FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatik

Perceptual Losses for Deep Learning on Fluid Simulations

Hanfeng Wu

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatik

Perceptual Losses for Deep Learning on Fluid Simulations

Perceptual Losses für Deep Learning von Flüssigkeitssimulationen

Author:	Hanfeng Wu
Supervisor:	Prof. Dr. Nils Thürey
Advisor:	M.Sc. Georg Kohl
Date:	15.09.2021

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 15.09.2021 Hanfeng Wu

Acknowledgements

Abstract

This thesis studies the integration of perceptual loss into several models that are related to fluid simulation. Perceptual losses are used to compare high level differences, while traditional loss functions like MSE and MAE only compare the pixel level differences which is more brute force.

Now some famous perceptual loss functions like comparing the intermediate layers of pretrained vgg network already improve its performance in some image related tasks. [JAFF16] We want to show that in the context of fluid simulation, the integration of some pretrained perceptual losses should also outperform the traditional loss functions.

We tested some fluid simulation tasks in SOL [UBF+20] to improve the performance of the percetual loss functions. Meanwhile we also integrate such loss functions into models like autoencoder and superresolution to compare their results with those from the same models but trained by MSE loss functions

Finally we combined the result of SOL and superresolution model to conduct some fluid simulations at a lower cost to show the advantage of integrating perceptual losses in such tasks.

Zusammenfassung

Diese Dissertation untersucht die Integration von Wahrnehmungsverlusten in verschiedene Modelle, die sich auf die Fluidsimulation beziehen. Wahrnehmungsverluste werden verwendet, um Unterschiede auf hohem Niveau zu vergleichen, während herkömmliche Verlustfunktionen wie MSE und MAE nur die Pixelniveauunterschiede vergleichen, was roher ist.

Einige berühmte Wahrnehmungsverlustfunktionen wie der Vergleich der Zwischenschichten eines vortrainierten VGG-Netzwerks verbessern bereits seine Leistung bei einigen bildbezogenen Aufgaben. [JAFF16] Wir wollen zeigen, dass die Integration einiger vortrainierter Wahrnehmungsverluste im Kontext der Fluidsimulation übertreffen auch die traditionellen Verlustfunktionen.

Wir haben einige Fluidsimulationsaufgaben in SOL [UBF+20] getestet, um die Leistung der Percetual Loss Functions zu verbessern. Inzwischen integrieren wir solche Verlustfunktionen auch in Modelle wie Autoencoder und Superresolution, um deren Ergebnisse mit denen aus den gleichen Modellen zu vergleichen, die jedoch mit MSE-Verlustfunktionen trainiert wurden

Schließlich haben wir die Ergebnisse von SOL und Superauflösungsmodell kombiniert, um einige Fluidsimulationen zu geringeren Kosten durchzuführen, um den Vorteil der Integration von Wahrnehmungsverlusten in solchen Aufgaben zu zeigen. *Note: Insert the German translation of the English abstract here.*

Contents

SOL Solver in the Loop

LSIM Learning Similarity Metrics for Numerical Simulations

Chapter 1

Introductions

1.1 Fluid simulation

1.1.1 Navier Stokes Equation

The state of art Fluid simulation is based on the famous incompressible equation Navier-Stokes equation

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{1}{\rho} \nabla p = \vec{g} + \nu \nabla \cdot \nabla \vec{u} \tag{1.1}$$

$$\nabla \cdot \overrightarrow{u} = 0 \tag{1.2}$$

where ∇ denotes the gradient, $\nabla \cdot$ denotes the Divergence, $\overrightarrow{u}$ denotes the velocity of the fluid, t denotes the time step, ρ denotes the density of the fluid, p denotes the pressure, $\overrightarrow{g}$ denotes the gravity and ν denotes the viscosity of the fluid.

The equation (1.1) is actually a transformation of the $\overrightarrow{F} = m\overrightarrow{a}$ and the equation (1.2) describes the incompressibility of the fluid.

1.1.2 Grid Stucture

The way of storing the velocity and the density is based on two different grid structure. We store the dencity in the center of each cell, we call it centeredgrid or scalar grid. However for storing the velocity, we sample them at the face centers of each cell [HW65]. In such way we can more easily compute the inflow and outflow of each grid for each direction, however as a trade off, the data format would be more complex than normal centeredgrid, and we usually have to stack each verlocity array of each dimension together and also adopt them individually in some computations(e.g. computing loss function in deep learning)

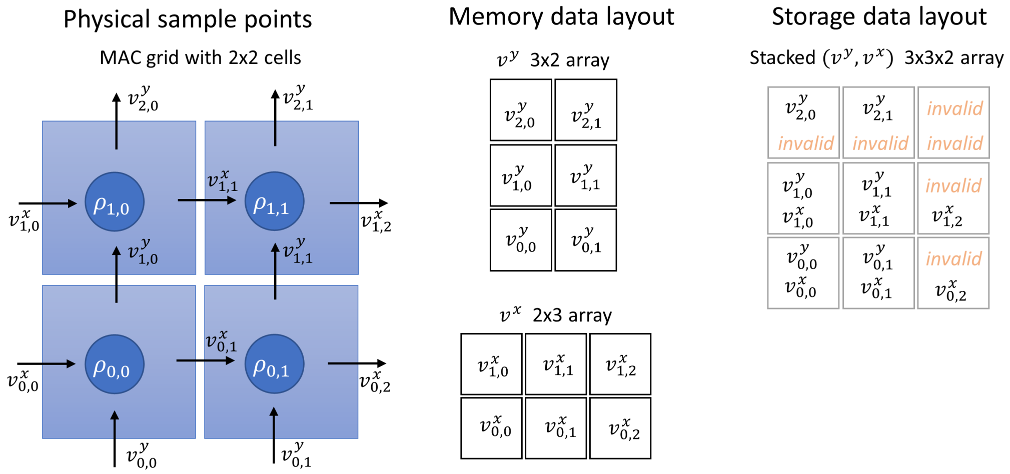

Figure 1.1: Staggered grid format

1.1.3 Advection

With the knowledge base of the NS-equation and grid structures we now need the advection algorithm to run the simulation. We adopt the semi-lagrangian advection algorithm. In the lagrangian point of view, if we want to compute the grid value of the n+1 time step of grid position $\overrightarrow{x_G}$, we should figure out which particle actually flows to the grid $\overrightarrow{x_G}$ from time step n+1. If we name the grid position of the particle as $\overrightarrow{x_P}$, then the new particle at time step n+1 in $\overrightarrow{x_G}$ should have the same physical property as the particle at time step n in $\overrightarrow{x_P}$

we first use the euler equation to calculate the $\overrightarrow{x_P}$ as a step before of $\overrightarrow{x_G}$

$$\vec{x}_P = \vec{x}_G - \Delta t \frac{d\vec{x}_G}{dt} \tag{1.3}$$

because $\frac{d\vec{x}}{dt} = \vec{u}(\vec{x})$ we have

$$\vec{x}_P = \vec{x}_G - \Delta t \vec{u}(\vec{x}_G) \tag{1.4}$$

where $\vec{u}(\vec{x}_G)$ denotes the velocity sampled at the position x_G, with the advection we can update the velocity, pressure and density of the whole vector field.

with all the knowledge above, we can formulate a sequence to conduct the basic fluid simulation.

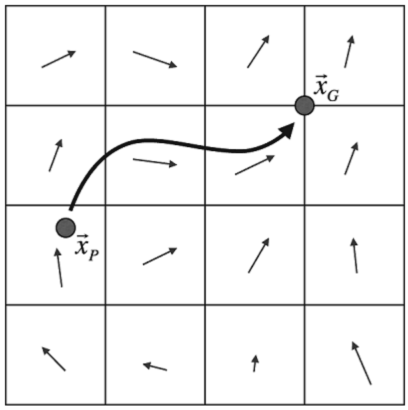

Figure 1.2: advection

- start with an initial velocity field $\overrightarrow{u}^0$ with the property $\nabla \cdot \overrightarrow{u} = 0$
- choose a time step Δt
- For time step n = 0, 1, 2, ...
  - $\overrightarrow{u}^A = advect(\overrightarrow{u}^n, \overrightarrow{u}^n, \Delta t)$
  - add the extra force$\overrightarrow{u}^B = \overrightarrow{u}^A + \Delta t \overrightarrow{g}$
  - $\overrightarrow{u}^{n+1} = make\_incompressible(\Delta t, \overrightarrow{u}^B)$

1.2 Deep learning tasks

There are also various ways of integrating deep learning model into the fluid simulation.

1.3 Perceptual loss functions

Note: Describe the problem that you like to address in your thesis to show the importance of your work. Focus on the negative symptoms of the currently available solution.

Chapter 2

Related Work

similar goal area methods

Chapter 3

Metrics and Perceptual Losses

Note: Describe each proven technology / concept shortly that is important to understand your thesis. Point out why it is interesting for your thesis. Make sure to incorporate references to important literature here.

3.1 triditional loss function

Note: This section would summarize the agile method Scrum using definitions, historical overviews and pointing out the most important aspects of Scrum.

3.2 Learning Similarity Metrics for Numerical Simulations

Note: This section would summarize the concept User Feedback using definitions, historical overviews and pointing out the most important aspects of User Feedback.

3.3 VGG-16 loss network

Note: This section would summarize the architectural style Representational State Transfer (REST) using definitions, historical overviews and pointing out the most important aspects of the architecture.

3.4 Lpips

Chapter 4

Datasets

4.1 Autoencoder

4.2 Superresolution

4.3 Solver in the Loop

Chapter 5

Tasks and Experiments Setup

5.1 Autoencoder

5.2 Superresolution

5.3 Solver in the Loop

Chapter 6

Results and Analysis

6.1 Comparison

6.1.1 Numerical evaluation

6.1.2 Time cost evaluation

6.2 Limitaions

Chapter 7

Conclusion

7.1 Future work

7.2 Summary

Appendix A

e.g. Questionnaire

Note: If you have large models, additional evaluation data like questionnaires or non summarized results, put them into the appendix.

List of Figures

List of Tables

Bibliography

[HW65] Francis H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.

[JAFF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[UBF+20] Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *Advances in Neural Information Processing Systems*, 2020.