Physics-Informed Style Transfer for CFD Simulations: CS 7643

Alexandra Stewart
Georgia Institute of Technology
Georgia Instiute of Technology Atlanta, GA 30332
astewart95@gatech.edu

Abstract

Neural style transfer is a powerful computer vision technique used to transfer the artistic style of an image to a new image while preserving the image's content. The process itself relies on a key assumption that the style of an image is represented by the Gram matrix of its feature vectors. Thus far most attempts at Style Transfer are purely for artistic effect in images and music and have obvious results in which the style is recognizable from the image, displaying the accuracy of results. However, the notion of learned "style" represented by a learned gram matrix containing only arbitrary artistic information seems overly simplistic. In this paper, we propose a potential physical meaning of the use of the Gram Matrix for style transfer which extends beyond the notion of texture, shape, and color in images to the relationships between feature vectors in simulations of physical systems. We incorporate domain knowledge of the use of CNNs in science applications with a focus on computational fluid dynamics in the lagrangian regime and suggest that the learned feature-vector Gram matrix may have physical significance in other domain applications. We propose future areas of study for style transfer extending beyond the realm of images and art into the relationship between learned feature vectors for physically-relevent data.

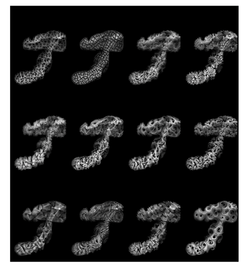

Figure 1. *Prior work of Byungsoo Kim et. al. on Style Transfer for smoke simulation for artistic effect. [5]*

1. Introduction/Background/Motivation

Neural style transfer is a powerful computer vision technique used to transfer the artistic texture or style of an image to a new image while preserving the content of the new image like symbols, objects, or shapes. The process itself relies on a key assumption which is often taken for granted: the style of an image is represented by the Gram matrix of its features, which is typically extracted from pre-trained CNNs (e.g., VGG-19). Generally most attempts at Style Transfer using an approximate or learned Gram matrix constructed from a fixed kernel are purely for artistic effect due to the somewhat vague notions of what constitutes "style" and "content" in the context of images.[2] However, in other domain areas, the Gram Matrix has physical significance and is used to compute linear independence, approximate or solve field equations, compute the exterior product of vectors, fast computations of PCA or other linear algebra applications. The notion of learned "style" represented by a gram matrix of the feature vectors containing only arbitrary artistic information seems overly simplistic. In this paper, we propose a potential physical meaning of the use of the Gram Matrix for style transfer which extends beyond the notion of texture, shape, and color in images to simulations of physical systems. We incorporate domain knowledge over the use of CNNs for physical systems with a focus on computational fluid dynamics in the Lagrangian frame and show that a learned gram matrix used in style transfer for images may have physical significance in the context of physical simulations. Finally, we propose future applications and areas of study for style transfer extending beyond the realm of images and art into the relationship between learned feature vectors.

1.1. Style Transfer: An Overview

Convolutional feature maps are generally a good representation of input image's features. They capture spatial information of an image without containing the style information given that a feature map is used as it is. Using feature maps of early convolutional layers represent the content much better, as they are closer to the input. They capture spatial information of an image without containing the

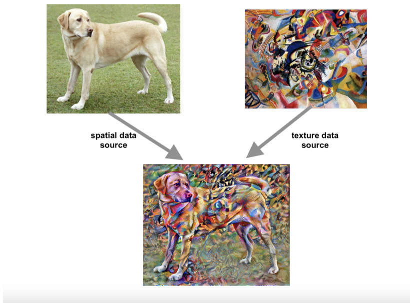

Figure 2. *Example of Image Style Transfer.*

style information. To preserve these features in the Content image, we keep the backbone weights fixed during back-propagation.

Starting from the network's input layer, the first few layer activations represent low-level features like textures, colors, and edges. Proceeding through the network, latter layers represent higher-level features like shapes, and recognizable large scale feature artifacts like animals, faces, or objects. We use the VGG19 network architecture, a pre-trained image classification network and extract intermediate layers to define the representation of content and style from particle data. Our goal is to show that physical meaning in these intermediate layers can result from preserving notions of spatial "nearness" in the input dataset. [2]

1.2. Importance of the Gram Matrix in Style Transfer

Consider two flattened feature vectors from a convolutional feature map of depth C which represent features of the input space. The dot product of these two vectors gives us the relation between them such that a small product indicates that the features represented by these vectors rarely co-occur. Conversely, the greater it is, the more the features represented by these two vectors occur together. The process removes all spatial information inherently by flattening. In summary, the Gram Matrix tells us about an image's texture and zero information about its spatial structure.

1.3. CNNs Beyond the Image: Lagrangian Fluid Flow, Particle Simulations, and Physics-Informed Loss Functions

The use of CNNs in Computational Fluid Dynamics (CFD) revolutionized the field in the past decade. Much speedup was achieved using CNNs for CFD using a bare-bones, data driven approach which was further advanced by physics-informed approaches [4],[1], [3],[7]. In particular, physically informed approaches to the loss function for CNNs using solutions to the Navier-Stokes Equation as the standard loss function rather than high-resolution images from simulations produced a considerable speedup. Finite-Differencing (FD) approaches to solutions to the Navier-Stokes equation are well-documented and provide enough accuracy to correctly train a Deep Neural Net to approximate solutions. Physics-Informed loss functions provide a great opportunity for speedup due to the low amount of training data required. [3]

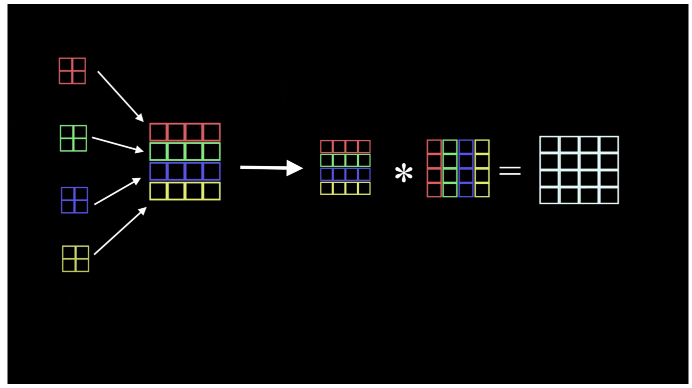

Figure 3. *Simplified visual representation of a Gram matrix. First feature vectors are flattened and placed into a matrix which is then multiplied by its transpose to result in the dot products of each feature vector with itself. For early feature vectors in the convolution network, these correspond to style features like color, texture, or shape. Source:Why Gram Matrix in Stlye Transfer*

One challenge in the area of physically-informed systems is testing the accuracy of the results. The learned behavior of the Neural Net still contains some uncertainty in its ability to product accurate results from optimization, and thus will never be as reliable as direct computed solutions. However, understanding the physical meaning behind the resulting features of CNNs and which are identified in CFD simulations proves to have significant meaning in the absence of exact solutions. [1] [3]

1.4. Gram Matrix and Physical Applications

A key principle in the process of Style Transfer is the separation of location-dependent and location-independent effects through the use of convolutional layers and flattening. The concept of "Style" can extend to simply location-independent relationships in spatially-related data. [2] Despite the fact that physically-informed loss functions have been studied in a wide variety of contexts, the idea of style transfer as a mechanism for learning small-order physical effects has been given much less attention. Fixing large-order effects, the relationships between feature vectors as represented by a gram matrix in the context of "Style" offers insight into purely location-independent features in the data.

1.5. Lagrangian Flow Particle Data

The dataset contains fluid dynamics simulations of a fluid block flowing inside a unit cube domain and time steps were computed using the Smoothed Particle Hydrodynam-

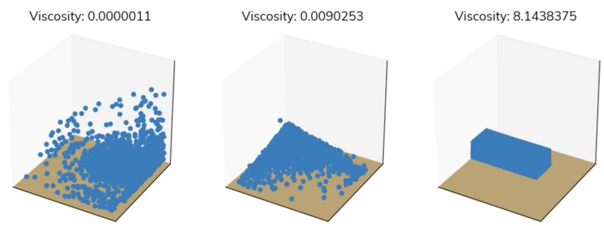

Figure 4. *Particle positions at a single time step of the SPH simulation with varying viscosity.*

ics (SPH) formulation which discretizes a fluid as a group of points in space, referred to as particles. [6] For more information about SPH formulation for calculating particle motion, see section 4.1 of the Appendix.

2. Approach

2.1. Style Transfer

2.1.1 Intermediate Layer Extraction for Style and Content

Intermediate outputs within our pre-trained image classification network allow us to define style and content representations. The networks in `tf.keras.applications` are designed so you can easily extract the intermediate layer values using the Keras functional API. To extract the intermediate layers for analysis, the vgglayers function builds a vgg19 model that returns a list of intermediate layer outputs.

2.1.2 Calculating Style & Gradient Descent

A Gram matrix is calculated from intermediate layer data that includes information represented by the values of the intermediate feature maps by taking the outer product of the feature vector with itself at each location and averaging that outer product over all locations. The Gram matrix for a particular layer l can be calculated as:

$$G_{cd}^{l} = \frac{\sum_{ij} F_{ijc}^{l}(x) F_{ijd}^{l}(x)}{IJ} \quad (1)$$

Finally, the mean square error of the image's output relative to each target can be used in gradient descent. Partially fixed weights at the later convolutional layers in the Content image preserve the content of the image while the weighted sum of the losses is used to change the Gram Matrix results for low-level features.

2.2. Dataset

2.2.1 Data pre-processing, Image or Graph?

To preserve a physical notion of spatial locality in the particle data, SPH particle position data was projected into one dimension with a single particle position over time placed as row data and nearby particles placed in nearby columns. This preserves some meaning to the spatial locality within the data, although the relationship between nearby points is not uniform as in an image. Nearby image information simply indicates a similarity in that region of particle starting position and time similarity. Thus visual artifacts in the data represent trends over time along the x-axis, and similar starting positions in the particles along the y-axis rather then the more traditional notions of spatial locality in an image.

2.2.2 Cube Flow Dataset

The dataset was obtained through publicly available, prerun Lagrangian SPH fluid simulations. For each simulation, the fluid block is set with different initial conditions like shape, position, velocity, and fluid viscosity. The results of each simulation are stored as a .npy array file with 4 dimensions inside a folder with the respective simulation ID, available for public download here Cube Flow Dataset [6].

3. Experiments and Results

To show recognizable patterns in the style/content images of the SPH data, we show the resulting style and context images after image pre-processing in Figure 5. In the content image, the fluid has relatively low viscosity while the Style image contains a high viscosity fluid with two velocity regimes. Figure 6 shows a visual Representation of Content/Style results where the R/G/B colors correspond to projected particle velocity and x and y position. Placing a lower weight on the style coefficient results in less abrupt changes in particle velocity and position over time, where time moves down the y-axis and nearby particle values correspond to the adjacent columns.

There are many opportunities to expand this work, which is simply opening the door to the possibility of a physical intuition of Style Transfer for simulation data. Fine-grained SPH simulations were used such that the particles have only undergone a few seconds of real-time motion, thus the starting point of the particles is sufficient metric along the x-axis, preserving some spatial relevance. For longer simulations, this method could pose problems due to the particles undergoing long-range motion and a better pre-trained model would be needed to capture the large-scale spatial features of the data when applying Style Transfer to extract small-order feature vectors. More work is needed on interpreting the results of the resulting Style-modified Context

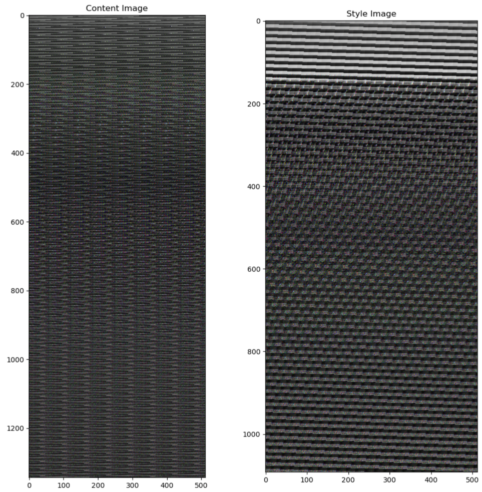

Figure 5. *Example of two different Style and Content images containing fluid transfer data. In the content image, the fluid has relatively low viscosity while the Style image contains a high viscosity fluid with two velocity regimes.*

image. The resulting dataset corresponds to a resulting simulation with modified fluid motion. In future work, movies could be produced comparing the style-modified fluid results to the original results.

4. Appendix

4.1. Standard SPH Formulation

SPH simulations involve several key steps:

1. **Particle Representation**: The core is divided into a large number of discrete particles, often referred to as "fluid particles." These particles represent small portions of the mass of the core material and are assigned physical properties based on the initial conditions.

2. **Initial Conditions**: The simulation starts with either a geometric distribution or arandom initialization of fluid particles and relaxes to an equilibrium. Researchers define the initial point properties such as density, temperature, composition, and initial velocity distribution.

3. **Equations of Motion**: The fundamental equations governing the behavior of the fluid (e.g., the Navier-Stokes Equation) are discretized and solved for each particle within the simulation. The interactions between particles are computed using kernel functions that account for physical properties like pressure, density, and viscosity.

4. **Time Integration**: SPH simulations evolve over time, advancing the system's state each time step. This integration process updates particle positions and velocities, taking into account the forces acting on them.

5. **Boundary Conditions**: Researchers impose boundary conditions to reflect the physical constraints of the system such as spacial boundaries and relational constraints. In our case, the motion of the SPH particles was contained to a unit cube.

Delving into the math and physics of SPH, we discover the following equations:

4.1.1 Smooth-Particle Hydrodynamics Density Estimation

The density ρ_i at the position of particle i is estimated using a kernel function W:

$$\rho_i = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h), \tag{2}$$

where:

$$\begin{aligned}
\rho_i &: \text{Density at particle } i, \\
m_j &: \text{Mass of particle } j, \\
\mathbf{r}_i &: \text{Position of particle } i, \\
\mathbf{r}_j &: \text{Position of particle } j, \\
h &: \text{Smoothing length}, \\
W(\mathbf{r}, h) &: \text{Kernel function.}
\end{aligned}$$

5. Work Division

There is no work division section of this paper due to the paper having one sole author/contributor.

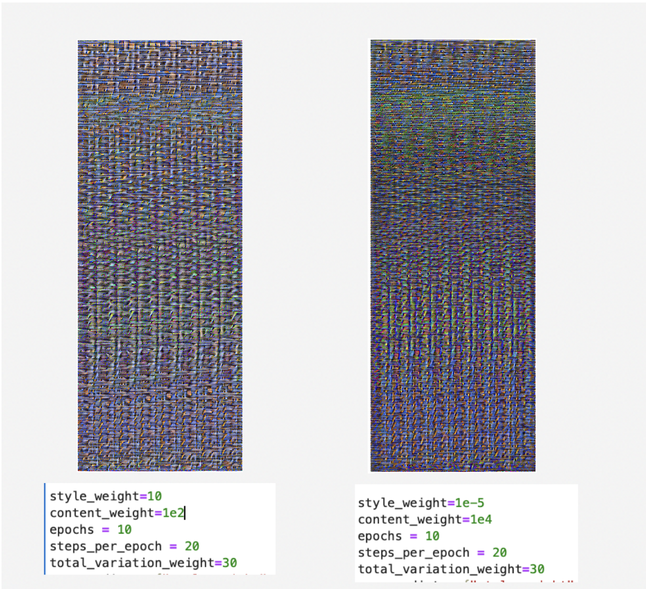

Figure 6. *A Visual Representation of Content/Style Results.* A visual Representation of Content/Style Results where the R/G/B colors correspond to projected particle velocity and x and y position. Placing a lower weight on the style coefficient results in less abrupt changes in particle velocity and position over time, where time moves down the y-axis and nearby particle values correspond to the adjacent columns.

References

[1] Jun Chen, Hui Duan, and Ming Tang. CNN-Based Fluid Motion Estimation Using Correlation Coefficient and Multiscale Cost Volume — mdpi.com. https://www.mdpi.com/2079-9292/11/24/4159. [Accessed 06-08-2024]. 2

[2] et. al. Gatys, Leon A. Texture synthesis using convolutional neural networks. https://arxiv.org/pdf/1505.07376, 2022. [Accessed 06-08-2024]. 1, 2

[3] Viktor Hermann Grimm. Physics-aware convolutional neural networks for computational fluid dynamics, Aug. 10 July 2023. 2

[4] Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. Lagrangian neural style transfer for fluids. *ACM Transactions on Graphics*, 39(4), Aug. 2020. 2

[5] Byungsoo Kim, Vinicius C. Azevedo, Markus H. Gross, and Barbara Solenthaler. Transport-based neural style transfer for smoke simulations. *CoRR*, abs/1905.07442, 2019. 1

[6] Maria-Castro. Fluid cube dataset (0.1), Apr 2023. 3

[7] Medium. Deep learning and fluid dynamics: Solving the Navier-Stokes equations with an optical neural... — medium.com. https://medium.com/optalysys/deep-learning-and-fluid-dynamics-solving-the-na 2023. [Accessed 06-08-2024]. 2