A Curious Exploration of Image Style Transfer — Using Color-Shape-Texture Cues, without ML

Hang Chu
University of Toronto
CSC2523 Final Project Report
chuhang1122@cs.toronto.edu

Abstract

Neural networks have been recently adopted in image style transfer and achieved impressive results. In this project, we takle the problem with a rather classic approach. In particular, we aim to transfer style between images without resorting to large-scale statistical learning. We believe this is beneficial for gaining insights about the nature of the problem, which provides useful references for future data-efficient methods.

1. Introduction

Image style transfer aims to compose a new image from a source image and a style image, where the composed image possesses the artistic style of the style image, while preserving the content of the source image. This enables entertaining applications such as Prisma [1], that has been used to editing user-uploaded photos creatively.

Despite the success of approaches based on deep neural networks, they suffer from the dependency of large-scale datasets. One important criticism to recent deep learning is that humans do not need equivalently large amount of training data to learn about a new task. This also applies to image style transfer.

The essential motivation of this project is to gain better understanding to the nature of image style transfer. We aim to discover the elements that affect our human perception of artistic style, and tackle with the challenges of imposing them computationally in the synthesis process.

Beyond this, there are practical values of pursuing non-deep image style transfer. First, it leads to algorithms that are data efficient and do not rely on expensive hardware. Second, it improves the explanability and hence customizability of the style transfer results. Finally, it helps us to reveal the underlying human perception mechanism, which could be extended to other problems, e.g. style transfer of 3D shapes.

2. Related Work

Deep neural networks have achieved great success in image style transfer. Gatys et al. [7] proposed to formulate this problem as gradient-guided reconstruction, where the output image suffices desired style specified by mid-level layer gram-matrix statistics. Johnson et al. [11] improved this idea by incoorperating perceptual loss, and achieved real-time transfer with a feed-forward stylization network. Furthermore, Huang et al. [10] proposed to realize style transfer without training new networks for each new style, via adaptive instance normalization.

Beyond artistic style, Luan et al. [15] adapted the idea to real-world photos. Li et al. [12, 13] extended this idea and achieved universal photo style transfer without per-style network training.

Before the widespread adoption of deep neural networks, texture synthesis has been a long-studied problem in both computer vision and computer graphics communities. Efros et al. [5] proposed texture synthesis methods, and further extended to a novel application named image quilting [4] that allows texture transfer between images. Hertzmann et al. [9] presented a method to replicate analogous image transformation, given an existing transformation image pair. This was revisited by Liao et al. [14] to incoorperate the recent advancement of semantic recognition technique. Extracting structural information from images has been studied in Evangelopoulos et al. [6] and Xu et al. [18].

Standing on the shoulders of gaints, our project also makes usage of multiple existing techniques. Including AMAT [17], color balancing [16], shape matching [3], SLIC superpixel [2], and guided filtering [8].

3. Method

Our method consists of three major steps in the order of high-level to low-level: color adjustment, shape deforma-

Code can be downloaded at: https://github.com/chuhang/StyleTransfer-noML

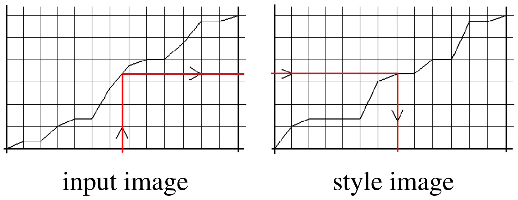


Figure 1. Illustration of color histogram equalization [16]. X-axis: color value; Y-axis: cumulated percentage of pixels.

tion, and texture imposition. We will describe these steps in detail, followed by brief discussion of other approaches we have attempted.

3.1. Color Adjustment

In this step, we aim to adjust the color palette of the input image, such that it matches the color distribution of the style image. To achieve this, we use a simple histogram equilization algorithm. Figure 1 shows an illustration to this. We first compute the color histogram of the input and style images. Next, we convert the histograms into Cumulative Distribution Functions (CDFs) of pixels. Each discretized color value in the input image is then mapped to a new value, where the new value is determined by the point that corresponds to the same CDF value in the style image.

We use the Hue-Saturation-Value (HSV) color encoding instead of default RGB, as it better reflects the color properties of an image. Figure 2 shows examples of histogram equalization. We emperically choose hue-only equalization as our color adjustment step, due to its relatively better visual quality.

3.2. Shape Deformation

In this step, we aim to adjust the mid-level stroke patterns of the input image, such that it possesses the same painting brush behavior of the style image. To achieve this, we first decompose both images into structural branches and medial disks using AMAT [17].

We observe that default AMAT does not produce ideal result, as many tiny branches exist in typical painting images. This opposes difficulties for representing the two shapes with the same detail level. To address this issue, we further apply three post-processing steps. First, we compute connected components in both branch images, and elliminate tiny branches that are smaller than 2 pixels. Next, we compute the histograms of branch length for both images. Then for each bin of the histograms, we identify the image that has more number of branches inside the length bin, and randomly discard medial-branches until both histograms are equal. Last, we re-compute the medial disk radius for both images, by using distance transform on the pruned branch images. Figure 3 shows an example of our pruning process.

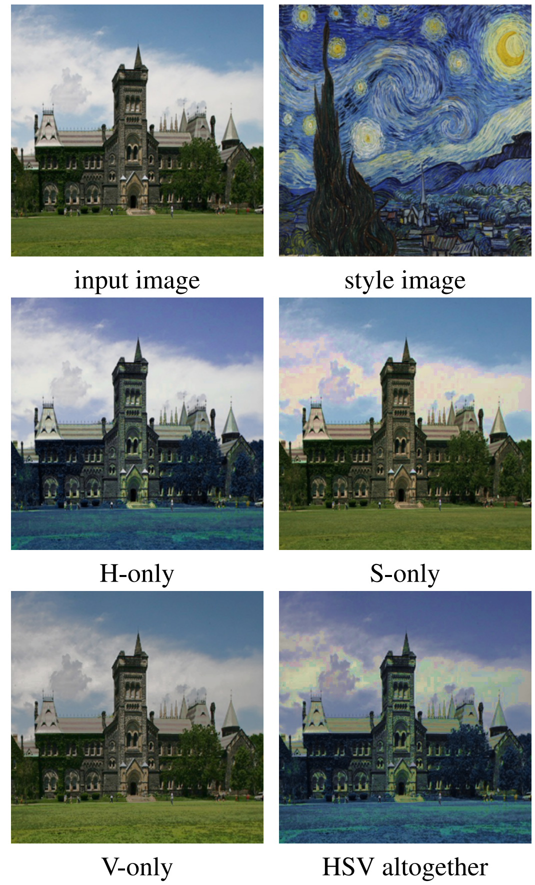


Figure 2. Results of applying histogram equalization on different channels.

Our next process aims to find similar shape correspondences between the two images. We use a classic shape matching approach [3] to achieve this goal. First, we randomly select medial points from both images, and compute a local shape descriptor for each selected point. We obtain the descriptor by overlaying log-polar bins centered at the point, and compute number of medial points located inside each bin. All numbers are then concatenated to form the descriptor vector. To match the computed descriptors between two images, we use the χ^2 statistics to measure the similarity between two descriptors, i.e.

$$C(i,j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)} \tag{1}$$

where $h(k)$ is the k^{th} dimension of the descriptor. In our implementation, we use 8 angular bins and 3 radial bins. Figure 4 shows an example of local shape matching.

As the last process of this step, we deform input image shapes locally such that they look like those of the style image. To achieve this, for the neighborhood around each

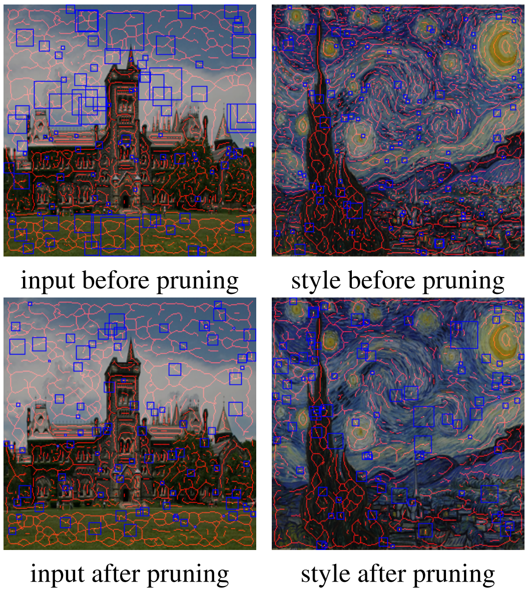


Figure 3. Default AMAT results in unbalanced distribution of medial disk radius. We apply a pruning post-processing to address this issue. Red: medial point branches; Blue: medial disk radius.

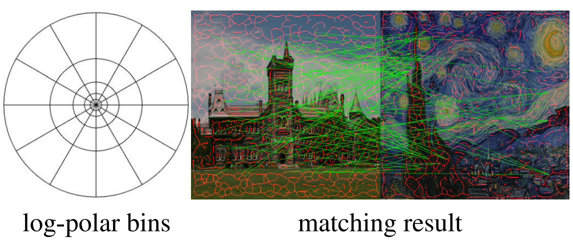


Figure 4. Shape matching descriptor and example result.

matching center point, we compute the (u, v) offset of each medial point, to its nearest medial point inside the matched neighborhood of the style image. This can be achieved using the nearest point index of distance transform. As shown in Figure 5, pixel-wise offset are typically non-smooth. Therefore, we further apply two filters to smooth the computed offset. First, we apply a Gaussian filter centered at each patch center, so that patch centers translate the most, and translation magnitude diminishes near patch boundaries. Second, we apply a branch-wise mean filter to enforce smooth offset within each medial point branch. The deformed image is shown in Figure 5.

3.3. Texture Imposition

In this step, we aim to impose the painting texture onto the result image. We provide two options to accomplish this. In the first option, we compute superpixels in the input and style images using SLIC [2]. This divides both images into regions while preserving important edges. We then pair up superpixels across the two images randomly, and blend

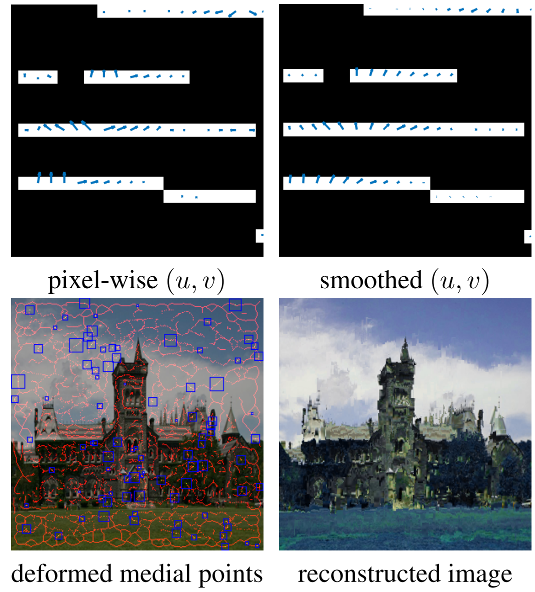


Figure 5. Shape deformation using matched local patches.

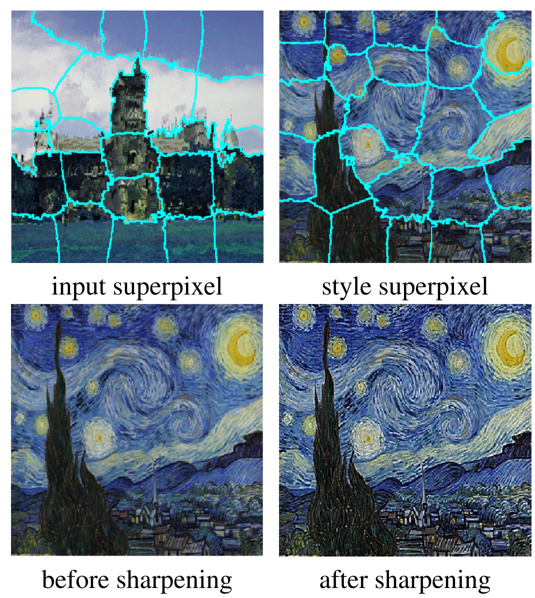
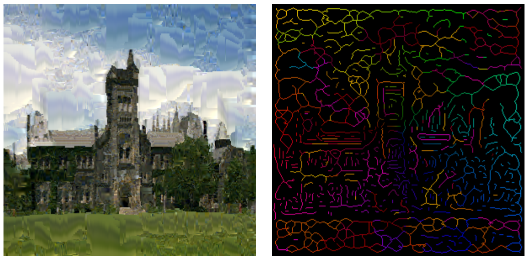


Figure 6. Example of SLIC [2] superpixel and sharpening that we use for texture imposition.

the two superpixels together to form the new superpixel in the result image. To further perceptually enhance the style image texture, we apply image sharpening to the style image before blending, and apply guided-filter using the style superpixel as gradient guidance after blending. Figure 6 shows an example.

As the second option, we directly apply Image Quilting [4]. Compared to the first option which can be computed instantly, this option requires significantly more computation.



direct medial disk copy geodesic dense match

Figure 7. Left: matching local shapes using Hamming distance and directly transfer medial disks; Right: dense cross image association based on geodesic distance, color represents associated (x, y) coordinates in the style image. Both methods do not yield successful result and are thus unused in our final algorithm.

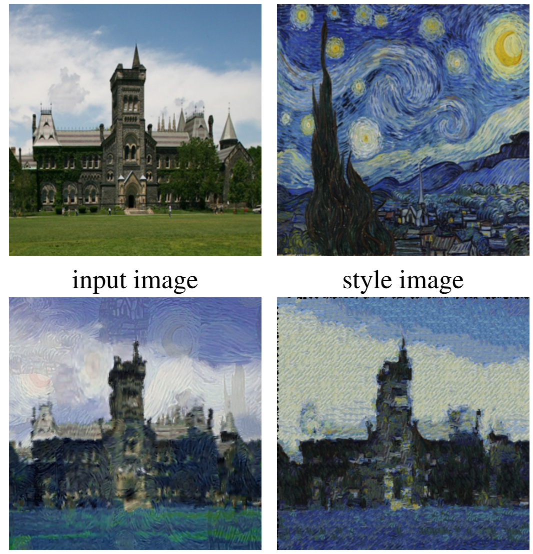
3.4. Other Attempted Steps

In this project, we come to realize the non-triviality of style transfer with classic image processing techniques. There are multiple approaches that we have attempted without success. These include e.g. directly match nearest local shape and transfer medial disk pixels while preserving the original medial disk's mean RGB color, as well as expanding patch correspondences to a dense medial point association map based on geodesic distance to the nearest patch center. These approaches do not work well, as they assume artistic style can be solely captured by medial disk statistics. However, higher-level modifications are necessary for the success of this task. Figure 7 shows intermediate results of these attempted, but eventually unused steps.

4. Results

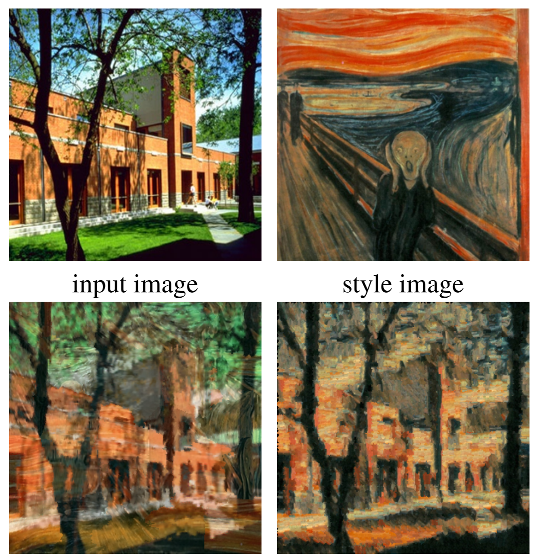
We tested our algorithm on random internet images. For texture imposition, we tried both options and demonstrate result in Figure 8 and Figure 9. Comparing the two texture imposition options, it can be seen that Option1 is perceptually better than Option2 in Figure 8, and two options are close in Figure 9. In both examples, it can be seen that our algorithm successfully transfers the color, shape, and texture properties of the style image, while preserving the content of the input image.

We measure the runtime of our algorithm on a Intel-i7 CPU. Our unoptimized single-thread Matlab prototype takes 11 seconds to finish our proposed main steps (excluding AMAT, using the fast Option1). For AMAT computation, it takes in average 246 seconds CPU time. For Option2 texture imposition using image quilting [4], it takes an extra 1489 seconds. We believe with the availability of a faster AMAT, our Option1 algorithm can be easily optimized to run in real-time on CPU.



result (Option1) result (Option2)

Figure 8. Example results of our algorithm. Option1 uses our proposed fast texture imposition, Option2 uses the slow image quilting [4] for texture imposition.



result (Option1) result (Option2)

Figure 9. Another example.

5. Conclusions & Discussions

Comparing our non-ML approach with current approaches based on deep neural networks, our method is capable of transfering color, shape, and texture properties of the style image, while requires less computational resource and is training and pre-training free. In the meantime, our method suffers in the lack of high-level semantic awareness (e.g. sky-to-sky and building-to-building transfer), and is sometimes lack of texture coherence.

The purpose of this project is not discrediting current style transfer approaches, as our method does not achieve significantly better performance. This said, we believe it is worth noting what we can achieve in the absence of any machine learning techniques, using only classic image processing techniques. We hope this project is helpful for better understanding of the nature of the problem, and contributes towards data-efficient computer vision.

References

[1] Prisma app. `https://prisma-ai.com/`.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2274–2282, 2012.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(4):509–522, 2002.

[4] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 341–346. ACM, 2001.

[5] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038. IEEE, 1999.

[6] G. Evangelopoulos and P. Maragos. Image decomposition into structure and texture subcomponents with multi-frequency modulation constraints. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016.

[8] K. He, J. Sun, and X. Tang. Guided image filtering. In *European Conference on Computer Vision (ECCV)*, pages 1–14. Springer, 2010.

[9] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 327–340. ACM, 2001.

[10] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1501–1510. IEEE, 2017.

[11] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[12] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 385–395, 2017.

[13] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. *arXiv preprint arXiv:1802.06474*, 2018.

[14] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *Transactions on Graphics (TOG)*, 36(4):120, 2017.

[15] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6997–7005. IEEE, 2017.

[16] L. Neumann and A. Neumann. Color style transfer techniques using hue, lightness and saturation histogram matching. In *Computational Aesthetics*, pages 111–122. Citeseer, 2005.

[17] S. Tsogkas and S. Dickinson. Amat: Medial axis transform for natural images. In *International Conference on Computer Vision (ICCV)*, pages 2727–2736. IEEE, 2017.

[18] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *Transactions on Graphics (TOG)*, 31(6):139, 2012.