SUPPLEMENTAL MATERIALS: Differentiable Modeling of Material Spreading in Inkjet Printing for Appearance Prediction

EMILIANO LUCI, Max Planck Institute for Informatics, Germany
FABIO PELLACINI, Università degli Studi di Modena e Reggio Emilia, Italy
VAHID BABAEI, Max Planck Institute for Informatics, Germany

In the main paper, we use a single-step particle simulation whose main component is a velocity field that takes particles from their initial position to their final position. In order to verify this simple setup, we also implement a significantly more complex simulator based on a Lattice Boltzmann Method (LBM), which is usually employed in fluid dynamics for solving advection-diffusion problems. The result of both simulation methods are given in the main paper. Here, we describe our LBM-based particle simulation used in our framework that models the spreading of printing materials.

CCS Concepts: • **Applied computing** → **Computer-aided manufacturing**; • **Computing methodologies** → **Shape Modeling**; **Simulation**.

Additional Key Words and Phrases: 3D printing, differentiable rendering, differentiable simulation, fluid dynamics, Lattice Boltzmann Methods

ACM Reference Format:
Emiliano Luci, Fabio Pellacini, and Vahid Babaei. 2024. SUPPLEMENTAL MATERIALS: Differentiable Modeling of Material Spreading in Inkjet Printing for Appearance Prediction. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3680528.3687598

1 LATTICE BOLTZMANN METHOD FOR MATERIAL SPREADING

We assume that the spreading that we observe in our prints occurs during the fluid phase of the material before being cured, that is, made solid, by UV light. External forces are applied by the movement of the print head and the jetting process. Internally, thermal energy, cohesion, and adhesion all contribute to a never-ending jiggling. Therefore we propose to approximate this behavior using a fluid simulator, namely the Lattice Boltzmann Method.

The Lattice Boltzmann Methods are a class of computational fluid dynamics methods. These methods originate from lattice gas automata [Chen and Doolen 1998] and provide a numerical solution to the Navier-Stokes equations. LBM characterizes a set of particles, statistically, by a probability density function

$$f(\mathbf{x}, \mathbf{v}, t)$$

which defines the probability of a particle at position $\mathbf{x}$ to possess velocity $\mathbf{v}$ at time t. We additionally define the density and momentum

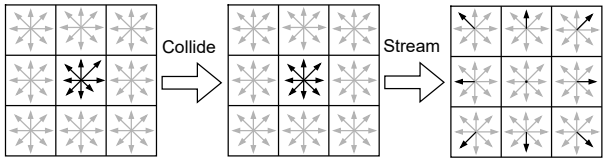

Fig. 1. The steps of the lattice Boltzmann simulation

of the fluid as

$$\rho(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t)\delta v, \tag{1}$$

$$\rho(\mathbf{x}, t)\mathbf{v} = \int \mathbf{v} f(\mathbf{x}, \mathbf{v}, t)\delta v. \tag{2}$$

In a fluid at rest, with no external forces and where inter-molecular forces can be ignored f reduces to the Boltzmann distribution. We model what happens when a force $\mathbf{F}$ is applied to a particle by Newton's law of motion:

$$f(\mathbf{x}, \mathbf{v}, t) = f(\mathbf{x} + \mathbf{v}\delta t, \mathbf{v} + \frac{\mathbf{F}}{m}\delta t, t + \delta t), \tag{3}$$

of which we can compute the derivative

$$\left(\frac{\delta}{\delta t} + \mathbf{v} \cdot \nabla_x + \frac{\mathbf{F}}{m} \cdot \nabla_v\right) f = 0. \tag{4}$$

But we would also like to account for collision between particles. We thus additionally introduce another term so that we have

$$\left(\frac{\delta}{\delta t} + \mathbf{v} \cdot \nabla_x + \frac{\mathbf{F}}{m} \cdot \nabla_v\right) f = \Omega(t), \tag{5}$$

where $\Omega(t)$ is aptly named the *collision operator* (Fig. 1). Evaluating $\nabla_v f$ is unfortunately very difficult. In the absence of external forces, however, Eq. 5 reduces to

$$\left(\frac{\delta}{\delta t} + \mathbf{v} \cdot \nabla_x\right) f = \Omega(t), \tag{6}$$

which is the material derivative of f. The new form is more treatable but since $\mathbf{x}$ and $\mathbf{v}$ are continuous it's still very hard.

This brings us to the core of LBM. This method discretizes space into a regular grid, i.e., a lattice, and at each point it assumes there is a certain number of particles. Furthermore, it also discretizes the direction and magnitude of the speeds at which particles can move such that at each time step they can only move into a neighbouring lattice point. Formally, for each lattice point we redefine the distribution function as

$$f(\mathbf{x}, e_i, t) = f_i(\mathbf{x}, t), \quad 0 \le i \le N. \tag{7}$$

Notice how the particles can only assume a finite set of velocities. These velocities are given by the choice of discretization. A popular

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12.
https://doi.org/10.1145/3680528.3687598

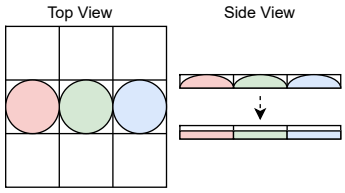

Fig. 2. Adjusting the density in 2D is a projection of a 3D effect. The material is always jetted in droplets which then spread to cover the whole area.

choice is a D2Q9 configuration where the 2 is for the dimensionality of the lattice and 9 for the possible velocities given by the 8 cardinal directions and the rest position. Now Eq. 6 becomes

$$\left(\frac{\delta}{\delta t} + \mathbf{e}_i \cdot \nabla_x\right) f = \Omega(t), \tag{8}$$

and therefore

$$f_i(\mathbf{x} + \mathbf{e}_i \delta t, t + \delta t) - f_i(\mathbf{x}, t) = \Omega_i(t). \tag{9}$$

There is an extensive literature on what $\Omega(t)$ must be. We use the Bhatnagar–Gross–Krook assumption [Bhatnagar et al. 1954] that simply states that f will eventually decay with a rate of $\frac{1}{\tau}$ toward an equilibrium distribution f^{eq}, that is

$$\Omega_i = \frac{1}{\tau}\left(f^{\text{eq}} - f_i\right), \tag{10}$$

where f^{eq} is

$$f^{\text{eq}} = \omega_i \rho \left(1 + 3\frac{\mathbf{e}_i \cdot \mathbf{u}}{c} + \frac{9}{2}\frac{(\mathbf{e}_i \cdot \mathbf{u})^2}{c^2} - \frac{3}{2}\frac{\mathbf{u} \cdot \mathbf{u}}{c^2}\right). \tag{11}$$

Here, ω_i is a term that depends on the choice of discretization, c is the speed of sound in the medium and $\mathbf{u}$ is the macroscopic velocity (vectorial sum of the discrete velocities) at each lattice point.

$$f_i = f_i - \frac{1}{\tau}\left(f^{\text{eq}} - f_i\right). \tag{12}$$

The relaxation factor τ and the number of iterations N are hyper-parameters to our simulation. The relaxation factor describes how *viscous* a fluid is while the number of iterations implicitly define how long the material should be allowed to stay fluid before they're cured. We choose $\frac{1}{\tau} = 0.03$ and $N = 100$.

It's worth noting that while there exists a mapping between the values used in the simulation and real physical values that is not of interest to us. A few key notes: fluids should be relatively slow otherwise the assumption that they can only move to neighbouring cells falls down and it's very important to consider the boundary conditions, that is, what happens at the edge of the simulations or at the interface between the fluid and an obstacle. We respect the first restriction but conveniently choose non-physical boundary condition. We can do that because we restrict our region of interest to a subset of the simulation domain.

Finally we're ready to introduce how we customize the LBM to model the material distribution. Our main assumption is that the pigments are suspended within the material and that they move with its flow. Furthermore we make the assumption that we only need to simulate one $X \times Y$ slice of material at a time, that is the previous slice is fully cured and thus particles cannot interact. We then define 3 fields

$$\boldsymbol{\nu} : \mathbb{R}^{X \times Y \times 2} \tag{13}$$

$$\boldsymbol{\rho} : \mathbb{R}^{X \times Y \times 9} \tag{14}$$

$$\boldsymbol{\mu} : \mathbb{R}^{X \times Y \times k}. \tag{15}$$

The velocity field $\boldsymbol{\nu}$ represents the initial continuous, macroscopic velocities of the particles. Since we're working in 2D the vectors are also 2-dimensional. The density field $\boldsymbol{\rho}$ represents the variable density of pigments in the print. This captures variations from the nominal amount of jetted material. As illustrated in Fig. 2, if a volume that is less than that of the parallelepipedical voxel is jetted, the material will always spread to fill the empty space but will be overall slightly smaller in height. Therefore what we're seeing is really a 2D view of a 3D effect. The density field is 9-dimensional, the same dimensionality as the discretized velocities for LBM. Finally, the mixing field represents the amount of physical mixing between the materials. In other words, $\boldsymbol{\mu}$ defines the proportion at each lattice points between the materials. If we only have 2, $k = 1$ and consequently if $\mu = 0$ or 1 it means there are only particles of one type of material. Formally we can now define our simulator function as

$$\mathcal{S} : (\boldsymbol{\nu}, \boldsymbol{\rho}, \boldsymbol{\mu}) \to (\boldsymbol{\nu}', \boldsymbol{\rho}', \boldsymbol{\mu}'). \tag{16}$$

We implement $\mathcal{S}$ using JAX [Bradbury et al. 2018], "an Autograd and XLA, brought together". We only write the code of the LBM in forward mode, and JAX performs automatic differentiation w.r.t. the inputs. Some glue code is used to manually backpropagate the gradients coming from the renderer.

REFERENCES

P. L. Bhatnagar, E. P. Gross, and M. Krook. 1954. A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.* 94 (May 1954), 511–525. Issue 3. https://doi.org/10.1103/PhysRev.94.511

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/google/jax

Shiyi Chen and Gary D. Doolen. 1998. LATTICE BOLTZMANN METHOD FOR FLUID FLOWS. *Annual Review of Fluid Mechanics* 30, 1 (1998), 329–364. https://doi.org/10.1146/annurev.fluid.30.1.329 arXiv:https://doi.org/10.1146/annurev.fluid.30.1.329