Visual Attribute Transfer through Deep Image Analogy *

Jing Liao[1], Yuan Yao[2] †, Lu Yuan[1], Gang Hua[1], and Sing Bing Kang[1]
[1]Microsoft Research, [2]Shanghai Jiao Tong University

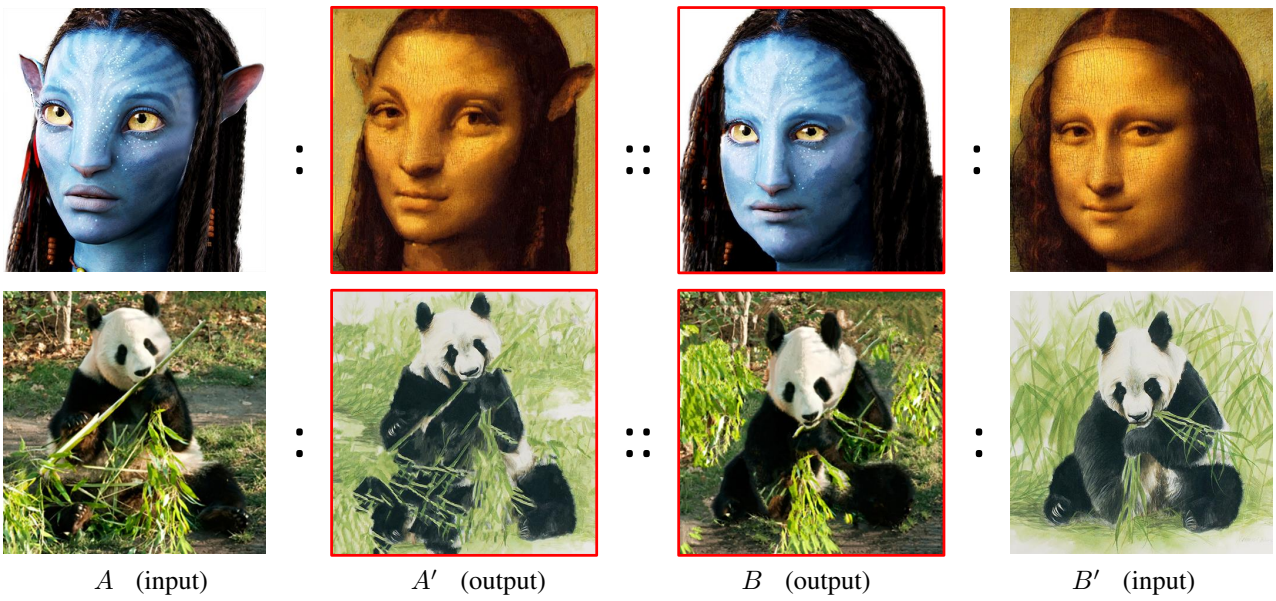


Figure 1: *Our technique allows us to establish semantically-meaningful dense correspondences between two input images A and B'. A' and B are the reconstructed results subsequent to transfer of visual attributes.*

Abstract

We propose a new technique for visual attribute transfer across images that may have very different appearance but have perceptually similar semantic structure. By visual attribute transfer, we mean transfer of visual information (such as color, tone, texture, and style) from one image to another. For example, one image could be that of a painting or a sketch while the other is a photo of a real scene, and both depict the same type of scene.

Our technique finds semantically-meaningful dense correspondences between two input images. To accomplish this, it adapts the notion of "image analogy" [Hertzmann et al. 2001] with features extracted from a Deep Convolutional Neutral Network for matching; we call our technique *deep image analogy*. A coarse-to-fine strategy is used to compute the nearest-neighbor field for generating the results. We validate the effectiveness of our proposed method in a variety of cases, including style/texture transfer, color/style swap, sketch/painting to photo, and time lapse.

1 Introduction

Many types of compelling image stylization effects have been demonstrated over the years, including color transfer, texture transfer, and style transfer. Their appeal is especially strong in the context of social media, where photo sharing and entertainment are important elements. A number of popular apps such as Prisma and Facetune have successfully capitalized on this appeal.

The applications of color transfer, texture transfer, and style transfer share a common theme, which we characterize as *visual attribute transfer*. In other words, visual attribute transfer refers to copying visual attributes of one image such as color, texture, and style, to another image.

In this paper, we describe a new technique for visual attribute transfer for a pair of images that may be very different in appearance but semantically similar. In the context of images, by "semantic", we refer to high-level visual content involving identifiable objects. We deem two different images to be semantically similar if both are of the same type of scene containing objects with the same names or classes, e.g., nature scenes featuring pandas and foliage, headshots, or indoor scenes featuring dining rooms with tables, chairs, walls, and ceiling. Figure 1 shows two semantically similar pairs. The images in each pair of inputs (A and B') look dramatically different, but have objects with similar identities.

Our technique establishes semantically-meaningful dense correspondences between the input images, which allow effective visual attribute transfer. Low-level matching methods, such as optical flow [Brox et al. 2004] and PatchMatch [Barnes et al. 2009], are designed to match local intensities. Hence, they fail to match under large visual variations. While other methods such as SIFT flow [Liu et al. 2011] or deep match [Weinzaepfel et al. 2013] are more reliable in matching sparse features, they are also not able to handle extreme visual variations, such as matching across an artistic painting and a real photograph. This is because these methods are still fundamentally based on low-level features.

Methods have been proposed to match different particular domains, such as drawings/paintings to photographs [Russell et al. 2011], sketches to photos [Chen et al. 2009], and photos under different illuminants [Chong et al. 2008]. However, these methods typically

*Supplemental material: https://liaojing.github.io/html/data/analogy_supplemental.pdf; source code: https://github.com/msracver/Deep-Image-Analogy.

†This work was done when Yuan Yao was an intern at MSR Asia.

are very domain-specific and do not easily generalize. Schechtman and Irani [2007] propose a more general solution using local self-similarity descriptors that are invariant across visual domains, and Shrivastava et. al. [2011] consider relative weighting between the descriptors for cross-domain image matching. Unfortunately, these methods do not produce dense correspondences between images of different domains.

We handle the dense correspondence problem using ideas related to *image analogies* [Hertzmann et al. 2001], which involve dense mappings between images from different domains. An image analogy is codified as $A : A' :: B : B'$, where B' relates to B in the same way as A' relates to A, and additionally, A and A' (also B and B') are in pixel-wise correspondences. In forming an image analogy, typically A, A', and either B or B' are given, and the goal is to solve for the sole remaining image. In contrast, for our scenario only a source image A and an example image B' are given, and both A' and B represent latent images to be estimated, imposing a bidirectional constraint to better match A to B'.

Solving the visual attribute transfer problem is equivalent to finding both unknown images A' and B. Instead of applying image analogy to image pixels directly, we use a Deep Convolutional Neutral Network (CNN) [Krizhevsky et al. 2012] to construct a feature space in which to form image analogies. It has been shown that such deep features are better representations for semantic structures [Zeiler and Fergus 2014]. We call our new technique *deep image analogy*.

Our approach leverages pre-trained CNNs for object recognition (e.g., VGG-19 [Simonyan and Zisserman 2014]) to construct such a feature space. A nice property of CNN representations is that they gradually encode image information from low-level details to high-level semantic content. This provides a good decomposition of semantic structure and visual attributes for transfer. Besides, the spatial correspondence between intermediate feature layers in CNN architectures is approximately maintained. Both properties facilitate a coarse-to-fine strategy for computing the nearest-neighbor field (NNF), a core component used in reconstructing images. To speed up the required nearest-neighbor search, we adapt PatchMatch [Barnes et al. 2009] to the CNN feature domain. Our method uses the bidirectional constraint in the patch similarity metric, which have previously been shown to work well on re-targeting problems [Simakov et al. 2008]. In the present context, the use of the bidirectional constraint introduces a useful symmetry and helps to mitigate the risk of mismatches.

Our major technical contributions are:

1. We present a new method "deep image analogy", which we show to be effective for visual attribute transfer across images in very different domains.

2. We extend PatchMatch and reconstruction from the image domain to the feature domain, which serves to guide semantically-meaningful visual attribute transfer.

We show how our deep image analogy technique can be effectively applied to a variety of visual attribute transfer cases, namely style/texture transfer, color/style swap, sketch/painting to photo, and time lapse. Our technique also has the effect of generating *two* results instead of the typical one generated from a one-way transfer. Such results can be seen in Figure 1. Our deep image analogy is designed to work on images with similar content composition. It is not effective for images which are semantically unrelated (e.g., a headshot and a countryside photo), and is not designed to handle large geometric variations (including scales and rotations).

2 Related Work

In this section, we review techniques that are related to visual attribute transfer (color, texture, and style transfer, as well as image analogy). We also briefly discuss two other topics very relevant to our work: dense image correspondence and neural style transfer.

2.1 Visual Attribute Transfer

Much work has been done on the transfer of various visual attributes (e.g., color, tone, texture, style) from one image to another, and we discuss only representative papers for brevity. Most of these approaches are, however, not general, in that they are designed to transfer a specific type of visual attribute. As we show later, our technique is designed to handle more types of visual attributes.

Color Transfer. Early color transfer techniques tend to be global, i.e., a global transformation is applied to a source image to match the color statistics of a reference image [Reinhard et al. 2001; Pitie et al. 2005]. They work well when the source and reference images are of similar scenes, even though the spatial layouts can be dissimilar. Other methods incorporate user input [An and Pellacini 2010; Welsh et al. 2002] or a large database [Dale et al. 2009; Laffont et al. 2014] to guide the color transfer. Local color transfer methods infer local color statistics in different color regions by establishing region correspondences [Tai et al. 2005]. More recently, Claudio et al. [2012] transfer local color between regions with the same annotated class; in a similar vein, Wu et al. [2013] transfer the color style across the same semantic regions.

Texture Transfer. Most early texture transfer algorithms rely on non-parametric sampling for texture synthesis while using different ways to preserve the structure of the target image. For instance, Efros and Freeman [2001] introduce a correspondence map that includes features of the target image such as image intensity to constrain the texture synthesis procedure. Ashikhmin [2003] focuses on transferring high-frequency texture information while preserving the coarse scale of the target image. Lee et al. [2010] improve this algorithm by additionally augmenting texture transfer with edge orientation information.

Style Transfer. Style transfer is used as a means to migrate an artistic style from an example image to a source image. The decomposition of content and style in artistic images is bound to the coupling between the source content and the example style. Zhang *et. al.* [2013] decompose the input images into three components: draft, paint, and edge; the style is transferred from the template image to the source image through the paint and edge components. Frigo *et. al.* [2016] view style transfer as the composition of local texture transfer and global color transfer, and suggest a simple yet efficient adaptive image partition for the decomposition of style and content. Shih *et. al.* [2014] robustly transfer the local statistics of an example portrait onto a new one using a multi-scale technique that separates local detail from structure.

Image Analogy. Texture or style transfer can also be done in a more supervised manner, where a pair of images A and A' are manually registered, and the analog of image B (similar in style with A) is to be found (resulting in B'). This approach, called image analogy, is first reported in [Hertzmann et al. 2001] and extended in various ways [Cheng et al. 2008; Bénard et al. 2013; Reed et al. 2015; Barnes et al. 2015]. In our work, we apply the concept of image analogy as a weak supervision, in a scenario where only an example image B' and an original image A are given.

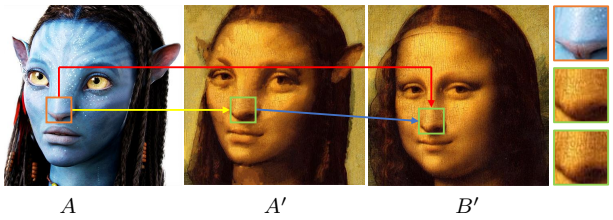


Figure 2: *Our method separates the difficult mapping $A \rightarrow B'$(red) into two tractable mappings: in-place mapping $A \rightarrow A'$ (yellow) and similar-appearance mapping $A' \rightarrow B'$ (blue).*

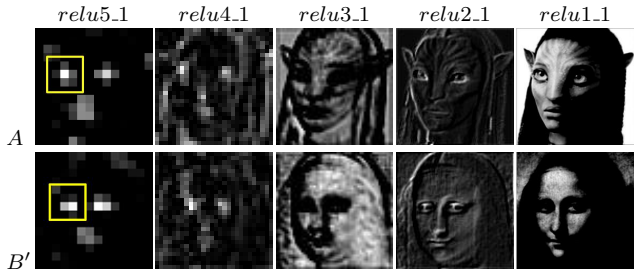


Figure 3: *The input image A (or B') is encoded by the CNN (e.g. VGG-19) as a set of filtered images at each layer. Here we visualize one representative filtered image for each layer.*

2.2 Dense Correspondence

Finding dense correspondences between two images is a fundamental problem in computer vision and graphics. Initial correspondence methods were designed for stereo matching, optical flow, and image alignment [Lucas and Kanade 1981]. These methods compute a dense correspondence field, but they assume brightness consistency and local motion, and may be hard to handle occlusion well.

The development of various local invariant features (e.g., SIFT [Lowe 2004]) has brought up significant progress. These features are robust to typical appearance variations (illumination, blur, compression), and a wide range of 2D transformations. Some methods combine sparse features with dense matching to cope with large-displacement optical flow [Brox et al. 2009], while others perform matching of visually different scenes (e.g., SIFT-flow [Liu et al. 2011], Deep flow [Weinzaepfel et al. 2013], Daisy flow [Yang et al. 2014], and Region foremost [Shen et al. 2016]). More recently, CNN-based features such as outputs of a certain convolution layer [Long et al. 2014]), object proposals [Ham et al. 2015]) or end-to-end trained network [Zhou et al. 2016] have been employed with flow algorithms, and shown potential to better align intra-class objects than handcrafted features (e.g., SIFT [Lowe 2004]). However, these methods assume locality and smoothness of the flow and thus may fail to align objects under large displacements or non-rigid deformations.

PatchMatch [Barnes et al. 2009] relaxes the rigidity assumption, and is a fast randomized algorithm for finding a dense NNF for patches. There are two extensions to handle patch variations in geometry and appearance. The Generalized PatchMatch [Barnes et al. 2010] algorithm allows these patches to undergo translations, rotations, and scale changes. NRDC [HaCohen et al. 2011] handles consistent tonal and color changes through iterative matching and refinement of appearance transformations. Recently, a multi-scale patch generazation called "Needle" [Lotan and Irani 2016] has been shown to facilitate reliable matching of degraded images. These approaches are still fundamentally based on low-level features, and as a result, fail to match images that are visually very different but semantically similar. Our proposed technique seeks to address this problem.

2.3 Neural Style Transfer

Our matching approach uses deep features generated by Deep Convolutional Neural Networks (CNN) [Krizhevsky et al. 2012]. It has been shown in high-level image recognition tasks that such deep features are better representations for images [Zeiler and Fergus 2014]. DeepDream [Alexander Mordvintsev 2015] is a recent attempt to generate artistic work using a CNN. This inspired work on neutral style transfer [Gatys et al. 2016b], which successfully applied CNN (pre-trained VGG-16 networks [Simonyan and Zisserman 2014]) to the problem of style transfer, or texture transfer [Gatys et al. 2015].

This method is able to produce more impressive stylization results than traditional texture transfer, since a CNN is effective in decomposing content and style from images. This idea is further extended to portrait painting style transfer [Selim et al. 2016] by adding face constraints. The most related work to ours is patch-based style transfer by combining a Markov Random Field (MRF) and a CNN [Li and Wand 2016a]. Yang et al. [2016] futher extend this idea to image inpainting. These two works also use patch similarity metric based on CNN features, but it only serves as a data term to optimize pixel values of the output image. They do not explicitly establish the correspondence between two images as our method does.

Another approach of neural style transfer is to directly learn a feed-forward generator network for a specific style. For example, Johnson et al. [2016] define a perceptual loss function to help learn a transfer network designed to produce results that match those of Gatys et al. [Gatys et al. 2016b]. Ulyanov et al. [2016] propose a texture network for both texture synthesis and style transfer. Li and Wand [2016b] introduce a Markovian Generative Adversarial Network to speed up their previous approach [Li and Wand 2016a]. Chen et al. [2017] and Dumoulin et al. [2016a] further extend the style transfer network from a single style to multiple styles.

Despite major progress, these methods are unable to guarantee that the transferred results are structure-preserving. In addition, they often generate stylization results with texture that is randomly distributed. In contrast, our technique transfers style in a more structure-preserving manner, and this is due to semantic-based dense correspondences.

3 Motivation

Given an image pair A and B', which may differ in appearance but have similar semantic structure, the goal is to find the mapping from A to B' (or from B' to A) for visual attribute transfer. It is assumed that the image pair has different visual attributes.

Analogy with Bidirectional Constraint. It is non-trivial to directly match A and B' due to their appearance differences. Our solution is to formulate the mapping as a problem of *image analogies* [Hertzmann et al. 2001]: $A : A' :: B : B'$, where A' and B are unknown latent variables. This analogy implies two constraints: 1) A and A' (also B and B') correspond at the same spatial position; 2) A and B (also A' and B') are similar in appearance (color, lighting, texture and etc.). As illustrated in Figure 2, the difficult mapping problem from A and B' (red arrow) can be separated into one in-place mapping from A to A' (yellow arrow) and one similar-appearance

mapping from A' to B' (blue arrow). The mapping from B' to A is achieved in the same way with the help of B.

These forward and reverse mapping functions between images A and B' are denoted as $\Phi_{a\to b}$ and $\Phi_{b\to a}$, respectively. More specifically, $\Phi_{a\to b}$ maps a pixel at location p from A to B'. Because of in-place mappings $A \to A'$ and $B \to B'$, $\Phi_{a\to b}$ is also the mapping from A to B, A' to B', and A' to B. It is found by imposing the constraints

$$A(p) = B(\Phi_{a\to b}(p)) \text{ and } A'(p) = B'(\Phi_{a\to b}(p)). \quad (1)$$

$\Phi_{b\to a}$ is the reverse mapping. To further achieve symmetric and consistent mappings, we consider the bidirectional constraint, namely $\Phi_{b\to a}(\Phi_{a\to b}(p)) = p$ and $\Phi_{a\to b}(\Phi_{b\to a}(p)) = p$. The latent images A', B and mapping functions $\Phi_{a\to b}$, $\Phi_{b\to a}$ are alternatively optimized in our objective function described in Section 4.

Reconstruction using CNN. The recovery of latent images A' and B is crucial in our method. Here we discuss the reconstruction of A'; B is recovered in the same way. The ideal A' should comprise the content structure from A and corresponding visual details from B'. We solve this problem using an image decomposition and reconstruction hierarchy. We adopt recent CNNs trained on an object recognition dataset, which compresses image information progressively from precise appearance (fine) to actual content (coarse) [Gatys et al. 2016b] (Figure 3). At the coarsest layer, A and B' may have very similar content information for better matching as indicated by yellow rectangles. As a result, we can assume A' to be A at this layer of the CNN. In contrast, this assumption fails in other image decompositions such as the Laplacian pyramid, where A and B' may still differ remarkably in colors at the coarsest layer.

At other layers of the CNN, we selectively extract content structures from features of A and detail information from features of B' by a weighted mask, which helps construct a linearly weighted combination to recover features of A'. We will describe the implementation details in Section 4.3. The updated latent images will carry the fusion information to the next layer for further mapping refinement.

Deep PatchMatch. Given latent images A' and B, inferring $\Phi_{a\to b}$ and $\Phi_{b\to a}$ is equivalent to computing a forward NNF and a reverse NNF between A and B as well as between A' and B'. More specifically, the forward NNF maps A to B; this NNF also maps A' to B'. The reverse NNF is similarly defined. PatchMatch [Barnes et al. 2009] is known as a randomized fast algorithm for computing approximate NNF between two images. Good patch matches can be found through random sampling, and spatial coherence in the imagery allows us to propagate such matches quickly to surrounding areas. Instead, we consider PatchMatch in a deep feature domain, which can provide better representations for semantic-level correspondences and be able to be incorporated into our CNN-based latent image reconstruction.

4 Deep Image Analogy

We achieve visual attribute transfer through image analogy and deep CNN features; an overall process we refer to as *deep image analogy*. Figure 4 shows the system pipeline. We first compute deep features for the input image pair A/B' through pre-trained CNN, and initialize feature maps of two latent images A'/B at the coarsest CNN layer in the preprocessing step (in Section 4.1). Then, at each layer, we compute a forward NNF and a reverse NNF that establish correspondences between feature maps of A and B as well as between feature maps of A' and B'. (Section 4.2). The extracted NNFs together with feature maps are used to reconstruct features

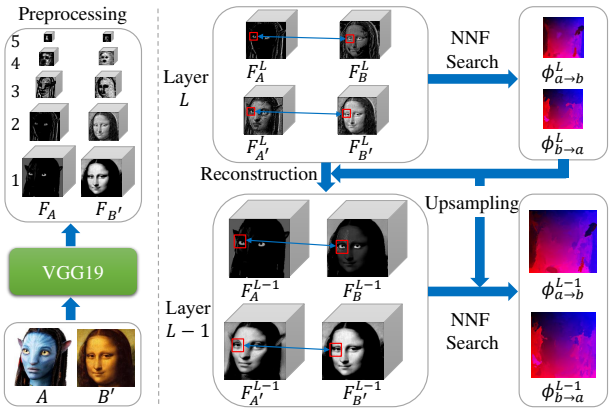


Figure 4: *System pipeline.*

of latent images A'/B at the next CNN layer (in Section 4.3). The NNFs obtained at the current layer are further upsampled to the next layer as their initialization (in Section 4.4). These three steps (NNF search, latent image reconstruction and NNF upsampling) are repeated at each layer, updating correspondences from coarse to fine.

4.1 Preprocessing

Our algorithm starts with precomputing feature maps by a VGG-19 network [Simonyan and Zisserman 2014] that is trained on the ImageNet database [Russakovsky et al. 2015] for object recognition. We obtain the pyramid of feature maps $\{F_A^L\}$ and $\{F_{B'}^L\}$ ($L = 1...5$) for the input images A and B'. The feature map of each layer is extracted from the *reluL_1* layer. It is a 3D tensor with $width \times heights \times channel$, and its spatial resolution increases from $L = 5$ to 1, as shown on Figure 4(*left*).

The features of A' and B are unknown. We estimate them in a coarse-to-fine manner, which needs a good initialization at the coarsest layer ($L = 5$). Here, we let $F_{A'}^5 = F_A^5$ and $F_B^5 = F_{B'}^5$ initially, that is, we view A and A' (also B and B') to be very similar at the top layer where the images have been transformed by the CNN into representations with the actual content, but being invariant to precise appearance (as shown in Figure 3).

4.2 Nearest-neighbor Field Search

At layer L, we estimate a forward NNF and a reverse NNF; they are represented by mapping functions $\phi_{a\to b}^L$ and $\phi_{b\to a}^L$, respectively. $\phi_{a\to b}^L$ maps a point in feature map F_A^L to another in feature map F_B^L. Note that $\phi_{a\to b}^L$ also maps $F_{A'}^L$ to $F_{B'}^L$. $\phi_{b\to a}^L$ is similarly defined in the reverse direction. $\phi_{a\to b}^L$ is computed by minimizing the following energy function:

$$\phi_{a\to b}^L(p) = \arg\min_q \sum_{x\in N(p), y\in N(q)} (||\overline{F}_A^L(x) - \overline{F}_B^L(y)||^2 + ||\overline{F}_{A'}^L(x) - \overline{F}_{B'}^L(y)||^2), \quad (2)$$

where $N(p)$ is the patch around p. We set the patch size to be 3×3 when $L = 5, 4, 3$, and 5×5 when $L = 2, 1$. For each patch around pixel p in the source A (or A'), we find its nearest neighbor position $q = \phi_{a\to b}^L(p)$ in the target B (or B'). The mapping function $\phi_{b\to a}^L(p)$ is similarly computed.

$F(x)$ in Equation (2) is a vector that represents all channels of the L-th feature layer at position x. We use normalized features

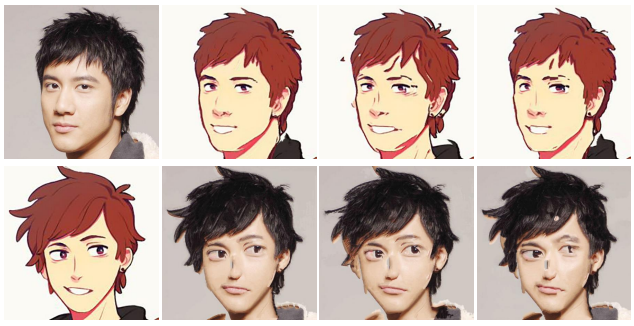


Figure 5: *Benefits of bidirectional constraint and deconvolution. (a)input images (b) bidirectional + deconvolution (c) single-direction + deconvolution (d) bidirectional + resampling.*

$\overline{F}^L(x) = \frac{F^L(x)}{|F^L(x)|}$ in our patch similarity metric, because relative values are more meaningful than absolute values in networks.

Equation (2) does not require a regularization term because local smoothness is implicitly achieved through aggregation of overlapping patches. Such a unary-only energy formulation can be efficiently optimized with the PatchMatch method [Barnes et al. 2009]. We adapt this approach of random search and propagation to support two pairs of multi-channel feature maps.

Our NNF search considers the constraints imposed by forward and reverse NNFs in the following manner. As indicated in Equation (2), the estimation of $\phi^L_{a\to b}$ relies on four feature maps. Among them, the reconstruction of F^L_B depends on the reverse mapping at previous layer $\phi^{L+1}_{b\to a}$ which will be described in Section 4.3. In other words, $\phi^L_{a\to b}$ is constrained by $\phi^{L+1}_{b\to a}$ through F^L_B; symmetrically $\phi^L_{b\to a}$ is constrained by $\phi^{L+1}_{a\to b}$ through $F^L_{A'}$. With these constraints, $\phi^L_{a\to b}$ and $\phi^L_{b\to a}$ usually will agree, and thus discourages ambiguous *1-to-n* mapping. In contrast, if we consider only single-direction mapping, Equation (2) becomes:

$$\phi^L_{a\to b}(p) = \arg\min_q \sum_{x\in N(p), y\in N(q)} ||\overline{F}^L_{A'}(x) - \overline{F}^L_{B'}(y)||^2 \quad (3)$$

for optimizing the mapping $\phi^L_{a\to b}$. Unfortunately, it is prone to misalignment. We show a comparison in Figure 5(b)(c).

At every layer, we have to recover features of A' and B before NNF search. In this section, we will mainly discuss the reconstruction of A'; B can be estimated likewise. As mentioned in Section 3, the reconstruction of $F^{L-1}_{A'}$ at layer $L-1$ is the fusion of content from F^{L-1}_A and details from $F^{L-1}_{B'}$. Let us define the feature map $R^{L-1}_{B'}$ to be the modified version of $F^{L-1}_{B'}$ to fit the structure of A. $F^{L-1}_{A'}$ is computed using a weighted sum:

$$F^{L-1}_{A'} = F^{L-1}_A \circ W^{L-1}_A + R^{L-1}_{B'} \circ (1 - W^{L-1}_A), \quad (4)$$

where $\circ$ is element-wise multiplication on each channel of feature map, and W^{L-1}_A is a 2D weight map (with the elements ranging from 0 to 1) used to separate content structures from details. As mentioned in Section 4.1, F^{L-1}_A is pre-computed. Next, we will introduce how to compute $R^{L-1}_{B'}$ and W^{L-1}_A respectively.

Ideally, the $R^{L-1}_{B'}$ should be equal to the warped $F^{L-1}_{B'}$ with $\phi^{L-1}_{a\to b}$: $R^{L-1}_{B'} = F^{L-1}_{B'}(\phi^{L-1}_{a\to b})$. However, $\phi^{L-1}_{a\to b}$ is initially unknown at layer $L-1$. A naïve method is to directly upscale $\phi^L_{a\to b}$ to the dimension at layer $L-1$, and then warp $F^{L-1}_{B'}$ with it. Unfortunately, this method cannot preserve mapped structures from previous layers (in Figure 5(d)). This is because spatial correspondence between two adjacent layers cannot be exactly maintained due to non-linear modules in CNNs (i.e., ReLU, Max-Pooling).

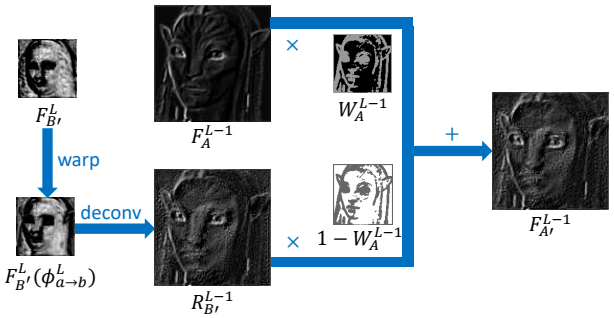


Figure 6: *Recovering features of latent image A' by weighted combination of the structures from A and the visual details sampled from B'.*

Figure 7: *Effect of different global offsets for weights $\{\alpha^L\}^4_{L=1}$.*

To address this problem, we present a new approach, which first warps the feature maps at the previous layer before deconvolving the warped features for the current layer. Specifically, we warp $F^L_{B'}$ by using $\phi^L_{a\to b}$, getting $F^L_{B'}(\phi^L_{a\to b})$. Let the sub-net of CNN including all computing units between layer $L-1$ and L be denoted as $\text{CNN}^L_{L-1}(\cdot)$. Our objective is to make the output of $\text{CNN}^L_{L-1}(R^{L-1}_{B'})$ be as close as possible to the target features $F^L_{B'}(\phi^L_{a\to b})$. Hence, $R^{L-1}_{B'}$, which is randomized initially, can be solved by minimizing the following loss function:

$$\mathcal{L}_{R^{L-1}_{B'}} = ||\text{CNN}^L_{L-1}(R^{L-1}_{B'}) - F^L_{B'}(\phi^L_{a\to b})||^2. \quad (5)$$

The gradient with respect to the feature values $\partial\mathcal{L}_{R^{L-1}_{B'}}/\partial R^{L-1}_{B'}$ can be used as input for some numerical optimization strategy. Here we use gradient descent (L-BFGS) [Zhu et al. 1994]. The optimization for feature deconvolution is similar to update the stylization result by pre-trained CNN, described in [Gatys et al. 2016b].

4.3 Latent Image Reconstruction

The role of the 2D weight map W^{L-1}_A is to separate content structures from details. Multiplying F^{L-1}_A (or $R^{L-1}_{B'}$) with W^{L-1}_A (or $1 - W^{L-1}_A$) helps extract content information from F^{L-1}_A (or details from $R^{L-1}_{B'}$). All feature channels share the same 2D weight map, which can be further be represented by

$$W^{L-1}_A = \alpha_{L-1} M^{L-1}_A, \quad (6)$$

where M^{L-1}_A is a function that specifies the magnitudes of neuron responses at layer L-1, in order to preserve content structures from A when they are present. We apply a sigmoid function to get $M^{L-1}_A(x) = \frac{1}{1+exp(-\kappa\times(|F^{L-1}_A(x)|^2-\tau))}$, with $\kappa = 300, \tau =$

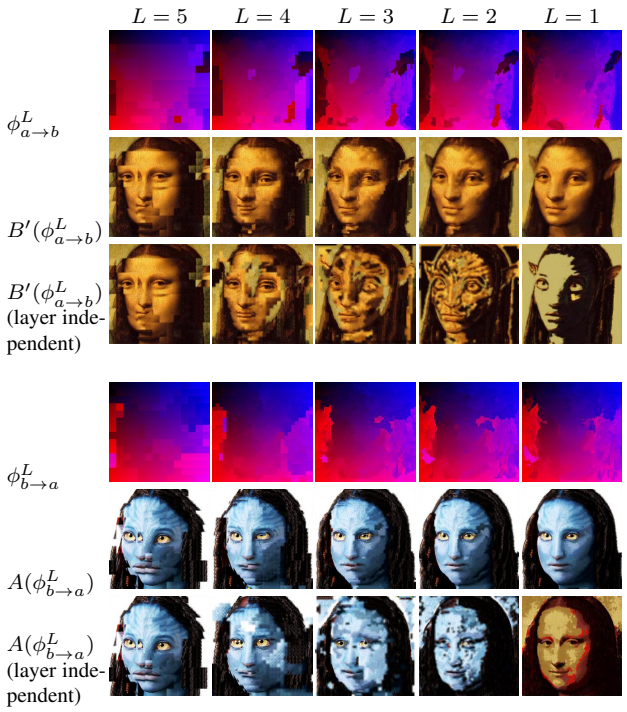


Figure 8: *Visualization of NNFs (top row), reconstructed results with NNFs (middle row) and reconstructed results based on layer-independent NNFs (bottom row) of each layer.*

0.05, and $|F_A^{L-1}(x)|^2$ being normalized to $[0, 1]$. In the reverse direction, $M_{B'}^{L-1}$ is computed using $F_{B'}^{L-1}$ instead of F_A^{L-1}.

The scalar α^{L-1} (in Equation (6)) controls the trade-off between content and attribute such as style, which is similar to [Gatys et al. 2016b]. When $\alpha^{L-1} \to 1$, it means that A' should be exactly consistent with A in structure. Reducing α^{L-1} would lead to structure variations in A' compared to A. The coarse layer represents more structure information than the fine layer due to higher-level abstraction, so we gradually decrease α_{L-1} from coarse to fine layer. For all of our visual attribute transfer results, the default setting is $\{\alpha^L\}_{L=4,3,2,1} = \{0.8, 0.7, 0.6, 0.1\}$. We only vary the default settings using two global offsets, namely $\{\alpha^L\}_{L=1}^4 \pm 0.1$. Figure 7 shows the effect of changing the global offset. Larger weights tend to produce results with more similar structure to inputs (A or B').

In summary, latent image reconstruction involves feature map warp at the current layer, deconvolution for the next layer, and fusion, as illustrated in Fig 6.

4.4 Nearest-neighbor Field Upsampling

Our NNFs are computed in a coarse-to-fine manner. At the coarsest layer, the mappings $\phi^5_{a\to b}$ and $\phi^5_{b\to a}$ are randomly initialized ($L = 5$). For other layers, as an initial guess, we upsample $\phi^L_{a\to b}$, $\phi^L_{b\to a}$, to layer $L-1$. Since the NNF obtained by PatchMatch is only piece-wise smooth, we use nearest-neighbor interpolation instead of linear interpolation.

The upscaled NNFs only serve as initial guesses, and they are further refined by NNF search (described in Section 4.2). Additionally, the initial guess serves as guidance to limit the random search space at every layer. A similar scheme is adopted in the extension of PatchMatch algorithm [Hu et al. 2016]. In our work, the search space is limited to be the receptive field of the network at each layer. For VGG-19, the random search radii of layers $\{4, 3, 2, 1\}$ are $\{6, 6, 4, 4\}$, respectively. Figure 8(top rows) shows how our mappings are gradually optimized from coarse to fine. Compared with the layer-independent matching results (bottom rows of Figure 8), our hierarchical matching scheme successfully propagates the correspondences from coarse levels to fine levels where matches are ambiguous, as shown in Figure 8(middle rows).

4.5 Output

After we obtain the final NNFs $\phi^1_{a\to b}$ (also $\phi^1_{b\to a}$) at the lowest feature layer, we let the pixel-location mapping functions $\Phi_{a\to b}$ and $\Phi_{b\to a}$ equal to $\phi^1_{a\to b}$ and $\phi^1_{b\to a}$ respectively, since the features at the lowest layer have the same spatial dimension as the input images. We then reconstruct A' by patch aggregation in the pixel layer of image: $A'(p) = \frac{1}{n} \sum_{x\in N(p)} (B'(\Phi_{a\to b}(x))$, where $n = 5\times5$ is the size of patch $N(p)$. B is reconstructed in a similar way.

ALGORITHM 1: Deep Analogy algorithm

Input : Two RGB images A and B'.
Output: Two pixel-location mapping functions: $\Phi_{a\to b}$, $\Phi_{b\to a}$; and two RGB images A', B.

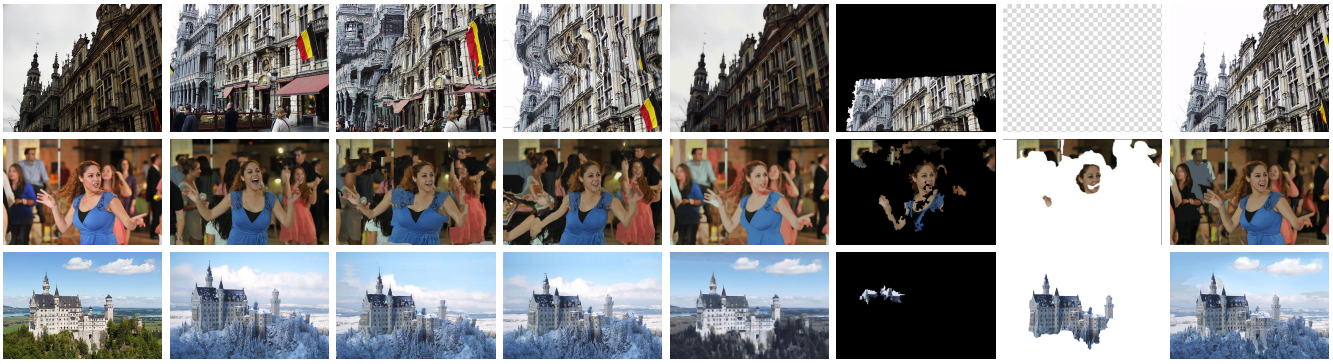
Preprocessing (Section 4.1):
$\{F_A^L\}_{L=1}^5, \{F_{B'}^L\}_{L=1}^5 \leftarrow$ feed A, B' to VGG-19 and get features.
$F_{A'}^5 = F_A^5$, $F_B^5 = F_{B'}^5$, and randomize mapping function $\phi^5_{a\to b}$, $\phi^5_{b\to a}$.
for $L = 5$ *to 1* **do**
  NNF search (Section 4.2):
    $\phi^L_{a\to b} \leftarrow$ map F_A^L to F_B^L, $F_{A'}^L$ to $F_{B'}^L$.
    $\phi^L_{b\to a} \leftarrow$ map F_B^L to F_A^L, $F_{B'}^L$ to $F_{A'}^L$.
  if $L > 1$ **then**
    Reconstruction (Section 4.3):
      Warp $F_{B'}^L$ with $\phi^L_{a\to b}$ to $F_{B'}^L(\phi^L_{a\to b})$.
      Deconvolve $R_{B'}^{L-1}$ with $F_{B'}^L(\phi^L_{a\to b})$ and $\text{CNN}_{L-1}^L(\cdot)$.
      $F_{A'}^{L-1} \leftarrow$ weighted blend F_A^{L-1} and $R_{B'}^{L-1}$.
      Warp F_A^L with $\phi^L_{b\to a}$ to $F_A^L(\phi^L_{b\to a})$
      Deconvolve R_A^{L-1} with $F_A^L(\phi^L_{b\to a})$ and $\text{CNN}_{L-1}^L(\cdot)$.
      $F_B^{L-1} \leftarrow$ weighted blend $F_{B'}^{L-1}$ and R_A^{L-1}.
    NNF upsampling (Section 4.4):
      Upsample $\phi^L_{a\to b}$ to $\phi^{L-1}_{a\to b}$
      Upsample $\phi^L_{b\to a}$ to $\phi^{L-1}_{b\to a}$
  end
end
$\Phi_{a\to b} = \phi^1_{a\to b}$, $\Phi_{b\to a} = \phi^1_{b\to a}$
$A'(p) = \frac{1}{n} \sum_{x\in N(p)} (B'(\Phi_{a\to b}(x))$,
$B(p) = \frac{1}{n} \sum_{x\in N(p)} (A(\Phi_{b\to a}(x))$

4.6 Algorithm and Performance

The pseudo code of our implementation is listed in Algorithm 1. Our core algorithm is developed in CUDA. All our experiments are conducted on a PC with an Intel E5 2.6GHz CPU and an NVIDIA Tesla K40m GPU. The runtime of each module is based on the input image with resolution 448×448. There are two bottlenecks. One is deep PatchMatch ($\sim$ 40 seconds), which needs to compute patch similarities on hundreds of feature channels. Another is feature deconvolution ($\sim$ 120 seconds), which may require hundreds of iterations to converge for Equation (5).

(a) Input (src) (b) Input (ref) (c) SIFT flow (d) DeepFlow2 (e) PatchMatch (f) Ours

Figure 9: *Comparison of different dense correspondences methods on input pairs with the same scene but slightly different views or motions. Some reconstruction errors are indicated by the red arrows.*



(a) Input (src) (b) Input (ref) (c) SIFT flow (d) DeepFlow2 (e) PatchMatch (f) NRDC (g) RFM (h) Ours

Figure 10: *Comparison of different dense correspondence methods on input pairs with the same scene but large variations in view, color, and tone. (For the first example, both public result and implementation of RFM are unavailable, so we have to make the result empty here.)*

(a) Input (src) (b) Input (ref) (c) SIFT flow (d) DeepFlow2 (e) PatchMatch (f) Daisy flow (g) Halfway (h) Ours

Figure 11: *Comparison of different dense correspondence methods on input pairs semantically-related but with vastly different styles.*

5 Evaluations on Matching

We evaluate the matching quality of our approach and state-of-the-art methods on three different categories of data: (I) the same scene, with very similar appearance and slight camera/object motions (e.g., neighboring frames in video); (II) the same scene, with variations in view, color, and tone (*e.g.*, two photos from different cameras or illumination); (III) semantically-related scene with vastly different styles (e.g., photograph and painting) or appearances. Our default parameter ranges are designed for images with very different appearance like those in category (II) and (III). So for the special case (I) which always requires two views to be perfectly aligned in structures, we specially set $\{\alpha^L\}_{L=4,3,2,1} = \{1.0, 1.0, 1.0, 1.0\}$ to achieve comparable quality to other methods. For category (II) we use default weights plus offset $+0.1$, and for category (III) we use the default weights. We keep automatic weights selection based on inputs in future work. All other approaches are based on author-provided implementations with the default settings.

	SIFT Flow	DeepFlow2	PatchMatch	ours
PSNR	27.97	30.04	35.18	34.16
Endpoint err.	0.91	0.34	22.40	3.21

Table 1: *Reconstruction accuracy and flow endpoint error of different dense correspondence estimation methods.*

Category (I) is tested on the Middlebury optical flow benchmark dataset, total 23 color image pairs. We compared with PatchMatch [Barnes et al. 2009], SIFT Flow [Liu et al. 2011] and DeepFlow2 [Weinzaepfel et al. 2013]. We evaluate the matching accuracy by both flow endpoint error and reconstruction errors (*i.e.*, PSNR metric). Similar to Patchmatch, our method does not assume any flow continuity, so it performs better than the flow methods (*i.e.*, Sift Flow and DeepFlow2) in reconstruction error but worse in flow endpoint error, as shown on Table 1. Since the task of this paper is image reconstruction rather than pure motion estimation, the recon-

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	**mean**
PatchMatch [Barnes et al. 2009]	6.5	6.3	2.6	2.9	2.3	4.7	3.3	12.5	2.0	0.0	4.2	2.6	4.2
SIFT Flow [Liu et al. 2011]	8.1	**14.3**	5.1	26.1	25	20.9	13.3	6.3	14.3	15.4	4.2	44.7	16.5
Cycle consistency [Zhou et al. 2016]	12.9	6.3	10.3	**39.1**	27.3	**23.3**	13.3	**12.5**	6.1	**19.2**	**12.5**	36.8	18.3
Ours	**19.4**	7.9	**15.4**	27.5	**47.7**	11.6	**20.0**	6.3	**18.4**	15.4	**12.5**	**50.0**	**21.0**

Table 2: *Correspondence accuracy measured in PCK. The test is conducted on randomly selected 20 pairs of each category of the PASCAL 3D+ dataset. Overall, our method outperforms the three baselines.*

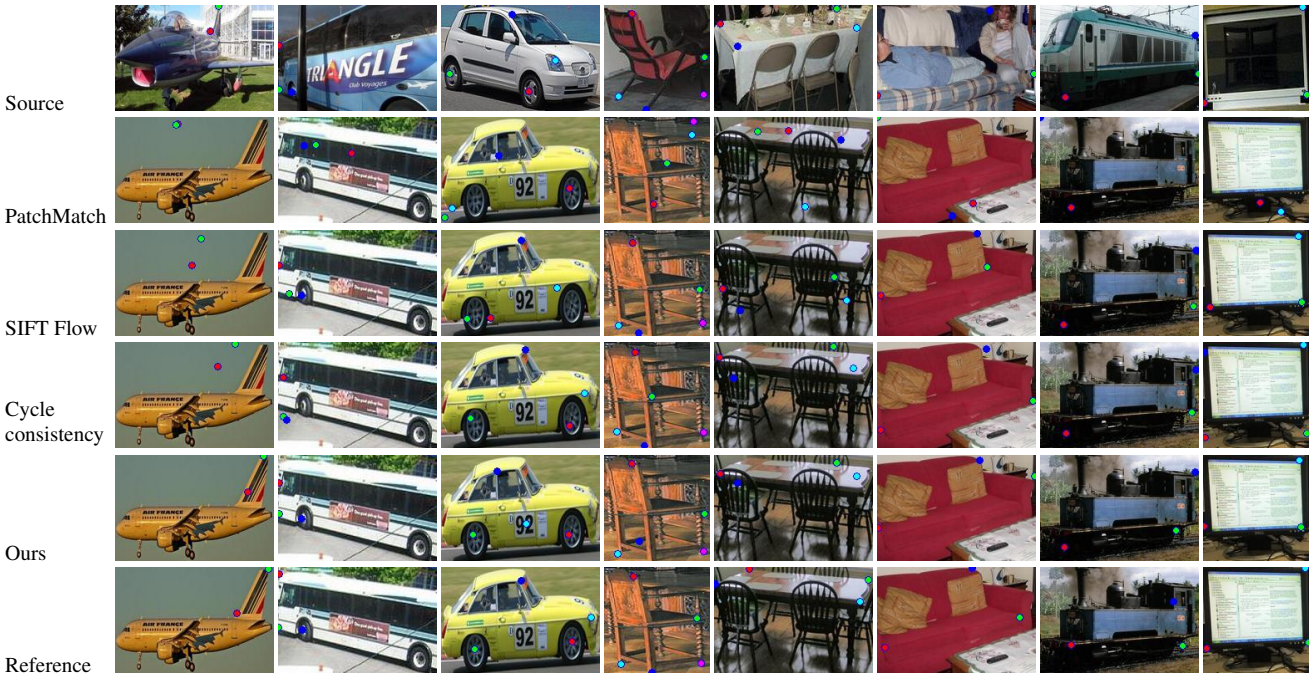


Figure 12: *Visual comparison of keypoint transfer performance for different methods on representative examples of the PASCAL 3D+ dataset.*

struction error makes more sense in our applications. Note that our reconstructed results look slightly worse than PatchMatch. This is because our approach does not consider color similarity as well as Patchmatch, which causes inconsistent color matches in our results. Figure 9 shows a visual comparison of the three methods and ours.

For category (II), we collect public data from [Yang et al. 2007; HaCohen et al. 2011; Shen et al. 2016]; we show some examples in Figure 10. The input pair of images differ significantly in object pose, illumination, and season. Since PatchMatch [Barnes et al. 2009] is based on color similarity, it may fail to match similar structures with different colors, leading to a result with appearance that is close to the source image (see Figure 10(e)). In contrast, SIFT flow [Liu et al. 2011] and Deepflow [Weinzaepfel et al. 2013] are better in matching structures (see Figure 10(c)(d)) because sparse features are matched instead of color. However, there are noticeable mismatches in areas without feature points. Regional foremost matching (RFM) [Shen et al. 2016] and NRDC [HaCohen et al. 2011] work well on these regions where matches are found. However, RFM fails for large non-rigid motion (Figure 10(g))) and NRDC can reliably match only a small fraction of the image in some cases (Figure 10(f)). Our results (Figure 10(h)) look visually comparable with either NRDC or RFM at regions with high confidence, and provide a better guess elsewhere.

For category (III), we collect pairs of photo and painting online by searching generic words such as “bird”, “girl”, and “portrait”. Two examples are shown in Figure 11. The task would be intractable for existing matching work [Liu et al. 2011; Weinzaepfel et al. 2013; Barnes et al. 2009; Yang et al. 2014], and we can see noticeable artifacts in their results (Figure 11(c)(d)(e)(f)). The Halfway morphing [Liao et al. 2014] addresses this kind of problem. However, their method relies on a modified SSIM term to find low-level feature correspondences, and needs user interaction to provide high-level matching. Without user input, they may fail as well on these cases (Figure 11(g)). By contrast, our approach is automatic and can produce visually acceptable results (Figure 11(h)).

We further evaluate on the Pascal 3D+ dataset [Xiang et al. 2014], which provides annotated sparse correspondences for semantically-related objects with remarkably different appearances. For each category in the Pascal 3D+ dataset, we randomly sample 20 image pairs from the training and validation datasets. Cycle consistency [Zhou et al. 2016] is a representative work considering high-level semantic information for dense matching. We conduct the same evaluation as that of Zhou et al. [Zhou et al. 2016] for all competing methods. The quality of correspondences is measured by the percentage of correct keypoint transfer (PCK) over all pairs suggested by Zhou et al. [Zhou et al. 2016]. The quantitative comparisons between different methods are shown in Table 2 and the visual comparisons on representative pairs are shown in Figure 12. Both our method and cycle consistency obtain better performance than methods based on low-level feature, e.g. SIFT Flow and PatchMatch. Overall, our method performs better than cycle consistency, even though the features we used are not trained on the Pascal 3D+

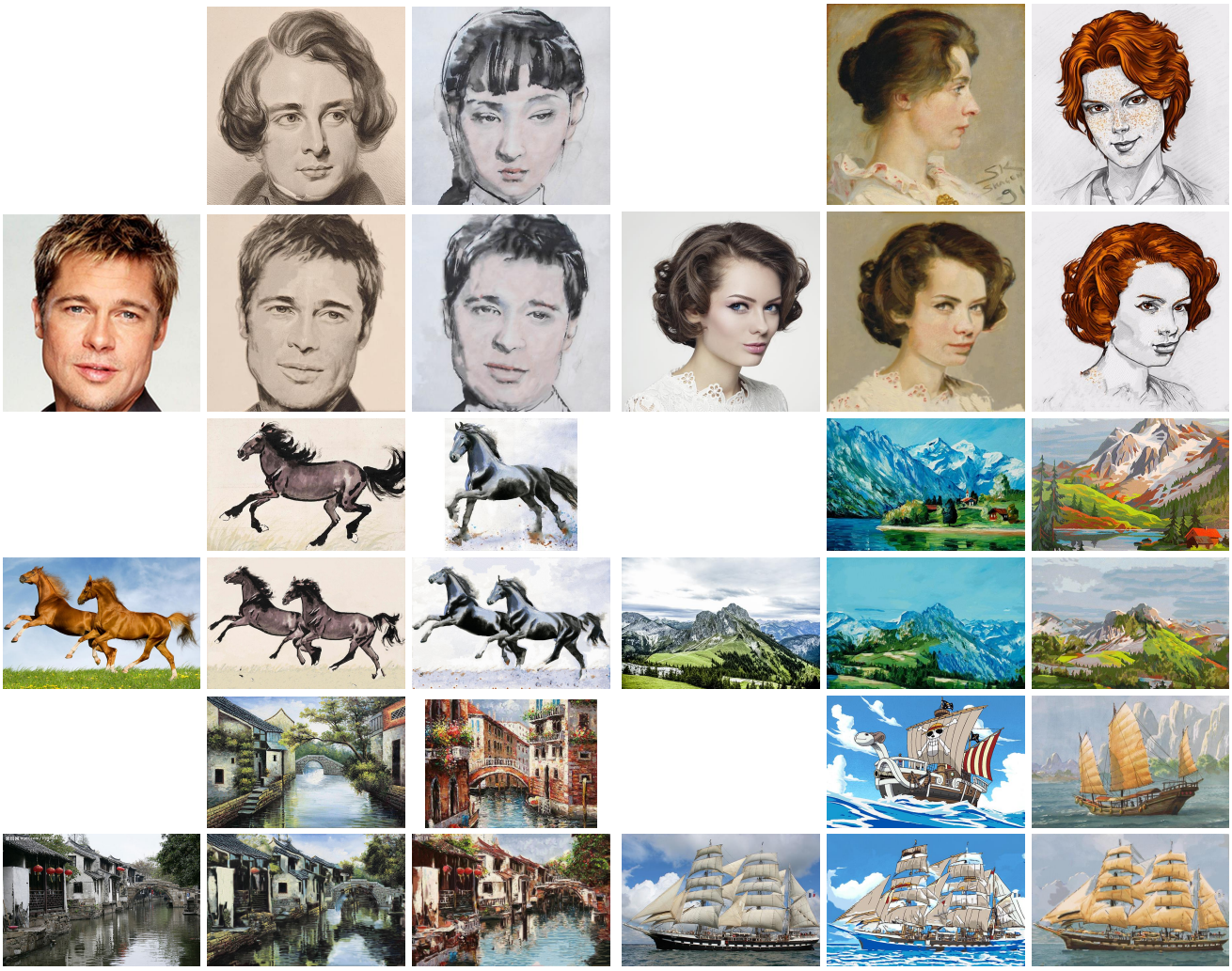


Figure 13: *Our photo-to-style transfer results. For each group, two images in the upper row are reference styles. The leftmost one in the lower row is the input photo, other two images are our results.*

dataset.

6 Applications

In this section, we show how our approach can be effectively used for four different tasks of visual attribute transfer: photo to style, style to style, style to photo, and photo to photo. For comparison with other methods, we again ran author-released implementations with default settings, or submit our images to their services or apps. More comparisons and results can be found in our supplemental material. All results are better viewed in electronic form.

6.1 Photo to Style

One key application of our approach is to transfer a photo to a reference artistic style. Users can easily borrow specific artwork stylization (possibly by famous or professional artists) to render their own photos for sharing and entertainment. A few examples are shown in Figure 13. The recent technique of neural style transfer [Alexander Mordvintsev 2015; Gatys et al. 2016b; Johnson et al. 2016; Li and Wand 2016a; Selim et al. 2016] generated very impressive stylization results, and some apps or services (*e.g.*, Prisma[1], Google Deep Style[2], Ostagram[3]) based on CNN are also very popular.

In comparison to these approaches, ours is capable of higher quality content-specific stylization that better preserves structures. Neural style [Gatys et al. 2016b] and perceptual loss methods [Johnson et al. 2016] rely on global statistics matching, and as a result, do not guarantee local structure-to-structure transfer. For example, the face in their results (top row of Figure 14) contains flower textures. If two inputs are totally unrelated as shown on top row of Figure 15, both ours and theirs are visually different but acceptable. Google Deep Style and Ostagram are two online consumer services. They also do not appear to transfer structures effectively in Figure 14. The portrait style approach [Selim et al. 2016] enforces constraints of face alignment to maintain the face structure. Ghosting artifacts on non-face regions are evident, as shown in bottom row of Figure 14. CNNMRF [Li and Wand 2016a] uses a non-parametric model, but different from ours, it is independent at each layer. In both methods, artifacts occur at misaligned regions. This can be

[1] http://prisma-ai.com/
[2] http://www.deepstylegenerator.com/
[3] https://ostagram.ru/

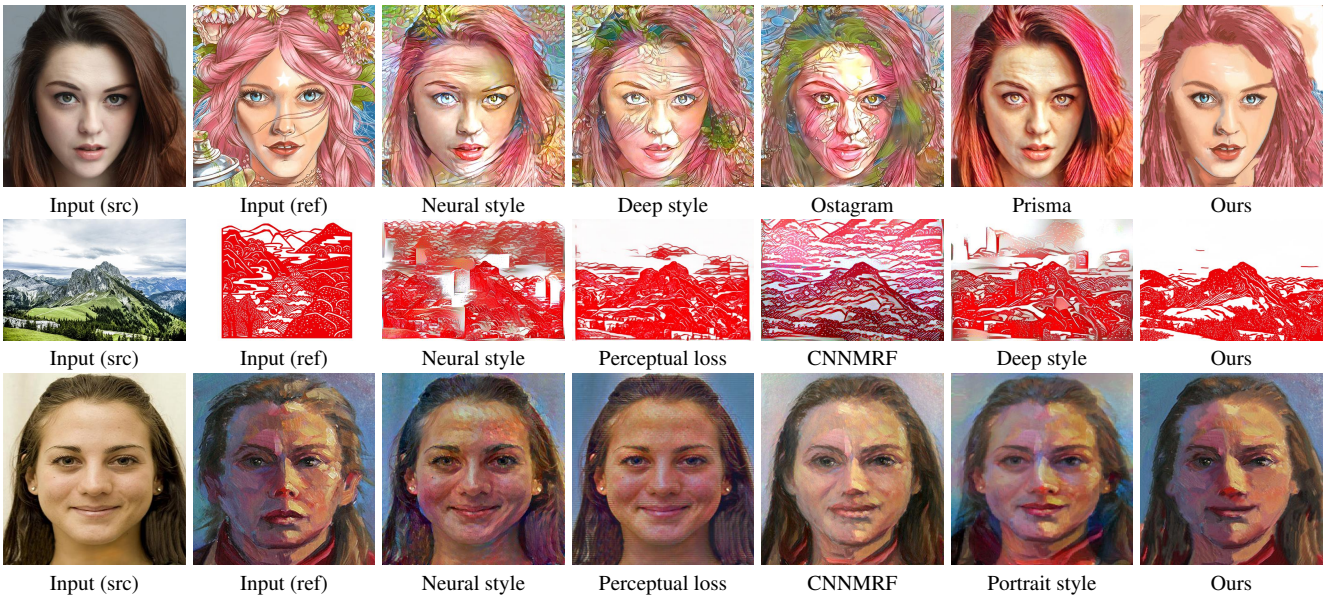


Figure 14: *Comparison with other style-transfer methods and apps based on neural network.*

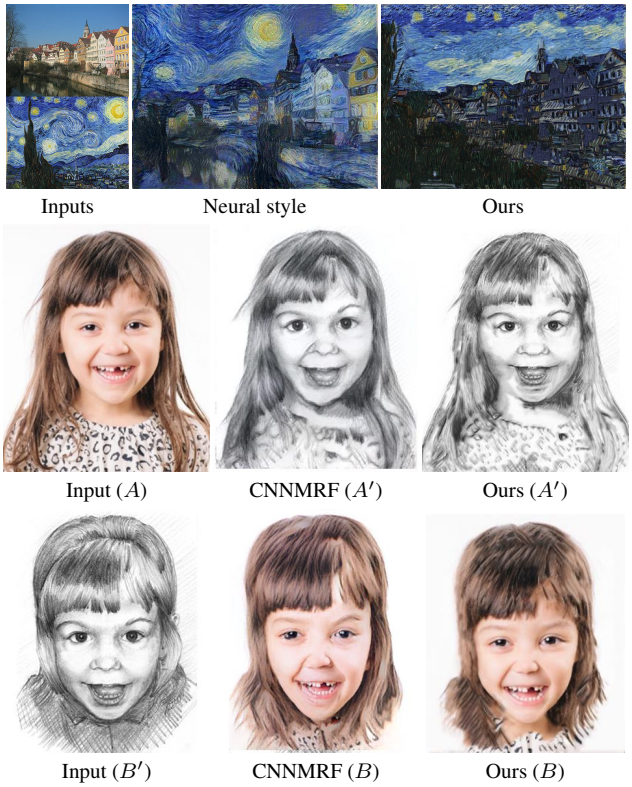


Figure 15: *Comparison with Neural style and CNNMRF on their examples.*

seen at the boundary between neck and shirt on ours and the left eye on theirs (Figure 15(*bottom row*)).

The other methods optimize for the pixel color using a loss function that combines content and style information, while ours directly samples reference patches using our estimated NNF. This is the primary difference between ours and other methods. Although our results less faithfully capture the content at times, they are able to better reproduce the style. As an example, while the Neural Style method has captured the appearance of the woman's face more faithfully than ours in Figure 14(*bottom row*), our version is a better reproduction of the style.

6.2 Style to Style

When input pairs of images are two content-related artworks but with vastly different styles, our method is able to swap the styles. To our knowledge, this is a new and fun effect, which has not been demonstrated before. For example, we are able to re-factor the "Pharaoh status" to the style of "Peking Opera facial makeup", as shown in Figure 16(*second row*). Our approach surprisingly works well for this application. Some results are shown in Figure 16.

6.3 Style to Photo

This can be seen as the inverse problem of photo to style. Unfortunately, this task is much more difficult than style transfer. One reason is that artworks tend to lack detail in favor of creativity. Another reason is that humans are very sensitive to slight structure misalignments; the quality bar for photorealism is known to be very high. Generally, our approach is less successful on this task compared with the photo-to-style application. However, our approach is still able to transfer sketches or paintings to real photographs, with more plausible results when both images are very related. Figure 17 show some example results.

We further compare our methods with the CG2Real [Johnson et al. 2011] on their provided examples, shown in Figure 18. They retrieve a small number of photos with similar global structure of the input CG, identify the corresponding regions by co-segmentation and transfer the regions to CG. In contrast to their region-to-region transfer, our approach builds a pixel-to-pixel mapping, which can better maintain the structures of the input CG image, shown in Figure 18(*first row*). Moreover, our results are visually comparable to theirs, even though we use only one reference photo while CG2Real uses multiple photos.

Figure 16: *Style-swap results.*

6.4 Photo to Photo

Photo to photo mapping can be in the form of color or tone transfer. Our approach enables local transfer in corresponding regions. Generally, only color or tone are borrowed from the reference, not allowing details. For this goal, we slightly modify our algorithm by applying a refinement on the reconstructed result. Specifically, we use weighted least squares filter (WLS) [Farbman et al. 2008] to smooth both input image A and reconstructed result A' with the guidance A. The final result $\widetilde{A}'$ is obtained by:

$$\widetilde{A}' = \text{WLS}(A', A) + A - \text{WLS}(A, A), \tag{7}$$

which implies that we only need to maintain colors of A' while discarding details of A', and filling in with details of A. It is similar for B. Without the refinement, the details from the reference will introduce distortion to the fine-grained structures of objects in the source image, for example, the eyes and nose in the comparison with and without the refinement (Figure 19).

We further show two examples compared with NRDC [HaCohen et al. 2011] in Figure 21(a). NRDC uses a global color mapping curve based on a small potion of matches. By contrast, our transfer is local, which may produce better region-to-region transfer than NRDC, as shown in the grass region of Figure 21(a) (*second row*). Such local transfer is sometimes sensitive to the quality of matching. For instance, we observe some inaccurate color mappings, like saturated sky region in Figure 21(a) (*first row*), caused by mismatched regions. Beyond the capability of NRDC, our approach can also transfer color across images with different scenes, such as the time-lapse example shown in Figure 20, due to the identified semantic correspondences, e.g., tree to tree, mountain to mountain.

Compared with other local color transfer methods, e.g. shih et al.[Shih et al. 2013] and Luan et al.[Luan et al. 2017] in Figure 21(c), our method can produce visually comparable results to theirs. However, our algorithm takes a single image as reference, while their methods require either a full time-lapse video [Shih et al. 2013] or optional precomputed segmentation masks [Luan et al. 2017]. When compared with Luan et al.[Luan et al. 2017] carefully, we can find some undesired posterization effects (e.g., forehead in Figure 21(b) and buildings in Figure 21(c)), which do not occur in ours. Compared with Shih et al. [Shih et al. 2014], both ours and Luan et al. [Luan et al. 2017] fail to transfer some details (e.g., eye highlights) as shown in Figure 21(b).

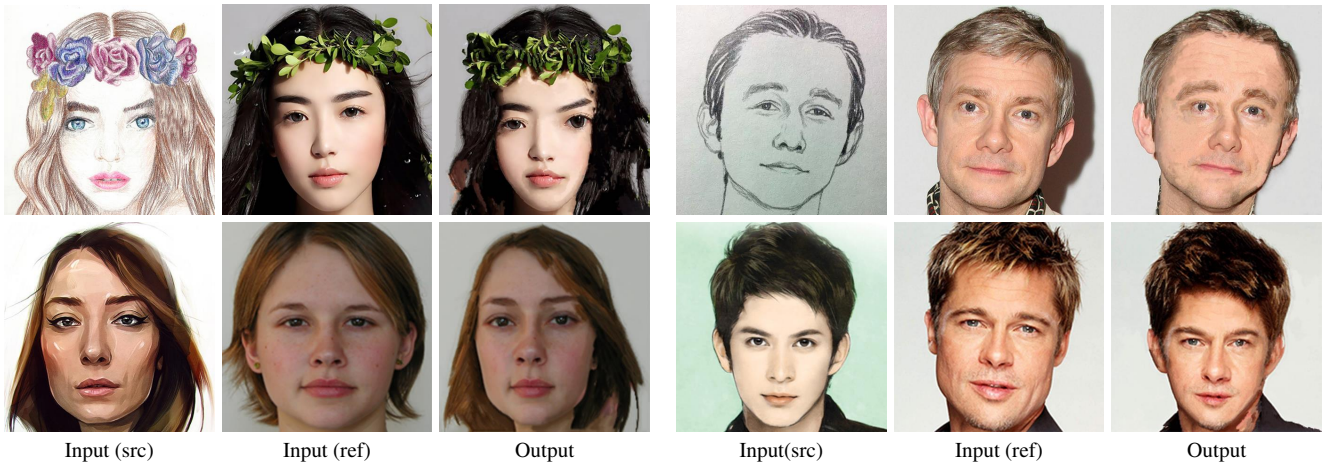


Figure 17: *Results of converting a sketch or a painting to a photo.*

Figure 18: *Comparisons of our CG-to-photo results with CG2Real method on their examples. For the inputs in each group, the source CG is used by both ours and theirs, but the reference photo is only for ours. Their multiple reference photos are not given in their paper.*

Figure 19: *Comparisons of transfer color with (middle column) and without (right column) WLS refinement.*

7 Discussion

Semantic-level matching is not fool-proof, since our deep match approach relies on a pre-trained VGG network that has limited capability in representing all objects or fine-grained structures. A possible improvement would be to train our network on a domain-specific dataset for better image representation. If an object is found in one image but not the other, it is not clear how semantic matching can be done. We show a failure case in Figure 22(*top-left*), which shows the mismatched hat region.

For the scenes which are semantically related but vary a lot in scales and view points, our method still fails to build correct correspondences as shown in Figure 22(*top-right*). Addressing these cases would require either pre-scaling images or allowing patches to be rotated and scaled in our NNF search like Generalized PatchMatch. However adding rotation or scale is nontrivial since geometric transformations in the image domain are not well preserved with those in feature domain because of non-linear modules (*e.g.*, ReLU).

Moreover, our method may fail to find correspondences in textureless regions that have very low neural activation, like the background in Figure 22(*bottom-right*). This problem may be addressed by either analyzing pixel intensity or by explicitly enforcing smoothness in our energy function.

Our photo-to-style transfer application is unable to produce geometry style transfer, like the case shown in Figure 22(*bottom-left*). The assumption in our work is to maximally preserve content structure.

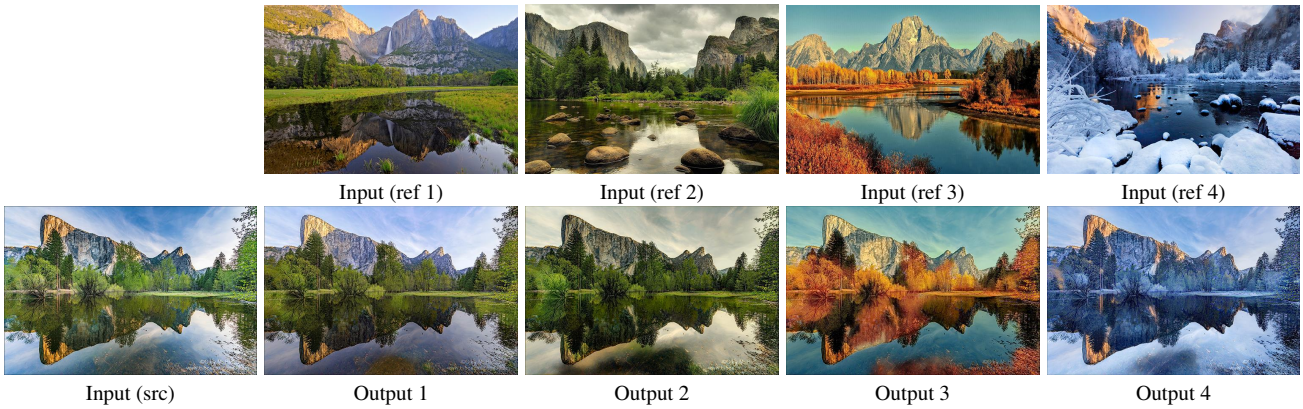


Figure 20: *Results of generating time-lapse sequences with references of another semantic-related scene.*

Figure 21: *Comparison of our photo-to-photo results with other state-of-art methods on their examples. For the inputs in each group, the upper one is the source photo and the lower one is the reference photo.*

We may relax the assumption in future work.

8 Concluding Remarks

We have demonstrated a new technique for transferring visual attributes across semantically-related images. We adapted the notion of image analogy to a deep feature space for finding semantically-meaningful dense correspondences. We show that our method outperforms previous methods where image pairs exhibit significant variance in their appearance including lighting, color, texture, and style. We have shown that our approach is widely applicable for visual attribute transfer in real-world images, as well as additional transfer challenges such as content-specific stylization, style to photo, and style to style. We believe this method may also be proven useful for a variety of computer graphics and vision applications that rely on semantic-level correspondences.

Figure 22: *Some examples of failure cases.*

Acknowledgements

We thank the anonymous reviewers for helping us to improve this paper, David Wipf for his help in proofreading. We also acknowledge to the authors of our image and style examples but we do not own the copyrights of them.

References

ALEXANDER MORDVINTSEV, CHRISTOPHER OLAH, M. T., 2015. Inceptionism: Going deeper into neural networks.

AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Computer Graphics Forum*.

ASHIKHMIN, M. 2003. Fast texture transfer. *IEEE Comput. Graph. and Appl. 23*, 4.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. (Proc. of SIGGRAPH) 28*, 3.

BARNES, C., SHECHTMAN, E., GOLDMAN, D. B., AND FINKELSTEIN, A. 2010. The generalized PatchMatch correspondence algorithm. In *Proc. ECCV*.

BARNES, C., ZHANG, F.-L., LOU, L., WU, X., AND HU, S.-M. 2015. Patchtable: Efficient patch queries for large datasets and applications. *ACM Trans. Graph. (Proc. of SIGGRAPH) 34*, 4.

BÉNARD, P., COLE, F., KASS, M., MORDATCH, I., HEGARTY, J., SENN, M. S., FLEISCHER, K., PESARE, D., AND BREEDEN, K. 2013. Stylizing animation by example.

BROX, T., BRUHN, A., PAPENBERG, N., AND WEICKERT, J. 2004. High accuracy optical flow estimation based on a theory for warping. In *Proc. ECCV*.

BROX, T., BREGLER, C., AND MALIK, J. 2009. Large displacement optical flow. In *Proc. CVPR*.

CHEN, T., CHENG, M.-M., TAN, P., SHAMIR, A., AND HU, S.-M. 2009. Sketch2photo: Internet image montage. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia) 28*, 5.

CHEN, D., YUAN, L., LIAO, J., YU, N., AND HUA, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *Proc. CVPR*.

CHENG, L., VISHWANATHAN, S., AND ZHANG, X. 2008. Consistent image analogies using semi-supervised learning. In *CVPR*.

CHONG, H. Y., GORTLER, S. J., AND ZICKLER, T. 2008. A perception-based color space for illumination-invariant image processing. *ACM Trans. Graph. (Proc. of SIGGRAPH) 27*, 3.

CLAUDIO, C., FRANCESCA, G., AND RAIMONDO, S. 2012. Color transfer using semantic image annotation. vol. 8299.

DALE, K., JOHNSON, M. K., SUNKAVALLI, K., MATUSIK, W., AND PFISTER, H. 2009. Image restoration using online photo collections. In *Proc. ICCV*.

DUMOULIN, V., SHLENS, J., AND KUDLUR, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

DUMOULIN, V., SHLENS, J., AND KUDLUR, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proc. ACM SIGGRAPH*, 341–346.

ELAD, M., AND MILANFAR, P. 2016. Style-transfer via texture-synthesis. *arXiv preprint arXiv:1609.03057*.

FARBMAN, Z., FATTAL, R., LISCHINSKI, D., AND SZELISKI, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Trans. Graph. (Proc. of SIGGRAPH)*, vol. 27, ACM.

FRIGO, O., SABATER, N., DELON, J., AND HELLIER, P. 2016. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proc. CVPR*.

GATYS, L., ECKER, A. S., AND BETHGE, M. 2015. Texture synthesis using convolutional neural networks. In *Proc. of NIPS*.

GATYS, L. A., BETHGE, M., HERTZMANN, A., AND SHECHTMAN, E. 2016. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*.

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2016. A neural algorithm of artistic style. In *Proc. CVPR*.

HACOHEN, Y., SHECHTMAN, E., GOLDMAN, D. B., AND LISCHINSKI, D. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Trans. Graph. (Proc. of SIGGRAPH) 30*, 4.

HAM, B., CHO, M., SCHMID, C., AND PONCE, J. 2015. Proposal flow. *arXiv preprint arXiv:1511.05065*.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. In *Proc. ACM SIGGRAPH*.

HU, Y., SONG, R., AND LI, Y. 2016. Efficient coarse-to-fine patchmatch for large displacement optical flow. In *Proc. CVPR*.

JOHNSON, M. K., DALE, K., AVIDAN, S., PFISTER, H., FREEMAN, W. T., AND MATUSIK, W. 2011. Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Trans. on Visualization and Computer Graphics 17*, 9.

JOHNSON, J., ALAHI, A., AND FEI-FEI, L. 2016. Perceptual losses for real-time style transfer and super-resolution. *Proc. ECCV*.

KINGMA, D., AND BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Proc. of NIPS*.

LAFFONT, P.-Y., REN, Z., TAO, X., QIAN, C., AND HAYS, J. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Trans. Graph. (Proc. of SIGGRAPH) 33*, 4.

LEE, H., SEO, S., RYOO, S., AND YOON, K. 2010. Directional texture transfer.

LI, C., AND WAND, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proc. CVPR*.

LI, C., AND WAND, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. *arXiv preprint arXiv:1604.04382*.

LIAO, J., LIMA, R. S., NEHAB, D., HOPPE, H., SANDER, P. V., AND YU, J. 2014. Automating image morphing using structural similarity on a halfway domain. *ACM Trans. Graph. 33*, 5.

LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLÁR, P., AND ZITNICK, C. L. 2014. Microsoft coco: Common objects in context. In *Proc. ECCV*, Springer.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 5.

LONG, J. L., ZHANG, N., AND DARRELL, T. 2014. Do convnets learn correspondence? In *Proc. of NIPS*, 1601–1609.

LOTAN, O., AND IRANI, M. 2016. Needle-match: Reliable patch matching under high uncertainty. In *Proc. CVPR*.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2.

LUAN, F., PARIS, S., SHECHTMAN, E., AND BALA, K. 2017. Deep photo style transfer. *arXiv preprint arXiv:1703.07511*.

LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. of Imaging Understanding Workshop*.

PETSCHNIGG, G., AGRAWALA, M., HOPPE, H., SZELISKI, R., COHEN, M., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. (Proc. of SIGGRAPH) 23*, 3.

PITIE, F., KOKARAM, A. C., AND DAHYOT, R. 2005. N-dimensional probablility density function transfer and its application to colour transfer. In *Proc. ICCV*.

REED, S. E., ZHANG, Y., ZHANG, Y., AND LEE, H. 2015. Deep visual analogy-making. In *Proc. of NIPS*.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. and Appl. 21*, 5.

RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M., ET AL. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision 115*, 3, 211–252.

RUSSELL, B. C., SIVIC, J., PONCE, J., AND DESSALES, H. 2011. Automatic alignment of paintings and photographs depicting a 3d scene. In *3D Representation for Recognition*.

SELIM, A., ELGHARIB, M., AND DOYLE, L. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph. (Proc. of SIGGRAPH) 35*, 4.

SHECHTMAN, E., AND IRANI, M. 2007. Matching local self-similarities across images and videos. In *Proc. CVPR*.

SHEN, X., TAO, X., ZHOU, C., GAO, H., AND JIA, J. 2016. Regional foremost matching for internet scene images. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia) 35*, 6.

SHIH, Y., PARIS, S., DURAND, F., AND FREEMAN, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph. (Proc. of SIGGRAPH) 32*, 6.

SHIH, Y., PARIS, S., BARNES, C., FREEMAN, W. T., AND DURAND, F. 2014. Style transfer for headshot portraits. *ACM Trans. Graph. (Proc. of SIGGRAPH) 33*, 4.

SHRIVASTAVA, A., MALISIEWICZ, T., GUPTA, A., AND EFROS, A. A. 2011. Data-driven visual similarity for cross-domain image matching. *ACM Trans. Graph. (Proc. of SIGGRAPH) 30*, 6.

SIMAKOV, D., CASPI, Y., SHECHTMAN, E., AND IRANI, M. 2008. Summarizing visual data using bidirectional similarity. In *CVPR*.

SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. 2015. Going deeper with convolutions. In *Proc. CVPR*, 1–9.

TAI, Y.-W., JIA, J., AND TANG, C.-K. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Proc. CVPR*.

ULYANOV, D., LEBEDEV, V., VEDALDI, A., AND LEMPITSKY, V. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*.

USTYUZHANINOV, I., BRENDEL, W., GATYS, L. A., AND BETHGE, M. 2016. Texture synthesis using shallow convolutional networks with random filters. *arXiv preprint arXiv:1606.00021*.

WEINZAEPFEL, P., REVAUD, J., HARCHAOUI, Z., AND SCHMID, C. 2013. Deepflow: Large displacement optical flow with deep matching. In *Proc. ICCV*.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. *Proc. ACM SIGGRAPH 21*, 3.

WU, F., DONG, W., KONG, Y., MEI, X., PAUL, J.-C., AND ZHANG, X. 2013. Content-Based Colour Transfer. *Computer Graphics Forum*.

XIANG, Y., MOTTAGHI, R., AND SAVARESE, S. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, IEEE, 75–82.

YANG, G., STEWART, C. V., SOFKA, M., AND TSAI, C.-L. 2007. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE transactions on pattern analysis and machine intelligence 29*, 11.

YANG, H., LIN, W.-Y., AND LU, J. 2014. Daisy filter flow: A generalized discrete approach to dense correspondences. In *Proc. CVPR*.

YANG, C., LU, X., LIN, Z., SHECHTMAN, E., WANG, O., AND LI, H. 2016. High-resolution image inpainting using multi-scale neural patch synthesis. *arXiv preprint arXiv:1611.09969*.

ZEILER, M., AND FERGUS, R. 2014. Visualizing and understanding convolutional networks. In *Proc. ECCV*.

ZHANG, W., CAO, C., CHEN, S., LIU, J., AND TANG, X. 2013. Style transfer via image component analysis. *IEEE Trans. on Multimedia*.

ZHOU, T., KRÄHENBÜHL, P., AUBRY, M., HUANG, Q., AND EFROS, A. A. 2016. Learning dense correspondence via 3d-guided cycle consistency. In *Proc. CVPR*.

ZHU, C., BYRD, R. H., LU, P., AND NOCEDAL, J. 1994. L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization.

ZHU, J.-Y., LEE, Y. J., AND EFROS, A. A. 2014. Averageexplorer: Interactive exploration and alignment of visual data collections. *ACM Trans. Graph. (Proc. of SIGGRAPH) 33*, 4.