2019-11-01

Light-Field Style Transfer

David Marvin Hart
Brigham Young University

Light-Field Style Transfer

David Marvin Hart

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Bryan Morse, Chair
Parris Egbert
Kent Seamons

Department of Computer Science

Brigham Young University

ABSTRACT

Light-Field Style Transfer

David Marvin Hart
Department of Computer Science, BYU
Master of Science

For many years, light fields have been a unique way of capturing a scene. By using a particular set of optics, a light field camera is able to, in a single moment, take images of the same scene from multiple perspectives. These perspectives can be used to calculate the scene geometry and allow for effects not possible with standard photographs, such as refocus and the creation of novel views.

Neural style transfer is the process of training a neural network to render photographs in the style of a particular painting or piece of art. This is a simple process for a single photograph, but naively applying style transfer to each view in a light field generates inconsistencies in coloring between views. Because of these inconsistencies, common light field effects break down.

We propose a style transfer method for light fields that maintains consistencies between different views of the scene. This is done by using warping techniques based on the depth estimation of the scene. These warped images are then used to compare areas of similarity between views and incorporate differences into the loss function of the style transfer network. Additionally, this is done in a post-training fashion, which removes the need for a light field training set.

Keywords: light field, style transfer, computational photography, multi-view imaging

ACKNOWLEDGMENTS

I would like to thank Dr. Morse for going above and beyond his duties as a research advisor. I also thank my parents, Vern and Amy Hart, for always encouraging me to pursue my academic dreams. Lastly, I thank my wife Jessica for her continual support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

When our eyes see an object in the world, our brains can automatically approximate a distance to that object. We can do this effectively and quickly because of two factors. First, we have two eyes that see the object from slightly different perspectives. Second, we move our heads frequently, providing additional views of the object. From the many views that our brain receives, it is able to analyze movement and parallax to triangulate a position for an object.

With a single digital photograph of a scene, determining the depth is not possible without additional information. At least one additional view is required to triangulate any true distances in the scene.

A light field camera seeks to resolve this problem by taking multiples shots of the same scene from multiple perspectives. This is done by using a special micro array of optics. When the light hits this micro array, it separates the light based on the angle from which it entered the camera. With some processing of the separated light, an image can be generated that is equivalent to taking the photograph from multiple, slightly offset locations. These locations are small translational shifts from a central view as shown in Figure 1.1.

With all this additional information, light fields allow for the estimation of depth in a scene and additional calculations that are not possible with standard photographs. These, in turn, can be used for novel effects unique to light field images. For example, a light field

Figure 1.1: Example light field views. Each image in the grid is taken from the same light field data and represents a small translational movement from its neighbors. This shift in viewpoint can be seen by comparing the locations of the leaves in the foreground in each view.

can generate focal stacks and refocus images even after the picture is taken. These novel effects are well explored [5, 29, 38, 39]. However, the task of editing a light field directly is very difficult. This is due to the many views that need to be edited at once. Any attempt to make a correction or augmentation in one view of the image needs to be appropriately propagated to each independent view in the light field.

Style transfer is an example of such an edit. The goal is to render a photograph in the artistic styling of a particular painting or photo. In previous years, this was done in a variety of ways such as computationally recreating brushstrokes to match a target image [17] or by analyzing patches of the source image and applying them optimally to generate a given photo [20]. In more recent years, style transfer has been revolutionized with neural networks that are trained to reproduce a photo in a particular style. This is a process that has been well studied and continues to be investigated, leading to further improvements in quality [9, 25, 50].

The naive approach to styling a light field would be to apply the style transfer to each view independently, but the style transferring neural network has no indication of spatial

Figure 1.2: Example naive stylization of two light field views. The two images on the left are two views from the same light field image. Even though there are minimal visual differences between these two images, the stylization of these two views (shown on the right) results in dramatic differences in the coloring and features that are present.

consistency between the images as shown in Figure 1.2. Thus, this approach leads to the same part of an object having different colors based on the view it is seen from. This inconsistency degrades the scene geometry that could normally be calculated from a light field. Without this intrinsic consistency and geometry, none of the effects that are normally associated with light fields can be processed in a visually coherent manner. In order to eliminate these inconsistencies, it is necessary to determine correspondences between views of the light field. This process is often used in stylization techniques that are designed to work in applications such as video [44] and stereo imaging [3, 14]. These methods, however, cannot be applied directly to light fields since they are dependent on specific correspondence algorithms and neural networks that were not designed or trained for light field data.

For this thesis, we implement a method of light field stylization that allows for consistencies in color between the different views of the light field. Our method uses the scene geometry to inform the neural network of locations that are the same between views. This is done in such a way that it does not require retraining the style transfer specifically for light fields. Thus, previously trained networks can be used and large light field databases do not need to be created.

In the following section, we present an overview of our method. In Chapter 2, we present a paper submitted to the Winter Conference on Applications of Computer Vision 2020. This paper goes into greater detail into the concepts and algorithms that were utilized in this

work. In Chapter 3, we present additional details in regards to our calibration and training techniques. In Chapter 4, we present additional work and studies that were conducted after the submission of the paper in Chapter 2.

1.2 Overview of Method

This section provides an overview of the light field styilization method presented in this work. A more complete description can be found in Chapter 2.

1.2.1 Disparity

The primary reason that a regular style transfer network cannot stylize a light field in a consistent manner is that it has no notion of similarities between views of the light field. If these correspondences could be determined, one image could be warped to the space of the other to find areas of similar content as shown in Figure 1.3. Thus, we wish to calculate correspondence between pixels in adjacent views. This is done by generating a disparity map, which indicates the amount of shift a pixel undergoes when the viewpoint shifts. An example of such a disparity map is shown in Figure 1.4.

We propose that the most effective way of generating these disparity maps is by using the depth map that is calculated from the epipolar images of the light field. This depth map is generally precomputed for light field images in standard light field collections and is easily accessible. Although this depth map is not calibrated, it provides the relative scene geometry and can be used to generate disparities through an optimization process. This process is described in detail in Chapter 3. While this concept is common to work in stereo vision, we extend it by describing pairwise disparity maps between any view in the light field and the central view of light field.

Figure 1.3: The central view (left), an adjacent view (middle), and the warped central view (right). The central view can be warped based on the disparity map to match the adjacent view. This warping has areas of unknown content since some material cannot be seen from the central view.

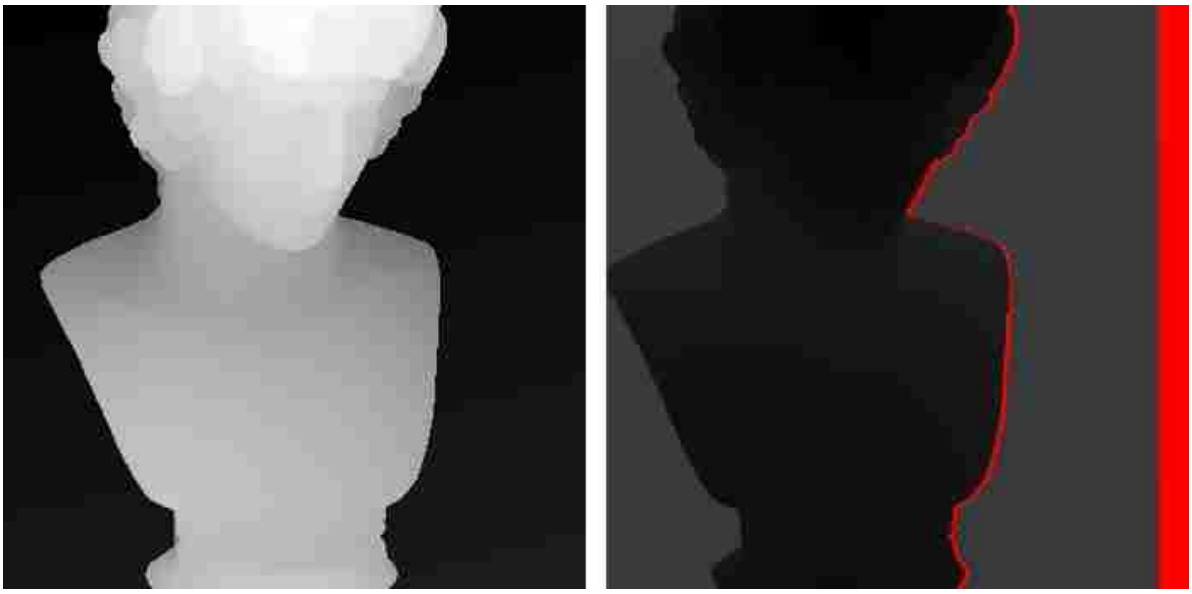


Figure 1.4: The known disparity of the central view (shown on the left) and the calculated disparity for an adjacent view (shown on the right). Red pixels denote areas of unknown disparity. Also, note that the disparities switch signs between images since the direction of shift changes to get from one view to the other.

Figure 1.5: An example consistency mask (shown on the right) for the central view compared to an adjacent view (left and middle respectively).

1.2.2 Consistency Masks

The purpose of all the calculation in the previous section is to provide a way of informing the style transfer network of areas of consistencies. Ideally, every pixel in a light field view would have a matching pixel in the central view. However, because of occlusions, pixels may have no corresponding partner in the central view. Additionally, an occlusion may be partial, giving some interpolated value between the value of the original pixel and the value of the pixel that is partially occluding it. To deal with both of these issues, we must generate a consistency mask for each view of the light field. This mask tells which pixels have a true match in the central view. It also handles partial occlusions by using a value between 0 and 1 describing the amount of consistency that should be present. Example consistency masks are shown in Figure 1.5 and Figure 1.6.

The consistency masks are generated for every view in the light field. With the disparity maps and consistency masks precomputed, consistency can now be enforced on the style transfer network.

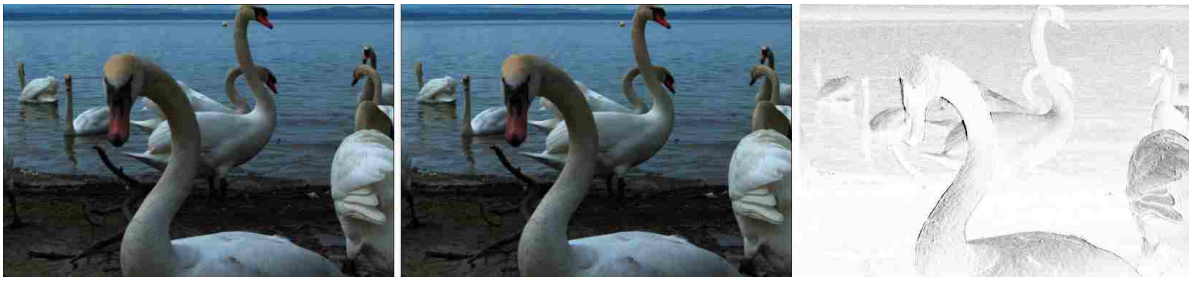


Figure 1.6: An example consistency mask (shown on the right) for the central view compared to an adjacent view (left and middle respectively). The grayscale value describes the percentage of consistency that should be enforced for each pixel. The grayscale values are amplified for visualization.

1.2.3 Style Transfer with Disparity Loss

Training a feed forward style transfer network is a time-consuming process that generally requires thousands or even millions of training images. Since light field images require tremendous amounts of storage space, light field datasets are very small and are not sufficient for the task of training a style-transfer network. Thus, any consistency constraints that are instituted must work within the framework of existing pretrained style transfer networks.

In order to stylize the light field in a consistent way, we utilize the disparity maps and consistency masks that are precomputed for the light field. We start by stylizing the central view through the feed-forward network proposed by Johnson *et al.* [25]. Then for every other view, we stylize it through the feed-forward network and compare it to a warped version of the central view. The mean squared error is computed, which we use as the disparity loss. The image is then optimized to reduce the loss until it converges.

Additional improvements include allowing the optimization to backpropogate through the stylization network, fusing features in the encoding space, and adding perceptual loss to the optimization. The full method is described in detail in Chapter 2, including experimental comparison of the effects of these different elements and variants.

Using this optimization process, we are able to generate stylized light field images that are angularly consistent across views as shown in Figure 1.7, marking substantial visual

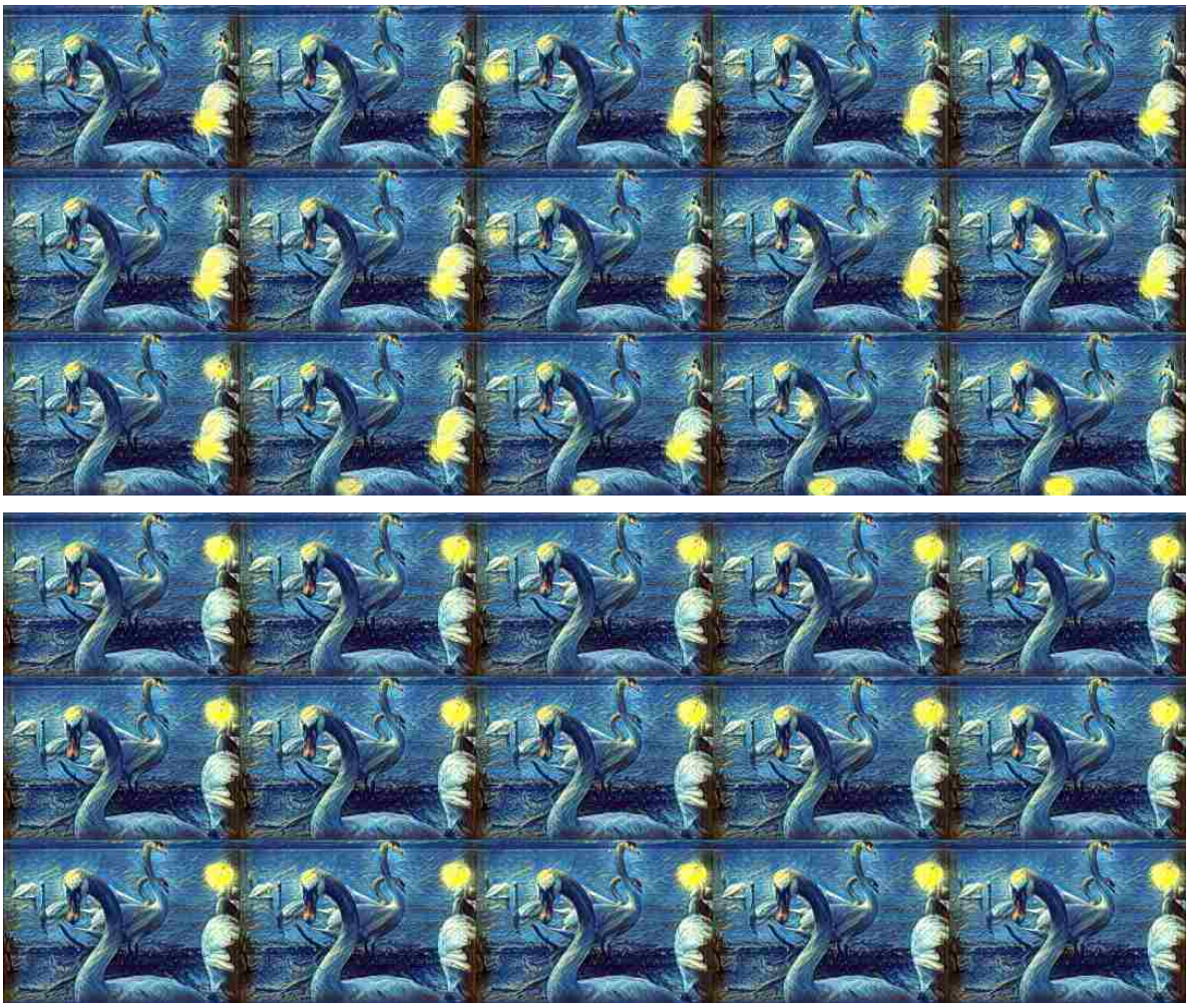


Figure 1.7: An example stylization of a light field (Top) done naively compared to (Bottom) our method.

improvements over the naive method. In addition to qualitative improvements, we can evaluate the stylize light field by its ability refocus and generate depth maps. As shown in Figures 1.8 and 2.6 respectively, the stylized light field can appropriately refocus and generate depth in manner similar to the original light field.

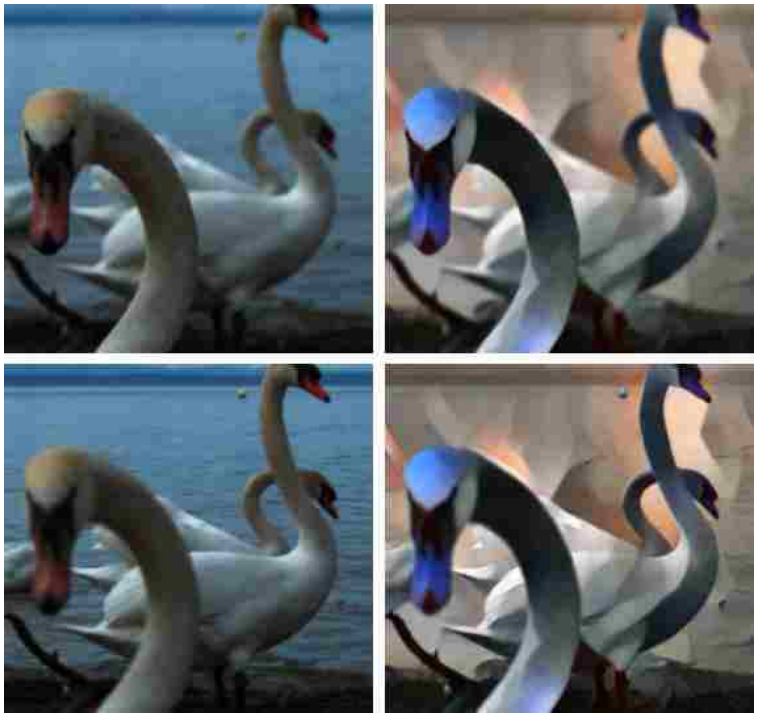
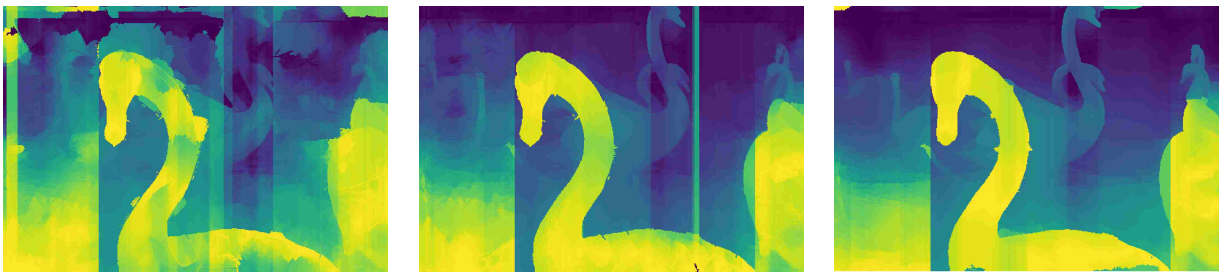


Figure 1.8: Example stylization of a focal stack. Top: Near focus for the original and stylized light field. Bottom: Far focus for the original and stylized light field.



a) Naive b) Proposed c) Unstylized

Figure 1.9: Depth maps computed [23] from stylized light fields. Depth maps computed from naively stylized light fields (a) demonstrate errors due to the lack of angular consistency while those reconstructed from light fields stylized using our method (b) are similar to those computed from original unstylized light fields (c).

Chapter 2

Light-Field Style Transfer

2.1 Introduction

Light field photography continues to be a technology that presents many challenges and problems to overcome including memory constraints and editing difficulties. However, it also presents fascinating capabilities that are not possible with regular images, such as novel view synthesis and focal stack generation. In recent years, we have seen light field technology adopted for more and more commercial applications, from virtual- and augmented-reality systems to dedicated light-field cameras such as those from Lytro (based on the work of [38]). Many variants on multiple-camera imaging configurations are beginning to become more commonplace in the commercial market (ex. dual-camera configurations increasingly found in cell phones), and methods for working with these images often draw from concepts in light-field literature.

"Painterly" and other forms of non-photorealistic rendering of one image based on the style of another have a long history in computer graphics. We refer the interested reader to the original work of [17, 19, 20] as well as more recent work in [34] and surveys in [16, 18]. This area has seen a resurgence in recent years due to the application of deep neural networks to the problem. Rather than trying to analyze brush strokes, texture, or other properties explicitly, recent methods for *neural style transfer* treat the problem as one of optimizing for preservation of the content of one image and the stylistic properties of another. The groundbreaking work of Gatys *et al.* [9] and the feed-forward method presented by Johnson *et al.* [25] have

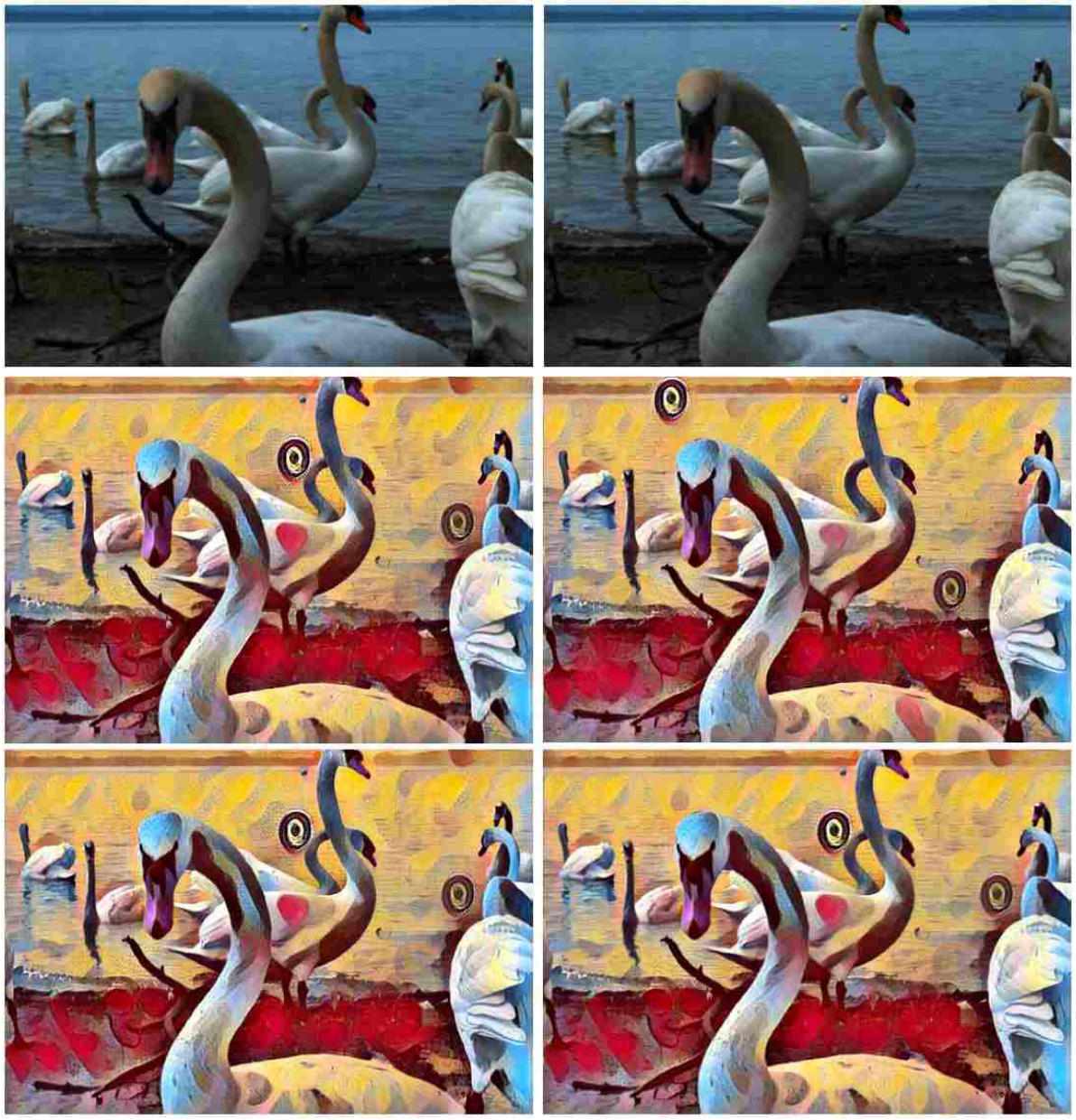


Figure 2.1: Stylization of two views from a light field with 81 images. Even though there are minimal visual differences between these two views from the same light field image (top), the stylization of these two views (middle) results in dramatic differences in the coloring and features that are not present in the original. Our proposed method results in consistency not only between these two views (bottom) but the entire set of 81 views.

opened the door to many variations on these ideas, including improved methods of direct optimization on the resulting image [27, 40], modifications to the feed-forward stylization network [28, 30, 31], stylizing for texture synthesis [46, 47], using depth information to inform stylization [33], training a single network to perform multiple stylizations [50], providing greater control over the stylization [10], stylizing video [2, 15, 21, 44], and stylizing stereo pairs [3, 14].

We seek to expand neural stylization to light-field images. This would present new possibilities in stylization not seen before, such as novel viewpoint generation and dynamic refocusing of stylized images. Expanding stylization to different types of photography has been done before for 360° video [44], RGB-D [33], and stereo imaging [3, 14], but light fields present their own challenges that cannot be solved by simply generalizing one of these methods.

The naive approach to stylizing a light field would be to use a single-image style transfer network for each view independently, but such a network has no notion of angular consistency between the images, which generally leads to visual differences as shown in Fig. 2.1. Both [14] and [3] note these inconsistencies when naively stylizing stereo images (what [14] refers to as the "baseline" method). This problem is only exacerbated when one goes from two images for a stereo pair to the much larger number of views in a light-field image (typically on the order of 50–200). This inconsistency degrades the scene geometry that could normally be calculated from a light field. Without this intrinsic consistency and geometry, none of the effects that are normally associated with light fields can be processed in a visually coherent manner.

This paper presents a method for stylizing light field images in a way that maintains angular consistency between the different views. We first demonstrate an effective way to generate disparity maps for each view of the light field given only a single depth map. Then, we exploit the scene geometry to inform the network of locations that are the same between views, extending the concept of multiple-image consistency used previously for video

sequences [44] and stereo pairs [3, 14]. Additionally, the proposed method does not require retraining the base feed-forward style transfer network [25] specifically for light fields. This allows previously trained networks to be used and avoids the need for large light-field datasets.

As light field technology continues to improve and be adopted for more applications, the need for methods of light field editing will continue to grow, especially as the capabilities of image-processing neural networks also continue to expand. Although this work is specific to neural style transfer, it potentially lays a foundation for light-field consistency optimizations that could generalize to other applications.

2.2 Related Work

Light field research continues to expand as light field cameras become increasingly available for commercial applications. Light fields can be used to create novel views and generate focal stacks [5, 29, 38]. Light fields can also be used to calculate more accurate depth estimates using epipolar images and light field features [4, 22, 23, 32]. The multiple angles and views of a light field also allow for separation of the diffuse and specular components of reflectance [1, 8]. Work has even been done to use light fields for classification, especially of materials [48].

As described in the introduction, the work in this paper seeks to extend the ideas of neural stylization to light field images, for which the key challenge is maintaining angular consistency between the multiple stylized views. Maintaining such consistency in the result is an essential element of any approach that edits multiple images with corresponding content, such as video sequences or stereo pairs [35, 37]. The key in these approaches is to identify or use existing methods to identify correspondences between the images (optical flow for video, stereo correspondence, etc.) and ensure that the results maintain this correspondence.

Ruder *et al.* [44] first introduced the idea of using such correspondences to extend neural style transfer to video sequences. They used optical flow to identify the correspondences and extended the optimization-based stylization approach of Gatys *et al.* [9] to include an

additional consistency loss term. These ideas were extended by Chen *et al.* [2] to train a feed-forward network (building on [25]) to produce similarly consistent video stylization.

Chen *et al.* [3] and Gong *et al.* [14] have each proposed methods for photo-consistent style transfer for stereo pairs, which can be thought of as a much smaller subset (two images) of a light field. The approach of Chen *et al.* [3] builds on a network structure similar to their earlier video-stylization work [2] to train a feed-forward network to learn to perform the stylization. Gong *et al.* [14] likewise train a feed-forward network to perform stylization.

This paper incorporates elements of both [3] and [14], but neither of these methods for stylizing image pairs directly generalize to the much larger number of views in light fields because 1) both methods rely on having pairwise disparity maps from each view to the other, 2) both depend on a single network to stylize all views, and 3) both rely on retraining the network on a large dataset. Extrapolating such an approach to a full light field is simply not viable.

2.3 Multiview Angular Consistency

To enforce angular consistency between multiple views, pixel-wise correspondence needs to be determined for each view in the light field. The most effective way of doing this is by using the depth map that is calculated from the epipolar images of the light field (e.g., [4, 22, 23, 32]), leveraging more information from the field than in two-image stereo correspondence. Using such methods, the depth map is generally precomputed for light field images in standard datasets [42, 43] and is easily accessible. However, such methods usually produce a depth map only for the central (reference) view and not for each separate view in the light field [24], which must be addressed for consistent stylization. The method proposed here is independent of the choice of method used to estimate depth and assumes that the depth map for the central view has been precomputed.

This paper adopts the notation of [23] and most other recent work by indexing the subaperture views by (s, t) and the pixels within each view by (x, y). Individual subaperture

views are thus denoted as $I_{s,t}$ with the central image as $I_{0,0}$ and others indexed using both positive and negative relative (s,t) indices.

Although the central-view depth map is often not calibrated, it provides relative scene geometry and can be inverted and calibrated to produce a pixel disparity map $D_{0,0}$ using a simple optimization algorithm to estimate the unknown scaling due to focal length, imaging pixel density, and the (effective) baseline separation of the subaperture views [7]. Specifically, this optimization inversely scales the depth map to produce the disparity map $D_{0,0}$ that maximizes the correspondence between the central view and the adjacent view to the right, giving us the mapping $I_{0,0} \rightarrow I_{1,0}$. For many light fields, including those shown in our results, there is also an additional translation and cropping for each view, resulting in a planar horopter at an unknown depth and a mix of both positive and negative disparities. To accommodate such cases, we add a second optimized calibration parameter that adds a translation bias. This allows for negative disparities even though the inverted depth map is all positive values.

Because stereo images are typically separated along a horizontal baseline, disparity is often mistakenly thought of solely as the degree of opposite horizontal movement as one moves in a horizontal direction. But it is important to remember that disparity is the degree of apparent opposite movement as one moves the camera in *any* direction. Thus, the reference disparity map $D_{0,0}$ thought of as horizontally mapping $I_{0,0} \rightarrow I_{1,0}$ can just as easily be used to provide the mapping $I_{0,0} \rightarrow I_{0,1}$ as one moves vertically. Similarly, the vector field that maps $I_{0,0} \rightarrow I_{s,t}$ can be calculated using $D_{0,0}(x,y)\ [s,t]^T$.

As noted previously, depth maps for light fields are often computed only for the central (reference) view, allowing computation of a disparity map for this view only. A disparity map for an image allows for forward-mapping of each pixel to where it maps to in another view, which can be many-to-one in the case of occlusion or none-to-one in the case of disocclusion. Instead of using forward warping, however, we desire to use backward warping of the central view to the other views, which requires disparity maps $D_{s,t}(x,y)$ for each of the other views.

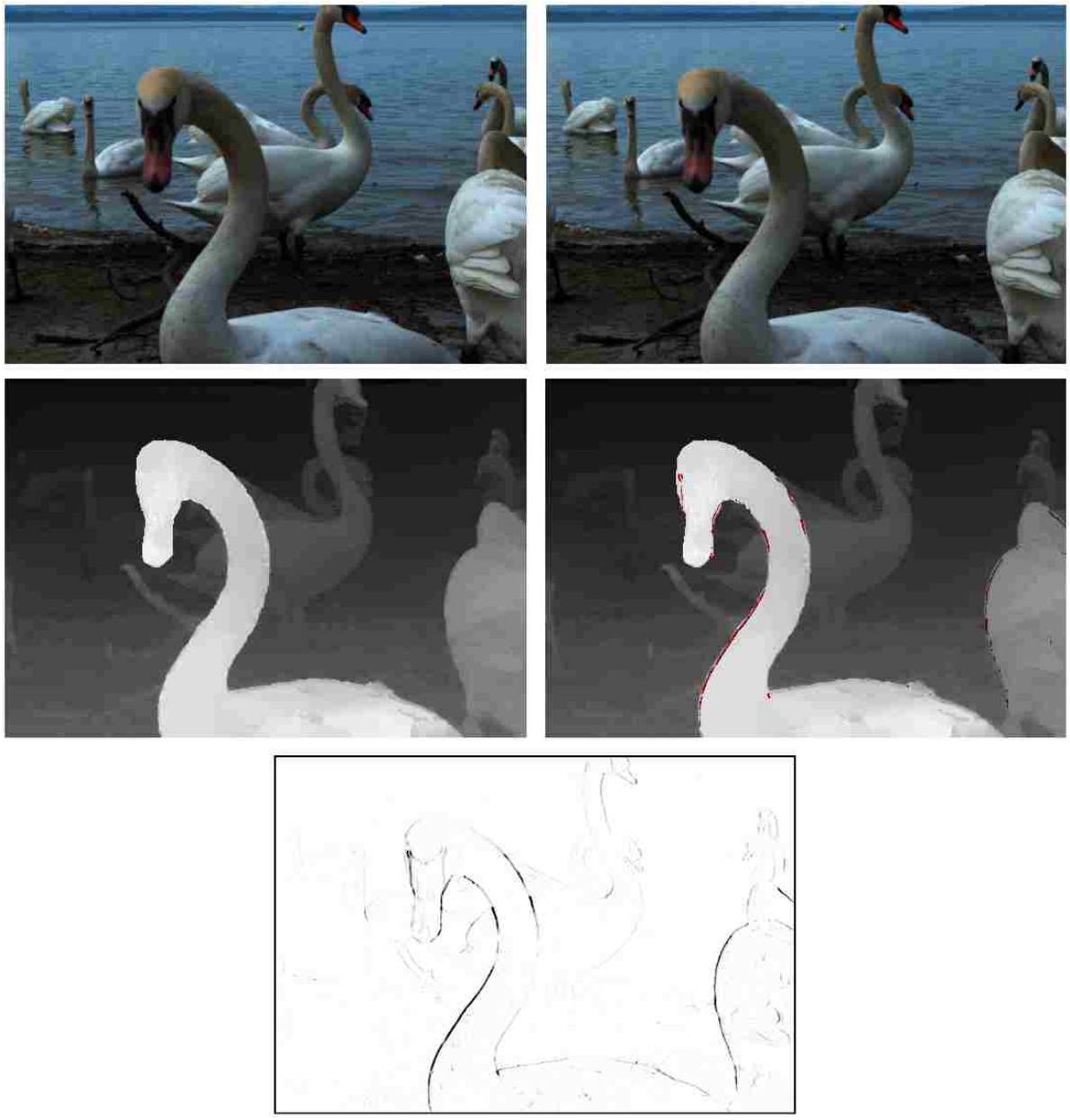


Figure 2.2: Reversing the central disparity map $D_{0,0}$ to produce (partial and masked) disparity maps $D_{s,t}$ for other viewpoints. Top: Views $I_{0,0}$ and $I_{2,0}$ of the light field. Middle: The disparity map from $I_{0,0} \rightarrow I_{2,0}$ (left) and the reversed disparity map from $I_{2,0} \rightarrow I_{0,0}$ (right). Red denotes areas of occlusion that are not seen in the central view. Bottom: The consistency mask $M_{s,t}$ with fuzzy values for partial occlusions or low-confidence correspondences, with zero (black) for points with no correspondence.

We assign disparity $D_{s,t}(x, y)$ for each pixel in each view through a simple search to find the set of pixels (potentially empty, one, or more than one) in the central image that map *to* that pixel (x, y) in view (s, t). While this might seem to be an expensive search, it can be constrained in multiple ways: 1) epipolar geometry constrains the corresponding points to match along the line $(x - s\ D_{s,t}(x, y), y - t\ D_{s,t}(x, y))$, and 2) the minimum and maximum disparities in the central disparity map $D_{0,0}$ can be used to bound the search range along, or near, the epipolar line.

For each candidate matching point (x', y'), we consider all candidate matches that satisfy

$$\|(x' + s\ D_{0,0}(x', y'), y' + t\ D_{0,0}(x', y')) - (x, y)\| < \epsilon \tag{2.1}$$

for some small value of ϵ large enough to account for discrete pixel sampling. (We use $\epsilon = 1.4$.)

Using the idea of stereo symmetries and plausible disparities from [45], we then select the potentially matching candidate with the largest disparity $D_{0,0}(x', y')$, which ensures that the front-most surface is chosen when occlusion causes a many-to-one forward mapping for the point. If no satisfactory match is found, this indicates the disocclusion that would result in a none-to-one forward mapping. An example of inverting the central disparity map can be found in Fig. 2.2, with "no correspondence" disoccluded regions indicated in red.

During this search we simultaneously compute a correspondence confidence map $M_{s,t}$ (as also shown in Fig. 2.2) where $M_{s,t}(x, y) \in [0, 1]$ is determined by comparing the quality of the pixel correspondences (using normalized RGB distance) determined through the just-described search process:

$$M_{s,t}(x, y) = 1 - ||I_{s,t}(x, y) - W(I_{0,0}, D_{s,t})(x, y)||/\sqrt{3} \tag{2.2}$$

where $W(I_{0,0}, D_{s,t})$ denotes the backward warping from image $I_{0,0}$ based on the disparity map $D_{s,t}$:

$$W(I_{0,0}, D_{s,t})(x, y) = \hat{I}_{0,0}(x - s\ D_{s,t}(x, y), y - t\ D_{s,t}(x, y)) \tag{2.3}$$

with $\hat{I}$ denoting interpolation of image I and all pixel values assumed to be in the range $[0, 1]$ for each color channel. If no correspondence was found through the search using Eq. 2.1,

the backward warping is undefined and $M_{s,t}(x, y)$ is set to 0. We use this confidence map as a mask when enforcing consistency (similar to the function of the "gate map" in [14]). This allows greater inconsistency where the original correspondences are uncertain, when partial-pixel effects near object edges produce imperfect correspondence, or when there is otherwise angular inconsistency (e.g., specular reflections [49]) and not enforcing consistency at all where no correspondence exists.

2.4 Style Transfer for Light Fields

Training a feed-forward style-transfer network is a time-consuming process that generally requires thousands or even millions of training images [25]. Since light field images require relatively large amounts of storage compared to typical single or even stereo images, existing light field datasets are very small and not sufficient for the task of training a feed-forward style-transfer network. Thus, any consistency constraints that are instituted must work within the framework of existing pre-trained style-transfer networks.

We propose a method for light-field style transfer that maintains angular consistency between views. For this method, we use a pre-trained feed-forward network as described in Johnson *et al.* [25], specifically the implementation found at [41]. While we choose to work with this specific implementation, our method could also be adapted to fit within the structures of more recent feed-forward stylization networks, such as those found in [28] and [30]. An overview of our architecture is shown in Fig. 2.3.

To stylize the light field, we first encode the features $F_{0,0}$ of the central image $I_{0,0}$ using the first (encoder) half of the stylization network. These features are then decoded using the second (decoder) half of the network to produce $I'_{0,0}$. These are then held fixed as we stylize the rest of the views.

For each other view $I_{s,t}$ of the light field, the features $F_{s,t}$ are also encoded and blended using the correspondence confidence map $M_{s,t}$ with a version of the central-view features

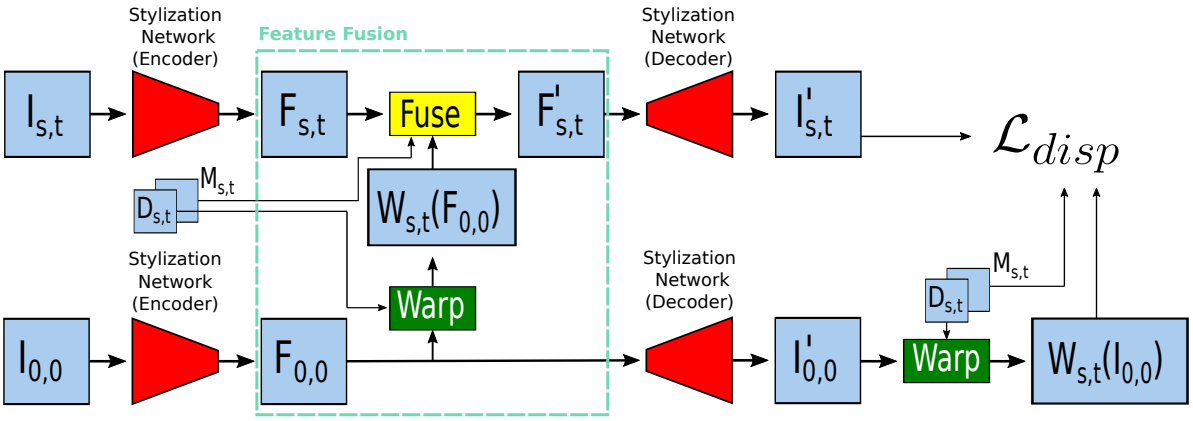


Figure 2.3: Network architecture for neural stylization of light field images. The lower channel stylizes $I_{0,0}$ to produce $I'_{0,0}$ with intermediate feature map $F_{0,0}$, all of which are then held fixed. The upper channel is repeated for each other view $I_{s,t}$. The encoded feature map for this image $F_{s,t}$ is then fused with the warped $F_{0,0}$ using the confidence map $M_{s,t}$ and decoded to produce $I'_{s,t}$. The masked disparity loss between $I'_{s,t}$ and a warped $I'_{0,0}$ is calculated and is backpropagated through the entire network updating only the encoding and decoding parts of the pre-trained feed-forward stylization subnetwork. This process is repeated for some number of epochs for each view $I_{s,t}$ to optimize the result.

warped to view (s,t) using $D_{s,t}$:

$$F'_{s,t} = M_{s,t} \odot W(F_{0,0}, D_{s,t}) + (1 - M_{s,t}) \odot F_{s,t} \tag{2.4}$$

This is similar to the process described in [14] (one of the dual channels) and [3] (single-directional variant). The warped-and-fused feature map is then decoded into the stylized image $I'_{s,t}$.

A disparity (angular consistency) loss [3, 14] is calculated using $I'_{s,t}$ and a warped version of $I'_{0,0}$, again modulated by correspondence confidence map $M_{s,t}$:

$$\mathcal{L}_{disparity} = ||M_{s,t} \odot (I'_{s,t} - W(I'_{0,0}, D_{s,t}))||^2. \tag{2.5}$$

The disparity loss is then backpropagated through the network. This repeats until convergence or a maximum number of iterations is reached.

Algorithm 1 Light-Field Style Transfer

Require: Pre-trained Style Transfer Network θ on Image S
Require: Light Field I, Disparity Maps D, and Consistency Masks M
Returns: Style Transferred Light Field I'

$F_{0,0} \leftarrow \theta_{encode}(I_{0,0})$
$I'_{0,0} \leftarrow \theta_{decode}(F_{0,0})$
for s, t in I **do**
    for *epochs* **do**
        $F_{s,t} \leftarrow \theta_{encode}(I_{s,t})$
        $F'_{s,t} \leftarrow M_{s,t} \odot W(F_{0,0}, D_{s,t}) + (1 - M_{s,t}) \odot F_{s,t}$
        $I'_{s,t} \leftarrow \theta_{decode}(F'_{s,t})$
        $\mathcal{L}_{disparity} \leftarrow ||M_{s,t} \odot (I'_{s,t} - W(I_{0,0}, D_{s,t}))||^2$
        $\theta \leftarrow$BackProp$(\theta, \mathcal{L}_{disparity})$
    $F_{s,t} \leftarrow \theta_{encode}(I_{s,t})$
    $F'_{s,t} \leftarrow M_{s,t} \odot W(F_{0,0}, D_{s,t}) + (1 - M_{s,t}) \odot F_{s,t}$
    $I'_{s,t} \leftarrow \theta_{decode}(F'_{s,t})$
return I'

This process is repeated for each subaperture view as described in Algorithm 1. In our implementation, we use a learning rate of 1e-2 and run it for a maximum of 50 epochs (though we found most views converge after 40 epochs approximately). We also use the overfit stylization network from one view as the initial stylization network for the next view rather than resetting the network. We have found that this works best when the shift from one view to the next is small, so we visit the different views $I_{s,t}$ by alternating the ordering on successive rows (i.e., boustrophedonically) so that the shift in viewpoint is always to an adjacent view. We also double the epochs for the first view visited to increase stability of stylization.

Although our integration of warped and fused feature is similar to the approaches in [3] and [14], there are distinct modifications necessary to allow such an approach to work for light fields beyond simply the number of views.

In [3], these features are warped to a common hypothetical view that is located halfway between the two images in the stereo pair, and then the two are fused. In a light field for which only a central-view depth map has been computed, however, the only reference point that can act as a common view for all $N \times N$ images is the central view. Thus, only the central view features are warped and fused with other view features. This is done using bilinearly

resized versions of the disparity maps and consistency masks to match the resolution of the features.

In [14], the features are warped and fused in both directions, providing a more consistent version of the features between the pair. While this process provides good results for an image pair, it does not generalize to light fields. Allowing all $N \times N$ views of a light field to have an influence on the central view leads to an aggressively averaged set of features, which leads to very blurry output images. Thus, in the method proposed here, the central view features are held fixed during all stages of the algorithm. This restriction on the central view is necessary in order to force the network to converge in a way that keeps consistency between all views of the light field while maintaining high-quality output.

It is important to note that our architecture is not used to train a feed-forward network for light fields. Instead, our algorithm provides a method for optimizing a single light field image in a reasonable amount of time. It is similar to the optimization presented by Gatys *et al.* [9], but warping of the features and initializing with feed-forward stylization allows the optimization to converge much faster and does not require training on perceptual loss.

2.5 Results and Evaluation

Since there are no other methods for neural stylization of light fields to compare against, we present qualitative results (visual examples) and quantitative evaluation of the degree to which the resulting stylization preserves both perceptual factors (content and style loss) and inter-view angular consistency (disparity loss).

2.5.1 Qualitative Evaluation

Our method works for a variety of models and images. Fig. 2.4 shows subsets of the views from light fields stylized with the proposed method. To more clearly see the consistency and shift between views, We also provide epipolar images of the stylized light fields in Fig. 2.5.

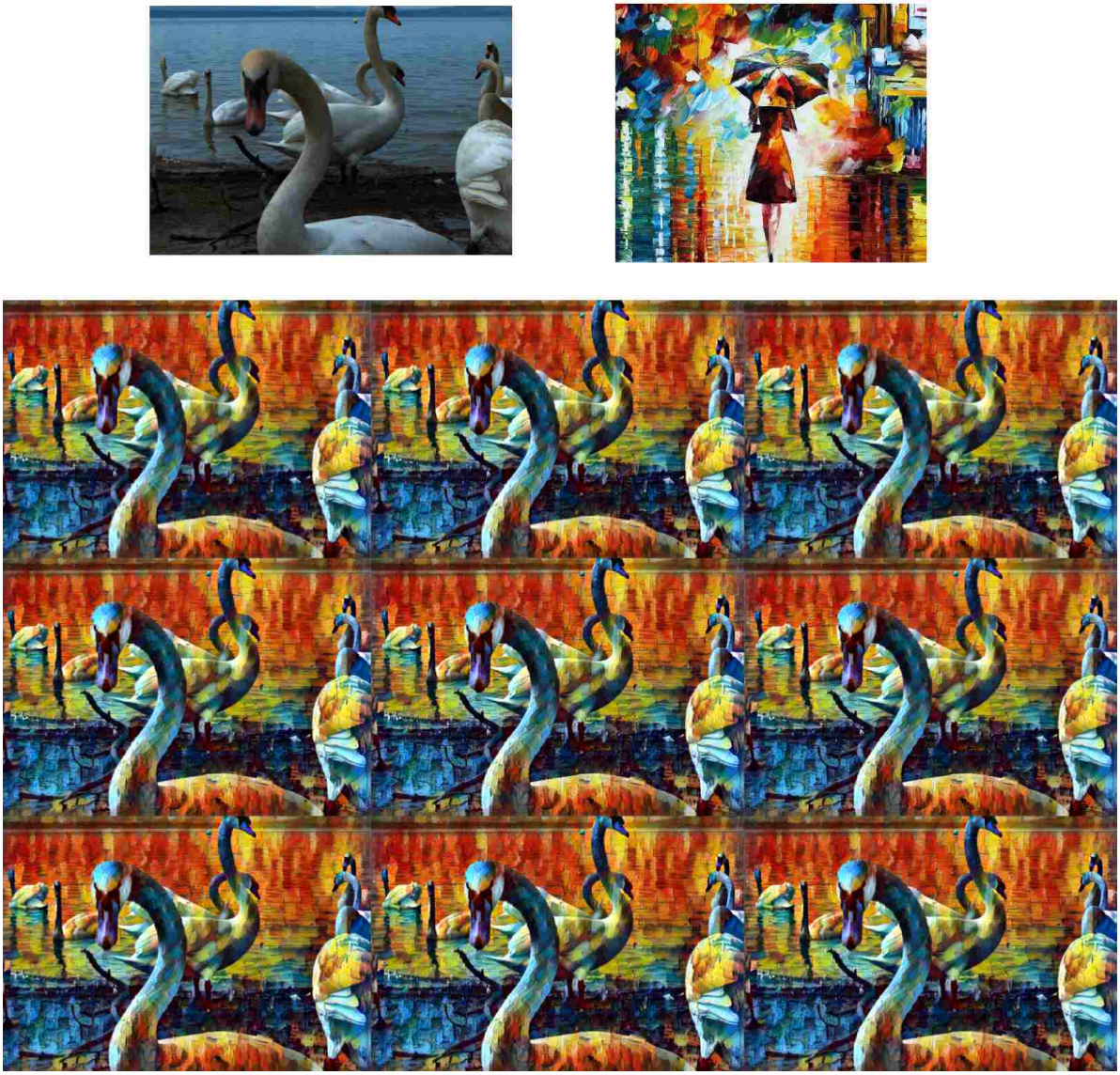


Figure 2.4: “Swans” light field image stylized with our method. A subset of the full set of stylized views is shown. The views are selected with a stride of 3 to each side of the central view.

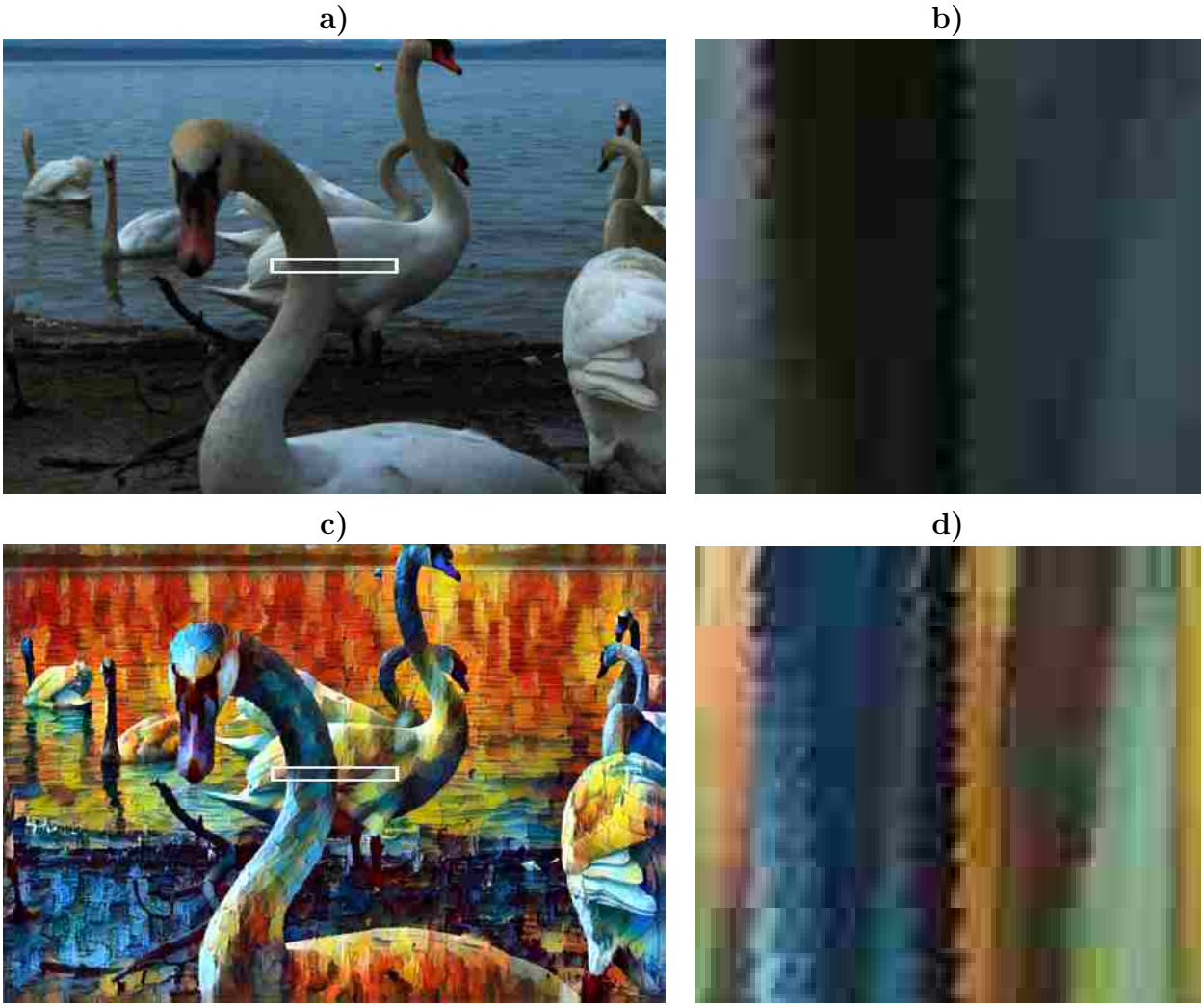
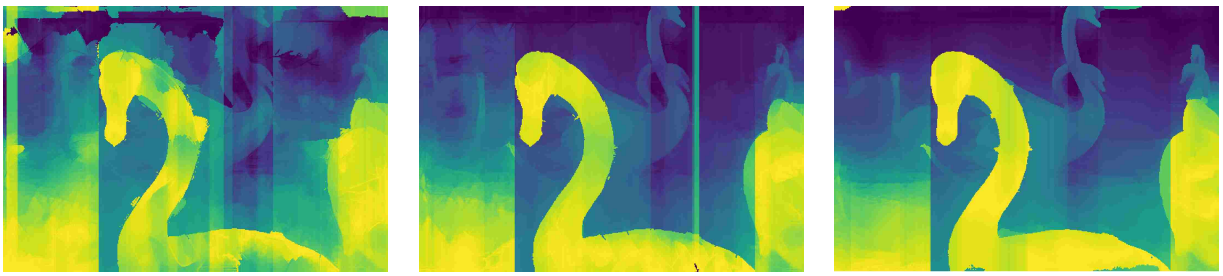


Figure 2.5: a) The central view of the "Swans" light field image. b) The epipolar image in the highlighted region part a. c) The central view of the stylized light field image. d) The epipolar image of the highlighted region part c.



a) Naive b) Proposed c) Unstylized

Figure 2.6: Depth maps computed [23] from stylized light fields. Depth maps computed from naively stylized light fields (a) demonstrate errors due to the lack of angular consistency while those reconstructed from light fields stylized using our method (b) are similar to those computed from original unstylized light fields (c).

Additional results can be found in the supplemental materials accompanying this paper, which include a video that better shows the angular consistency between shifting subaperture views.

In addition to visually inspecting the individual views for angular consistency, we can also determine how well the stylized light field maintains geometric properties of the original. One way to verify this is to recompute depth maps from the stylized light fields and compare them to those of the unstylized light fields. Fig. 2.6 shows an example of this using the "Swans" light field, the "Mosaic" style image, and the depth computation method from [23]. As shown in Fig. 2.6a, naively stylized light fields do a poor job retaining depth properties due to the lack of angular consistency. Light fields stylized with our method do a better job preserving depth properties (2.6b) and are similar to those of the unstylized original (2.6c).

Light field images are often used to render dynamically refocused images of the captured scene as first described in [38]. If the light field is angularly consistent, it should maintain the ability to refocus even in the stylized format. This ability to refocus the stylized light field is demonstrated in Fig. 2.7.

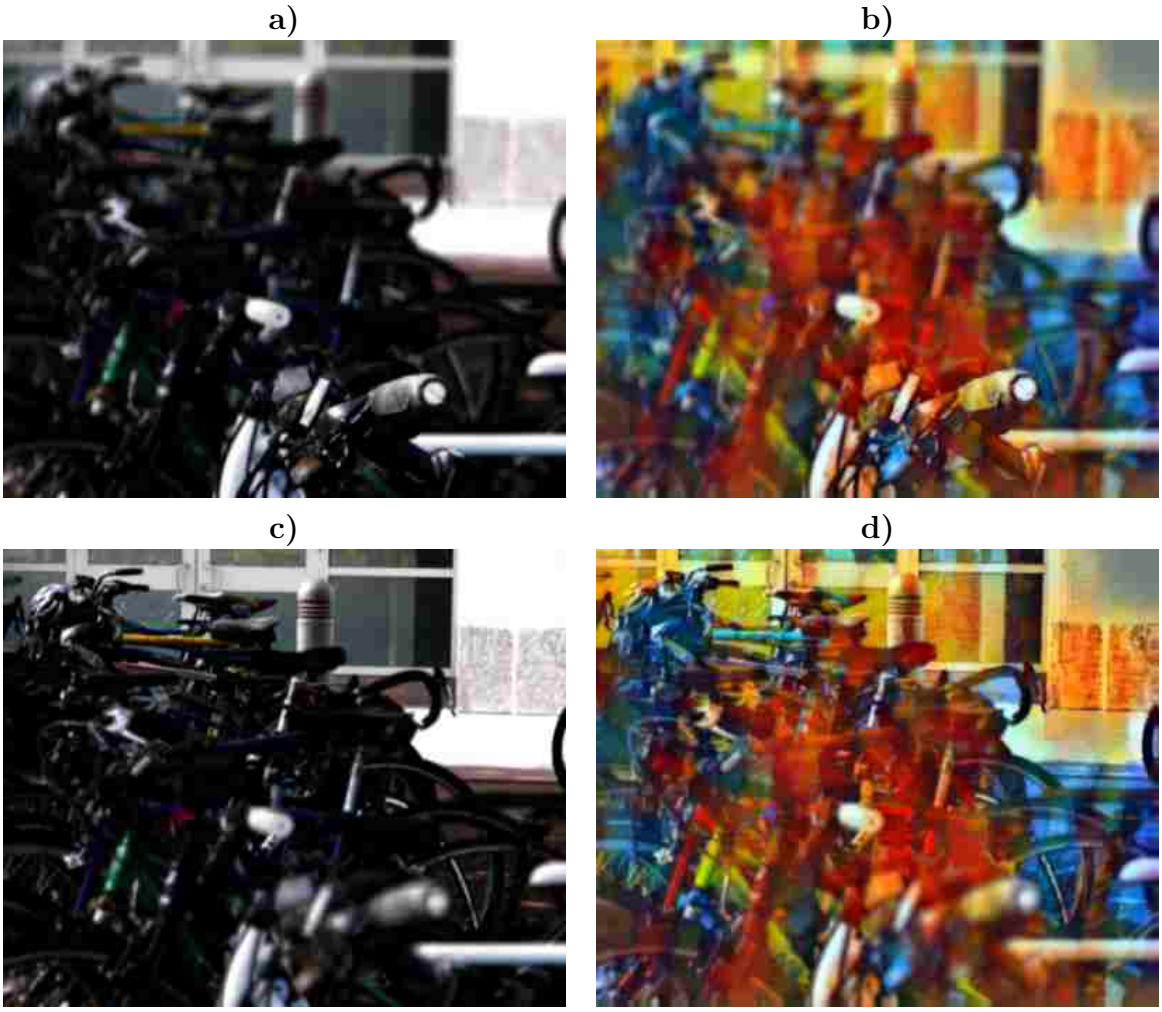


Figure 2.7: The focal stack of an example stylization. Near focus for the a) original and b) stylized light field. Far focus for the c) original and d) stylized light field.

	Perceptual Loss			
	Candy	Mosaic	Rain Princess	Udnie
Naive	3954244	4520050	3628793	830148
Ours	3954242	4520043	3628792	830148
	Disparity Loss			
Naive	6044	8807	3845	690
Ours	113	139	102	40

Table 2.1: Evaluation of perceptual and disparity loss for multiple stylization models. Our method keeps similar perceptual loss to the naive method while greatly decreasing the disparity loss.

2.5.2 Quantitative Evaluation

The primary metric for neural style transfer is perceptual loss [25], combining the ideas of content loss and style loss from [9]. As in [3] and [14], for multiview stylization we can combine this metric with disparity loss (Eq. 2.5) to evaluate angular consistency. An ideal light field stylization method should be able to minimize the disparity loss without increasing the perceptual loss. In Table 2.1, we compare our results to the baseline of naively stylizing each view independently. This evaluation uses four different styles and presents the per-view average loss. The proposed method causes only an extremely small increase in perceptual loss across all four styles, which is to be expected since this is balanced against disparity loss.

The most significant change is in the disparity (angular consistency) loss, which drops by an order of magnitude or more, quantitatively validating the visual consistency demonstrated in Figs. 2.1 and 2.4.

2.6 Variations and Experiments

In addition to evaluating the proposed method, we also explore several variations and simplifications to evaluate the relative contributions of various elements of the approach.

As noted in Section 2.4, our approach begins with a pre-trained style transfer network and then iteratively optimizes the network variables to reduce the disparity loss for a single

image rather than trying to train a single network to function in a purely feedforward way that generalizes to other images. This raises the question of whether one could simply use a purely optimization-based approach such as a Gatys-like network that incorporates perceptual loss and the additional disparity loss term to encourage consistency. We have explored that option and found that although it produces good results, the optimization does all of the work from scratch instead of being able to leverage a pre-trained stylization network and explicitly warped-and-fused feature maps. As such, it requires significantly more iterations and typically takes about twice as long to run as the proposed method.

We explore other variations of the full method proposed in Section 2.4 and illustrated in Fig. 2.3. These variations are described in the following subsections and summarized in Table 2.2. For comparison of the possible variations, we analyze the average per-view perceptual loss and disparity loss for a set of light fields, the results of which are given in Table 2.3.

For the Naive method (independently stylized views), we again see that the disparity loss is high because no angular consistency was enforced. For a consistently stylized light field, we would expect the disparity loss to greatly decrease while the perceptual loss remains unchanged. Table 2.3 shows that all of the variations explored here produce light fields with nearly identical perceptual loss. This is to be expected since any given view considered in isolation maintains the properties of the stylization, even if inconsistent with the other views. Thus, rather than focusing on perceptual loss, we use disparity loss as the main comparison metric when comparing and discussing the following variations.

	Warp/Fuse Features	Warp/Fuse Images	No Fusion
Full BP	BPFuseFeatures[1]	BPFuseImg	BPNoFuse
Post-optimize	OptFuseFeatures	OptFuseImg	OptNoFuse
No iteration	NaiveFuse	WarpBlend	Naive[2]

[1] The full method proposed in Section 2.4
[2] What [3] refers to as "baseline"

Table 2.2: Variations on the proposed method explored in Section 2.6

	Perceptual Loss		
	Warp/Fuse Features	Warp/Fuse Images	No Fusion
Full BP	3193389	3193389	3193388
Post-optimize	3193386	3193388	3193380
No iteration	3193392	3193389	3193391
	Disparity Loss		
Full BP	98	16	363
Post-optimize	15	14	14
No iteration	2089	25	4846
	Execution Time (seconds)		
Full BP	308	304	304
Post-optimize	135	137	136
No iteration	43	45	42

Table 2.3: Comparison of perceptual loss, disparity loss, and execution time for variations of the proposed method

2.6.1 Fusion Variations

Our method, like [3] and [14], pairwise fuses elements of two images in the feature map domain. This raises the question of whether such feature-map fusion is preferable to image-space fusion (i.e., fusing $I'_{0,0}$ and $I'_{s,t}$ instead of $F_{0,0}$ and $F_{s,t}$) or is even required at all. When we analyze the methods that use fusion in the image domain (represented as the middle column in Table 2.3), it is clear that these methods have the lowest masked disparity loss. However, visual analysis of the stylized light fields produced with these methods shows that artifacts appear frequently in the unmasked regions, *which are not factored into the quantitative masked disparity loss.* Warping and fusing in the image domain also relies heavily on the notion that this process is done with perfect disparity maps. In reality, noise in the image data, featureless regions, ambiguous matches, discrete pixel sampling, and other factors cause imperfect depth or disparity estimates, all of which are well known issues with stereo, multi-view stereo, and light-field depth estimation. This over-reliance on the accurate disparities can cause additional artifacts that are undesirable in the final images as shown in Fig. 2.8. We believe that fusing feature maps and then decoding them results in visually

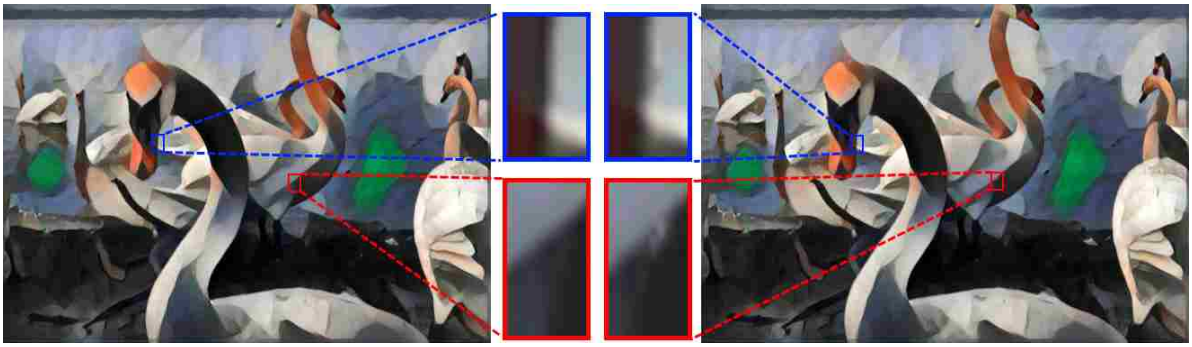


Figure 2.8: Two views of "Swans" stylized using the WarpBlend variant (image-space fusion without subsequent optimization). Methods that fuse in the image domain are highly dependent on accurate disparity maps. Errors in the disparity maps lead to visual artifacts in the final stylized light field, such as those shown in the red and blue callouts.

better (more artifact-free) stylizations than image-space fusion after decoding because the decoding of the feature maps mitigates such artifacts.

Since the fusion step takes time, we also consider whether fusion is even necessary for the optimization to converge and whether it could be discarded in order to save processing time. While the disparity loss is comparable to that of other backpropagating methods, the lack of a fusion step causes the network to take longer to converge on each individual view, especially the earliest optimized views. If this method is trained with the same learning rate, number of epochs, and optimization sequence as described in Section 2.4, it results in some views having ghosting artifacts, especially in areas with high frequency content. Thus, the number of optimization epochs must be increased to produce results comparable to the proposed method, more than offsetting any potential time savings.

2.6.2 Optimization Variations

Our method performs optimization by backpropagating all the way back through the stylization encoder/decoder, essentially the same as extreme overfitting of the network to a single set of light-field views. Another option would be to use the feed-forward stylization network, including warping and fusion of the feature maps, to produce initial estimates of $I'_{s,t}$ and

then post-optimize with the pixels of $I'_{s,t}$ as the only updated variables. Such a method is essentially the same as generating the $I'_{s,t}$ images using warped and fused feature maps and then running it through a Gatys-like optimization including the additional disparity loss term. It is also worth considering if a light field can be stylized without the need for an optimization at all, but relying solely on warping and fusing feature maps to create consistency. Upon experimenting with these optimization variations, we find three key findings.

First, fusing features and then post-optimizing $I'_{s,t}$ to reduce disparity loss without backpropagating through the network (OptFuseFeatures) results in worse angular consistency than propagating the loss back through the stylization encoder/decoder (BPFuseFeatures, our primary method). This can be attributed to the initial fusing of features $F_{0,0}$ with $F_{s,t}$, which essentially alpha-blends features maps from a network that has not trained on disparity loss. Allowing the feed-forward stylization encoder/decoder to train on disparity loss allows these to learn to produce more consistent feature maps.

Second, the WarpBlend method, which uses image-space warping and blending with no subsequent optimization, gives reasonable results with a roughly 12x speedup compared to the method proposed in this paper since it requires no iterative optimization. This method essentially involves independently stylizing each view using the pre-trained stylization network and then employing a purely image-space approach to warp the stylized central view $I'_{0,0}$ to each of the other views and blending it with the initial stylization $I'_{s,t}$ for those views using the correspondence confidence map $M_{s,t}$. This means it could serve as an alternative to the proposed method if greater speed is desired. However, we reiterate that because it relies *entirely* on accurate disparity-based warping in image space, it is susceptible to qualitative visual artifacts from disparity errors as discussed earlier in Section 2.6.1 and shown in Fig. 2.8. These are not factored into the masked disparity loss.

Third, we find all methods of post-optimization on the output image to be undesirable. Since it can only optimize on the disparity loss, it eventually converges to give essentially the

	Perceptual Loss		
	BPFuseFeatures	BPFuseImg	BPNoFuse
Both Loss Terms	3193389	3193389	3193388
Disp. Loss Only	3193389	3193389	3193388
	Disparity Loss		
	BPFuseFeatures	BPFuseImg	BPNoFuse
Both Loss Terms	98	16	363
Disp. Loss Only	99	16	366
	Execution Time (seconds)		
	BPFuseFeatures	BPFuseImg	BPNoFuse
Both Loss Terms	566	565	565
Disp. Loss Only	308	304	304

Table 2.4: Comparison of backpropagating / optimizing using combined perceptual and disparity loss to using disparity loss alone

same results as the WarpBlend method (which can be thought of as the minimization of the disparity loss), and thus retains all the same visual artifacts.

2.6.3 Loss Function

For the results described so far in this section, we either backpropagate through the stylization network or post-optimize the output image using only disparity loss. Given that the stylization network has been pre-trained to minimize perceptual loss [25], we consider the question of whether it is effective to include perceptual loss to minimize visual artifacts along occlusion boundaries. However, we have found that backpropagating perceptual loss and disparity loss produces results that are nearly indistinguishable visually from those created using disparity loss alone, as shown in Fig. 2.9 and in Table 2.4. This is also evident when analyzing the disparity loss for each view of the light field as shown in Fig. 2.10. Excluding perceptual loss from the optimization also avoids having to backpropagate through the VGG-16 network at the end of the overall network, reducing the computation required. We have found that this reduces the execution time by approximately 35% with comparable results.

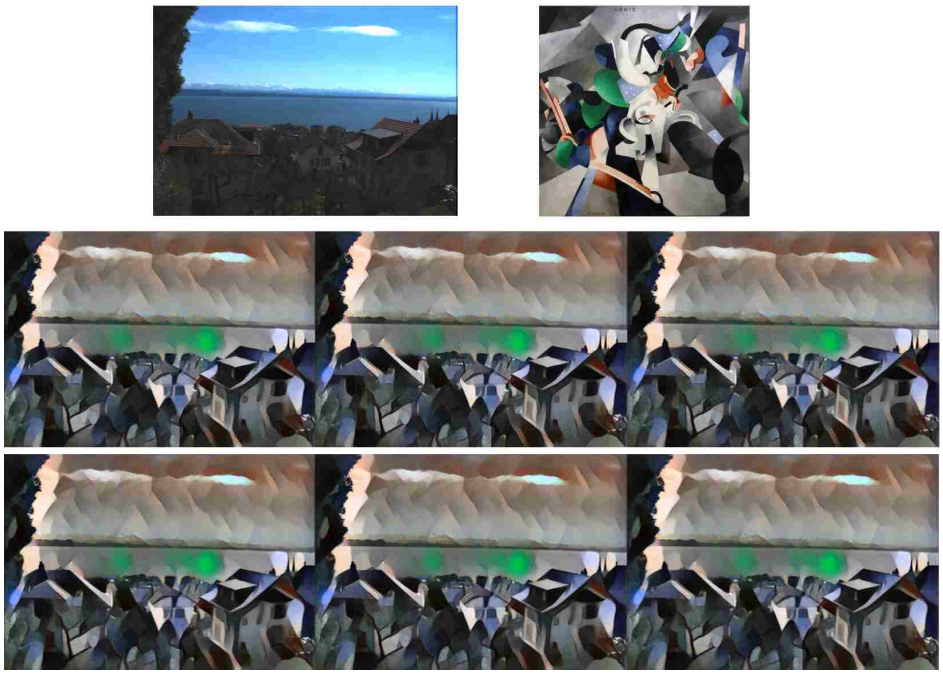


Figure 2.9: "Lake" light field stylized using BPFuseFeatures optimized with the disparity loss and perceptual loss (top) and the disparity loss only (bottom). Results are visually indistinguishable.

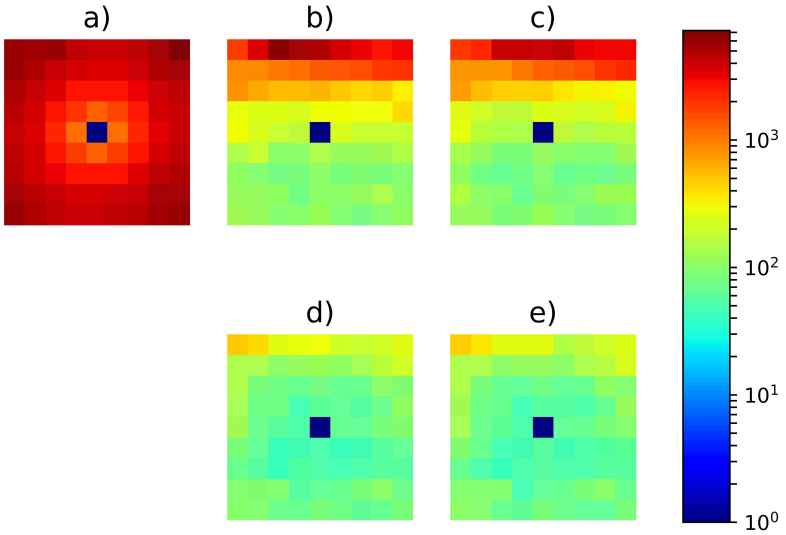


Figure 2.10: Visualization of disparity loss by view for a) Naive, b) BPNoFuse optimized on disparity loss only, c) BPNoFuse optimized on disparity and perceptual loss, d) BPFuseFeatures optimized on disparity loss only, and e) BPFuseFeatures optimized on disparity and perceptual loss.

2.7 Conclusion

This paper presents the first neural style-transfer method for light fields that achieves high-quality visual results while maintaining angular consistency. The method uses a given central-view depth map to create masked and confidence-weighted disparity maps for each other view, allowing backward warping from the central view to all other views. This warping and masking is vital to the optimization process and consistently stylized results. As with recent methods for stereoscopic neural stylization, we fuse warped feature maps using a confidence-weighted mask. Unlike these methods, we do not try to train a single network to stylize different light fields in a purely feed-forward fashion. Instead, we incorporate pre-trained monocular style-transfer networks and iteratively optimize them for each view.

These results are validated both qualitatively (visually) and quantitatively. We also present variants of this method that allow for trade-offs between angular consistency, sensitivity to errors in the original depth map, and execution time.

Chapter 3

Implementation Details

Since WACV requires that all papers remain under a particular page length, there are important details to this work that could not be included in the paper submission. In this chapter, we present details specific to implementation that will be helpful to those that wish to implement or build on this method in the future.

3.1 Disparity Calibration

As stated in Section 2.3, the light field depth map typically comes uncalibrated, and thus, the disparity map is uncalibrated since it is simply treated as an inversion of the depth map. Fortunately, the true disparity map D is linearly related to the uncalibrated disparity map $\tilde{D}$. Thus, there is some multiplicative factor k and offset b that can be applied to solve for the true disparity map (*i.e.*, $D = k\tilde{D} - b$). Thus, the process of calibration becomes an optimization of these two parameters for each view.

In our current implementation, a random search is performed to find these two parameters. Values for k are sampled randomly from a log-normal distribution that starts at 0 and peaks at 1.0. Values for b are sampled from a standard normal distribution.

In order to determine the best k and b values, our optimization algorithm assumes the sampled k and b values and warps the central view to an adjacent view based on those values. If the values are optimal, we would expect the warped central view to be similar to the adjacent view, but we would expect some mismatch to occur since there are occlusions in the adjacent view that are not present in the central view. Thus, we optimize on the

Algorithm 2 Disparity Calibration

Require: Light Field I, Uncalibrated Disparity Map $\tilde{D}$ **Returns**: Calibrated Disparity Map D

$k \leftarrow 1$
$b \leftarrow 0$
for *epochs* **do**
    $\hat{I_{0,0}} \leftarrow W(I_{0,\lambda 1}, k * \tilde{D} - b)$
    errors $\leftarrow ||\hat{I_{0,0}} - I_{0,0}||^2$
    errors $\leftarrow$ first(sort(errors),95%)
    error $\leftarrow \sum$ errors
    if error $<$ error' **do**
        error' $\leftarrow$ error
        $k' \leftarrow k$
        $b' \leftarrow b$
    $k \leftarrow$ sample($e^{\mathcal{N}(0,1)}$)
    $b \leftarrow$ sample($\mathcal{N}(0,1)$)
$D \leftarrow k'D - b'$
return D

difference between the warped central view and the adjacent view, but we exclude the top 5% of mismatch to account for errors along occlusion boundaries. The k and b values with the lowest error are used for the final calibration. This full process is shown in Algorithm 2.

Once the k and b values are known, the disparity is calibrated from the central view to an adjacent view (what was referenced as $D_{0,0}$ in Section 2.3). To calculate the other disparity maps (*i.e.*,, from the central view to any other view in the light field), the following assumptions can be made:

- The disparity from the central view to the view two images to the right is the same as twice the disparity from the central view to the adjacent view to the right (*i.e.*, $dx_{2i} = 2dx_i$).
- The disparity from the central view to the adjacent view on the left is the same as the negative of the disparity from the central view to the adjacent view to the right (*i.e.*, $dx_{-i} = -dx_i$).
- The disparity from the central view to the adjacent view above is the same as the disparity from the central view to the adjacent view to the right (*i.e.*, $dy_i = dx_i$).

Thus, finding each disparity map becomes the simple task of taking the previously calculated disparity and multiplying by a constant factor based on the current view of interest. Note that this simple multiplication is possible because the spacing is equal between all light field views in our data set. If another method of acquisition was used, such as using a robotic gantry arm, location information would be required to determine the exact relation between disparities in the different views.

The process described above is the process that was used to generate the results found in Chapter 2. However, we did explore the possibility of fine-tuning the k and b values in certain ways. For example, instead of calibrating to the immediately adjacent view, the optimization could be done on a different view of the light field. Alternatively, the k and b values could be optimized to minimize error across the whole light field instead of a single view. The algorithm could also allow for different k and b values that are fined-tuned for each view. An example of k values selected for each when fine tuned is shown in Figure 3.1.

In exploring these variations, we did find that certain practices improved overall calibration, albeit with minimal impact on the final result. Calibrating on a view that is further from the center of the light field tended to avoid artifacts along edge views. A global optimization across all views had minimal effect and took far too long to be pragmatic. Allowing for each view to fine-tune the k and b values actually caused overfitting of the error function, which caused a jitter effect as the viewer scanned across views. Fine-tuning only k for each view was more stable, but again takes far too long to be preferred over a single calibration method.

Overall, we empirically found our single-view calibration algorithm to be the fastest method to generate angularly consistent results.

Figure 3.1: Chart of k values when tuned for each view in the "Swans" image.

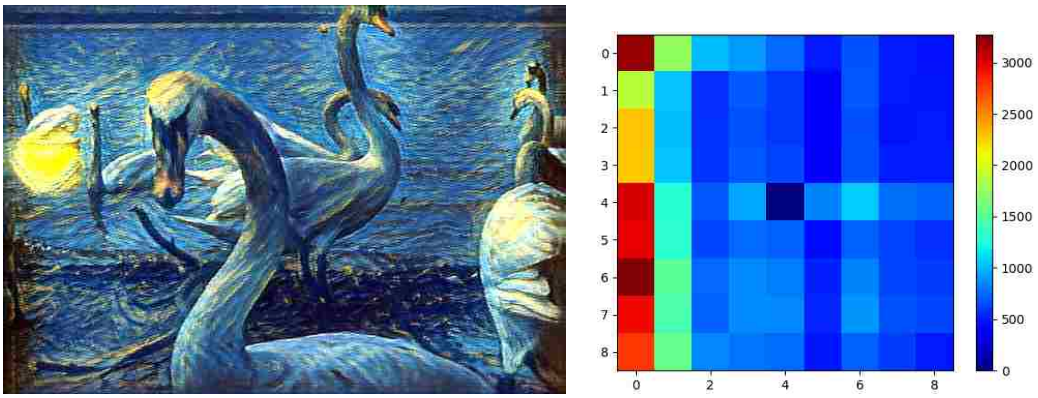


Figure 3.2: The stylization for a view on the left side of the light field (Left) and the associated disparity loss for each view (Right) when using naive scanning order. Note that the left side view is blurred. This is caused by high disparity loss on left side views as the network has difficulty transitioning from a far right view to a far left view.

3.2 Optimization

As Section 2.4 points out, we used hyperparameter values of $\alpha = 1$, $\beta = 5$, and $\gamma = 500$ and a learning rate of 1e-2 for 50 epochs. We empirically found these values to work the best. However, some light field stylizations converged to angular consistency as soon as 30 epochs. We also found the light field never converges to an angularly consistent result if it is trained with a learning rate above 1e-2.

The order that the light field views are visited are very important. Initially, we were optimizing the views in a naive scanline order. However, we found large amounts of error on the left side views of the light field, as shown in Figure 3.2. This artifact is caused by the sudden spatial jump that the network must learn. Thus, it is more effective to select the next view to optimize to be adjacent to the previous view. Thus, we optimize in a boustrophedonical order (*i.e.*, scanning right on odd rows and scanning left on even rows). Additionally, we found that the network has the most to learn on the first view it optimizes, so we optimize this view for twice as many epochs. Combining this longer initial training with the specific ordering of views gives much improved results, as shown in Figure 3.3.

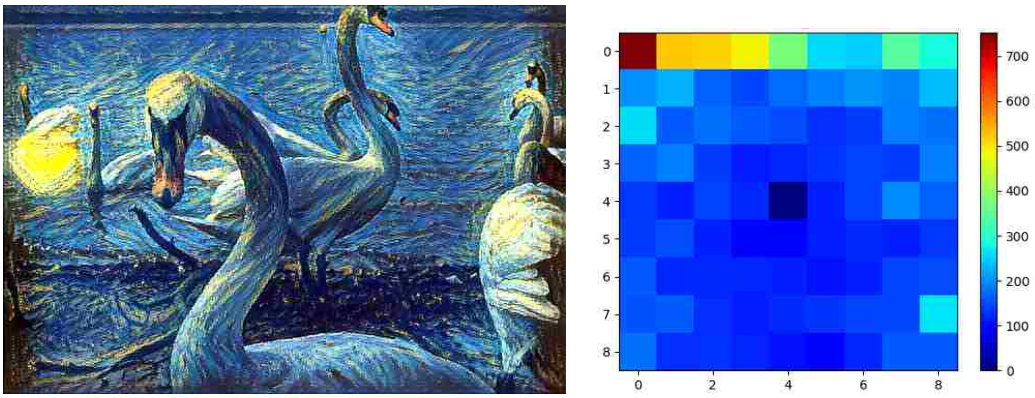


Figure 3.3: The stylization for a view on the left side of the light field (Left) and the associated disparity loss for each view (Right) when using our scanning order. Note the improved image quality compared to the results shown in Figure 3.2. Also note the high disparity loss that remains in the first view visited (*i.e.*,Top Left). This suggests that increasing the number of training epochs for the first view is preferable.

3.3 Evaluation

In addition to visually and qualitatively analyzing the generated light fields, we introduced three quantitative metrics in Chapter 2. Specifically, we analyzed the resulting loss values for perceptual and disparity loss, recomputed depth maps from the stylization, and simulated refocusing on the stylized light field.

Our reported perceptual loss comes from the first two terms of our loss function (*i.e.*, $L_{content}$ and L_{style}). However, rather than weighting them by α and β, the reported perceptual loss values is the simple sum of the content loss and the style loss (*i.e.*, $L_{content} + L_{style}$). We decided to report in this fashion since a fully stylized image has near zero content loss. Thus, the style loss is the main contributor to the value. Since we wanted this to remain unscaled, we decided to use an unweighted sum rather than a weighted sum.

The recomputed depth maps were calculated using the algorithm described by Jeon *et al.* [23]. We used a MATLAB implementation found at the author's website [6].

Refocusing of images was coded using the original refocus formula (Equation 4.1) found in Ren Ng's disseration [38]. We reimplemented this refocusing algorithm for our work.

Chapter 4

Extensions and Future Work

4.1 Gatys Style Training

As stated in Section 2.6, we also explored optimizing a stylized light field with an additional disparity term in a fashion similar to Gatys *et al.* [9] rather than starting with a pre-trained feed forward network. In actuality, our research initially used a Gatys-like optimization, and it does generate results comparable to our method presented in Chapter 2 as shown in Figure 4.1. However, it maintains the same weakness of the Gatys *et al.* method, which is that it is much slower than the feed-forward networks such as those proposed by Johnson *et al.*[25]. We thus chose in the end to utilize a feed forward network to decrease the run time of the overall algorithm.

4.2 Depth Loss

Although our method provides angularly consistent stylization for a light field, we found that our stylizations would look unnatural when switching between views. This was caused by "brush strokes" going across edges that mark the transition from a foreground object to a background object. In other words, our method like most stylization methods doesn't properly segment the image for depth discontinuities.

In addition to the disparity loss and perceptual loss described in Section 2.4, we also experimented with a depth loss term similar to that proposed by Liu *et al.* [33]. This term is calculated as the squared difference of the computed depth for the original light field to

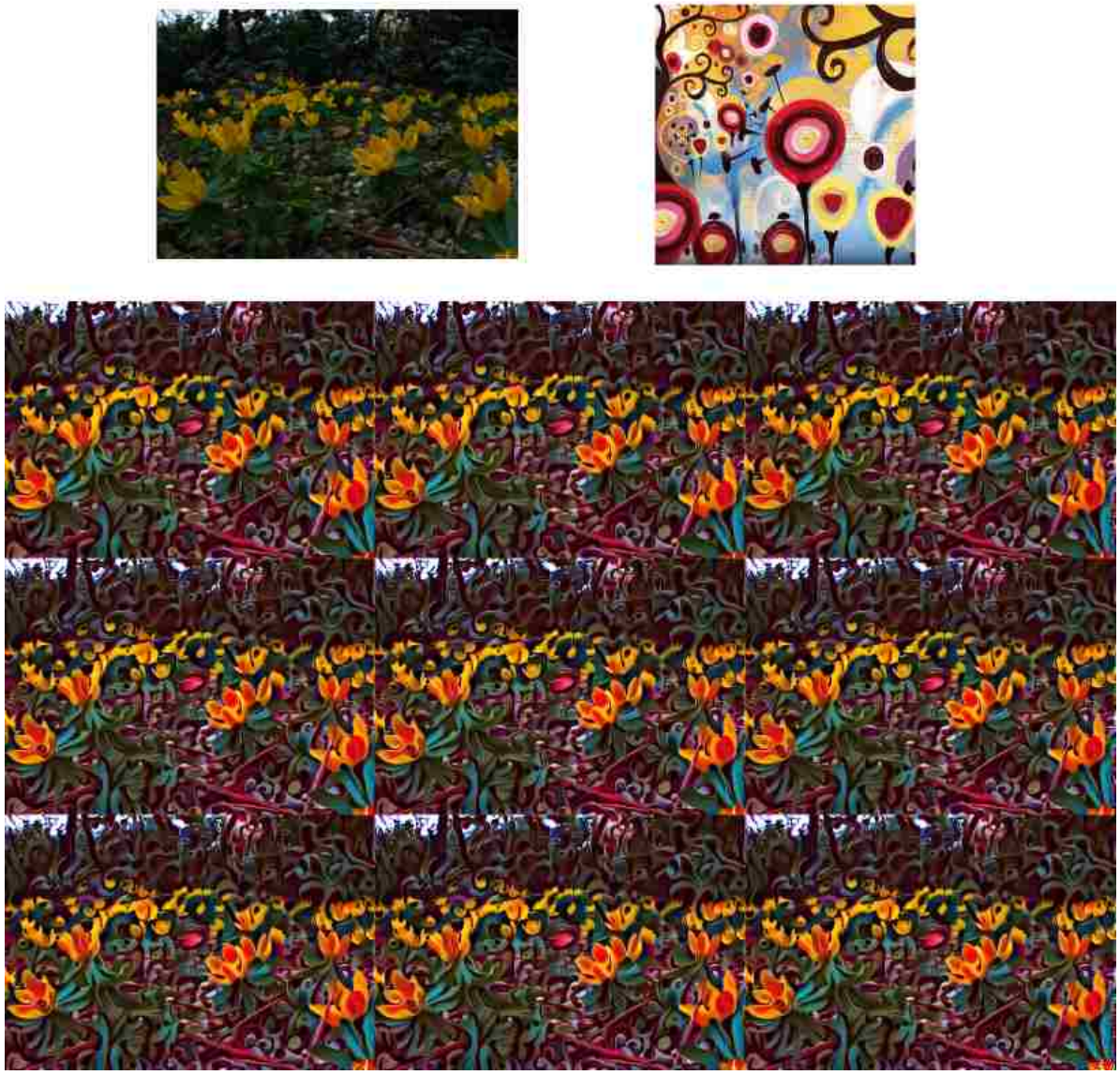


Figure 4.1: An example light field stylization optimized without a feed forward network.

the computed depth for the stylized light field (*i.e.*, $\mathcal{L}_{depth} = ||D(\hat{I}_{s,t}) - D(I_{s,t})||^2)$) . The purpose of this additional loss term is to prevent artifacts along depth discontinuities in the stylization, which should lead to smoother transitions between views.

Rather than going through an expensive and potentially non-differentiable depth calculation, the depth of both the original and stylized views of the light field are predicted with a depth prediction network. We used a Pytorch implementation [36] of the monocular depth estimation network proposed by Godard *et al.* [13]. The network was trained on the KITTI dataset [11].

As an experiment, we added the depth loss term to the loss function on our original optimization method. When this term is added, it affects the stylization by changing the relative sizes of "brush strokes". However, it is severely limited by the accuracy of the depth estimation network used. The KITTI dataset contains primarily cityscape images, so the depth prediction network does poorly when predicting depths for other types of scenes. This leads to the stylization being ill-informed as shown in Figure 4.2.

While a better depth prediction network could be used, there is an additional hurdle that must be overcome. If a pretrained feed forward network is used in the optimization, it will have no indication of depth. Thus, it would have to be initially trained to minimize depth loss before it could be used in the full algorithm.

The need for two specialized depth networks led us to abandon this line of work.

4.3 Gibbs Loss

When analyzing our stylized light fields, we found that most artifacts were unsurprisingly in the masked occlusion regions. These artifacts occur because there is no information from the central view to inform the stylization. In order to combat this, we attempted to add an additional term that computed the Gibbs energy [12] along boundaries in the masked region.

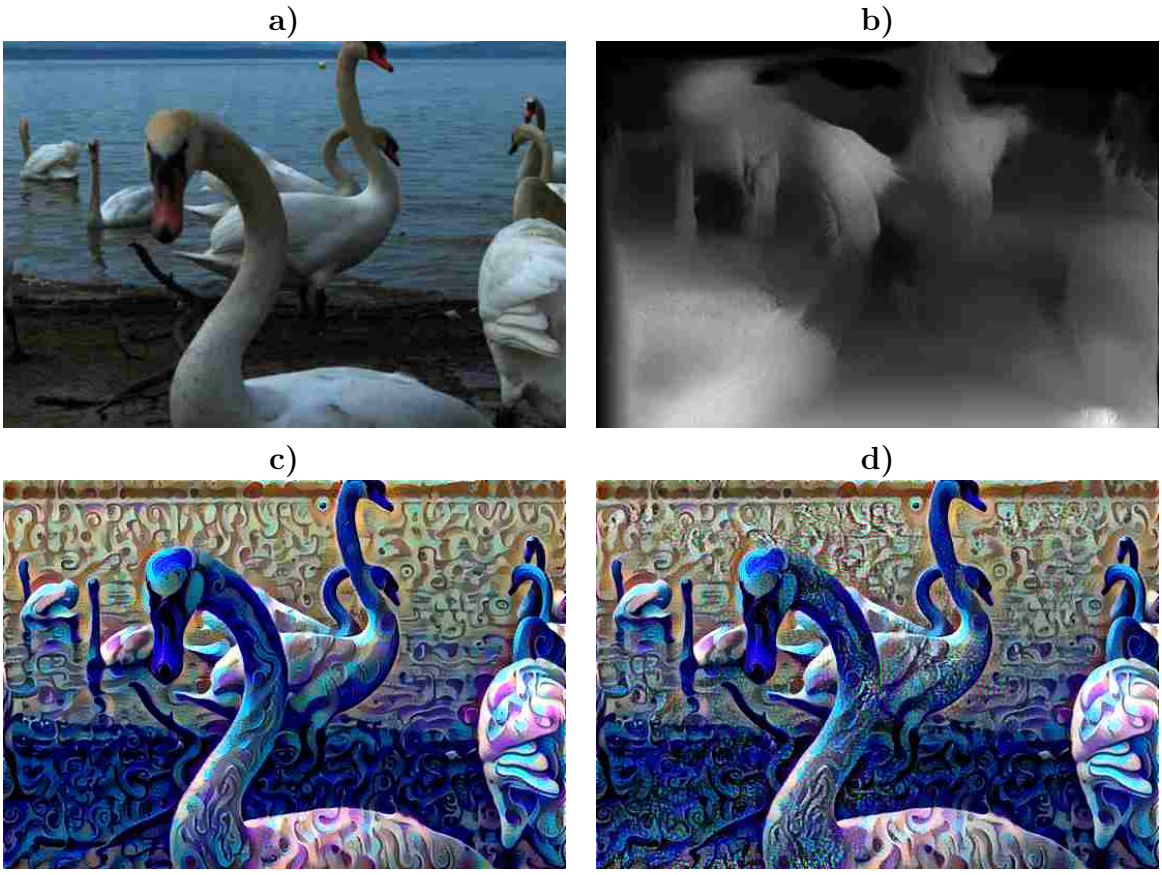


Figure 4.2: a) The central view of the light field, b) The predicted depth, c) The stylization with no depth loss term, and d) The stylization with the depth loss term. Note that the depth loss does affect the stylization, but a poor depth estimation causes many undesirable artifacts.

Figure 4.3: A light field stylization with a high Gibbs loss term. The high value illustrates the smoothing that occurs along occlusion boundaries.

This was done by multiplying the Gibbs energy at each pixel by the gradient magnitude of the mask:

$$\mathcal{L}_{Gibbs} = \sum_i ||\nabla \hat{M}||_i^2 \sum_{k \in N(i)} (\hat{I}_i - \hat{I}_k)^2. \tag{4.1}$$

We call this loss term *Gibbs Loss*. In theory, this should smooth results in uninformed areas, giving a more visual appealing transition along occlusion boundaries as shown in Figure 4.3. We found in our results that while Gibbs loss removed artifacts along occlusion boundaries, it generally led to undesirable effects that gave the stylization a unauthentic visual feel, as shown in Figure 4.4. Thus, we did not include this loss term in our final results.

4.4 Feed-Forward Network

A very apparent weakness of our method is the computation time required to optimize the stylization for the light field. It remains an open question whether this process could be

Figure 4.4: An example stylization with the included Gibbs loss term. The central view (left) is compared to the view two to the right of center (right). Even small movements between views causes unrealistic blurring in the final result.

learned by a network so that consistent light field stylization can be done with a forward pass. Similar to the way Johnson *et al.* [25] trained a network over the stylizations generated by the work of Gatys *et al.* [9], our results might be used as the training set for a future feed-forward light-field stylization method.

One major obstacle that remains to be overcome in general is using a light field as input for a neural network. Given current technology and network architectures, it is impractical to input the entire network at one time. However, alternatives that would train on a subset of views at a time might be more plausible. For example, conditional information could be appended to layers of the network to inform it of the current view being stylized. It has also been shown that a light field can be reconstructed from just a small subset of views of the light field [26]. These and other modifications may soon allow for a fully feed-forward light field stylization network as well as other light-field neural networks.

This line of work, along with general editing of light fields using neural networks, is the current aim of our ongoing research.

Chapter 5

Conclusion

This thesis has presented a method for light field stylization. This is done through informing the stylization network of correspondences between images. We found our results to be visually appealing while minimizing artifacts. This work could be extended by finding more computationally efficient methods through processes such as training a feed-forward network specifically for light fields.

Additionally, this work marks a step forward in light field processing methods. As light field technology continues to improve and be adopted for more computer vision applications, the need for simple methods of light field editing will continue to grow, especially as the capabilities of image processing neural networks continue to expand. Although this work is specific to style transfer, it potentially sets a foundation of consistency optimizations that will generalize to more applications.

References

[1] Anna Alperovich, Ole Johannsen, Michael Strecke, and Bastian Goldluecke. Light field intrinsics with a deep encoder-decoder network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[3] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stereoscopic neural style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[4] J. Chen, J. Hou, Y. Ni, and L. Chau. Accurate light field depth estimation with superpixel regularization over partially occluded regions. *IEEE Transactions on Image Processing*, 27(10):4889–4900, Oct 2018. ISSN 1057-7149. doi: 10.1109/TIP.2018.2839524.

[5] Michael Cohen, Steven J. Gortler, Richard Szeliski, Radek Grzeszczuk, and Rick Szeliski. The lumigraph. In *ACM SIGGRAPH*, August 1996.

[6] DepthfromLF. Accurate light field depth estimation code. `https://drive.google.com/file/d/0B2553ggh3QTcS01zU0RjOG5FTjQ/view`, Source: `https://sites.google.com/site/hgjeoncv/publications?authuser=0`.

[7] Maximilian Diebold, Oliver Blum, Marcel Gutsche, Sven Wanner, Christoph S. Garbe, Harlyn Baker, and Bernd Jähne. Light-field camera design for high-accuracy depth estimation. In *Videometrics, Range Imaging, and Applications XIII*, June 2015.

[8] Elena Garces, Jose I. Echevarria, Wen Zhang, Hongzhi Wu, Kun Zhou, and Diego Gutierrez. Intrinsic light field images. *Computer Graphics Forum*, 36(8):589–599, 2017. doi: 10.1111/cgf.13154. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13154`.

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016. doi: 10.1109/CVPR.2016.265.

[10] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul 2017. URL `https://arxiv.org/abs/1611.07865`.

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[12] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, Nov 1984. ISSN 0162-8828. doi: 10.1109/TPAMI.1984.4767596.

[13] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[14] Xinyu Gong, Haozhi Huang, Lin Ma, Fumin Shen, and Wei Liu. Neural stereoscopic image style transfer. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[15] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] Siddharth Hegde, Christos Gatzidis, and Feng Tian. Painterly rendering techniques: a state-of-the-art review of current approaches. *Computer Animation and Virtual Worlds*, 24(1):43–64, 2012. doi: 10.1002/cav.1435. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/cav.1435`.

[17] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, SIGGRAPH '98, pages 453–460, New York, NY, USA, 1998. ACM. ISBN 0-89791-999-8. doi: 10.1145/280814.280951. URL `http://doi.acm.org/10.1145/280814.280951`.

[18] Aaron Hertzmann. Image stylization: History and future. Blog post, June 2018. https://research.adobe.com/image-stylization-history-and-future/.

[19] Aaron Hertzmann and Ken Perlin. Painterly rendering for video and interaction. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*, NPAR '00, pages 7–12, New York, NY, USA, 2000. ACM. ISBN 1-58113-277-8. doi: 10.1145/340916.340917. URL `http://doi.acm.org/10.1145/340916.340917`.

[20] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 327–340, New York, NY, USA, 2001. ACM. ISBN 1-58113-374-X. doi: 10.1145/383259.383295. URL `http://doi.acm.org/10.1145/383259.383295`.

[21] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[22] Sunghoon Im, Hae-Gon Jeon, Hyowon Ha, and In So Kweon. Depth estimation from light field cameras. In *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 190–191, Oct 2015. doi: 10.1109/URAI.2015.7358863.

[23] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[24] Ole Johannsen, Katrin Honauer, Bastian Goldluecke, Anna Alperovich, Federica Battisti, Yunsu Bok, Michele Brizzi, Marco Carli, Gyeongmin Choe, Maximilian Diebold, Marcel Gutsche, Hae-Gon Jeon, In So Kweon, Jaesik Park, Jinsun Park, Hendrik Schilling, Hao Sheng, Lipeng Si, Michael Strecke, Antonin Sulc, Yu-Wing Tai, Qing Wang, Ting-Chun Wang, Sven Wanner, Zhang Xiong, Jingyi Yu, Shuo Zhang, and Hao Zhu. A taxonomy and evaluation of dense light field depth estimation algorithms. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46475-6.

[26] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016.

[27] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[28] Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Bjorn Ommer. A content transformation block for image style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM. ISBN 0-89791-746-4. doi: 10.1145/237170.237199. URL `http://doi.acm.org/10.1145/237170.237199`.

[30] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[31] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, 2017.

[32] Fei Liu, Guangqi Hou, Zhenan Sun, and Tieniu Tan. High quality depth map estimation of object surface from light-field images. *Neurocomput.*, 252(C):3–16, August 2017. ISSN 0925-2312. doi: 10.1016/j.neucom.2016.09.136. URL `https://doi.org/10.1016/j.neucom.2016.09.136`.

[33] Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L. Rosin. Depth-aware neural style transfer. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, NPAR '17, pages 4:1–4:10, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5081-5. doi: 10.1145/3092919.3092924. URL `http://doi.acm.org/10.1145/3092919.3092924`.

[34] Jingwan Lu, Pedro V. Sander, and Adam Finkelstein. Interactive painterly stylization of images, videos and 3D animations. In *Proceedings of I3D 2010*, February 2010.

[35] Sheng-Jie Luo, Ying-Tse Sun, I-Chao Shen, Bing-Yu Chen, and Yung-Yu Chuang. Geometrically consistent stereoscopic image editing using patch-based synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 21(1):56–67, 2015.

[36] MonoDepth. PyTorch, Monocular Depth Estimation. `https://github.com/ClubAI/MonoDepth-PyTorch`.

[37] Bryan Morse, Joel Howard, Scott Cohen, and Brian Price. PatchMatch-based content completion of stereo image pairs. In *Proceedings 3D Image Modeling, Processing, Visualization, and Transmission*, October 2012.

[38] Ren Ng. *Digital Light Field Photography*. PhD thesis, Stanford University, Stanford, CA, USA, 2006. AAI3219345.

[39] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Stanford tech report: Light field photography with a hand-held plenoptic camera, 2005.

[40] Gilles Puy and Patrick Perez. A flexible convolutional solver for fast style transfers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[41] pytorch. PyTorch, Fast Neural Style. `https://github.com/pytorch/examples/tree/master/fast_neural_style`.

[42] A. Raj, M. Lowney, R. Shah, and G. Wetzstein. Stanford lytro light field archive. http://lightfields.stanford.edu, 2016.

[43] Martin Rerabek and Touradj Ebrahimi. New light field image dataset. http://mmspg.epfl.ch/EPFL-light-field-image-dataset, 2016.

[44] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, Nov 2018.

[45] Jian Sun, Yin Li, S.B. Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 399–406, 2005.

[46] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1349–1357. JMLR.org, 2016. URL `http://dl.acm.org/citation.cfm?id=3045390.3045533`.

[47] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[48] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei Efros, and Ravi Ramamoorthi. A 4D light-field dataset and CNN architectures for material recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[49] Ting-Chun Wang, Manmohan Chandraker, Alexei Efros, and Ravi Ramamoorthi. SVBRDF-invariant shape and reflectance estimation from light-field cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[50] Keiji Yanai and Ryosuke Tanno. Conditional fast style transfer network. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 434–437, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4701-3. doi: 10.1145/3078971.3079037. URL `http://doi.acm.org/10.1145/3078971.3079037`.