

Controlling Perceptual Factors in Neural Style Transfer

Leon A. Gatys¹ Alexander S. Ecker¹ Matthias Bethge¹ Aaron Hertzmann² Eli Shechtman²
¹University of Tübingen ²Adobe Research

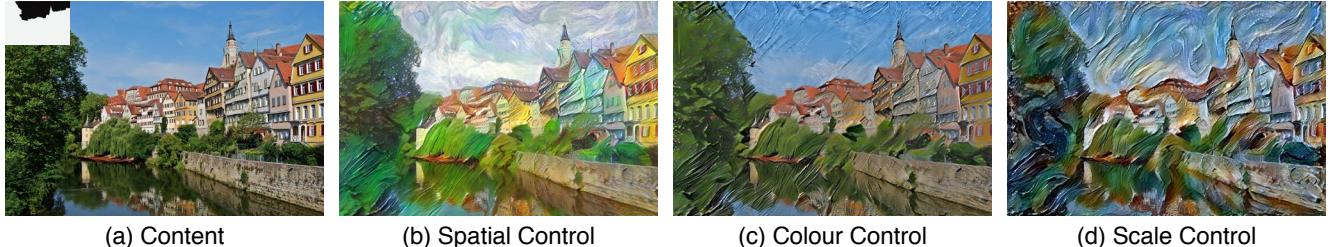


Figure 1: Overview of our control methods. **(a)** Content image, with spatial mask inset. **(b)** Spatial Control. The sky is stylised using the sky of Style II from Fig. 2(c). The ground is stylised using Style I from Fig. 4(b). **(c)** Colour Control. The colour of the content image is preserved using luminance-only style transfer described in Section 5.1. **(d)** Scale Control. The fine scale is stylised using using Style I from Fig. 4(b) and the coarse scale is stylised using Style III from Fig. 4(b). Colour is preserved using the colour matching described in section 5.2.

Abstract

Neural Style Transfer has shown very exciting results enabling new forms of image manipulation. Here we extend the existing method to introduce control over spatial location, colour information and across spatial scale¹². We demonstrate how this enhances the method by allowing high-resolution controlled stylisation and helps to alleviate common failure cases such as applying ground textures to sky regions. Furthermore, by decomposing style into these perceptual factors we enable the combination of style information from multiple sources to generate new, perceptually appealing styles from existing ones. We also describe how these methods can be used to more efficiently produce large size, high-quality stylisation. Finally we show how the introduced control measures can be applied in recent methods for Fast Neural Style Transfer.

1. Introduction

Example-based style transfer is a major way to create new, perceptually appealing images from existing ones. It takes two images \mathbf{x}_S and \mathbf{x}_C as input, and produces a new image $\hat{\mathbf{x}}$ applying the style of \mathbf{x}_S to the content of \mathbf{x}_C . The concepts of “style” and “content” are both expressed in terms of image statistics; for example, two images are said

to have the same style if they embody the same correlations of specific image features. To provide intuitive control over this process, one must identify ways to access perceptual factors in these statistics.

In order to identify these factors, we observe some of the different ways that one might describe an artwork such as Vincent van Gogh’s *A Wheatfield with Cypresses* (Fig. 2(c)). First, one might separately describe different styles in different regions, such as in the sky as compared to the ground. Second, one might describe the colour palette, and how it relates to the underlying scene, separately from factors like image composition or brush stroke texture. Third, one might describe fine-scale spatial structures, such as brush stroke shape and texture, separately from coarse-scale structures like the arrangements of strokes and the swirly structure in the sky of the painting. These observation motivates our hypothesis: image style can be perceptually factorised into style in different spatial regions, colour and luminance information, and across spatial scales, making them meaningful control dimensions for image stylisation.

Here we build on this hypothesis to introduce meaningful control to a recent image stylisation method known as Neural Style Transfer [8] in which the image statistics that capture content and style are defined on feature responses in a Convolutional Neural Network (CNN) [22]. Namely, we introduce methods for controlling image stylisation independently in different spatial regions (Fig. 1(b)), for colour and luminance information (Fig. 1(c)) as well as on different spatial scales (Fig. 1(d)). We show how they can be applied

¹Code: github.com/leongatys/NeuralImageSynthesis

²Supplement: bethgelab.org/media/uploads/stylecontrol/supplement/

to improve Neural Style Transfer and to alleviate some of its common failure cases. Moreover, we demonstrate how the factorisation of style into these aspects can gracefully combine style information from multiple images and thus enable the creation of new, perceptually interesting styles. We also show a method for efficiently rendering high-resolution stylisations using a coarse-to-fine approach that reduced optimisation time by an approximate factor of 2.5. Finally, we show that in addition to the original optimisation-based style transfer, these control methods can also be applied to recent fast approximations of Neural Style Transfer [13, 23].

2. Related Work

There is a large body of work on image stylisation techniques. The first example-based technique was Image Analogies [12], which built on patch-based texture synthesis techniques [4, 26]. This method introduced stylisation based on an example painting, as well as ways to preserve colour, and to control stylisation of different regions separately. The method used a coarse-to-fine texture synthesis procedure for speed [26]. Since then, improvements to the optimisation method and new applications [20, 6] have been proposed. Patch-based methods have also been used with CNN features [16, 2], leading to improved texture representations and stylisation results. Scale control has been developed for patch-based texture synthesis [9] and many other techniques have been developed for transferring colour style [5]. There are also many procedural stylisation techniques that provide extensive user control in the non-photorealistic rendering literature, e.g., [1, 15, 18]. These procedural methods provide separate controls for adjusting spatial variation in styles, colour transformation, and brush stroke style, but cannot work from training data.

More recently, Neural Style Transfer [8] has demonstrated impressive results in example-based image stylisation. The method is based on a parametric texture model [14, 10, 19] defined by summary statistics on CNN responses [7] and appears to have several advantages over patch-based synthesis. Most prominently, during the stylisation it displays a greater flexibility to create new image structures that are not already present in the source images [16].

However, the representation of image style within the parametric neural texture model [7] allows far less intuitive control over the stylisation outcome than patch-based methods. The texture parameters can be used to influence the stylisation but their interplay is extremely complex due to the complexity of the deep representations they are defined on. Therefore it is difficult to predict their perceptual effect on the stylisation result. Our main goal in this work is to introduce intuitive ways to control Neural Style Transfer to combine the advantages of that method with the more fine-grained user control of earlier stylisation methods. Note

that concurrent work [27] independently developed a similar approach for spatial control as presented here.

3. Neural Style Transfer

The Neural Style Transfer method [8] works as follows. We define a content image \mathbf{x}_C and a style image \mathbf{x}_S with corresponding feature representations $\mathbf{F}_\ell(\mathbf{x}_C)$ and $\mathbf{F}_\ell(\mathbf{x}_S)$ in layer ℓ of a CNN. Each column of $\mathbf{F}_\ell(\mathbf{x})$ is a vectorised feature map and thus $\mathbf{F}_\ell \in \mathcal{R}^{M_\ell(\mathbf{x}) \times N_\ell}$ where N_ℓ is the number of feature maps in layer ℓ and $M_\ell(\mathbf{x}) = H_\ell(\mathbf{x}) \times W_\ell(\mathbf{x})$ is the product of height and width of each feature map. Note that while N_ℓ is independent of the input image, $M_\ell(\mathbf{x})$ depends on the size of the input image.

Neural Style Transfer generates a new image $\hat{\mathbf{x}}$ that depicts the content of image \mathbf{x}_C in the style of image \mathbf{x}_S by minimising following loss function with respect to $\hat{\mathbf{x}}$

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} \quad (1)$$

where the content term compares feature maps at a single layer ℓ_C :

$$\mathcal{L}_{content} = \frac{1}{N_{\ell_c} M_{\ell_c}(\mathbf{x}_C)} \sum_{ij} (\mathbf{F}_{\ell_c}(\hat{\mathbf{x}}) - \mathbf{F}_{\ell_c}(\mathbf{x}_C))_{ij}^2 \quad (2)$$

and the style term compares a set of summary statistics:

$$\mathcal{L}_{style} = \sum_{\ell} w_{\ell} E_{\ell} \quad (3)$$

$$E_{\ell} = \frac{1}{4N_{\ell}^2} \sum_{ij} (\mathbf{G}_{\ell}(\hat{\mathbf{x}}) - \mathbf{G}_{\ell}(\mathbf{x}_S))_{ij}^2 \quad (4)$$

where $\mathbf{G}_{\ell}(\mathbf{x}) = \frac{1}{M_{\ell}(\mathbf{x})} \mathbf{F}_{\ell}(\mathbf{x})^T \mathbf{F}_{\ell}(\mathbf{x})$ is the Gram Matrix of the feature maps in layer ℓ in response to image \mathbf{x} . As in the original work [8], we use the VGG-19 Network and include “conv4_2” as the layer ℓ_C for the image content and Gram Matrices from layers “conv1_1”, “conv2_1”, “conv3_1”, “conv4_1”, “conv5_1” as the image statistics that model style.

4. Spatial Control

We first introduce ways to spatially control Neural Style Transfer. Our goal is to control which region of the style image is used to style each region in the content image. For example, we would like to apply one style to the sky region and another to the ground region of an image to either avoid artefacts (Fig. 2(d),(e)) or to generate new combinations of styles from multiple sources (Fig. 2(f)). We take as input R spatial guidance channels \mathbf{T}^r for both the content and style image (small insets in (Fig. 2(a)-(c)). Each of these is an image map of values in $[0, 1]$ specifying which styles should be applied where: regions where the r^{th} content guidance channel is equal to 1 should get the style from

regions where the r^{th} style guidance channel is 1. When there are multiple style images, the regions index over all the example images. The guidance channels are propagated to the CNN to produce guidance channels \mathbf{T}_ℓ^r for each layer. This can be done by simple re-sampling or more involved methods as we explain later in this section. We first discuss algorithms for synthesis given the guidance maps.

4.1. Guided Gram Matrices

In the first method we propose, we multiply the feature maps of each layer included in the style features with R guidance channels \mathbf{T}_ℓ^r and compute one spatially guided Gram Matrix for each of the R regions in the style image. Formally we define a spatially guided feature map as

$$\mathbf{F}_\ell^r(\mathbf{x})_{[:,i]} = \mathbf{T}_\ell^r \circ \mathbf{F}_\ell(\mathbf{x})_{[:,i]} \quad (5)$$

Here $\mathbf{F}_\ell^r(\mathbf{x})_{[:,i]}$ is the i^{th} column vector of $\mathbf{F}_\ell^r(\mathbf{x})$, $r \in R$ and \circ denotes element-wise multiplication. The guidance channel \mathbf{T}_ℓ^r is vectorised and can be either a binary mask for hard guidance or real-valued for soft guidance. We normalise \mathbf{T}_ℓ^r such that $\sum_i (\mathbf{T}_\ell^r)_i^2 = 1$. The guided Gram Matrix is then

$$\mathbf{G}_\ell^r(\mathbf{x}) = \mathbf{F}_\ell^r(\mathbf{x})^T \mathbf{F}_\ell^r(\mathbf{x}) \quad (6)$$

Each guided Gram Matrix is used as the optimisation target for the corresponding region of the content image. The contribution of layer ℓ to the style loss is then:

$$E_\ell = \frac{1}{4N_\ell^2} \sum_{r=1}^R \sum_{ij} \lambda_r (\mathbf{G}_\ell^r(\hat{\mathbf{x}}) - \mathbf{G}_\ell^r(\mathbf{x}_S))_{ij}^2 \quad (7)$$

where λ_r is a weighting factor that controls the stylisation strength in the corresponding region r .

An important use for guidance channels is to ensure that style is transferred between regions of similar scene content in the content and style image. For example, Figure 2 shows an example in which the sky in the content image has bright clouds, whereas the sky in the style image has greyish clouds; as a result, the original style transfer stylises the sky with a bright part of the ground that does not match the appearance of the sky. We address this by dividing both images into a sky and a ground region (Fig. 2(a),(b) small insets) and require that the sky and ground regions from the painting are used to stylise the respective regions in the photograph (Fig. 2(e)).

Given the input guidance channel \mathbf{T}^r , we need to first propagate this channel to produce guidance channels \mathbf{T}_ℓ^r for each layer. The most obvious approach would be to down-sample \mathbf{T}^r to the dimensions of each layer's feature map. However, we often find that doing so fails to keep the desired separation of styles by region, e.g., ground texture still appears in the sky. This is because neurons near the boundaries of a guidance region can have large receptive fields

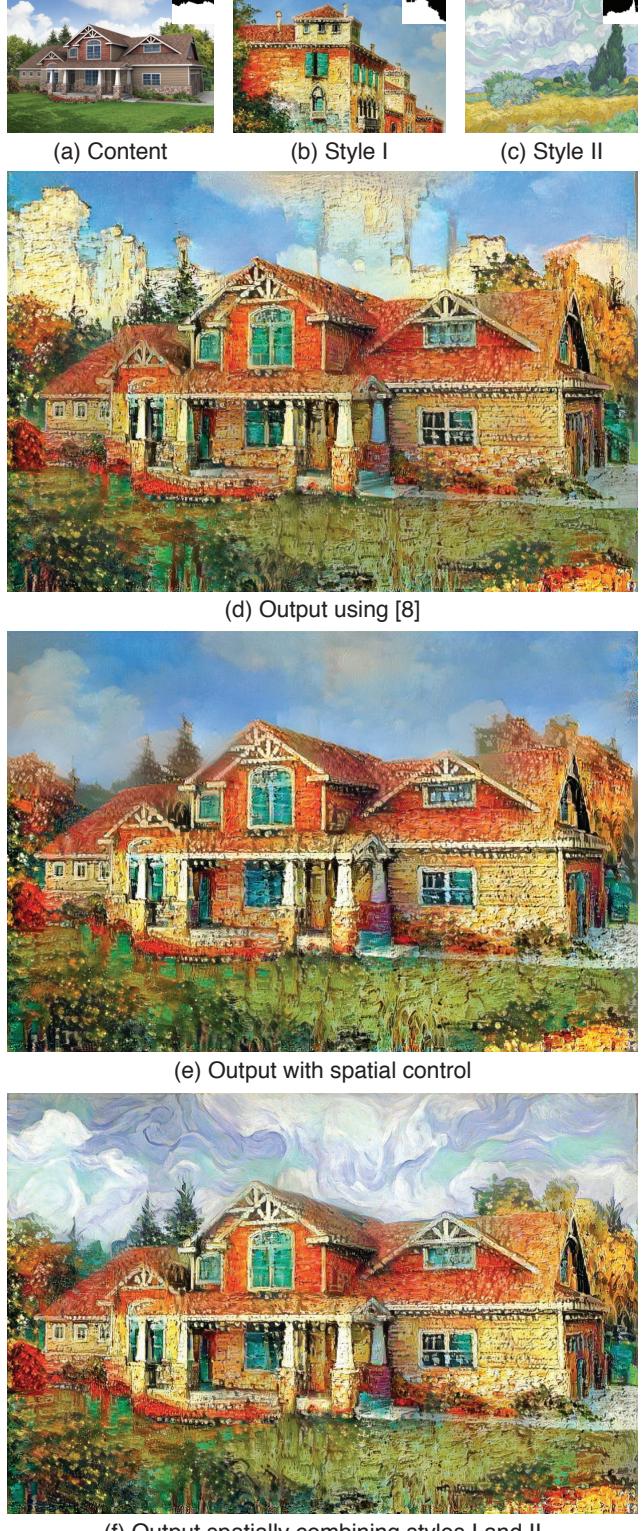


Figure 2: Spatial guidance in Neural Style Transfer. (a) Content image. (b) Style image I. (c) Style image II. Spatial mask separating the image in sky and ground is shown in the top right corner. (d) Output from Neural Style Transfer without spatial control [8]. The clouds are stylised with image structures from the ground. (e) Output with spatial guidance. (f) Output from spatially combining the the ground-style from (b) and the sky-style from (c).

that overlap into the other region. Instead we use an eroded version of the spatial guiding channels. We enforce spatial guidance only on the neurons whose receptive field is entirely inside the guidance region and add another global guidance channel that is constant over the entire image. We found that this soft spatial guidance usually yields better results. For further details on the creation of guidance channels, see the Supplementary Material, section 1.1.

Another application of this method is to generate a new style by combining the styles from multiple example images. Figure 2(f) shows an example in which the region guidance is used to use the sky style from one image and the ground style from another. This example demonstrates the potential of spatial guidance to combine many example styles together to produce new stylisations.

4.2. Guided Sums

Alternatively, instead of computing a Gram Matrix for each guidance channel, we can also just stack the guidance channels with the feature maps as it is done in [2] to spatially guide *neural patches* [16]. The feature representation of image \mathbf{x} in layer ℓ is then $\mathbf{F}'_\ell(\mathbf{x}) = [\mathbf{F}_\ell(\mathbf{x}), \mathbf{T}_\ell^1, \mathbf{T}_\ell^2, \dots, \mathbf{T}_\ell^R]$ and $\mathbf{F}'_\ell(\mathbf{x}) \in \mathcal{R}^{(N_\ell+R) \times M_\ell(\mathbf{x})}$. Now the Gram Matrix $\mathbf{G}'_\ell(\mathbf{x}) = \frac{1}{M_\ell(\mathbf{x})} \mathbf{F}'_\ell(\mathbf{x})^T \mathbf{F}'_\ell(\mathbf{x})$ includes correlations of the image features with the non-zero entries of the guidance channels and therefore encourages that the features in region r of the style image are used to stylise region r in the content image. The contribution of layer ℓ to the style loss is simply

$$E_\ell = \frac{1}{4N_\ell^2} \sum_{ij} (\mathbf{G}'_\ell(\hat{\mathbf{x}}) - \mathbf{G}'_\ell(\mathbf{x}_S))_{ij}^2 \quad (8)$$

This is clearly more efficient than the method presented in Section 4.1. Instead of computing and matching R Gram Matrices one only has to compute one Gram Matrix with R additional channels. Nevertheless, this gain in efficiency comes at the expense of texture quality. The additional channels in the new Gram Matrix are the sums over each feature map spatially weighted by the guidance channel.

$$\mathbf{G}'_\ell(\mathbf{x}_S)_{i,N_\ell+r} = \sum_j (\mathbf{T}_\ell^r \circ \mathbf{F}_\ell(\mathbf{x}_S)_{[:,i]})_j \quad (9)$$

Hence this method actually interpolates between matching the original global Gram Matrix stylisation and the spatially weighted sums over the feature maps. While the feature map sums also give a non-trivial texture model, their capacity to model complex textures is limited [7]. In practice we find that this method can often give decent results but also does not quite capture the texture of the style image – as would be expected from the inferior texture model. Results and comparisons can be found in the Supplementary Material, section 1.2.

5. Colour Control

The colour information of an image is an important perceptual aspect of its style. At the same time it is largely independent of other style aspects such as the type of brush strokes used or dominating geometric shapes. Therefore it is desirable to independently control the colour information in Neural Style Transfer. A prominent use case for such control is colour preservation during style transfer. When stylising an image using Neural Style Transfer, the output also copies the colour distribution of the style image, which might be undesirable in many cases (Fig. 3(c)). For example, the stylised farmhouse has the colours of the original van Gogh painting (Fig. 3(c)), whereas one might prefer the output painting to preserve the colours of the farmhouse photograph. In particular, one might imagine that the artist would have used the colours of the scene if they were to paint the farmhouse. Here we present two simple methods to preserve the colours of the source image during Neural Style Transfer — in other words, to transfer the style without transferring the colours. We compare two different approaches to colour preservation: colour histogram matching and luminance-only transfer (Fig. 3(d,e)).

5.1. Luminance-only transfer

In the first method we perform style transfer only in the luminance channel, as done in Image Analogies [12]. This is motivated by the observation that visual perception is far more sensitive to changes in luminance than in colour [25].

The modification is simple. The luminance channels L_S and L_C are first extracted from the style and content images. Then the Neural Style Transfer algorithm is applied to these images to produce an output luminance image \hat{L} . Using a colour space that separates luminance and colour information, the colour information of the content image is combined with \hat{L} to produce the final colour output image (Fig. 3(d)).

If there is a substantial mismatch between the luminance histogram of the style and the content image, it can be helpful to match the histogram of the style luminance channel L_S to that of the content image L_C before transferring the style. For that we simply match mean and variance of the content luminance. Let μ_S and μ_C be the mean luminances of the two images, and σ_S and σ_C be their standard deviations. Then each luminance pixel in the style image is updated as:

$$L_{s'} = \frac{\sigma_C}{\sigma_S} (L_S - \mu_S) + \mu_C \quad (10)$$

5.2. Colour histogram matching

The second method we present works as follows. Given the style image \mathbf{x}_S , and the content image \mathbf{x}_C , the style image's colours are transformed to match the colours of the

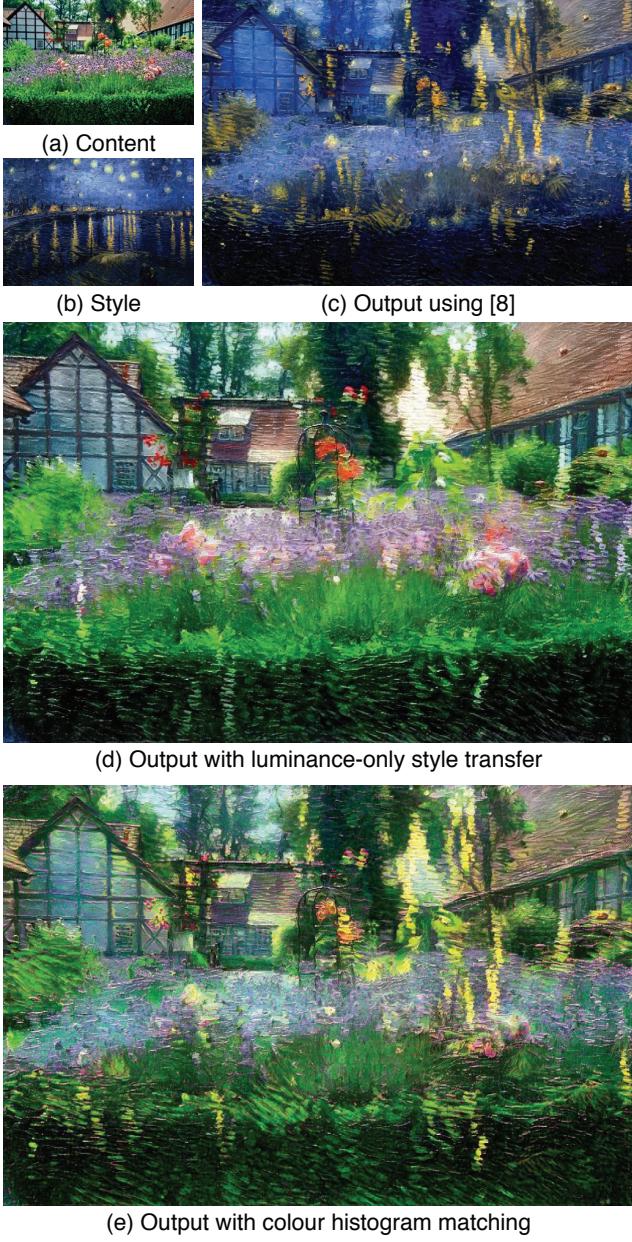


Figure 3: Colour preservation in Neural Style Transfer. **(a)** Content image. **(b)** Style image. **(c)** Output from Neural Style Transfer [8]. The colour scheme is copied from the painting. **(d)** Output using style transfer in luminance domain to preserve colours. **(e)** Output using colour transfer to preserve colours.

content image. This produces a new style image \mathbf{x}'_S that replaces \mathbf{x}_S as input to the Neural Style Transfer algorithm. The algorithm is otherwise unchanged.

The one choice to be made is the colour transfer procedure. There are many colour transformation algorithms to choose from; see [5] for a survey. Here we use linear methods, which are simple and effective for colour style transfer.

Given the style image, each RGB pixel \mathbf{p}_S is transformed

as:

$$\mathbf{p}'_S = \mathbf{A}\mathbf{p}_S + \mathbf{b} \quad (11)$$

where \mathbf{A} is a 3×3 matrix and \mathbf{b} is a 3-vector. This transformation is chosen so that the mean and covariance of the RGB values in the new style image \mathbf{p}'_S match those of \mathbf{p}'_C [11] (Appendix B). In general, we find that the colour matching method works reasonably well with Neural Style Transfer (Fig. 3(e)), whereas gave poor synthesis results for Image Analogies [11]. Furthermore, the colour histogram matching method can also be used to better preserve the colours of the style image. This can substantially improve results for cases in which there is a strong mismatch in colour but one rather wants to keep the colour distribution of the style image (for example with pencil drawings or line art styles). Examples of this application can be found in the Supplementary Material, section 2.2.

5.3. Comparison

In conclusion, both methods give perceptually-interesting results but have different advantages and disadvantages. The colour-matching method is naturally limited by how well the colour transfer from the content image onto the style image works. The colour distribution often cannot be matched perfectly, leading to a mismatch between the colours of the output image and that of the content image.

In contrast, the luminance-only transfer method preserves the colours of the content image perfectly. However, dependencies between the luminance and the colour channels are lost in the output image. While we found that this is usually very difficult to spot, it can be a problem for styles with prominent brushstrokes since a single brushstroke can change colour in an unnatural way. In comparison, when using full style transfer and colour matching, the output image really consists of strokes which are blotches of paint, not just variations of light and dark. For a more detailed discussion of colour preservation in Neural Style Transfer we refer the reader to the Supplementary Material, section 2.1.

6. Scale Control

In this section, we describe methods for mixing different styles at different scales and efficiently generating high-resolution output with style at desired scales.

6.1. Scale control for style mixing

First we introduce a method to control the stylisation independently on different spatial scales. Our goal is to pick separate styles for different scales. For example, we want to combine the fine-scale brushstrokes of one painting (Fig. 4(b), Style I) with the coarse-scale angular geometric shapes of another image (Fig. 4(b), Style II).

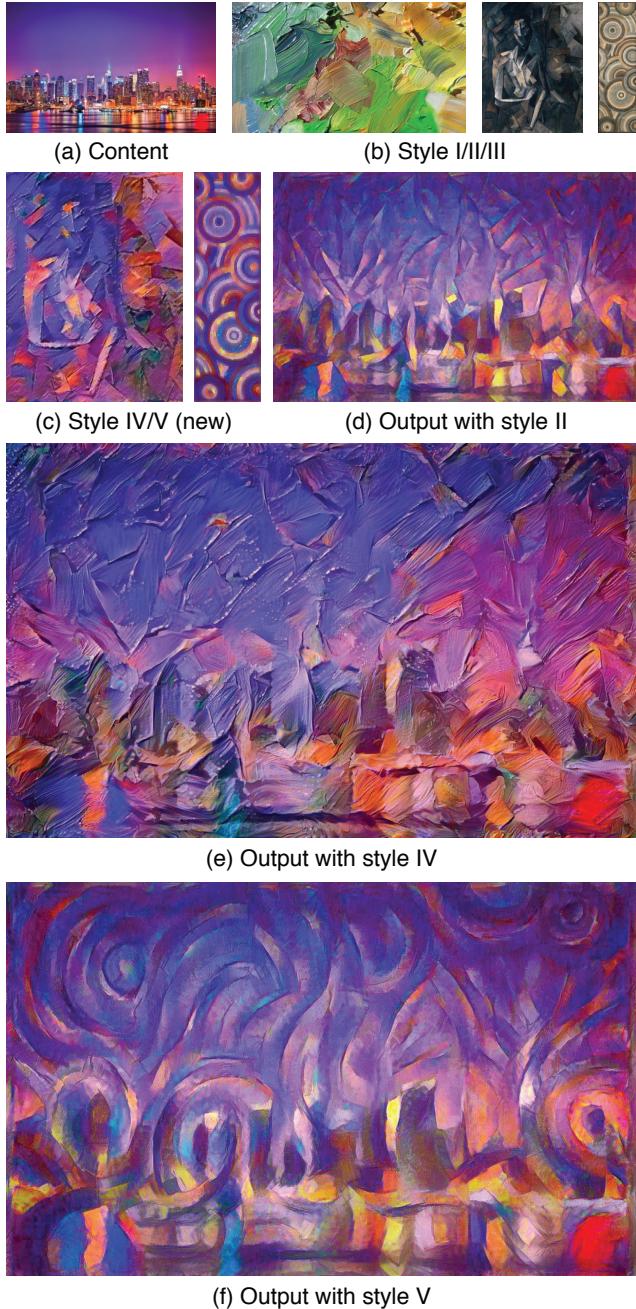


Figure 4: Scale control in Neural Style Transfer. **(a)** Content image. **(b)** Collection of styles used. Style I has dominant brush strokes on the fine scale. Style II has dominant angular shapes on the coarse scale. Style III has dominant round shapes on the coarse scale. **(c)** New styles obtained from combining coarse and fine scales of existing styles. Style IV combines fine scale of Style I with coarse scale of Style II. Style V combines fine scale of Style II with coarse scale of Style III. **(d)** Output using original Style II. **(e)** Output using the new Style IV. **(f)** Output using the new Style V. All stylisations preserve the colour of the photograph using the colour matching method described in section 5.2

We define the style of an image at a certain scale as the distribution of image structures in image neighbourhoods of a certain size f . In that sense, the colour separation introduced in the previous section can be thought of a special case of scale separation, since image colours are “structures” on one-pixel neighbourhoods. To model image style on larger scales, we use the Gram Matrices from different layers in the CNN. In particular, a Gram Matrix at layer ℓ represents the second-order statistics of image neighbourhoods of size corresponding to the receptive field size f_ℓ .

Unfortunately, this representation is not factorised over scale. In general, a Gram Matrix $\mathbf{G}_\ell(\mathbf{x})$ at a given spatial scale also captures much of the image information on smaller spatial scales and thus shares a lot of information with the Gram Matrix $\mathbf{G}_{\ell-k}(\mathbf{x})$ at a lower layer in the CNN (see Supplementary Material, section 3.1 for more details). Therefore, simply combining Gram Matrices from different scales of different images does not give independent control over the different scales.

Here we show a way to combine scales that avoids this problem. We first create a new style image that combines fine-scale information from one image with coarse scale information from another (Fig. 4(c)). We then use the new style image in the original Neural Style Transfer. We do this by applying Neural Style Transfer from the fine-scale style image to the coarse-scale style image, using only the Gram Matrices from lower layers in the CNN (e.g., only layer “conv1_1” and “conv2_1” in Fig. 4). We initialise the optimisation procedure with the coarse-style image and omit the content loss entirely, so that the fine-scale texture from the coarse-style image will be fully replaced. This is based on the observation that the optimisation leaves images structures intact when they are of larger scale than the style features. While this is not guaranteed, as it depends on the optimiser, we empirically find it to be effective for the L-BFGS method typically used in Neural Style Transfer. The resulting images (Fig. 4(c)) are used as the input to the original Neural Style Transfer to generate a new stylisations of the cityscape photograph. For example, we combine the fine scale of Style I with the coarse scale of Style II to re-paint the angular cubistic shapes in Fig. 4(d) with pronounced brushstrokes (Fig. 4(e)). Or we combine the fine scale of Style II with the coarse scale of Style III to replace the angular shapes by round structures, giving the image a completely different “feel” (compare Fig. 4(d) with Fig. 4(f)).

This method enables the creation of a large set of perceptually appealing, new styles by recombining existing ones in a principled way. It also allows for interesting new ways to interpolate between styles by interpolating across spatial scales. For more examples of new styles and results of interpolating between styles, we refer the reader to the Supplementary Material, sections 3.2 and 3.3.

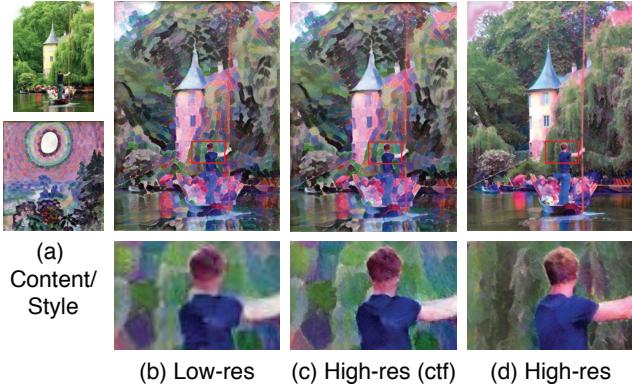


Figure 5: Neural Style Transfer in high resolution. **(a)** Content and style images. **(b)** Output in low-resolution with total number of pixels equal to 450^2 . **(c)** Output in high-resolution generated in a coarse-to-fine fashion from **(b)**. **(d)** Output in high-resolution without coarse-to-fine procedure. For both high-resolution images the total number of pixels is 3000^2 and they can be found in the Supplemental Material.

6.2. Scale control for efficient high resolution

The existing Neural Style Transfer method does not work well for high-resolution outputs. Since the receptive fields in a CNN have a fixed size, the stylisation outcome depends on the resolution of the input images: stylisation happens only up to the scale of the receptive fields in the output. In practice, we find that for the VGG-19 network, there is a sweet spot around 500^2 pixels for the size of the input images, such that the stylisation is appealing but the content is well-preserved (Fig. 5(b)). For a high-resolution image, however, the receptive fields are typically very small compared to the image, and so only very small-scale structures are stylised (Fig. 5 (d)).

Here we show that the same scale separation principle from the previous section can be used in order to produce high-resolution outputs with large-scale stylisation. We are given high-resolution content and style images \mathbf{x}_C and \mathbf{x}_S , both having the same size with N^2 pixels in total. We down-sample each image by a factor k such that N/k corresponds to the desired stylisation resolution, e.g., 500^2 for VGG, and then perform stylisation. The output is now low-resolution of size N/k . We can then produce high-resolution output from this image by up-sampling the low-resolution output to N^2 pixels, and use this as initialisation for Neural Style Transfer with the original input images \mathbf{x}_C and \mathbf{x}_S . The style features now capture and can fill-in the high-resolution information from the style image while leaving the coarse-scale stylisation intact (Fig. 5(c)).

This coarse-to-fine procedure has the additional advantage of requiring fewer iterations in the high-resolution optimisation and thus increasing efficiency. In our experiments we used 2.5 times fewer iterations for the high-resolution

optimisation. We also noticed that this technique effectively removes low-level noise that is typical for neural image synthesis. In fact, all figures shown in this paper, except for Fig. 6, were enhanced to high-resolution in that way. The low/high-resolution pairs can be found in the Supplement. Applying this technique iteratively also enables the generation of very high-resolution images that is only limited by the size of the input images and available memory.

7. Controlling Fast Neural Style Transfer

A major drawback of Neural Style Transfer is that image generation is relatively slow. Recently, a number of works have shown that one can train a feed-forward CNN to perform stylisation [13, 23, 17]. We now show how to apply the spatial and colour control described above to these Fast Neural Style Transfer methods. Applying scale control to Fast Neural Style Transfer is trivial, as it entails simply training on the new style image that combines multiple scales. We use Johnson’s excellent publicly-available implementation of Fast Neural Style Transfer [13]³. The networks we train all use the well-tuned default parameters in that implementation including Instance Normalization [24] (for details see Supplementary Material, section 4). For comparability and to stay in the domain of styles that give good results with Fast Neural Style Transfer, we use the styles published with that implementation.

7.1. Colour control

The simplest way to preserve the colour of the input image is to just use an existing feed-forward stylisation network [13], and then combine the luminance channel of the stylisation with the colour channels of the content image (Fig. 6(c)). An alternative is to train the feed-forward network exclusively with the luminance channel of the style and content images. This network then produces a luminance image that can be combined with the colour channels from the input content image (Fig. 6(d)). For both methods we match the mean luminance of the output image to that of the content image. In general, we find that colour preservation with the luminance network better combines stylisation with structures in the content image (Fig. 6(c),(d)).

7.2. Spatial control

We now describe training a feed-forward network to apply different styles to different regions. We show that this can be done with a surprisingly small modification to Johnson’s training procedure [13], which we illustrate with the following example. We create the style image by vertically concatenating the *Candy* and *Feathers* images shown in Fig. 6(b). Two additional binary guidance channels are added to the style image, i.e., one for the top of the image

³github.com/jcjohnson/fast-neural-style

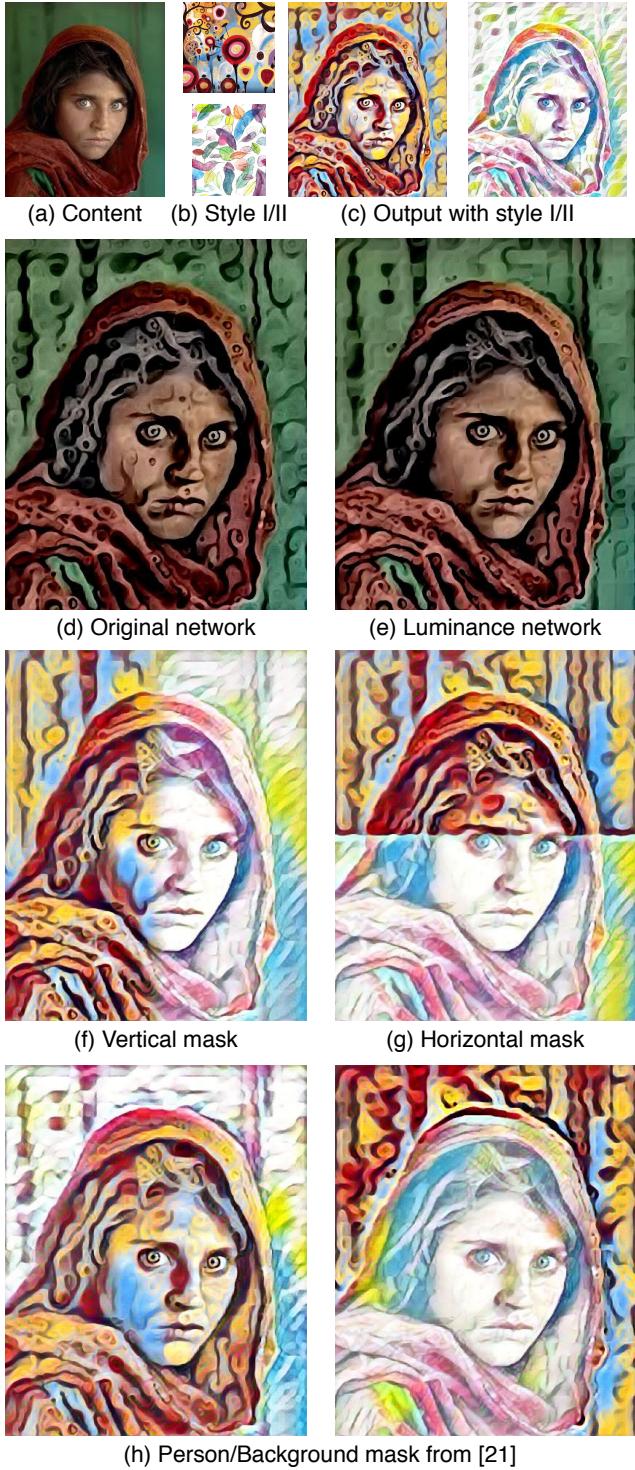


Figure 6: Colour and spatial control in Fast Neural Style Transfer. (a) Content image. (b) Styles *Candy* and *Feathers*. (c) Outputs from [13], trained with styles shown in (b). (d) Simple colour preservation. Luminance channel from (c) is combined with colour channels from (a). (e) Colour preservation with luminance network. Output from luminance network is combined with colour channels from (a). (f) Vertical separation of styles. (g) Horizontal separation of styles. (h) Separation of styles into person and background using [21].

and one for the bottom. The style loss function is based on the guided Gram Matrices (Eq. 7). During training, the feed-forward network takes as input the content image and two guidance channels. The input guidance channels are passed to the loss network to evaluate the spatially-guided losses. Surprisingly, we find that the guidance channels can be kept constant during training: during training we required the feed-forward network to always stylise the lower half of the image with one style and the upper half with another. However, the network robustly learns the correspondence between guidance channels and styles, so that at test time we can pass arbitrary masks to the feed-forward network to spatially guide the stylisation (Fig. 6(f)-(h)). By providing an automatically-generated figure-ground segmentation [21] we can create an algorithm that performs fast spatially-varying stylisation automatically. (Fig. 6(g),(h))

8. Discussion

In this work, we introduce intuitive ways to control Neural Style Transfer. We hypothesise that image style includes factors of space, colour, and scale, and present ways to access these factors during stylisation to substantially improve the quality and flexibility of the existing method.

One application of the control methods we present is to combine styles in an interpretable fashion. This contrasts with the alternative approach of combining styles by linearly interpolating in the style representation as, for example, is done in the concurrent work of Dumoulin et al. [3]. A possible concern with that approach is that if the directions in the style representation do not correspond to perceptual variables, it becomes difficult to generate appealing new styles. Still, even with our methods the selection of which inputs to combine for aesthetically pleasing results can be challenging. An exciting open research question is to predict what combinations of styles will combine nicely into new, perceptually pleasing styles.

Neural Style Transfer is particularly appealing because it can create new image structures based on the source images. This flexibility arises from the representation of style in terms of spatial summary statistics, in contrast to patch-based methods [12, 20, 6]. However, because it is not clear how the perceptual aspects of style are represented in the summary statistics, it is hard to achieve meaningful *parametric* control over the stylisation. For that it may be necessary to encourage appropriate factorisations of the CNN representations during network training, for example, to learn representations that factorise the image information over spatial scales. In fact, this touches a fundamental research question in machine vision: to obtain interpretable yet powerful image representations that decompose images into the independent factors of human visual perception.

References

- [1] L. Benedetti, Winnemöller, M. H., Corsini, and R. Scopigno. Painting with bob: Assisted creativity for novices. In *Proc. UIST*, 2014.
- [2] A. J. Champandard. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *arXiv:1603.01768 [cs]*, Mar. 2016. arXiv: 1603.01768.
- [3] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. In *Proc. ICLR*, 2017.
- [4] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, 1999.
- [5] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, E. Reinhard, D. Kuzovkin, and A. Tremeau. Colour mapping: A review of recent methods, extensions and applications. *Computer Graphics Forum*, 35(1):59–88, 2016.
- [6] J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sýkora. StyLit: Illumination-guided example-based stylization of 3d renderings. *ACM Trans. Graph.*, 35(4), July 2016.
- [7] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture Synthesis Using Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 28*, 2015.
- [8] L. A. Gatys, A. S. Ecker, and M. Bethge. Image Style Transfer Using Convolutional Neural Networks. In *Proc. CVPR*, 2016.
- [9] C. Han, E. Risser, R. Ramamoorthi, and E. Grinspun. Multi-scale texture synthesis. *ACM TOG*, 27(3):51:1–51:8, 2008.
- [10] D. J. Heeger and J. R. Bergen. Pyramid-based Texture Analysis/Synthesis. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’95*, pages 229–238, New York, NY, USA, 1995. ACM.
- [11] A. Hertzmann. *Algorithms for Rendering in Artistic Styles*. PhD thesis, New York University, 2001.
- [12] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image Analogies. In *Proc. SIGGRAPH*, 2001.
- [13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, number 9906 in Lecture Notes in Computer Science, pages 694–711. Springer International Publishing, Oct. 2016. DOI: 10.1007/978-3-319-46475-6_43.
- [14] B. Julesz. Visual Pattern Discrimination. *IRE Transactions on Information Theory*, 8(2), Feb. 1962.
- [15] R. D. Kalnins, L. Markosian, B. J. Meier, M. A. Kowalski, J. C. Lee, P. L. Davidson, M. Webb, J. F. Hughes, and A. Finkelstein. Wysiwyg npr: Drawing strokes directly on 3d models. *ACM Trans. Graph.*, 21(3), July 2002.
- [16] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. CVPR*, 2016.
- [17] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proc. ECCV*, 2016.
- [18] P. O’Donovan and A. Hertzmann. Anipaint: Interactive painterly animation from video. *IEEE TVCG*, 18(3), 2012.
- [19] J. Portilla and E. P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, 40(1):49–70, Oct. 2000.
- [20] G. Ramanarayanan and K. Bala. Constrained texture synthesis via energy minimization. *IEEE TVCG*, 2007.
- [21] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. *Computer Graphics Forum*, 35(2):93–102, 2016.
- [22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015.
- [23] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. *arXiv:1603.03417 [cs]*, Mar. 2016. arXiv: 1603.03417.
- [24] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv:1607.08022 [cs]*, July 2016. arXiv: 1607.08022.
- [25] B. Wandell. *Foundations of Vision*. Sinauer Associates Inc., 1995.
- [26] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proc. SIGGRAPH*, 2000.
- [27] P. Wilmot, E. Risser, and C. Barnes. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. *arXiv preprint arXiv:1701.08893*, 2017.