Rethinking Style Transfer: From Pixels to Parameterized Brushstrokes

Dmytro Kotovenko Matthias Wright Arthur Heimbrecht Bjorn Ommer

Presenters:

Sohan Patnaik and Kishan Pandey

Reading Session XIII

Kharagpur Data Analytics Group and AI Reading Group
IIT Kharagpur

October 31, 2021

Important Notes

- We expect you all to have certain queries regarding the presentation, certain intricate doubts maybe... **Please put them in the chat of this meeting (if you feel shy) else unmute and speak**.
- Finally, a **Google Form** [Link] will be released for feedback but most importantly, we ask you to put up name of any AI-related paper to be presented in the upcoming sessions!

Link to GitHub Repo : Click Here
Link to join Slack Workspace : Click Here
Link to KDAG YouTube Channel : Click Here
Link to doc on good AI Blogs/Resources/Topics : Click Here

Outline

- Introduction
- Background
  - Content Loss
  - Style Loss
- Approach
  - Implementation Details
  - Differentiable renderer
  - Idea - Basic Case Scenario
  - Renderer
  - Controlling flow of brushstrokes with user input
- Experiments
  - Deception Rate
  - Differentiable Renderer
- Conclusion

Introduction

- Style and texture transfer have been around for a while now! And some very popular.
- Contributions from Gatys are marked outstanding in these areas including the work of synthesis of an image with combination of style and content of seperate images.
- Regularization on final image and intermediate latent representation is confined to pixel domain.
- The paper argues that the pixel representation is unnatural for the task of artistic style transfer:
    - artists compose their paintings with brushstrokes, not with individual pixels.
    - So the paper explores the representation with parameterized brush strokes instead of pixels

Pixels

Parametrized
Brush-strokes

Background

- The original style transfer paradigm was proposed by Leon A. Gatys.
- The approach revolved around the joint minimisation of content and style losses.

- **Content Loss**: Euclidean distance between rendered image and content image.

$$\mathcal{L}_{\text{content}} = ||\phi_l(I_r) - \phi_l(I_c)||_2,$$

- **Style Loss**: The style loss is defined as follows:

$$\mathcal{L}_{\text{style}} = \sum_{l=0}^{L} w_l E_l$$

- Where,

$$E_l = \frac{1}{N_l^2 M_l^2} ||G_r^l - G_s^l||_F$$

- G is the Gram matrix capturing correlations between the feature spaces and Nl is the number of feature maps and, Ml is width times height of the feature map.,

Approach: Implementation Details

- **VGG Nomenclature:** VGG network has block of convolutional layers, where each block is given cardinality starting from 1. Further, there are several convolutional layers inside each block with similar cardinality pattern. So, each layer is uniquely represented as conv1_1, conv1_2, etc.

- **For content loss** - conv4_2, conv5_2 output feature maps were used.
- **For style loss** - conv1_1, conv2_1, conv3_1, conv4_1, and conv5_1 output feature maps were used.

- **Adam** optimizer was used with a learning rate of 0.1.

Approach: Differentiable Renderer

- A dataset of brushstrokes were generated using FluidPaint environment.
- StyleGAN network was trained to generate an image conditioned on the brushtroke parameters.
- Disadvantage - Memory intensive approach, large number of brushstrokes could be run in parallel.
- A differentiable function was constructed which transforms a collection of brushstrokes parameterized by location, shape, width and color into pixel values on a canvas.
- Formally, the renderer is function mapping as

$$\mathcal{R} : \mathbb{R}^{N \times F} \rightarrow \mathbb{R}^{H \times W \times 3}$$

Approach: Idea - Basic case scenario

- Consider a flat disk parameterized with color, radius, and location.
- Obtain a binary mask, assign 1 to those points lying inside the disc and 0 to points lying outside the disc.
- Multiply the mask with a colour value to incorporate colour.
- For two discs, the region lying outside the overlap between the discs has straight forward computation of pixel values i.e., the matrices can simply be added..
- The case is a bit complex for overlapping region.

Approach: Idea – Basic case scenario

- For overlapping region, assign each pixel to the nearest disc.
  - Can be done by computing distance between each pixel and the disc centers.

- Mathematically,

$$A := \{1 \text{ if } D_1 \leq D_2,\ 0 \text{ otherwise}\}$$

- Generalising to n discs,

$$A(i,j,n) := \begin{cases} 1 & \text{if } D_n(i,j) < D_k(i,j)\ \forall k \neq n, \\ 0 & \text{otherwise.} \end{cases}$$

- Once assignment matrix is obtained, the final image can be computed as follows:

$$I(i,j) := \sum_{n=1}^{N} I_n(i,j) * A(i,j,n)$$

- Note: Final image is computed by weighted sum of renderings, weighted by the assignment matrix

Approach: Renderer

- Compute a matrix (Db) of distances from every point in a 2D image to the nearest point on the Bezier curve.
- Mask points that are closer than the brushstroke width and multiply them by a color value.
- Approximation of distance from a point p to a Bezier curve was calculated by sampling S equidistant points p1', p2', p3, ..., ps' along the curve. The minimum pairwise distance between between p and p1', p2', p3', ..., ps' was assigned as the approximate distance.
- Compute individual rendering of brushstrokes and the assignment matrix from the equation as stated in the previous slide..

Approach: Renderer

- Oops! Assignment matrix and masking operation are discontinuous! To solve this problem they implement the masking operation with a SIGMOID function.

- To make the assignment matrix computation differentiable they computed the assignment values with a Softmax operation with high temperature.

Approach: Controlling Flow of Brushstrokes with User Input

- **Previous approach**: Bezier Curve control points were randomly assigned.
- A user can draw arbitrary curves on the content image through user interface.
- Each curve drawn was represented as a set of points P1, P2, ... Pn. (don't confuse with the control points of bezier curve)
- For Pi, the approximate tangent vector is given as (Q was set 3 in all experiments)

$$\mathbf{v}_i = \frac{\tilde{\mathbf{v}}_i}{||\tilde{\mathbf{v}}_i||}, \quad \tilde{\mathbf{v}}_i = \left(\frac{1}{Q} \sum_{j=1}^{Q} P_{i+j} \right) - P_i$$

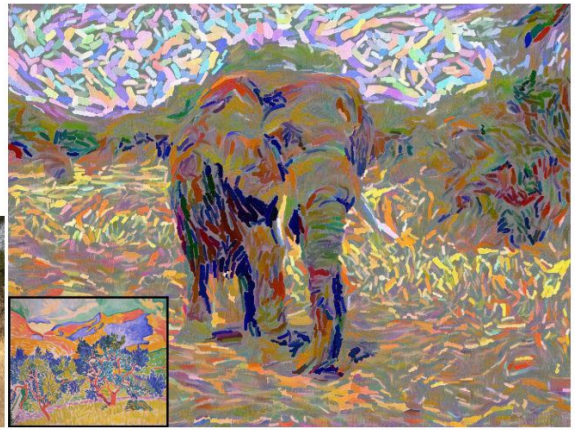
Approach: Controlling Flow of Brushstrokes with User Input

- A new loss called projection loss was incorporated along with the style and content losses.

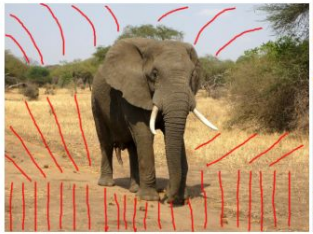
- Projection loss was computed in the following steps:
  - Compute for each brushtroke the vector from start to end point of the bezier curve.
  - For each tangent vector, compute L-nearest brushstrokes on the canvas.
  - For each tangent vector, compute the projection of the orientation vectors from the nearest brushstrokes onto the tangent vector
  - The projection loss encourages the absolute value of these projections to be 1. (absolute value will be 1 if the vectors are aligned)

Approach: Controlling Flow of Brushstrokes with User Input

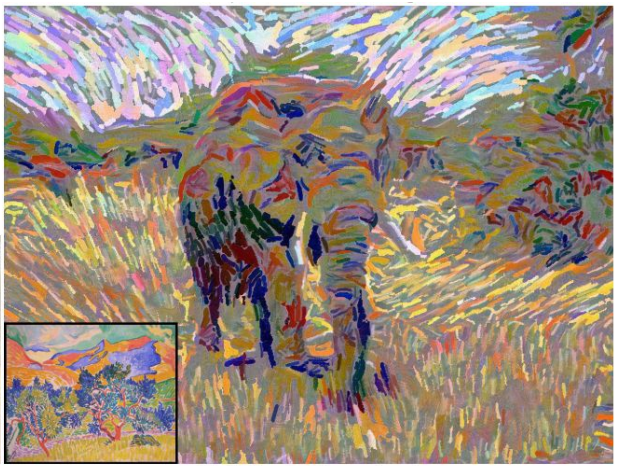
(a) Content



(b) Stylized without user input



(d) Content with user input



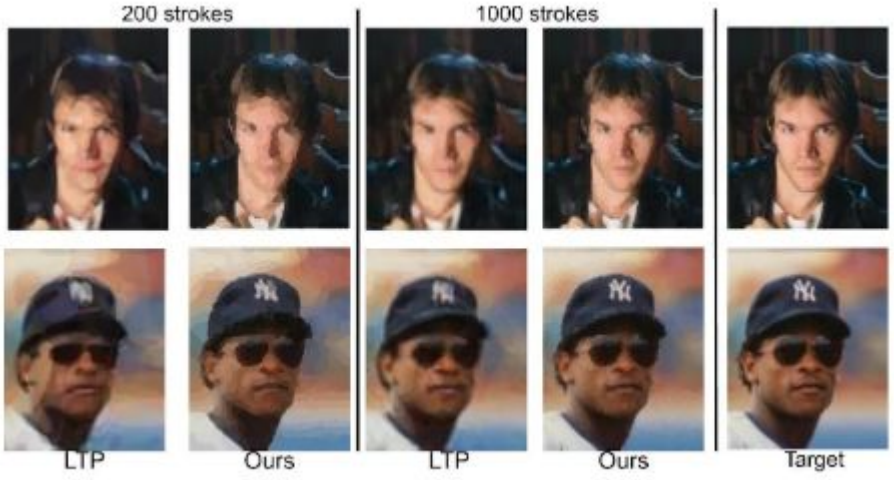
(e) Stylized with user input

Experiments: Deception Rate

- The deception rate is the fraction of stylized images that the network has assigned to the artist,
- A high deception score indicates high similarity to the target image.
- A human subject was shown 4 crop outs where he/she was asked to assign the cropout to real artwork or generated art.

Method	Mean deception score ↑	Mean human deception rate ↑
AdaIN [21]	0.08	0.035
WCT [35]	0.11	0.091
Gatys et al. [13]	0.389	0.139
AST [47]	0.451	0.146
Ours	**0.588**	**0.268**
Wikiart test	0.687	-
Photos	0.002	-

Experiments: Differentiable Renderer



- This approach achieved 20% lower mean squared error for 200 strokes and 49% lower mean square error for 1000 strokes.
- The image reconstruction was done by minimising the l2 distance between input target image and rendered image.

Conclusion

- Switch the representation of style transfer from pixels to parameterized brush strokes
  Benefits: Improved visual quality of the stylisation and enabling additional control.

- An explicit rendering mechanism solves the purpose and goes far and beyond the field of style transfer.

- Author proposes to incorporate the limitations of this approach by potentially alleviating the more sophisticated brush stroke blending procedures.

Thank You