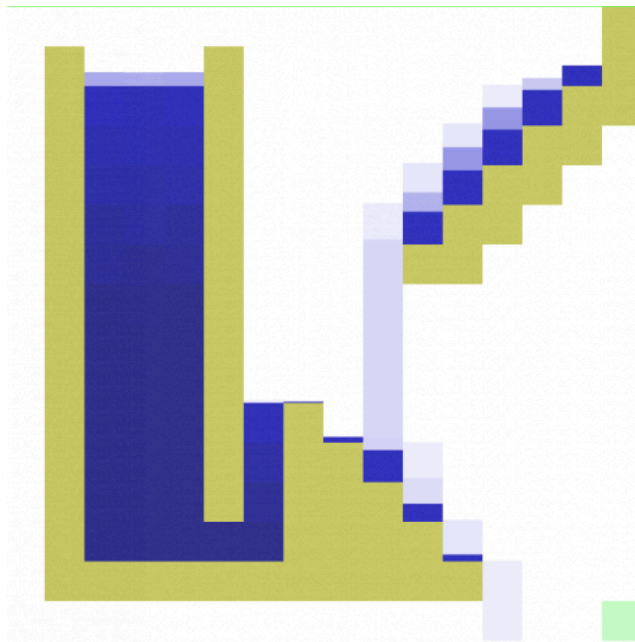


Simulação de Dinâmica de Fluidos com Autômatos Celulares

Simulações e Modelação



Joshua Dourado Capistrano n° 115799

Alexandre Gonçalves Ferreira n° 120527

Afonso Jorge Manata Salazar n° 119340

Tiago Filipe Miranda Soure n° 108769

Maio, 2024

Aveiro

Resumo.....	3
Introdução.....	3
Métodos.....	3
Regras de Simulação de Autômatos Celulares:.....	3
Fluxo para a Célula Vizinha Inferior:.....	3
Fluxo para Cima com Pressão:.....	3
Rastreamento da Direção do Fluxo:.....	3
Renderização de Células.....	3
Descrição do Código:.....	4
Inicialização:.....	4
Inicialização das Direções de Fluxo.....	4
Loop de Simulação:.....	4
Função de Sombreamento.....	5
Função de Sombreamento:.....	6
Resultados:.....	6
Discussão.....	6
Conclusão.....	6
Referências.....	6
Apêndice.....	6

Resumo

O objetivo deste trabalho é desenvolver uma simulação de fluidos utilizando autômatos celulares no MATLAB. A simulação implementa regras específicas para a movimentação do líquido dentro de uma grade. Os resultados demonstram a eficácia do método em replicar diversos comportamentos fluidos, como quando o fluxo sofre ação gravidade e influência de obstáculos.

Introdução

A simulação de fluidos é crucial para entender o comportamento de líquidos e gases em diferentes condições físicas. Este projeto utiliza uma abordagem inovadora para a representação de líquidos em um ambiente baseado em grade, utilizando o conceito de autômatos celulares. Inspirada pelo "Jogo da Vida" de Conway, onde células evoluem com base em um conjunto de regras predefinidas, esta técnica permite criar simulações visuais interativas de fluidos. Cada célula na grade contém um valor, representando a quantidade de líquido, sendo que cada célula evolui de acordo com regras específicas para simular o movimento dos fluidos. *A priori*, sabia-se que autômatos celulares poderiam ser utilizados para simulações interativas, mas faltava explorar sua aplicação específica na dinâmica de fluidos. O objetivo é criar uma ferramenta que permita visualizar e entender a dinâmica dos fluidos por meio de simulação baseada em autômatos celulares.

Métodos

Regras de Simulação de Autômatos Celulares:

Fluxo para a Célula Vizinha Inferior:

- Implementação de uma regra que simula o efeito da gravidade.
- A regra básica da gravidade em autômatos celulares é que o líquido, se houver, tende a se deslocar para baixo. Cada célula verifica continuamente o estado da célula inferior. Se essa célula estiver vazia (ou tiver menos fluido que a célula atual), o líquido na célula superior flui para a inferior. Essa transferência permanece até que a célula inferior esteja preenchida ou o líquido na célula atual se esgote.

Fluxo para as Células Vizinhas à Esquerda e Direita:

- Distribuição do líquido para células adjacentes horizontais.
- O líquido presente em uma célula não só desce para células inferiores, mas também se distribui lateralmente para as células adjacentes à esquerda e à direita. Se ambas as células vizinhas horizontais estiverem vazias, o líquido se divide igualmente entre elas. Se apenas uma das células vizinhas estiver vazia, todo o líquido escorre para essa célula. Esse movimento lateral simula a dispersão natural do líquido em um ambiente, promovendo uma distribuição mais uniforme.

Fluxo para Cima com Pressão:

- Simulação de líquidos que se movem contra a gravidade devido à pressão.
- Em determinadas condições, especialmente quando há uma alta concentração de líquido ou pressão na parte inferior, o líquido pode se mover para cima, contra a gravidade. Isso é comum em simulações de líquidos sob pressão, como em tubos ou recipientes fechados. As células superiores recebem líquido das células inferiores devido à pressão acumulada, criando um fluxo ascendente.

Rastreamento da Direção do Fluxo:

- Monitoramento contínuo da direção do fluxo de cada célula para ajustes visuais e precisão na simulação.
- Cada célula monitora e registra a direção do fluxo do líquido, seja para baixo, lateralmente, ou para cima. Esse rastreamento é crucial para ajustar a simulação em tempo real, garantindo que o movimento do líquido seja preciso e que os efeitos visuais reflitam corretamente as mudanças no fluxo. O rastreamento também ajuda a detectar padrões e anomalias no comportamento do líquido, permitindo ajustes finos na simulação.

Renderização de Células

- Ajustes na visualização para melhorar a percepção de líquidos fluindo, especialmente em pequenas quantidades.
- Para que a simulação seja visualmente clara e precisa, é importante ajustar a renderização das células. Células com pequenas quantidades de líquido devem ser visíveis e diferenciadas das células completamente cheias ou vazias. Isso pode ser feito ajustando a opacidade, cor, ou outros parâmetros visuais das células. A renderização adequada facilita a observação do fluxo e da distribuição do líquido, tornando a simulação mais intuitiva e informativa.

Descrição do Código:

O código desenvolve uma simulação de fluidos utilizando uma abordagem baseada em autômatos celulares. A seguir, é apresentada uma descrição detalhada das suas componentes:

Inicialização:

```
rows = 32;
columns = 32;
grid = zeros(rows, columns);
grid_direction = cell(rows, columns);
grid(31,2)=100;
thresholds = linspace(0,1,10);
A= size(thresholds);
numTones = A(2);
COLOR_SCALE = uint8([zeros(numTones, 2), linspace(255, 160, numTones)']);
COLOR_SCALE(1, :) = [255, 255, 255];
COLOR_SCALE = reshape(COLOR_SCALE, [numTones, 1, 3]);
coloredImage = uint8(zeros(rows, columns, 3));
```

Inicializa os parâmetros da simulação, incluindo a grade de células, a direção do fluxo e a escala de cores para visualização.

Adiciona obstáculos na grade, marcados com -1, para simular barreiras fixas.

Inicialização das Direções de Fluxo

```
grid_direction{y, x} = [0, 0]; % Sem direção inicial
% x,y pertencem a 1 a N
```

Define a direção inicial do fluxo para todas as células como nula.

Loop de Simulação:

```
for i=0:1000
    hold off;
    [grid,grid_direction]=update(grid,grid_direction);

    for i = 1:rows
        for j = 1:columns
            color = shade_of_white(grid(i, j), COLOR_SCALE,thresholds);
            coloredImage(i, j, :) = color;
        end
    end

    imagesc(coloredImage);
    hold on
    U = zeros(size(grid)); % Componente vertical das setas
    V = zeros(size(grid)); % Componente horizontal das setas
    for y = 1:rows
        for x = 1:columns
            U(y, x) = grid_direction{y, x}(1);
            V(y, x) = grid_direction{y, x}(2);
        end
    end
    quiver(1:columns, 1:rows, V, U, 'AutoScale', 'off');
    drawnow;
end
```

Executa o loop de simulação, atualizando a grade e a direção do fluxo, e renderizando a visualização em cada iteração.

Função de Sombreamento

Define a cor das células com base no valor do líquido, utilizando uma escala de cinza e um esquema de limiares.

Função de Atualização:

```

function [grid, grid_direction] = update(grid, grid_direction)
    div = 0.25;
    [rows, cols] = size(grid);

    % Limpeza
    for y = 1:rows
        for x = 1:cols
            if grid(y, x) <= 0
                grid_direction{y, x} = [0,0];
            end
        end
    end

    % Gravidade
    for y = rows:-1:1
        for x = 1:cols
            if y < rows && grid(y, x) > 0 && grid(y+1, x) < 1 && grid(y+1, x) ~= -1
                if grid(y, x) == grid(y+1, x)
                    value = grid(y, x);
                else
                    value = min(grid(y, x) - grid(y+1, x), div);
                end
                grid(y+1, x) = grid(y+1, x) + value;
                grid(y, x) = grid(y, x) - value;
                grid_direction{y+1, x} = [1, 0]; % Para baixo
            end
        end
    end

    % Divisão esquerda/direita
    for y = 1:rows
        for x = 1:cols-1
            a = randi([-1, 1]);
            if x==1
                a=1;
            end
            if grid(y, x+a) < 1 && grid(y, x+a) >= 0 && grid(y, x) > 0 && ((y < rows &&
grid(y+1, x) ~= 0) || y == rows) && grid(y, x+a) ~= -1
                value = min((grid(y, x) - grid(y, x+a)), div);
                grid(y, x+a) = grid(y, x+a) + value;
                grid(y, x) = grid(y, x) - value;
                if a > 0
                    grid_direction{y, x} = [0, 1]; % Para direita
                else
                    grid_direction{y, x} = [0, -1]; % Para esquerda
                end
            elseif grid(y, x+a) >= 0 && grid(y, x) > 1 && ((y < rows && grid(y+1, x) ~=
0) || y == rows) && grid(y, x+a) ~= -1
                value = min((grid(y, x) - grid(y, x+a)), div);
                grid(y, x+a) = grid(y, x+a) + value;
                grid(y, x) = grid(y, x) - value;
                if a > 0
                    grid_direction{y, x} = [0, 1]; % Para direita
                else
                    grid_direction{y, x} = [0, -1]; % Para esquerda
                end
            end
        end
    end
end

```

```

        end
    end
end

% Pressão
for y = 1:rows-1
    for x = 2:cols-1
        if (grid(y+1, x-1) >= 0 || grid(y+1, x-1) < 0) && (grid(y+1, x+1) >= 0 ||
grid(y+1, x+1) < 0) && grid(y+1, x) > 1 && grid(y, x) >= 0
            value = min((grid(y, x) - grid(y+1, x)), div);
            grid(y+1, x) = grid(y+1, x) + value;
            grid_direction{y+1, x} = [-1, 0]; % Para cima
        end
    end
end
end
end

```

Define a atualização da grade e das direções do fluxo, tendo em conta a gravidade, a divisão horizontal e a pressão.

Este código implementa de forma completa a simulação de fluidos utilizando autômatos celulares, demonstrando a interação dos líquidos com obstáculos e a movimentação em diferentes condições.

Resultados:

Os resultados das simulações realizadas com o método autômatos celulares mostraram:

- **Comportamento do fluido:** A simulação consegue replicar o movimento de líquidos sob a influência da gravidade, fluxo lateral e pressão.
Interação com Obstáculos: Fluido interagindo com obstáculos móveis e fixos foram simulados com precisão.
- **Visualização Melhorada:** Pequenas quantidades de líquido foram renderizadas de maneira mais visualmente agradável, mostrando unidades completas de líquido em células apropriadas.

Discussão

Os resultados indicam que a utilização de autômatos celulares é eficaz para simular a dinâmica de fluidos em um ambiente baseado em grade. As regras implementadas permitem uma boa representação do comportamento real dos fluidos, incluindo a influência da gravidade e a pressão. No entanto, algumas limitações foram observadas, como a necessidade de um alto número de iterações para manter a fluidez visual. Os erros associados são principalmente por causa do tempo de computação necessário.

Além disso, como o método de autômatos celulares utiliza conceitos simples e é basicamente composto por declarações condicionais (if), ele encontra dificuldades em simular fenômenos mais complexos. Por exemplo, a simulação do gotejamento de uma gota ou a colisão de um fluido com um sólido apresenta desafios significativos. Esses eventos envolvem dinâmicas mais sutis e detalhadas que exigem um nível de precisão e realismo difícil de alcançar com regras simples baseadas em condições discretas.

O gotejamento de uma gota, por exemplo, envolve a formação, a separação e o impacto da gota, processos que são difíceis de capturar com precisão usando apenas regras locais simples. Da mesma forma, a interação de fluidos com sólidos, como a difusão de um líquido ao bater em uma superfície sólida, requer uma modelagem mais sofisticada para representar corretamente as forças de colisão, a redistribuição do líquido e os efeitos de ricochete.

Estas limitações realçam a necessidade de métodos mais avançados ou híbridos para situações que exigem maior detalhe e precisão. Técnicas como a utilização de autômatos celulares em conjunto com métodos de partículas ou simulações baseadas em física contínua podem ser exploradas para melhorar a representação de tais fenômenos. Isso permitirá uma simulação mais robusta e realista, capaz de lidar com a complexidade inerente a muitas dinâmicas de fluidos.

Conclusão

Este trabalho atingiu com sucesso o objetivo de desenvolver uma simulação interativa de fluidos utilizando autômatos celulares no MATLAB. A simulação criada fornece uma visão detalhada e intuitiva da dinâmica dos fluidos, tendo um grande valor tanto para fins educacionais quanto para pesquisas. Futuras extensões incluem a melhoria dos algoritmos e a expansão para simulações tridimensionais.

Referências

Documentação do MATLAB.

[2d-liquid-simulator-with-cellular-automaton-in-unity](#)
[Real-Time Fluid Dynamics for Games](#)
[fluid-simulation-for-dummies](#)
[simple-fluid-simulation](#)

Apêndice

Inclui dados detalhados e informações adicionais que são úteis para a compreensão do trabalho, mas que não foram incluídos no texto principal para manter a fluidez da leitura. O códigos-fonte do FluidAutomata e outro Trabalho que usa Equações de Navier-Stokes para simular os fluidos.