulação com uma boa taxa de FPS (Frames por Segundo). Na etapa de avaliação de resultados, serão testados objetos com superfícies arbitrárias e serão feitas medições com algumas variações em função da quantidade de pontos calculados na cena e da quantidade de iterações feitas na simulação.

### 3.1.3 Elaboração da dissertação e artigos

A elaboração da dissertação partirá da consolidação do levantamento bibliográfico, do refinamento das anotações realizadas em todas etapas da experimentação, análise de dados, bem como o aproveitamento de parte da estrutura e do texto da qualificação.

A partir da etapa de qualificação, serão gerados dois artigos: o primeiro com base na proposta e resultados iniciais da qualificação, e o segundo com a pesquisa e resultados finais gerados até a conclusão do mestrado.

A estratégia para submissão de artigos se dará selecionando os congressos e revistas nacionais e internacionais relevantes na área da computação gráfica, priorizando os que possuam melhor qualificação.

## 3.2 CRONOGRAMA

A tabela 3.1 lista as atividades que serão feitas até a conclusão do mestrado.

**Tabela 3.1** Cronograma das atividades do mestrado.

Atividade	2018.2						2019.1					
	7	8	9	10	11	12	1	2	3	4	5	6
Estudo da literatura:	X	X	X	X	X							
Definição da proposta	X	X										
Implementação			X	X	X	X	X	X	X			
Análise dos dados							X	X	X			
Escrita da dissertação							X	X	X	X	X	
Escrita de artigos				X	X							
Qualificação			X	X								
Logística e produção de material gráfico	X		X		X		X		X			X
Preparação da apresentação										X	X	X
Apresentação da dissertação												X

Segue o detalhamento das atividades do cronograma:

- Estudo da literatura: levantamento bibliográfico de trabalhos que serão tomados como referência para a elaboração da pesquisa.
- Definição da proposta: identificação da problemática, do problema, gerar hipóteses, formular o objetivo, delimitar escopo.
- Implementação: Concepção da simulação, modelagem do problema, proposição de solução técnica.
- Análise dos dados: Verificação dos resultados da simulação, comparação da simulação com o experimento físico, verificação do desempenho.
- Escrita da dissertação: Escrever o texto final que será apresentado na dissertação.
- Escrita de artigos: Escrever dois artigos e submetê-los para publicação em revistas ou congressos qualificados.
- Qualificação: Escrever e submeter o projeto de pesquisa para qualificação do trabalho.
- Logística e produção de material gráfico: Aquisição de material para realização do experimento. Realização de impressões na película hidrográfica.
- Preparação da apresentação: Criação da apresentação para apresentação final. Impressão e entrega do trabalho escrito.
- Apresentação da dissertação: Apresentação final do trabalho perante banca para avaliação.



*Síntese deste texto, com algumas considerações preliminares*

## **CONCLUSÕES**

### **4.1 CONCLUSÕES PRELIMINARES**

Os resultados iniciais da simulação que foi implementada são bastante promissores. Ainda não houve uma avaliação entre a simulação e o experimento físico. Contudo, acredita-se que com alguns ajustes importantes, será possível desenvolver uma simulação próxima do fenômeno real.

Será necessário aprimorar a técnica, para que seja possível simular a pintura em qualquer objeto e não somente uma esfera. A ideia será implementar um sistema de detecção de colisão entre os triângulos de uma malha qualquer (representando o objeto) com os pontos da malha plana (que representa o filme). Será preciso projetar um sistema que seja eficiente em termos de desempenho.

O controle da cor também não foi implementado. Durante o experimento físico, percebe-se que algumas regiões do filme sofrem um maior esticamento do que outras, o que no leva a um clareamento da cor. É desejável que após a simulação, seja impressa no filme uma imagem que possa amenizar estas distorções, gerando uma compensação inversa (aumentando a pigmentação nas regiões em que se preveja um maior esticamento). Para isto, será necessário pesquisar métodos de simular a variação da espessura do filme e propor uma função que torne a cor mais escura na medida em que o filme esteja em uma região de maior esticamento.

A simulação permite que o usuário veja como ficaria um objeto se submetido a pintura hidrográfica. Porém, ainda não permite preparar o filme com a imagem distorcida, que será aplicada ao objeto. Um passo importante seria gerar o mapeamento dos pontos do objeto no filme (chamado usualmente de função inversa de mapeamento). Assim, tendo um objeto conhecido, seria possível mapear no filme os pontos-chave do objeto em que se deseje imprimir uma textura. Para isto, a proposta seria determinar uma função de aproximação, que leve pontos médios de correspondência entre os vértices da malha deformada e os vértices do objeto.

A pesquisa tem o potencial de proporcionar um desenvolvimento nas técnicas de dar acabamento a objetos criados em impressoras 3D. O potencial da impressão 3D colorida

é bastante amplo, tornando-a aplicável em inúmeras situações em que se necessite automatizar a reprodução precisa de formas complexas em 3D a um baixo custo. Protótipos criados em impressão 3D têm sido constantemente utilizados como prova de conceito, viabilizando projetos. Talvez por, por este motivo, esta tecnologia foi ganhando outras aplicações: na área médica para geração de próteses, na indústria de entretenimento gerando miniaturas de personagens de filmes e jogos, na confecção de vestimentas e calçados.

No contexto da preservação de artefatos culturais, a impressão 3D colorida vem crescendo, sobretudo no auxílio a preservação de ativos históricos. Atualmente, é possível capturar um objeto histórico em três dimensões e representá-lo em um modelo 3D. Com este objeto capturado e representado virtualmente, é possível recriá-lo em três dimensões. No caso de uma peça que esteja incompleta pela ação do tempo, é possível modelar partes faltantes da peça e recompô-la com o auxílio da tecnologia.

Nos dias de hoje, museus que possuem um acervo digitalizado em 3D podem facilmente reproduzir peças deste acervo e colocá-las em exposição, mantendo a peça original resguardada em um outro ambiente, evitando a degradação pela ação do tempo. Peças originais e portanto, raras, devem estar protegidas de qualquer intempérie. Artefatos importantes devem estar em segurança, evitando um dano por algum tipo de acidente ou catástrofe natural. De forma complementar, um usuário pode tocar uma peça impressa em 3D parecida com a original, tendo contato com uma experiência sensorial muito mais completa. Museus históricos podem oferecer como souvenir a réplica da miniatura de um artefato representativo, difundindo ainda mais a experiência tida em uma visita.

## REFERÊNCIAS BIBLIOGRÁFICAS

- BARAFF, D.; STUDIOS, P. A. Physically based modeling. 1999. Disponível em: <http://www.joinville.udesc.br/portal/professores/jacimar/materiais/pixar.pdf>.
- BATHE, K.-J. Finite element method. *Wiley encyclopedia of computer science and engineering*, Wiley Online Library, p. 1–12, 2007.
- BATTY, C. et al. Discrete viscous sheets. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)*, v. 31, n. 4, p. 113:1–113:7, 2012.
- FRATARCANGELI, M.; PELLACINI, F. A gpu-based implementation of position based dynamics for interactive deformable bodies. *Journal of Graphics Tools*, Taylor & Francis, v. 17, n. 3, p. 59–66, 2013.
- GAO, Z.; GIBSON, I. Haptic sculpting of multi-resolution b-spline surfaces with shaped tools. *Computer-aided design*, Elsevier, v. 38, n. 6, p. 661–676, 2006.
- HAEBERLI, P.; SEGAL, M. Texture mapping as a fundamental drawing primitive. In: *Fourth Eurographics Workshop on Rendering*. [S.l.: s.n.], 1993. v. 259, p. 266.
- HORMANN, K.; LÉVY, B.; SHEFFER, A. Mesh parameterization: Theory and practice. 2007.
- JIMÉNEZ, P.; THOMAS, F.; TORRAS, C. 3d collision detection: a survey. *Computers & Graphics*, Elsevier, v. 25, n. 2, p. 269–285, 2001.
- MACKLIN, M.; MÜLLER, M.; CHENTANEZ, N. Xpbd: position-based simulation of compliant constrained dynamics. In: ACM. *Proceedings of the 9th International Conference on Motion in Games*. [S.l.], 2016. p. 49–54.
- MILLINGTON, I. *Game physics engine development*. [S.l.]: CRC Press, 2007.
- MOORE, M.; WILHELMS, J. Collision detection and response for computer animation. In: ACM. *ACM Siggraph Computer Graphics*. [S.l.], 1988. v. 22, n. 4, p. 289–298.
- MÜLLER, M. et al. Position based dynamics. *Journal of Visual Communication and Image Representation*, Elsevier, v. 18, n. 2, p. 109–118, 2007.
- PANOZZO, D. et al. Texture mapping real-world objects with hydrographics. *Computer Graphics Forum (proceedings of EUROGRAPHICS Symposium on Geometry Processing)*, v. 34, n. 5, p. 65–75, 2015.

REEVES, W. T. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics (TOG)*, ACM, v. 2, n. 2, p. 91–108, 1983.

SHREINER, D. et al. *OpenGL programming guide: The Official guide to learning OpenGL, version 4.3*. [S.l.]: Addison-Wesley, 2013.

SHREINER, D. et al. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*. 8th. ed. [S.l.]: Addison-Wesley Professional, 2013. ISBN 0321773039, 9780321773036.

ZHANG, Y. et al. Computational hydrographic printing. *ACM Transactions on Graphics (TOG)*, ACM, v. 34, n. 4, p. 131, 2015. Disponível em: <http://www.cs.columbia.edu/~cxz/publications/hydrographics.pdf>.



## Apêndice

# A

*Aqui serão detalhados os materiais e os passos para execução do experimento preliminar com a pintura hidrográfica*

## DETALHAMENTO DO EXPERIMENTO PRELIMINAR

### A.1 MATERIAL UTILIZADO

Foi adquirido um material básico para realização do experimento, conforme a lista que segue:

- a) Container plástico de 20 litros retangular
- b) Máscara de proteção
- c) Luvas de proteção
- d) 200 ml de ativador tipo A
- e) Borrifador
- f) 1 rolo de fita crepe
- g) 3 folhas de película hidrográfica com fundo transparente e estampa prateada de carbono (dimensões 100 x 50 cm)

### A.2 ETAPAS DO EXPERIMENTO

1. Encher o container plástico com água morna entre 27°C a 30°C. Esperar o movimento da água cessar.
2. Obter um pedaço de película com dimensões compatíveis com container. Deve-se colocar uma moldura com fita crepe nas bordas da película, afim de conter a sua expansão quando ativada.

3. Dispor a película suavemente no tanque. Sem deixar formar bolhas por debaixo da película. Uma técnica prática consiste em segurar a película pelas extremidades de sua diagonal e ir dispondo-a suavemente.
4. Observar o movimento inicial da película: inicialmente ela ficará enrugada, depois ficará mais plana.
5. Borrifar aos poucos uma pequena quantidade de ativador tipo A em toda extensão da película. No experimento realizado, foram dadas 6 borrifadas. Observar a reação da película. A película deve se dissolver gradativamente, até adquirir um aspecto brilhante na superfície da água. Neste momento, estará pronta para a imersão do objeto.
6. Submergir lentamente o objeto de forma oblíqua em relação a linha d'água. Recomenda-se que o objeto esteja a um ângulo de aproximadamente  $30^\circ$  em relação à linha d'água.
7. Ao finalizar a imersão, agitar o objeto para romper a película residual (que não será útil na pintura).
8. Aguardar de quatro a seis horas a secagem completa da peça.

Como o ativador possui um odor forte semelhante a um solvente, fez-se necessário o uso de máscara de proteção.