

Style Transfer By Texture Synthesis

By: Michael Elad *

March 19, 2017

This work was done during the summer 2016 in Google-Research Mountain-View, CA

* Joint work with



Peyman Milanfar

Terminology

Content Image





Style-Transfer Algorithm Result Image



Style Image



Is the Goal Clear ?

This task is NOT well-defined, and different interpretation of its goals lead to qualitatively different results





More Specifically:

- □ Which parts of the content should be preserved and how?
- □ Are we allowed local contrast changes as part of the transfer?
- □ Should elements in the content be allowed to shift?
- □ Which color palette should the output adopt?
- □ How far can the hallucination be allowed to go?
- □ Which parts from the style-image qualify as style to be used?
- □ Where do we draw the line between copying of style and hallucination?
- □ What constitutes a successful style-transfer result?



Spoiler

Here is what we get in our work:



Content

Style

- 4 -

Spoiler

Here is what we get in our work:



Content

Style

Result



This Talk



Part 1

CNN Based Methods The Core Ideas

[Presented for the Completeness of the Story]

CNN Based Style-Transfer [Gatys, Ecker, & Bethge, 2016]

The core ideas in this work are

Use an image-recognition pre-trained Convolutional Neural Network (CNN) as a feature extractor (VGG-16).

- □ Form the style-transfer problem as an energy minimization, where the final result should be
 - Close in the feature-domain to the content image, and
 - Close in the correlation-domain to the style one.

 $E_{\text{Total}}(X) = \alpha \|VGG_k\{X\} - VGG_k\{C\}\|_2^2$ $+\beta \|Corr[VGG_j\{X\}] - Corr[VGG_j\{S\}]\|_2^2$

The results are beautiful, leading to the renewed interest in styletransfer.

Gatys' Results



Gatys et. al. [2016]

The Concept of CNN Inversion

aken from Gatys' 2016 Paper



Observation: The upper path ||codescribescarCodescribescarCodescribescarCodescribescarCodescribescarCodescribes]||² based texturesynthesis algorithm

We will come back to this later on

Gatys' Style-Transfer Algorithm



Taken from Gatys' 2016 Paper

11 -

Gatys' Style-Transfer Algorithm

Observation: This method poses a highly demanding optimization procedure that runs back and forth over the VGG – this is the main shortcoming of this work.

 $\vec{a} =$

input



Taken from Gatys' 2016 Paper

- 12 -

One trivial approach: Create many triplets of content-, style-, and result-images, and train a CNN to accomplish the same style-transfer task → Tough.

 $E_{\text{Direct}}(\theta) = \|\mathbf{T}_{\theta}\{\mathsf{C},\mathsf{S}\} - \mathbf{Gatys}\{\mathsf{C},\mathsf{S}\}\|_{2}^{2}$

 □ A simplification: Fix the style image and train the CNN to match the input content image to the output stylized result → Still Tough.

 $\mathbf{E}_{\mathbf{S}}(\mathbf{\theta}) = \|\mathbf{T}_{\mathbf{\theta}}\{\mathbf{C}\} - \mathbf{Gatys}\{\mathbf{C}, \mathbf{S}\}\|_{2}^{2}$

Improving this Algorithm

Improved idea: Instead of the above, use the VGG itself + the energy minimized by [Gatys, Ecker, & Bethge, 2016] to perform the training:

 $E_{\text{Total}}(X) = \alpha \|VGG_k\{X\} - VGG_k\{C\}\|_2^2$ $+\beta \|Corr[VGG_j\{X\}] - Corr[VGG_j\{S\}]\|_2^2$

Observation:

Note that this way we have moved from Per-Pixel to Perceptual loss

 $E_{S}(X) = \alpha \|VGG_{k}\{T_{\theta}\{C\}\} - VGG_{k}\{C\}\|_{2}^{2} + \beta \|Corr[VGG_{j}\{T_{\theta}\{C\}\}] - Corr[VGG_{j}\{S\}]\|_{2}^{2}$

This is the idea behind the following: [Johnson, Alahi, & Li, 2016] and [Ulyanov, Lebedev, Vedaldi, &Lempitsky 2016].

A Closer Look



Comments:

1. The above has been suggested also for single-image super-resolution.

2. The trained "Transform Net" is restricted to one chosen style image.

Part 2

Texture Synthesis Methods and Their Relevance

Texture Synthesis as a Path to the Transfer

- Style-transfer via CNN by Gatys is a generalization of an earlier work that targeted texture-synthesis [Gatys, Ecker & Bethge, 2015]
- □ The idea was to minimize: $\left\| \operatorname{Corr} [\operatorname{VGG}_{j} \{X\}] - \right\|_{2}^{2}$ $\operatorname{Corr} [\operatorname{VGG}_{j} \{S\}] \right\|_{2}^{2}$
- We see here a path of generalizing a texture-synthesis method to treat the style-transfer.



So, What is Texture Synthesis?

- Definition: Texture Synthesis is the process of synthesizing a new texture image from a given texture sample image, while avoiding a direct copy.
- Texture synthesis can be done in various ways (indeed, CNN is a new comer to this game).
- The shown results are taken from [Kwatra, Essa, Bobick,& Kwatra, 2005], based on multi-scale patchmatching and fusion.



Note: No one says that these algorithms are restricted to texture only

- 18 -

How Texture-Synthesis is Done?

- There are many existing algorithms out there, and most share the same ingredients:
 - Multi-resolution treatment
 - Patch-matching (in some feature domain)
 - Deployment of Approximate Nearest Neighbor (ANN)
 - Drawing pixels at random from a conditional distribution
 - Fusing patches by Belief-Propagation, or other methods.
- Key work in this field: [Efros & Leung 1999] [Efros & Freeman 2001]
 [Wei & Levoy 2000] [Liang, Liu, Xu, Guo, & Shum, 2001]
 [Kwatra, Essa, Bobick, & Kwatra 2005] [Lefebvre & Hoppe 2006]
- Variants over these: texturizing 3D shapes, texture-synthesis on surfaces, parametric models for textures, …

Underlying Rationale of Our Work

□ An underlying assumption behind our work is:

Style-Transfer can be obtained by taking any good performing Texture Synthesis algorithm and modifying it to take the content image into account



Our plan: Taking Kwatra's Texture Synthesis algorithm [2005] and modifying it to meet the styletransfer goal

Desired Property: Style-transfer applied with an empty content image should reduce to plain texture-synthesis.

Background: Lets Go Back in Time ...

- Ideas similar to Style-Transfer appeared in the early 2000's, although in a less daring way (and with less impressive results).
- Those came as a side-results (anecdotal) to texture-synthesis work.
- One of the earliest is [Efros and Freeman, 2001]
 "quilting" patches for texture-synthesis.
- As a by-product to the above, they offered a Texture-Transfer process, by augmenting the synthesis by a matching of the local shades of the content.





Background: "Analogies"

- ❑ Another closely related idea was the concept of "analogies": Given a pair of input image and stylized form, apply the stylization on a new image [Hertzmann, Jacobs, Oliver, Curless, & Salesin, 2001].
- The core algorithm is reminiscent of "classic" texture-synthesis methods, using motives such as pyramidal decomposition, patchmatching, approximate nearest neighbor, ...



- 22 -

Texture-transfer is achieved as a special case of the algorithm.

Kwatra's Texture-Synthesis in a Nutshell

□ Core engine: Matching patches from the given texture S to the image X being built, iteratively minimizing an energy functional:

$$\min_{\mathbf{X}} \mathbf{E}(\mathbf{X}) = \min_{\mathbf{X}} \sum_{ij \in \Omega} \min_{kl} \|\mathbf{R}_{ij}\mathbf{X} - \mathbf{Q}_{kl}\mathbf{S}\|_{2}^{2}$$

- Assume that at the t-th iteration we have the image X_t. The algorithm consists of two steps that remind of the EM Algorithm:
 - Patch Matching: for each patch in location [i,j] in X_t, find the NN patch from the style. Perform this over a sampled grid covering X.
 - Aggregation: Given all the matched patches, merge them to one image X_{t+1} by averaging them.

Kwatra's Texture-Synthesis In a Nutshell

□ As said, the above process is only the core engine of the algorithm.

□ On top of this, the following key ideas are used:

- Operating in several scales over a Gaussian Pyramid of the desired image, generate from coarse to fine.
- Within each resolution level, using patches of varying sizes from large to small.
- Initializing the algorithm with random assignment of patches.
- Performing Nearest-Neighbor searech with the tree K-Means.
- Replacing plain averaging by a robust-statistics one.

Kwatra's Texture-Synthesis In a Nutshell

As said, the above process is only the core engine of the algorithm.

On top of this, the f

- Derating in seve
 c rate f m co
 V 30
 mitializing , e alg
- Performing Neare
- Replacing plain a

This algorithm produces high-quality texturesynthesis results and thus can be relied upon for being migrated to the style-transfer task.

As we shall see hereafter, we adopt all these ingredients in our algorithm, and accompany them with the necessary modifications to get the desired transfer.

Frigo's Style-Transfer Algorithm [Frigo, Sabater, Delon & Hellier, 2016]

- This work proposes a novel style-transfer method based on texture-synthesis, leading to much faster algorithm (compared to Gatys).
- □ It relies on patch-matching and multi-scale treatment ideas, but it has a very different reasoning, which leads to a substantially different algorithm.
- □ The algorithm is best explained by this figure, which exposes the core essence of their work: A quad-Tree partition of the image:





Example



Adaptive partition



Stylization with our method

Frigo's Method: More Details

A given patch is matched to the Style Image, to find its nearest-neighbor. If the distance is small enough AND the inner variance of the patch is small, stop the division. Otherwise, divide to four and proceed further down the tree.

 At the end of the above process, every patch has several potential neighbors from the style image. A Belief Propagation (BP) is ran in order to fit all these patches into one coherent image.
 Overlaps are simply averaged.



Part 3

The Proposed Scheme

Our Algorithm: Core Features

- Being a follow-up to Kwatra's method, our algorithm contains most of the ingredients mentioned before (patch-matching, multi-res., multi-size patches, robust and iterative energy minimization, ...).
- Moreover when our algorithm is ran over a flat (no content) image, it reduced to a plain texture-synthesis process.
- On the above foundations, we introduce several key modifications that turn the process into a style-transfer algorithm.
- Comment: Our algorithm works in the RGB domain, and thus 3D patches are manipulated in it.

Our Algorithm: Objective Function(s)

Patch-matching quality Match to the Goodmeasured w.r.t. the content in quality Style image selected regions image $E_{L,n}(X) = \frac{1}{c} \sum_{kl} \min_{kl} \|\mathbf{R}_{ij}X - \mathbf{Q}_{kl}S\|_{2}^{2} + \|X - C\|_{W}^{2} + \lambda \rho(X)$ ij∈Ω There is one such energy Obtained by segmentation function for each image' resolution L (over the pyramid)

and the chosen patch size, n.

Overall Process



The Inner Minimization

$$E_{L,n}(\mathbf{X}) = \frac{1}{c} \sum_{ij \in \Omega} \min_{kl} \left\| \mathbf{R}_{ij} \mathbf{X} - \mathbf{Q}_{kl} \mathbf{S} \right\|_{2}^{r} + \left\| \mathbf{X} - \mathbf{C} \right\|_{\mathbf{W}}^{2} + \lambda \rho(\mathbf{X})$$

□ We address the above minimization task using:

- ADMM [Boyd, Parikh, Chu, Peleato, & Eckstein, 2011] and
- The Plug-and-Play-Prior approach

[Venkatakrishnan, Bouman, & Wohlberg, 2013]

- □ This approach essentially decouples the above into a series of updates, per each term separately.
- An interesting feature no need for an explicit choice of ρ(X)
 it is replaced by a denoising algorithm.

The Inner Minimization

$$E_{L,n}(X) = \frac{1}{c} \sum_{ij \in \Omega} \min_{kl} \left\| \mathbf{R}_{ij} X - \mathbf{Q}_{kl} S \right\|_{2}^{r} + \left\| X - C \right\|_{W}^{2} + \lambda \rho(X)$$

Instead, we approximate the solution by several iterations of the ADMM:

Applying the patch matching by Tree-K-Means Approximate NN on PCA-projected patches.

□ Fusing the patches as in Kwatra's alg. via IRLS (handling $r \neq 2$).

□ Blending back the original content with the outcome using

 $\widehat{\mathbf{X}} = (\mathbf{I} + \mathbf{W})^{-1} (\mathbf{X} + \mathbf{W}\mathbf{C})$

Applying post-process smoothing on the outcome (Domain-Transform filtering).

Segmentation and Its Role

- Segmentation is used to direct the style-transfer process which parts of the content to be (more) preserved.
- We experimented with several options and their combinations:
 - Edge-based methods + region filling.
 - Affinity-based (bilateral) segmentation.
 - Face detection + grabcut.
- The segmentation leads to the creation of W, which is used in the content blending stage in the algorithm.



No Segmentation?

- Due to the sensitivity w.r.t. the segmentation, there is a desire to avoid segmentation altogether. Could it be done in our scheme?
- Our answer: Sometimes it can be done.
- \Box We experimented with $\mathbf{W} = \alpha$ as a replacement to segmentation.



Color (Palette)-Transfer

- ❑ We have the flexibility of using ANY palette simply bring the contentand the style-images to common grounds, and proceed regularly.
- □ All our experiments use the style-palette: artistic result, and richer.
- Various methods to transfer a given image to a new palette we used the built-in Matlab function of histogram matching (not the best option).
- Pre-processing the style image could help increasing the success of the style-transfer task (removal of too dark/bright areas, ...).
- Our algorithm projects to the palette, both as a pre-processing of the content image, and also within each iteration, due to the desire to force the algorithm to preserve the diversity of the style image.
Summary: Key Modifications over Kwatra's ...

In the initialization of the algorithm:

- Optional: Apply strong edge-preserving spatial filtering (Domain-Transform) on the content Image, in order to remove small details from the final artistic result.
- Apply color-transfer from the style image to the content one, prior to initiating the algorithm.
- Build a mask image to define the content regions to be preserved.
 This mask is based on several segmentation methods.
- □ Initialize the whole algorithm by the content image + strong noise.

Summary: Key Modifications over Kwatra's ...

Within the iterative part:

- □ After each aggregation step, we fuse the content Image with the result by a weighted average, using the mask image.
- □ We apply color-transfer from the style image to the result, in order to prevent a drift from the color richness existing in the style.
- Our (approximate) nearest-neighbor is based on dimensionality reduction + tree-based search.

Part 4

Results and Discussion

 \Box All images are of size 400 × 400 pixels.

- □ The patch-sizes are [33, 21, 13, 9, (5)].
- □ The sub-sampling gaps are [28, 18, 8, 5, (3)].
- \Box The pyramids built have Lmax = 3 resolution layers.
- □ The robust fusion uses exponent r = 0.8.
- \Box Per each energy-function $E_{L,n}(X)$ we apply 5 iterations.

Results (0): A Closer Look at the Process



Color-Transferred image

Segmentation mask obtained using the edge-based method



Results (0): A Closer Look at the Process



The error shown is

$$\sum_{ij\in\Omega} \left\| \mathbf{R}_{ij} \mathbf{X} - \mathbf{Q}_{kl} \mathbf{S} \right\|_{2}^{\mathbf{r}}$$

Results (0): A Closer Look at the Process





Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
 Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect

Segmentation and its effectModifying content?

- 49 -



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
 Palette-transfer and its effect
 Modifyin



Face parts from the style?
 Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
 Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
 Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect

Segmentation and its effectModifying content?

Last

64 -



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
Palette-transfer and its effect



Face parts from the style?
 Palette-transfer and its effect




























































Results (5): Tendency to Randomness



- 98 -

Results (5): Tendency to Randomness



- 99 -

Results (6): Comparison to Gatys'



Results (7): Comparison to Frigo's



Content Image

Frigo's Results

Our Results

- 101 -

Similarities/Differences to Our Scheme:

- □ Their Belief-Propagation is replaced by the iterative IRLS refinement.
- □ The quadtree division is replaced by the segmentation in our scheme.
- □ In our work, every pixel is covered by various patch sizes and not just one.
- □ We (optionally) filter the input image in order to get a "more artistic" result.
- Our work allows better 'hallucination', as we may deviate from the content in permitted regions, and simply match the current solution to the style image.
- If Frigo's method is ran on an empty content image, it simply copies a portion of the style image.

- 102 -



Reasons for Failing:

- □ Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 103 -



Reasons for Failing:

- □ Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 104 -



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- □ Mismatch Between Content and Style

- 105 -

- Poorly Chosen Style Image
- Poor Results without Segmentation



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style

- 106 -

- Poorly Chosen Style Image
- Poor Results without Segmentation



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 107 -



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 108 -


Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 109 -



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation

- 110 -



Reasons for Failing:

- Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style

- 111 -

- Poorly Chosen Style Image
- Poor Results without Segmentation



Reasons for Failing:

- □ Poor Palette-Transfer
- Poor Segmentation
- Mismatch Between Content and Style
- Poorly Chosen Style Image
- Poor Results without Segmentation



The higher the pyramid, the larger the elements taken from the style image, but at the risk of getting large pieces copied



The higher the pyramid, the larger the elements taken from the style image, but at the risk of getting large pieces copied

- 114 -



 $[33, 21, 13, 9, 5] \qquad [33, 21, 13, 9] \qquad [33, 21, 13] \qquad [33, 21]$

The smaller the patches used, the more refined the result obtained. In the content areas, this means getting closer to the original content



 $[33, 21, 13, 9, 5] \qquad [33, 21, 13, 9] \qquad [33, 21, 13] \qquad [33, 21]$

The smaller the patches used, the more refined the result obtained. In the content areas, this means getting closer to the original content



Varying density of subsampling from very coarse (left) to relatively dense

The conclusion: It does not influence the results so much, and therefore we prefer coarse overlaps, as it reduced dramatically the computational cost of the algorithm.

Well, sorry, we cannot give the code !!

BUT

Elias Wang and Nicholas Tan can (students from Stanford doing their final project in EE368 who simply replicated our work to the letter)

ewang314/EE368_Final_Project

Part 5

Summary and What Next?



- We have introduced an extension of Kwatra's texture-synthesis work, to handle the style-transfer problem.
- □ The results are appealing, and the algorithm is relatively simple and fast. The main computational load is in the patch-matchings.
- One major benefit of the proposed scheme is that it is explicit, and as such the results are relatively easy to interpret and modify.
- Another benefit is the potential of this algorithm to have a very short run-time, while being able to operate on any pair of style+content pair.

Details of the Proposed Method

Preserving content regions?

 \rightarrow Segmentation is used to define the regions to preserve.

□ Local contrast changes?

 \rightarrow Yes, although we did not incorporate this into our scheme.

□ Shifts in the content image?

 $\rightarrow\,$ We control these shifts by the size of the patch sizes

□ Color palette to adopt?

 \rightarrow We are not limited, but we adopted the style one due to its expected richness.

□ Hallucination allowed?

 \rightarrow As much as we want – we have control over it.

□ Parts qualify as style?

 $\rightarrow\,$ Easily defined. We used all.

□ Copying vs. hallucination?

 \rightarrow The maximal patch-size and the richness of the style control this effect

□ Predicting success?

 \rightarrow We have guidelines for success (matched-scale, palette-transfer, style richness)

Immediate Improvements

Better designing the palette-transfer algorithm:

- Avoid the creation of too dark/bright areas
- Create false-colors that do not exist in the style image
- Lead to better matching between content and style.
- Controlling the sharpness in the segmented content regions, to be robust to the palette-transfer stage.
- Modify the nearest-neighbor strategy to use image-search (e.g. PatchMatch) instead of the current method.
- Enrich the style-patch database by rotations, scaled versions, mirroring, and contrast/brightness modifications. How about running texture-synthesis (by CNN?) on the style to create newer versions of it to add.
- Can we exploit the case where the style image is given off-line? _ 122 -

Longer Term Steps

- Modifying the metric to include orientation in edge areas, in order to lead to content modification of the form obtained by CNN methods.
- Seek ways to bypass the segmentation altogether while producing good results.
- □ Using different stopping criteria for each region, so that the smallest patch sizes does not operate uniformly in the image domain.
- Fixing automatically the style- and the content-images resolutions for a best fit.
- □ Handling video?

Better energy minimization formulation?



Thank You

- 124 -