Outline

**Block 1 (9:00 - 10:30)**
- Foundations of SPH
- Governing equations
- Time integration
- **Example:** Our first SPH solver
- Neighborhood Search

**Block 2 (11:00 - 12:30)**
- Enforcing incompressibility
  - State equation solvers
  - Implicit pressure solvers
- Boundary Handling
  - Particle-based methods
  - Implicit approaches

**Block 3 (13:30 - 15:00)**
- Multiphase fluids
- Highly-viscous fluids
- Vorticity and turbulent fluids
- **Demo:** Splash

**Block 4 (15:30 - 17:00)**
- Deformable solids
- Rigid body simulation
  - Dynamics and coupling
- Data-driven/ML techniques
- Summary and conclusion

**Coffee break (30min)**

**Lunch break (60min)**
Smoothed Particle Hydrodynamics
Techniques for the Physics Based Simulation of Fluids and Solids

Part 4
Data-driven / ML Techniques

Dan Koschier
Jan Bender
Barbara Solenthaler
Matthias Teschner
Motivation

• Substantial improvements in speed, robustness, versatility...

Incompressibility

Multi-scale simulations

• Potential of data-driven approaches?
  – PhysicsForest: Real-time SPH simulations
  – Deep Learning & Fluids: Related work and Outlook

• Computation time
• Trial & error, parameters
• Data reuse
• Edit & control simulations
• ...
Real-time prediction of fluids with Regression Forests

Ladicky et al. 2015, Apagom AG
Physics Forest

Current State \( S^n \) → \( S^{n+1} \) Next State

Simulation training data

Regression Model

Data size:
165 scenes x 6s x 30fps x 1-6M particles

Training
1) Regression method?
2) Input and output of regression?
3) Feature vector?

Physics Forest

Regression Model

Current State

Next State
Physics Forest

1) Regression method?
2) Input and output of regression?
3) Feature vector?

Regression Model

Regression Forest
[Breiman 2001]
Physics Forest

1) Regression method?
2) Input and output of regression?
3) Feature vector?

Current State

Regression Forest
[Breiman 2001]

Next State

Regression Model

Test
**Learning Strategies**

Learn velocity or acceleration?
Problem: no self-correction possible

**Naïve approach**

Learn accelerations
-> mimics standard SPH (no incompressibility)
Correction approach

Correction from Adveceted States

$S^n$ → External Forces → Advection → Collision Detection

Feature Vector → Regression → Apply Correction → Collision Detection $\rightarrow S^{n+1}$

Learn acceleration corrections
$\rightarrow$ mimics PCISPH (incompressibility)

Learn velocity corrections
$\rightarrow$ mimics PBD (incompressibility)
Feature Vector

1) Regression method?
2) Input and output of regression?
3) Feature vector?

Navier-Stokes Equations

\[ \Phi_{\text{pres}}(x_i), \Phi_{\text{visc}}(x_i), \Phi_{\text{surf}}(x_i), \Phi_{\text{comp}}(x_i) \]

Regression Forest

[Breiman 2001]

\[ \Phi = \{ F_{R0}, F_{R1}, F_{R2}, \ldots, F_{Rk} \} \]

Integral features:
Flat-kernel sums of rectangular regions around particle

- Regional forces and constraints over the set of boxes
- Fast evaluation
- Robust to small input deviations
- Evaluation in constant time (linear in number of particles)
Training Data and Performance

- Data size: 165 scenes x 6s x 30fps x 1-6M particles
- Training: 4 days on 12 CPUs
- Size of trained model: 40MB
- Only use most discriminative features (pressure, compressibility)

1-1.5M particles in real-time
Varying Material Properties

- Viscosity
- Surface Tension
- Static Friction
- Adhesion
- Drag
- Vorticity Confinement

Ladicky et al. 2015
Real-time Simulations with PhysicsForests
Related Work

- RegressionFluid: fast, but hand-crafted features -> Deep Learning (DL)
- Using DL for fluids (physics) is largely unexplored!

We evaluated the robot on 12 pouring sequences. Figure 5 shows 2 example frames from 2 different sequences and the result of both SPNets and SPNets with perception. The yellow pixels indicate where the model and ground truth agree; the blue and green where they disagree. From this it is apparent that SPNets with perception is significantly better at matching the real liquid state than SPNets without perception. We evaluate the intersection-over-union (IOU) across all frames of the 12 pouring sequences. SPNets alone (without perception) achieved an IOU of 36.1%. SPNets with perception achieved an IOU of 56.8%. These results clearly show that perception is capable of greatly improving performance even when there is significant model mismatch. Here we can see that SPNets with perception increased performance by 20%, and from the images in Figure 5 it is clear that this increase in performance is significant. This shows that our framework can be very useful for combining real perceptual input with fluid dynamics.

Conclusion & Future Work

In this paper we presented SPNets, a method for computing differentiable fluid dynamics and their interactions with rigid objects inside a deep network. To do this, we developed the ConvSP and ConvSDF layers, which allow the model to compute particle-particle and particle-rigid object interactions. We then showed how these layers can be combined with standard neural network layers to compute fluid dynamics. Our evaluation in Section 5 showed how a fully differentiable fluid model can be used to 1) learn, or identify, fluid parameters from data, 2) control liquids to accomplish a task, 3) learn a policy to control liquids, and 4) be used in combination with perception to track liquids. This is the power of model-based methods: they are widely applicable to a variety of tasks. Combining this with the adaptability of deep networks, we can enable robots to robustly reason about and manipulate liquids. Importantly, SPNets make it possible to specify liquid identification and control tasks in terms of the desired state of the liquid; the resulting controls follow from the physical interaction between the liquid and the controllable objects. This is in contrast to prior approaches to pouring liquids, for instance, where the relationships between controls and liquid states have to be specified via manually designed functions.

We believe that by combining model-based methods with deep networks for liquids, SPNets provides a powerful new tool to the roboticist’s toolbox for enabling robots to handle liquids. A possible next step for future work is to add a set of parameters to SPNets to facilitate learning a residual model between the analytical fluid model and real observed fluids, or even to learn the dynamics of different types of substances such as sand or flour. SPNets can also be used to perform more complex manipulation tasks, such as mixing multiple liquid ingredients in a bowl, online identification and prediction of liquid behavior, or using spoons to move liquids, fluids, or granular media between containers.
SPNets - Smoothed Particle Network for PBF

- PBF with a deep neural network
  -> can compute full analytical gradients (differentiable solver)
- Two new layers: ConvSP for particle-particle interactions
  ConvSDF for particle-object interaction
- Robots interacting with liquids (learning parameters, control)

Schenk & Fox 2018
Latent Space Physics – Learning Temporal Evolution

- LSTM network to predict changes of pressure field over time (3D + time) within the latent space
- Uses a history of 6 steps to infer next [1...x] steps, followed by a regular sim step
- 155x speed-up

Talk tomorrow 10:00

Examples from training data set liquid128
Resolution 128

Wiewel et al. 2019
DeepFluids: Generative Net for Parameterized Simulation

- Input parameterizable data set
- Generative network with supervised training
- Latent space time integration network
- >1300x compression, >700x speed-up, trained model 30MB
TempoGAN - Superresolution Fluids

- Infer high-resolution details
- Generator, guided during training by two discriminator networks (space and time)
- Training data: low- and high-res density pairs (density, velocity, vorticity)

$$\begin{align*}
G(x_a) &\rightarrow D_s \\
G(x_t) &\rightarrow D_t \\
G(x_{t+1}) &\rightarrow D_t
\end{align*}$$

$\text{Input}$

$\text{Result}$

---

**Related Work**

In the area of computer vision, deep learning techniques have achieved significant results, especially in tasks such as image classification and object detection. They have been particularly powerful in re-creating the distributions of complex data sets such as high-resolution images.

**TempoGAN**

To achieve super-resolution details, researchers have explored transformation tasks, such as image translations problems. GANs can be separated into unconditional and conditional. The formers generate realistic data from samples of a synthetic domain, while the latter are conditioned for generic natural images. Depending on the kind of input data they take, GANs can be separated into unconditional and conditional. The conditional GANs were introduced by Mirza and Osindero [Mirza and Osindero 2014], as well as a second network, the conditional discriminator, to generate consistent and temporally coherent images.

In some way related to the target function in order to control the training process aware of the underlying transport phenomena, such as images of human faces. Depending on the kind of input data they take, GANs can be separated into unconditional and conditional. The conditional GANs were introduced by Mirza and Osindero [Mirza and Osindero 2014], as well as a second network, the conditional discriminator, to generate consistent and temporally coherent images.

**Conclusion**

As long as the output matches the correlated spatial and temporal characteristics of the training data, we have found data augmentation crucial to avoid overfitting. As such, we not only employ this adversarial approach for the smoke simulation, but we also train a specialized, discriminative and compatible with the training process aware of the underlying transport phenomena, such that the training process can alleviate this problem. Our approach is the first generative adversarial network (GAN) that learns to judge how closely the generated output matches the ground truth data. In this way, we train a specialized, discriminative and compatible with the training process aware of the underlying transport phenomena, such that the training process can alleviate this problem.

**Capturing the Intrinsic Details of Turbulent Fluids**

For physics functions, we refer the readers to corresponding books [Bishop 2006; Goodfellow et al. 2016], and additionally, in the area of content creation. For more in-depth reviews of neural networks and deep learning techniques, we refer the readers to corresponding books [Bishop 2006; Goodfellow et al. 2016].
FlowStyle – Neural Stylization of Flows

- Transfer low- and high-level style features from images to 4D fluid data
- Structurally and temporally coherent
- Pre-trained networks on images, 3D reconstruction
Potential and Challenges of Data-driven Fluids

Unexplored area
Exciting research, triggers research and collaborations across disciplines

What is the potential of data-driven simulations?
Computational speed, data compression, novel applications: quick simulation previews, interpolation of simulations, image-based modeling and control...

Use DL as a black box?
No; synergistic combination of mathematical models and data

What are the challenges?
Loads of data (expensive, lack of data sets), training time / re-training, visual quality (memory limitations), 4D data, network architecture and parameters