Style Transfer for 360 images

Xin Zhang, B.Tech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

(Augmented and Virtual Reality)

Supervisor: Prof. Aljosa Smolic

September 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Xin Zhang

September 6, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Xin Zhang

September 6, 2020

Acknowledgments

I would like to express my gratitude to my supervisor, Prof. Aljosa Smolic, for providing valuable suggestions and feedback. I also want to thank my second reader, Prof. Michael Manzke, who is also very valuable to my thesis. I am appreciative of Mr. Koustav Ghosal for communicating with me every week and providing a lot of support and guidance for the promotion of my project.

I would like to thank Trinity College Dublin for giving me the opportunity to experience different humanities and gain theoretical and practical knowledge. I shall extend my gratefulness to my family and friends for their help in my study and life throughout the year.

Last but not least, my sincere appreciation also goes to all frontline workers who continue to provide essential services during the Covid-19 pandemic. Their dedication is saving countless lives and making thousands of differences.

Xin Zhang

University of Dublin, Trinity College
September 2020

Style Transfer for 360 images

Xin Zhang, Master of Science in Computer Science

University of Dublin, Trinity College, 2020

Supervisor: Prof. Aljosa Smolic

The creation of art requires the work of a professional and an investment of time. There are clear advantages to be derived from deep learning-based style transfer, a practice that entails the extraction of content and style characteristics from one form and applying it to another. Currently, VR is increasing in popularity and 360 images are being readily utilised for their capacity to capture rich scene information. As such, style transfer for 360 images will undoubtedly be a crucial tool used to address the future visual needs of users. However, in light of the special spatial distribution inherent in 360 images, special attention must be paid to ensure that the consistency of pixels and boundaries after style transfer is appropriate to create a realistic image.

The contribution of this thesis is to experiment with these problems and evaluate the solutions devised to address such problems. On the basis of fast style transfer, four distinct methods are adopted in the present study: direct processing, simple extension followed by re-cutting, SIFT, and filling with mean value after Cubemap and Equirectangular. The user's perceptual opinions will then be analyzed by carrying out standard statistical analysis. The results of this research will demonstrate that these various methods can be differentiated based on their border effects as SIFT is greater than simple, mean, and direct. However, it should be noted that some differences between different images and styles will persist.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In the creation of art, style is an abstract representation of artistic characteristics. The same objects can be represented using a vast range of styles, which have different connotations relating to context or culture. The imitation or re-representation of different styles is a highly demanding and time-consuming process requiring high levels of artistic ability and technical skill. The difficulty of this task can be reduced and satisfactory results obtained with the help of computer technology.

The process of taking the content features of an image and re-representing it in a different extracted style to redesign effect is called Style Transfer. Style transfer using computers can be traced back to the image texture generation technology which was in use before 2000. Researchers used complex mathematical models and formulas to interpret and generate textures. However, manual modelling is time-consuming and laborious, and computing power at that time was comparatively limited. Gatys et al. [1][2][3] changed this situation in 2015. They proposed the style transfer algorithm based on Convolutional Neural Networks (CNN). By processing high-level abstract features, it can be effectively implemented and relatively ideal results obtained with obvious improvements in attributes such as texture, colour, and structure. Image features are combined in accordance with human visual habits, with excellent versatility and ease of use, eliminating the needs to repeat complicated mathematical processes.

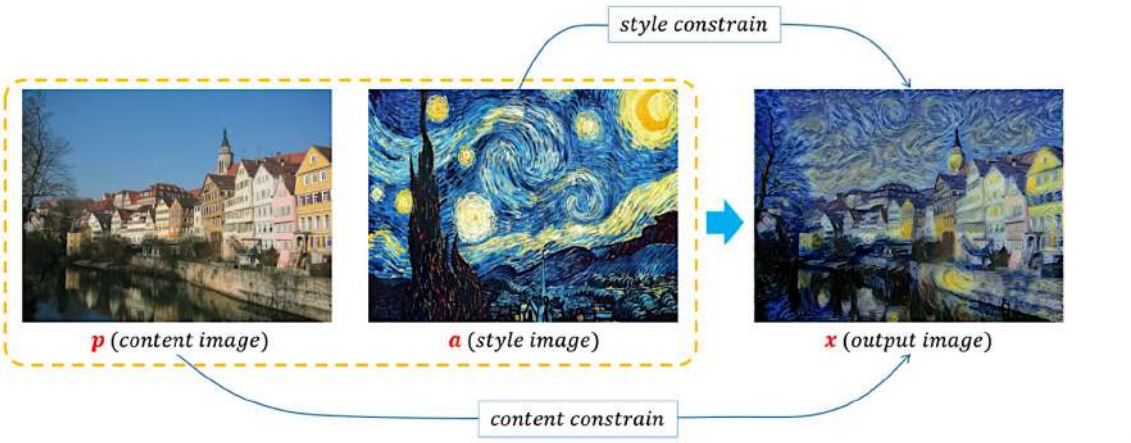


Figure 1.1: Style Transfer using deep learning

Virtual reality and panoramic imaging are technologies that construct virtual environments based on 360-degree images. The functionality of VR is highly valued and is widely used in the entertainment, medical, education, film, and television fields. The perceivers can observe the scene from any perspective and experience almost complete visual immersion. Compared with traditional images, panoramic images can capture more scene information in greater detail. Users can use head-mounted displays (HMD) or other ways to render different perspective images in real-time. Image stitching technology is used to combine multiple images with overlapping parts into a single wide-angle image. The stitched images need to be as close as possible to the original image, without obvious seams. Due to the special distribution of 360 images, It is particularly important for 360 image slices after style transfer to maintain consistency of edge style and splicing with adjacent slices.

1.2 Motivation

In today's rapidly developing information age, digital images have penetrated every corner of social life as a common and effective information carrier. In the field of cartoon animation, for example, many of the animated works of the famous Japanese director Makoto Shinkai are created based on natural scenes. The animated scenes are highly similar to real scenes, with distinctive colours and styles, and are universally admired.

Figure 1.2: Animated works of Japanese famous director Makoto Shinkai

In the field of computer vision, the traditional image style transfer method has many drawbacks in practical operation. The earlier approach was complex and laborious, involving several stages. First the image to be transferred had to be analysed in great detail, and the features and aspects to be altered precisely mapped and quantified. Mathematical formulas and complex models would then have to be constructed, frequently on a bespoke basis, to alter all of the mapped points and features in a precise way to achieve the desired result. Generally, the end results were disappointing, and the effort expended largely fruitless as the formulas and models could rarely be re-used. The breakthrough came as the result of the development of Deep Learning, in which computers themselves learn as they refine and repeat processes, with transferrable results, avoiding the need for constant human intervention to improve processes. The application of deep learning to image style transfer has produced excellent results, effectively analysing, altering and regenerating sophisticated high-level abstract features of images in a manner that mimics human vision. Systems based on deep learning are comparatively easy to use and can be applied in a variety of situations.

Famous image processing applications such as prism, Instagram, and DeepForge, which are very popular with users, apply style transfer technology to convert users' photos into pictures with brushstrokes in the styles of famous artists. What is more, special effects have become an indispensable element in movies. On the premise of ensuring high migration speeds in image stylization, style transfer can be used in special effects production. Compared with traditional film and television productions, it offers fascinating visual effects and is more appealing than real videos. For example, neural style

transfer technology was used to shoot the movie 'Come Swim', which was visually a perfect integration of impressionist painting style and film content.

Virtual Reality (VR) applications are becoming more and more popular. There is a growing demand for image processing methods suitable for spherical images and video. Although style transfer is a relatively mature technology in the field of computer vision, research on 360 imaging has been limited. Spherical media are usually presented using two-dimensional projection. The most commonly used tool is Equirectangular Projection (ERP). However, because of the two poles of a sphere, there may be considerable distortions of the procedure. This inconsistent deformation is problematic for style transfer. If a 360 image is directly style-converted, the algorithm will run based on the distorted content. At the same time, it may be difficult and require large amounts of computing power to train entire 360 images due to the limited pixel-bearing capacity of the transfer model. As a result, the style will change and deform while using HMDs. Moreover, if the 360 images are cropped and stitched, there will be problems such as unwanted border artifacts and inconsistent styles.

It is therefore extremely important to research style transfer in dealing with 360 images using CNN. Combining VR and style transfer can quickly beautify, edit, and render 360 images with colourful and diverse results, which will satisfy users' visual requirements. Expanding the applicable scenarios for 360 image style transfer is of great significance for the promotion of its commercial application. The advancement of art is closely related to human aesthetics. As there is currently no absolutely objective method to measure the visual quality of one picture, in the final method evaluation stage, I introduced the pairwise comparison based on human tasteful decisions to acquire subjective judgments.

1.3 Dissertation Structure

The main content of this thesis consists of six chapters, the specific content of each chapter is as follows:

Chapter 1: Introduction. This chapter introduces the research background of style transfer of 360 images based on deep learning and CNNs, the significance of the

problem addressed and the motivations for exploring and extending the transformation on areas of 360 images.

Chapter 2: Background research and literature review. This chapter discusses the basic concepts of the entire research, summarizes the current mainstream style transfer, and briefly introduces 360 images.

Chapter 3: Method. This chapter introduces the theory of fast style transfer algorithm and then we describe the principles of various methods to improve the border consistency towards 360 images after style transfer, as well as the relevant qualitative and quantitative methods used to evaluate its performance. By processing between adjacent slices of stylized images, the cracks of the stylized images can be improved.

Chapter 4: Experiments. This chapter implements the methodology in detail, analyzes the boundary results of 360 stylized images, and then conducts user study and applies professional statistical methods to the standard analysis.

Chapter 5: Results. This chapter presents the results of the project. We examine the perceptual subjective observed results and furthermore analyse the data of user study for evaluating the performances of these different methods.

Chapter 6: Conclusion. This chapter summarizes my work, discusses the issues that arise in the subject's research, with suggestions for improvement and future work to extend this study.

Chapter 2

Background Research

This chapter describes the main image style transfer methods based on deep learning, including Image-based Iteration and Model-based Iteration.[4][5] The former refers to optimize and iterate directly on noise image; the second is iterative optimization of neural network model, which can realize fast style transfer by network feedforward. Then, the related technologies are analyzed and discussed systematically. Finally, the 360 images are briefly introduced.

	Methods	Representative work	Pros and Cons
Image-based Iteration	Maximum Mean Discrepancy (MMD)	Gatys et al.[1][2][3]	The stylized images have good quality with well-controllable texture and color, and no training data is required. However, the calculating time is long and highly dependent on the pre-trained model.
	Markov Random Field (MRF)	Li et al.[6]	
	Deep Image Analogy (DIA)	Liao et al.[7]	
Model-based Iteration	Generative model	Johnson et al.[8] [9]	The computing speed is fast and suitable for industrialization. But the image quality needs to be improved, and a lot of data is required for highly-cost training.
	Image reconstruction	Li et al.[10]	

Table 2.1: Image style transfer methods based on deep learning

2.1 Convolutional Neural Networks (CNNs)

From the perspective of the deep learning and neural network development, McCulloch and Pitts[11] first proposed the famous M-P Model in 1943. Rosenblatt[12] put forward the Single-layer Perceptron Model in the late 1950s, which brought the research from theory to practical application and triggered the first upsurge of related research. The second research craze was the Backpropagation Algorithm implemented by Rumelhart[13] in 1986. Lecun[14] et al. proposed LeNet5, which used the Gradient-based BP algorithm to supervise the training of the network. This was successfully applied in the field of handwritten text recognition. In 2012, AlexNet, proposed by Krizhevsky et al.[15], won the championship of ImageNet. After that, new CNNs models such as VGGNet from Oxford University, GoogleNet from Google, and ResNet from Microsoft continued to appear.

CNNs is a supervised neural network, including convolutional layers, pooling layers, fully connected layers, etc., each layer contains multiple feature maps of multiple neurons. These layers can achieve the process of vector feature extraction through filter, and use gradient descent method to minimize the loss function, so as to iteratively adjust the parameters and improve the accuracy of the model. The convolutional layers refer to learn the feature representations of the input data and are composed of many convolutional kernels, which are used to calculate different feature maps. By reducing the connections between the convolutional layers using the pooling layer, including average pooling and maximizing pooling, the structure is not prone to overfitting. Finally the fully connected layer functions as a classifier.

2.2 Feature Extractor

The VGG-19 model can be used to analyse and map image features as pioneered by Gatys et al.[1][2][3] Abstract features are processed in the middle layer of the application, where they are represented as filtered feature vectors, namely abstract feature representations. The result is a representations of the high-level abstract features of the image and features can be reconstructed according to the representations of a specific intermediate layer, in which the feature information extracted from the lower layer

is more refined and the feature information extracted from the higher layer is more granular. In VGG19 Network, content features can be processed using '*conv*4_2', and style features using *conv*1_1','*conv*2_1',and so on.

2.3 Image-based Iteration

The main idea of Image-based Iteration is to use the pre-trained CNN model as the feature extractor so that the white noise image can match the content features and style features at the same time, and finally obtains a stylized synthesis image. The following content will discuss three representative methods appearing in the development process: Maximum Mean Discrepancy, Markov Random Field and Deep Image Analogy.

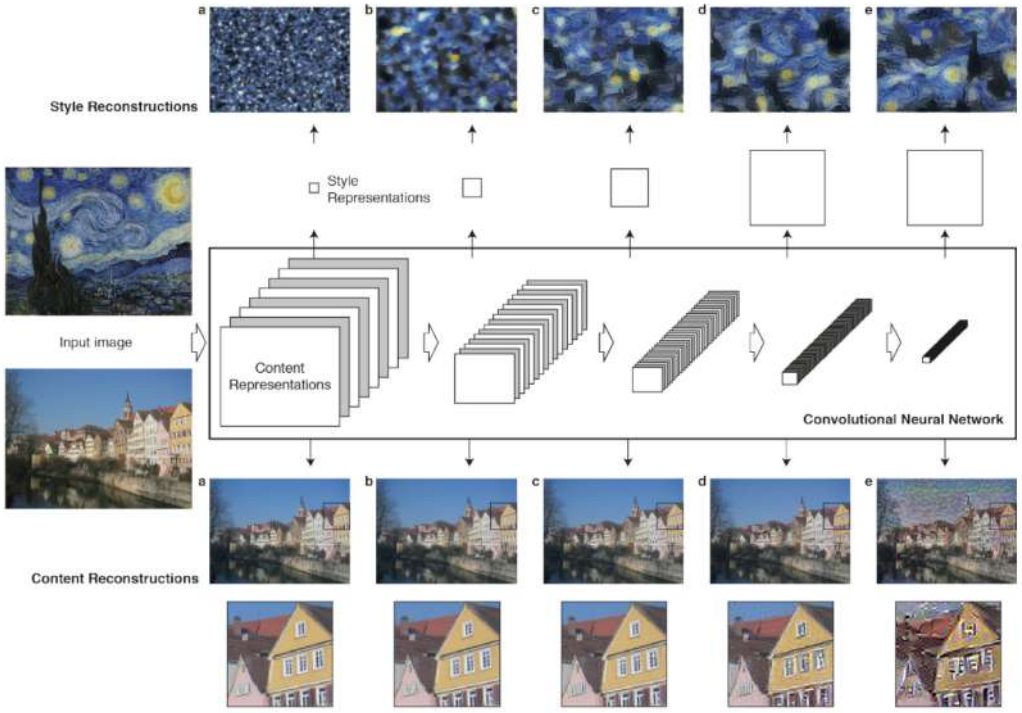


Figure 2.1: Style Transfer process including Style Reconstructions and Content Reconstructions

2.3.1 Maximum Mean Discrepancy (MMD)

Gatys et al.[1][2][3] initially found that by reconstructing the abstract feature representations of the middle layer of the CNN network, the abstract content and style representations can be extracted from any image by constructing the eigenvector Gram Matrix. The core idea of the Gram Matrix is equivalent to minimizing the maximum mean difference of a specific representation. Specifically, given a randomly generated white noise image x along with the content image x_c, style image x_s. The total loss function can be formulated as:

$$\mathcal{L}_t(x, x_c, x_s) = \alpha\mathcal{L}_c(x, x_c) + \beta\mathcal{L}_s(x, x_s)$$

where α and β individually symbolize the weighting factors for content and style reconstruction. The total loss function of content $\mathcal{L}_c$ and style images $\mathcal{L}_c$ are as follows:

$$\mathcal{L}_c(x, x_c) = \frac{1}{2}\sum_{i=1}^{N_l}\sum_{j=1}^{M_l}(F_{ij}^l - P_{ij}^l)^2$$

$$\mathcal{L}_s(x, x_s) = \sum_{l=0}^{L} w_l E_l$$

where F_l and P_l respectively are their generated and original feature representations in layer l. F_{ij}^l is the activation which is the ith filter at position j in layer l. ω_l represents the weight coefficient and E_l represents the loss function of style feature in layer l. So the contribution of layer l to the total loss is then:

$$E_l = \frac{1}{4N_l^2M_l^2}\sum_{i=1}^{N_l}\sum_{j=1}^{M_l}(G_{ij}^l - A_{ij}^l)^2$$

A layer with N_l distinct filters means it has N_l feature maps with size M_l, that is, the height times the width of the feature map. G_{ij}^l is the inner product in layer l that between the vectorized feature maps i and j. And G_{ij}^l, A_{ij}^l respectively represent Gram Matrix of original image and generated images in layer l. The Gram Matrix can be detailed described as:

$$G_{i,j}^l = \sum_k F_{ik}^l F_{jk}^l$$

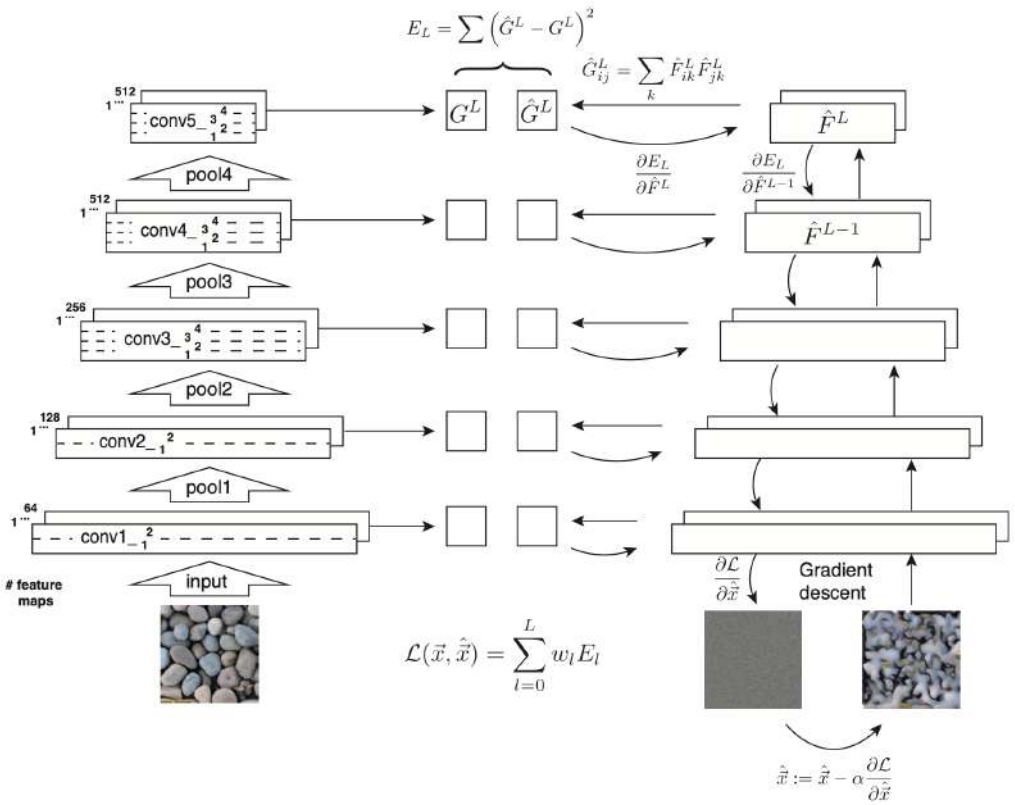


Figure 2.2: Synthesis method using Gram Matrices through the CNN

2.3.2 Markov Random Field (MRF)

Markov Random Field[16] predicts the conditional distribution between the current adjacent synthesized pixels by searching and finding all approximate neighborhoods in the sample image data, so as to retain the local structure as much as possible, and then synthesize the image texture with reasonable distribution and good effect. Li et al.[6] initially applied MRF to the CNN, and the image feature mapping is divided into regional blocks and matched to improve the visual performance of the composite image. Given a style image x_s and a content image x_c. The synthesized image x minimizes the following function:

$$x = \arg\min_x E_s(\Phi(x), \Phi(x_s)) + \alpha_1 E_c(\Phi(x), \Phi(x_c)) + \alpha_2 \Upsilon(x)$$

Among them, E_s and E_c denote the loss function of reconstructed style and content feature, the additional regularizer $\Upsilon(x)$ is a penalized squared gradient norm so as the smoothness prior on the reconstruction that can encourage smoothness in the synthesized image. $\Phi(x)$ is a set of abstract feature maps that are reconstructed and segmented in the feature extractor, α_1 and α_2 represent the weight coefficients of the content features loss function and the weight coefficients of the regularization respectively. The style loss function and content loss function are as follows:

$$E_s(\Phi(x), \Phi(x_s)) = \sum_{i=1}^{m} \|\Psi_i(\Phi(x)) - \Psi_{NN(i)}(\Phi(x_s))\|^2$$

$$E_c(\Phi(x), \Phi(x_c)) = \|(\Phi(x)) - (\Phi(x_c))\|^2$$

Here m is the cardinality of $\Psi(\Phi(x))$, which is the number of segmented feature blocks. The best matching patch $\Psi_{NN(i)}(\Phi(x_s))$ was found using normalized cross-correlation over all patches in $\Psi_i(\Phi(x))$.

Later, in order to enhance the controllability of the style effects, Champandard et al.[17] combined image semantic mapping tags based on the work of Li et al., which further improved the rationality and observability of the stylized images in terms of texture and structure.

2.3.3 Deep Image Analogy (DIA)

Among the non-parametric image synthesis methods, another classical framework is the Image Analogy algorithm proposed by Hertzmann et al.[18]. With the powerful feature learning ability of deep learning, we can effectively learn semantically mapping between two given images. Liao et al.[7] proposed a Deep Image Analogy method that aims to find a deep mapping relationship in image semantics between pairs of images. DIA is applied on style transfer that iteratively reconstruct the matching blocks so that the color, hue, texture, style and other visual information can be transferred to another image.

The mapping relationship in DIA can be expressed as $A : A* :: B : B*$. The iter-

ative optimization and reconstruction of deconvolution are carried out from the middle layer of the high-level to low-level. It can finally synthesize the stylized image $A*$ with the content from A and the style from $B*$, and vise versa. DIA can generate high-quality stylized images, however, multi-level iterative optimization in the hyper-parametric space is very unstable and the amount of calculation is high.

2.4 Model-bases Iteration

Methods based on image iteration have the problems such as low computational efficiency. While in the model iterative optimization algorithm, the image transfer network and the perceptual loss network are designed to train the generative model using a large number of images. Then we can only use the image transformation network to performs the style transfer task, thus speeding up the generation of stylized images.

2.4.1 Generative Model

Johnson et al.[8] proposed a method by training a generative model, also known as fast style transfer. The regularization of the model is based on works of Gatys et al..[1] which uses the loss function of image style perception as the regularization term, and then trains a feedforward generative model for a specific style. Compared with loss function that focuses on per-pixel when training model, this perceptual loss function squares the high-level abstract feature representation extracted by the pre-trained VGG model. Specifically, Johnson et al. used residual blocks as the core structure of the generative model, using the MS-COCO dataset[19] as the training dataset of the feedforward network. An image $\hat{y}$ is generated by solving the problem :

$$\hat{y} = \underset{f(x)}{\operatorname{argmin}} \lambda_c \ell_c(f(x), y_c) + \lambda_s \ell_s(f(x), y_s) + \lambda_{TV}(f(x))$$

Among them, hyperparameters λ_c, λ_s, and λ_{TV} respectively represent the weight coefficient of the content loss function, style loss function and the image smoothing function; x represents the network input, $f(x)$ represents the generative model; y_c represents content target and y_s represents the style target.

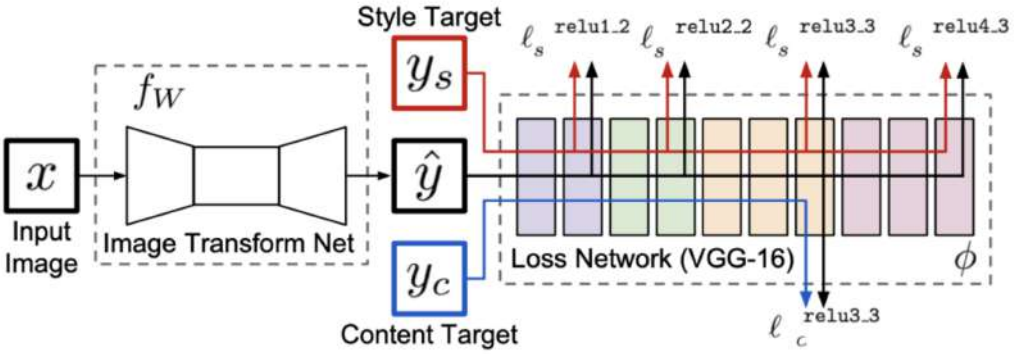


Figure 2.3: Fast Style Transfer Architecture

In addition, the Generative Adversarial Networks (GAN)[20] also have an excellent performance in style transfer. Li et al.[21] combined GAN and MRF, generated realistic images and improved reasonableness of the local blocks. Subsequently, CycleGAN, DualGAN, and so on were proposed successively which achieved unsupervised style transfer. Overall, since GAN is based on the iterative optimization of the image divergence distribution rather than the content and texture, the process is somehow difficult to control.

2.4.2 Image Reconstruction

Li et al.[10] proposed an algorithm based on an Image Reconstruction Decoder, which uses a multi-level encoding and decoding strategy. This method does not require models for specific styles and avoids the problems of parameter adjustment. Specifically, given the content image and style image, the abstract feature representations in a specific layers are extracted then the vectorization features of the content image and style image are obtained by whitening and coloring transformation. Finally, the trained decoder is used to decode the stylized encoding result of the corresponding layer, so stylized image of the corresponding layer can be obtained. Perform the next cycle until the encoding and decoding operations of all layers are completed.

2.5 360-degree Images

Panoramic 360-degree images was first introduced in 1787, when Irish painter Robert Barker concocted the expression "panorama" for his compositions displayed on a cylindrical surface. A 360 picture is a plane view generated by mapping the surrounding scene with a certain geometric relationship. It can form pseudo-3D visual effects after the correction processing of the panoramic player. By using the Head Mounted Display (HMD) to render images from different perspectives in real-time, the observer can watch the scenes in any direction, thus giving people a three-dimensional sensory experience, enabling the observer to have a real sense of presence and interactive experience, as if being among them.

In recent years, 360 images have been widely used in entertainment, medical care, education, film and television, and other fields. It is also playing an increasingly important role in the fields of scientific research, teaching, and campus publicity in colleges and universities to promote the process of information construction. For example, the panoramic campus roaming system[22] is applied to the campus display, which can truly reproduce the scenes in front of viewers, allowing them to have a comprehensive and intuitive understanding of the campus scenery and surroundings.

Chapter 3

Methodology

The chapter introduces the theory of fast style transfer architecture, my fast style transfer Tensorflow implementation is an extension on Logan Engstrom's work[35], it is based on a combination of Gatys'[1], Johnson's[8], and Ulyanov's[9] work. Then we describe the principles of different methods to improve the border consistency towards 360 images after style transfer, along with the relevant qualitative and quantitative methods to evaluate their performance.

3.1 Style Transfer

3.1.1 Frameworks and VGG Network

Tensorflow[23] is an open-source library with high portability, performance and flexibility between devices and platforms. It calculates through data flow graphs, where nodes represent mathematical activities and edges speak to tensors that are interconnected between nodes. Tensorflow in Python will be used for my experiment of style transfer for 360 images and its border effect improvements. User study will be carried out by Python as well along with the data analysis actualized by Matlab.

The VGG network won the prize on ImageNet Large Scale Visual Recognition Challenge in the 2014. The feature extractors used in the overall deep learning image style transfer are mainly the pre-trained VGG19 network and VGG16 network. Its simple structure, astounding effect make it suitable for its learning and generalization ability.

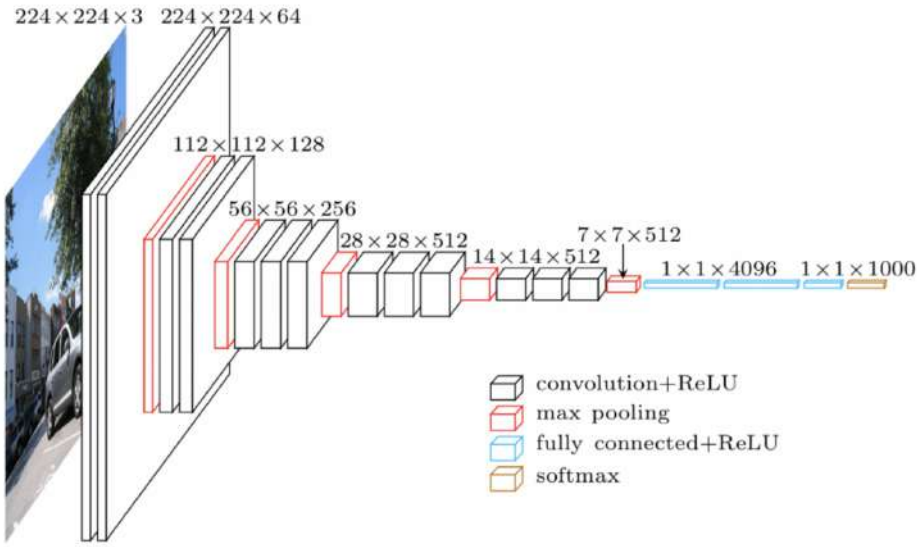


Figure 3.1: VGG-19 model structure

3.1.2 Fast Style Transfer

Style transfer can use feed-forward CNN in a Supervised Manner with per-pixel loss that only the shallow features can be extracted and perceptual similarity might be ignored. In order to achieve fast and high-quality style transfer, feedforward CNN will optimize perceptual loss[8] and generate styled images which are not synthetized by random noise but transformed by input image. This method is called Fast Style Transfer and only needs one forward calculation to get the output instead of retraining the CNN network, which meets the needs of real-time system.

The basic architecture of fast style transfer is mainly composed of two neural networks, the first is Image Transform Network and the second is Loss Network, which is used to extract features.

3.1.3 Image Transform Network

Image Transform Network is a Deep Residual Network. The purpose is to reduce the loss function and use gradient descent to optimize to produce better results. In the process of forward propagation, the mapping residual network is actually a continuous addition operations. Compared with the continuous multiplication operations of the traditional CNNs, there is no doubt that the calculation speed of residual network is better.

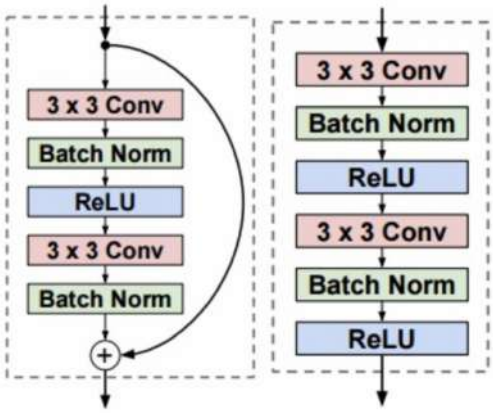


Figure 3.2: Residual block in fast style transfer network and an equivalent convolutional block

3.1.4 Perceptual Loss

Loss network defines the content reconstruction loss $l_{content}$ and style reconstruction loss l_{style}. The loss network will remain unchanged throughout the training process so different styles can use the same pre-trained network. Using feature map $\phi_j(x)$ with shape $C_j * H_j * W_j$ as the activation of the layer j of the loss network ϕ for the input x in order to generate the output $\hat{y}$ with as much content as possible. The content reconstruction loss can be represented as Euclidean distance of content feature representations:

$$l^{\theta,j}_{content}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

We also need to get the style reconstruction loss function, the Gram Matrix is used to calculate the texture differences between the two images[3]. Define Gram matrix $G_j^{\phi}(x)$ is a matrix with shape $C_j * C_j$:

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \theta_j(x)_{h,w,c'}$$

The style reconstruction loss can be expressed as the Squared Frobenius Norm between Gram Matrics of two different images:

$$l_{style}^{\phi,j}(\hat{y}, y) = \|G_j^{\phi}(\hat{y}) - G_j^{\phi}(y)\|_F^2$$

In response to obtain stylized images, we need to minimize the weighted combinations of the two loss functions and use the variation regularization $l_{TV}(y)$ to make the spatial structure smoother:

$$\hat{y} = \underset{y}{\operatorname{argmin}} \lambda_c l_{content}^{\phi,j}(y, y_c) + \lambda_s l_{style}^{\phi,j}(y, y_s) + \lambda_{TV} l_{TV}(y)$$

3.1.5 Instance Normalization

It was Ulyanov et al.[9] called attention that style transfer quality can be enhanced by Instance Normalization rather than Batch Normalization. The stylization relies upon the contrast differentiation of the style images, so the generator must reform the contrast information of the content images by changing the regularization in the CNN network. Instance Normalization calculates the mean deviation and normalize over each channel in each preparation model. Exploratory outcomes show that Instance Normalization performs better on style transfer when supplanting Batch Normalization.

3.2 Border Consistency

The border of 360 images formed by the Head-Mounted Display will be magnified by presenting the cylindrical shape, the pixels near the top and bottom of the images will be deformed. Additionally, the consequences of the style transfer algorithm are arbitrary, and there is no assurance that the styles on the left and right sides of the

360 images can be totally incidental. Any distinctions, even little ones, will cause an observable and undesirable crease. Therefore, style transfer needs to be applied in a way that matches the deformation or we get to grips with inconsistent areas.

Figure 3.3: The unpleasant scene may occur after style transfer when using HMD

A simple solution is to broaden the length of the two sides of the images[24], that is, paste part of the content on the right side to its left side and play out a similar operation on the right side. From that point onward, we manage the style transfer operation for the stitched images and erase the redundant part at the end. This method can significantly improve the effect of 360 images stitching.

However, it still cannot take care of the issues of losing edge style due to excessively large pixels at the border. In response to this phenomenon, I will try to eliminate this boundary effect through some methods introduced below, analyze and compare the points of interest and drawbacks of of these methods.

3.3 Scale-Invariant Feature Transform (SIFT)

Aimed at the problem of inconsistent borders and changes in pixels on both sides after style transfer for 360 images. The traditional stitching method Scale-Invariant Feature Transform[25] can be carried out to handle the problem. The main concept of SIFT

algorithm is to locate the key points of adjacent layers for images. The feature points generated by the SIFT have the characteristics of scale invariance and rotation invariance, which is conducive to the generation of panoramic pictures.

After feature vectors are generated, the Euclidean distance of the vectors can be used as the similarity measurement of the key points in the next step. If the number that the closest distance divided by the suboptimal distance among the two key points, is less than a certain threshold, it is determined as a pair of matching points.[26][27][28]

3.3.1 Scale Space and Keypoint Location

Firstly, SIFT downsamples the original image to continuously reduce the size of the image. Images of different sizes constitute each layer of the image pyramid. Gaussian Blur with various blurring degrees are performed to deal with each layer of the pyramid. In other words, the images are convoluted according to Gaussian Blur Template. In the event that $I(x, y)$ is the pixel coordinates of image I, and σ is the scale factor of Gaussian Blur $G(x, y, \sigma)$, then the scale space of the image is:

$$L(x, y, \sigma) = G(x, y, \sigma)I(x, y)$$

The Difference of Gaussian (DoG) pyramid is obtained by subtracting the adjacent upper and lower images in each group of Gaussian Pyramid. Each pixel of the DoG is compared with all the pixels in its neighborhood and the extreme point is the key feature point of the image in this scale.

In order to eliminate the key points that have no correlating relations because of picture occlusion and background disarray. Lowe proposed a strategy that analyzes the proportion of the nearest and the second nearest point, if the ratio is less than a certain threshold T, then this pair of matching points are acknowledged. The number of points will decrease when decreasing T, but it will be more stable, and vice versa.

3.3.2 Assigning orientations and Keypoint Descriptors

With the purpose of keeping the rotation invariance, the gradient directions of the pixels in the feature points domain are calculated, thereby assigning directions to all feature points. Each feature point is depicted by its position, scale and direction. Let $m(x, y)$ be the modulus of gradient at (x, y), $\theta(x, y)$ be the direction of gradient at (x, y), and L be the scale of all feature points. The formulas of the gradient with its direction are as follows:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = arctan(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)})$$

Within the $4*4$ neighborhood of the scale space where the extreme point is located, the gradient of each pixel in 8 directions is interpolated to describe the key point, thereby generating the key point feature descriptor, namely the feature point. The feature descriptor is shown in the Figure 3.4. Each minor square serves as a pixel, the length of the arrow represents the amplitude of the pixel, and the arrow direction means the direction of the pixel gradient.

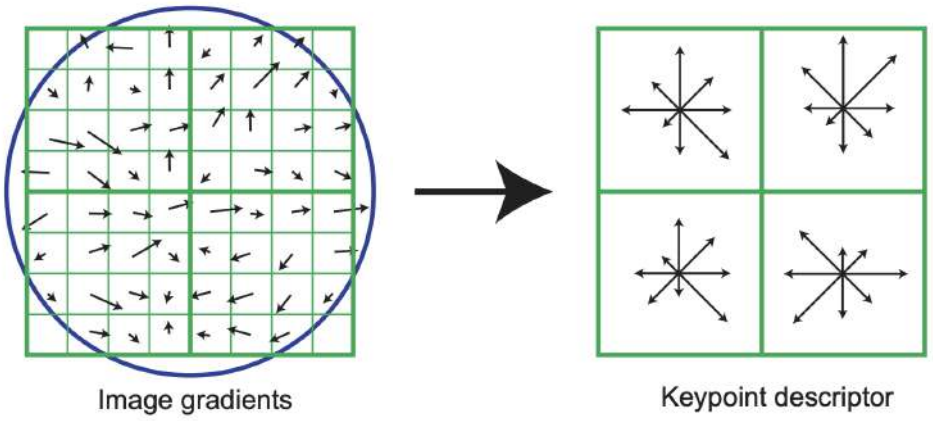


Figure 3.4: Generating a keypoint descriptor

3.3.3 RANSAC estimate Homography Matrix

In an attempt to improve the stability of feature point matching, Random Sample Consensus (RANSAC)[29] is requested to calculate the Homography Matrix between two images. The inner points should be retained and the outer points should be removed as much as could reasonably be expected, so that the calculated Homography relationship can satisfy feature points as much as possible. The specific steps of RANSAC algorithm are as per the following:

(1) Randomly select 4 pairs matched Keypoints, and calculate the transform matrix by Linear Least Square Algorithm as the initial image transform matrix.

(2) Calculate and count the point pairs that match the Homography Matrix H. If the distance among the pairs after H transformation is less than the threshold, the point pair is recorded as an inner point, otherwise it is an outer point.

(3) The transform model with the most matching points is the optimal model and update it to a new matrix after rehashing the initial two stages. Thus we will have the final H generated between images.

3.3.4 Mosaicing

Provided that the images are directly stitched even after SIFT, there will be obvious traces of the overlapping area, which affects the image quality and visual experience. Therefore, an image fusion algorithm is required to merge the overlapping parts. In this paper, we will use linear weighted algorithm, which is easy to calculate and has a good stitching effect. The formula is:

$$P(x, y) = \alpha P_1(x_1, y_1) + (1 - \alpha) P_2(x_2, y_2)$$

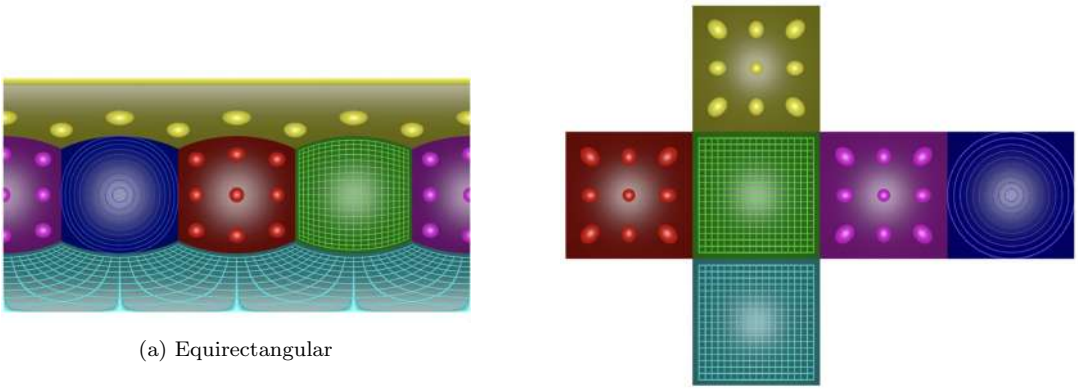
$P(x, y)$, $P_1(x_1, y_1)$and $P_2(x_2, y_2)$ are the pixel values corresponding to the overlapping areas after fusion and before fusion respectively, α is the weight, which varies linearly along with the distance from the pixel point to the overlapping edge, ranging from 0 to 1.

3.4 Border Artefacts

3.4.1 Cubemap and Equirectangular

Coping with subdivided 360 images containing multiple rectilinear projections. We will use Cubemap projection[30], which has a spherical image of six undistorted square faces. Each point on each face, i.e., the plane in Cartesian coordinates, needs to be converted to spherical coordinates, and the pixel values are obtained correspondingly. The generated images must be consistent along the boundaries of faces of cubes that are neighbouring. Every cube face has four corresponding neighbors.

The restoration from six crops to 360 images entails Equirectangular Reprojection, we are required to calculate the polar coordinates relating to each pixel in the spherical image and use polar coordinates to frame a vector and discover the pixels comparing to the vectors, then the average value is applied to avoid aliasing effects.



(a) Equirectangular

(b) Cubemap

Figure 3.5: Conversion between Cubemap and Equirectangular

3.4.2 Filling missing regions

To achieve the consistency constraint, we extend the cubes so that they can overlap with the adjacent face. Then the six cubes are stylized in sequence to make sure that the cut edges are consistent, that is, the Equirectangular Reprojection cannot introduce erroneous discontinuities along the edges of the cube after style transfer.[31]

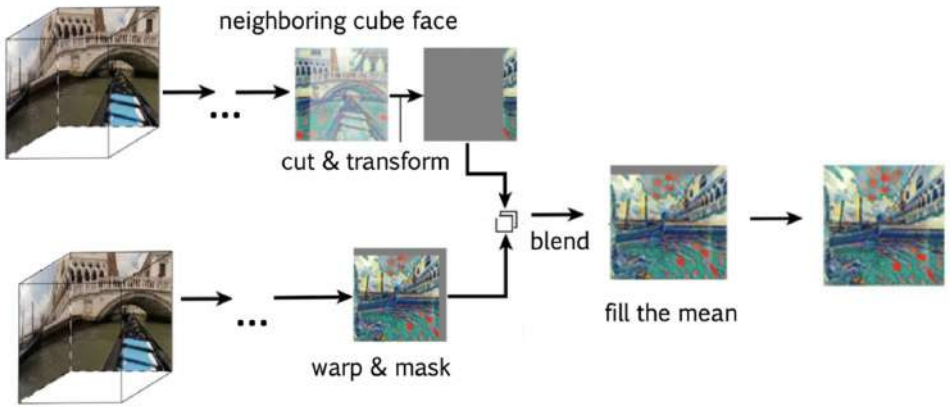


Figure 3.6: Outline of the 360 images processing pipeline and illustration of crops generation process to adapt to border regions

The illustration of the process can be shown in Figure 3.6. Introducing edge consistency significantly diminishes false gradients along with crops. Nonetheless, it brings in new gradients for the inner edge of the previous images, especially in the areas that have little structure. To further revise this problem, we experiment with filling mean values in missing regions in the prior image. This fine-tuning technique put style forward to cover false edges. The visual quality has improved even the gradient magnitude of quantitative measurement is still increased, and it will only be noticed for untextured parts, for example, the blue sky on the top.

3.5 User Study

3.5.1 Pairwaise Comparision

Evaluating the quality of such a stylization is challenging, there are no standard quality metrics available, to the best of our knowledge, for comparing the effects of style transfer. Sometimes the visual qualities of natural eyes are not considered. Therefore, it is often found that the assessment results are conflicting with individuals' feelings of subjectivity.

Hence, we will ask participants to sort a set of images and indicate which the higher quality image is, i.e. pairwise comparison[32][33]. For example, if we want to analyze the best methods out of three approaches (A, B, and C), we can present the images in pairs (AB, BC, AC), and then ask the observer to choose. Consequently, we can rank the methods from the best to the worst, estimate the confidence and scale the ranking scores with the goal that it can be easily explained based on the probability of perceived quality.

3.5.2 Data Analysis

Data processing is required for the obtained csv data. The user's perceptual opinions will then be analyzed by carrying out standard statistical analysis. The observer's perceptual sentiments will at that point be dissected via standard statistical analysis. The aftereffects of this examination will demonstrate that these different techniques can be differentiated dependent on their border effects.

The first step is to convert the data into a set of comparison matrices M, one for each onlooker. The value $c_{ij} = n$ indicates that the selection condition i is n times the selection condition j, with the columns and rows correspond to the comparison conditions. Outlier analysis is performed to identify potential observers who behave differently from others. Then we will scale the results and calculate the confidence intervals. Maximum likelihood estimation accounts for the number of comparisons.

3.5.3 JNDs and JODs

We can use Just-Noticeable Difference (JNDs)[34] to scale the pair comparisons. However, the quality can be measured as Just-Objectionable-Differences (JODs). JOD is more similar to visual equivalence, which is the quality of the average scores of differences.

When the observer votes of the two conditions are equal, the JOD between the these two conditions is 0. 1 JOD, 2 JOD and 3 JOD respectively indicates that the probability of A being better than B is 0.75, 0.91 and 0.97. A negative JOD value indicates that more observers incline toward B to A.

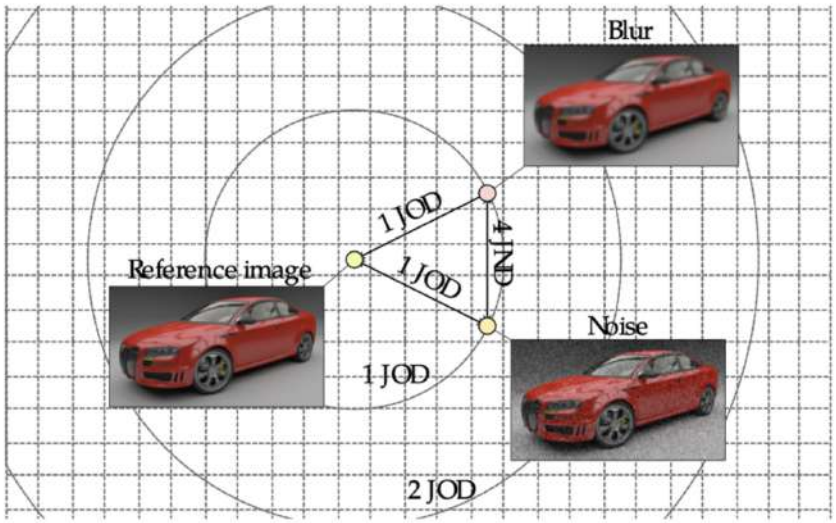


Figure 3.7: Representation of the contrast among JODs and JNDs

3.6 Method Pipeline

As per all the above techniques, we can sort out a reasonable processing pipeline. The particular operations will be clarified in the following section.

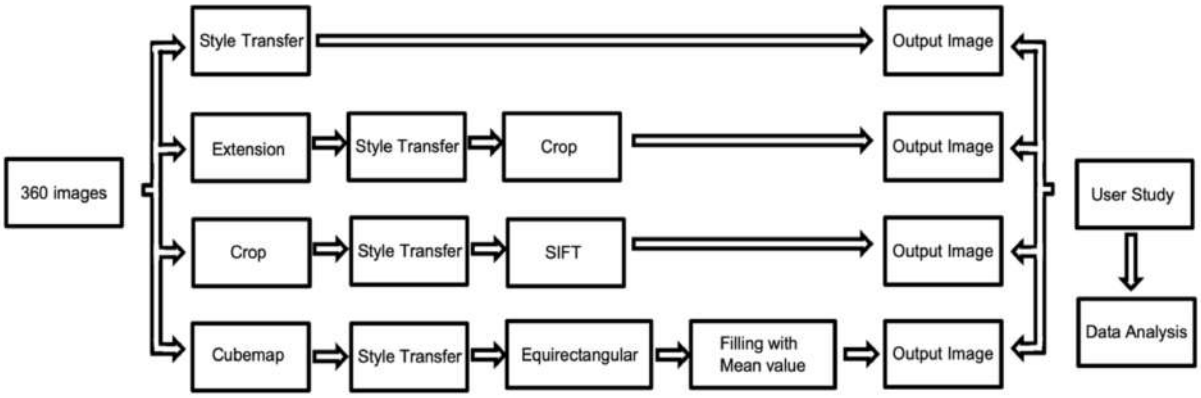


Figure 3.8: Style transfer for 360 images pipeline

Chapter 4

Experiment

In this chapter, we carry on the concrete implementations as indicated by the hypothesis clarified previously. We first discuss the setup needed for the experiment and the preparation of the dataset. Then we discuss the specific operation of using VGG19 to achieve fast style transfer. Next, we will deal with the inconsistent boundaries of 360 images. Finally, we will make use of the pairwise comparison with actualize the user study for conducting data analysis experiments applicable to the methods result. The entire implementation code and datasets can be found in the attached appendix.

4.1 Setup and Dataset

The implementation of the fast style transfer architecture used in this research is based on Logan Engstrom's work[35], using Tensorflow as the framework of deep learning application. Other Python libraries used in the training process are Pillow, scipy, numpy. As for the processing towards 360 images, matplotlib and opencv-python are further required.

Regarding the training steps of fast style transfer, a decent GPU and all the required NVIDIA software to run Tensorflow on a GPU (cuda, etc) are preferred. We only need to use the trained transfer network to handle the following steps of 360 images. This can be operated on my own laptop, which is a MacBook with 2.4GHz Quad-Core Intel Core i5 processor.

The fast style transfer was trained on the COCO2014 training dataset [2], which contains more than 83k images and is enough to deliver good results. In relation to the subsequent treatments of 360 images, V-SENSE laboratory provides me with a total of 3000 panoramic pictures.

4.2 Style Transfer

4.2.1 Fast style transfer Architecture

As stated by the theory of fast style transfer in Chapter 3, Image Transform Network and Perceptual loss Network are necessitous. We generally use the same transformation network as described in Johnson[8][36], except that the Batch Normalization is replaced by Ulyanov's Instance Normalization[37] and the offset of the output tanh layer is slightly different. The perceptual loss function is calculated and reconstructed through the high-level image features that are extracted from the pre-trained CNN, which is essentially steady with the strategy of Gatys[1]. I use VGG19 rather than VGG16, and for the most part use the shallower layer than in Johnson's implementation, for instance, use $relu1\_1$ instead of $relu1\_2$, which gives rise to a greater proportion of style features in the transformation.

In my code, style.py trains networks that can transfer a new style from artwork into images. evaluate.py evaluates trained networks given a checkpoint directory. Some indispensable parameter settings are as follows:

- batch-size: Batch size for training. Default: 4.
- checkpoint-iterations: Number of iterations between checkpoints. Default: 2000.
- content-weight: Weight of content in loss function. Default: 7.5e0.
- style-weight: Weight of style in loss function. Default: 1e2.
- tv-weight: Weight of total variation term in loss function. Default: 2e2.
- learning-rate: Learning rate for optimizer. Default: 1e-3.

4.2.2 Networks

The input of Image Transform Net should be the image to be transformed. After the model is trained, only this part of the network is used to generate style transfer. The overall Image Transform Network also is a residual network, which is made out of 3 convolutional layers, 5 residual blocks, and 3 convolutional layers.

These five residual blocks can make the network learn the identify function, thus the input and output images can share the same structural features. The five residual blocks use the same number of convolution kernels and each residual block has two $3 * 3$ convolution layers. There is no standard 0 padding in the convolution layer so it can reduce serious artifacts on the boundaries of the generated image caused by using 0 padding. Here we did not use pooling, but used strided and fractionally strided convolution for down-sampling and up-sampling. The size of the images stay at $3 * 256 * 256$. The down-sampling is performed at the beginning second and third layer of the convolutional layer, and the up-sampling is acted in the last 3 convolutional layers, so that it can reduce the computational complexity and improve its performance. Also, it can refine the perceptual domain of each pixel with the benefits that the greater the field of view in relating to each pixel in the image, the better.

```
def net(image):
    conv1 = _conv_layer(image, 32, 9, 1)
    conv2 = _conv_layer(conv1, 64, 3, 2)
    conv3 = _conv_layer(conv2, 128, 3, 2)
    resid1 = _residual_block(conv3, 3)
    resid2 = _residual_block(resid1, 3)
    resid3 = _residual_block(resid2, 3)
    resid4 = _residual_block(resid3, 3)
    resid5 = _residual_block(resid4, 3)
    conv_t1 = _conv_tranpose_layer(resid5, 64, 3, 2)
    conv_t2 = _conv_tranpose_layer(conv_t1, 32, 3, 2)
    conv_t3 = _conv_layer(conv_t2, 3, 9, 1, relu=False)
    preds = tf.nn.tanh(conv_t3) * 150 + 255./2
    return preds
```

The output images from the transformation network are fed to perceptual loss network.

```
def net(data_path,input_image):
    layers = (
        'conv1_1','relu1_1','conv1_2','relu1_2','pool1',
        'conv2_1','relu2_1','conv2_2','relu2_2','pool2',
        'conv3_1','relu3_1','conv3_2','relu3_2',
        'conv3_3','relu3_3','conv3_4','relu3_4','pool3',
        'conv4_1','relu4_1','conv4_2','relu4_2',
        'conv4_3','relu4_3','conv4_4','relu4_4','pool4',
        'conv5_1','relu5_1','conv5_2','relu5_2',
        'conv5_3','relu5_3','conv5_4','relu5_4',
    )
```

As described in Chapter 3, we aggregate the content reconstruction loss and the style reconstruction loss into the final estimated loss:

```
loss = content_loss + style_loss + tv_loss
```

For all styles of transfer experiments, we calculate the $relu4\_2$ layer of VGG19 for content reconstruction loss and style reconstruction loss is determined at layer $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ and $relu4\_1$ of the VGG19, which are shallower layer than Johnson's work.

```
STYLE_LAYERS = ('relu1_1', 'relu2_1', 'relu3_1', 'relu4_1', 'relu4_1')
CONTENT_LAYERS = 'relu4_2'
DEVICES = 'CUDA_VISIBLE_DEVICES'
```

4.3 Procedures of 360 images

4.3.1 Simple Treatment

The simple solution is to extend the length of both sides of the images, copy a part of the left side and paste it to its right side, and perform the same operation for the right side, then remove the pasted part in the transferred image. As should be obvious from the Figure 4.1, the left and right sides of the pictures have overlapping parts i.e. the whole white cars appeared on both sides.

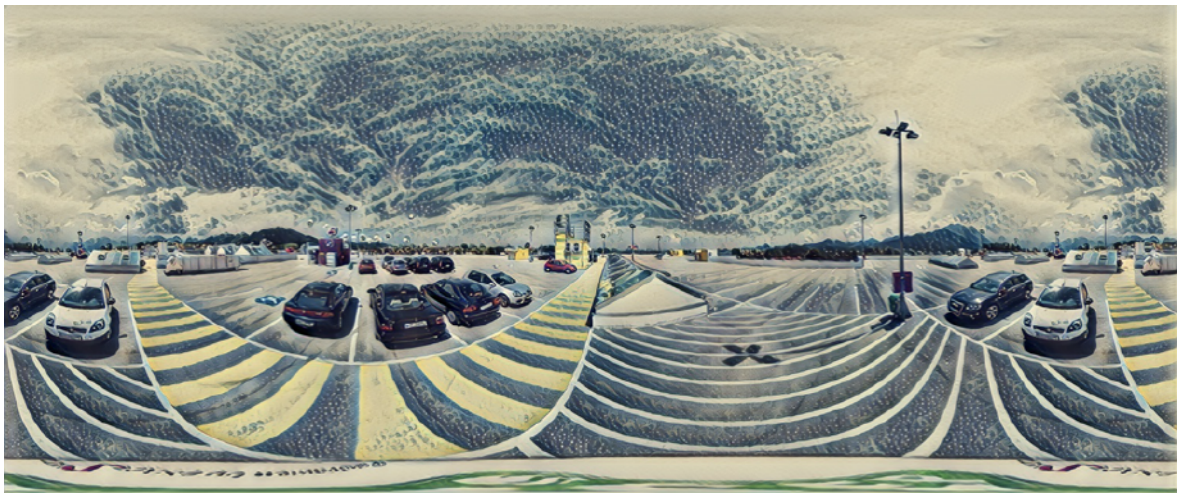


Figure 4.1: 360 images after extension and style transfer

4.3.2 SIFT

In this treatment, first 360 images are cut into 3 crops with a certain degree of overlap maintained at the boundary. Then we stylized the crops. After that, use SIFT algorithm which explained in Chapter 2 to discover feature points in different scale-spaces, and ascertain the direction of the key points, match the feature points of adjacent blocks.

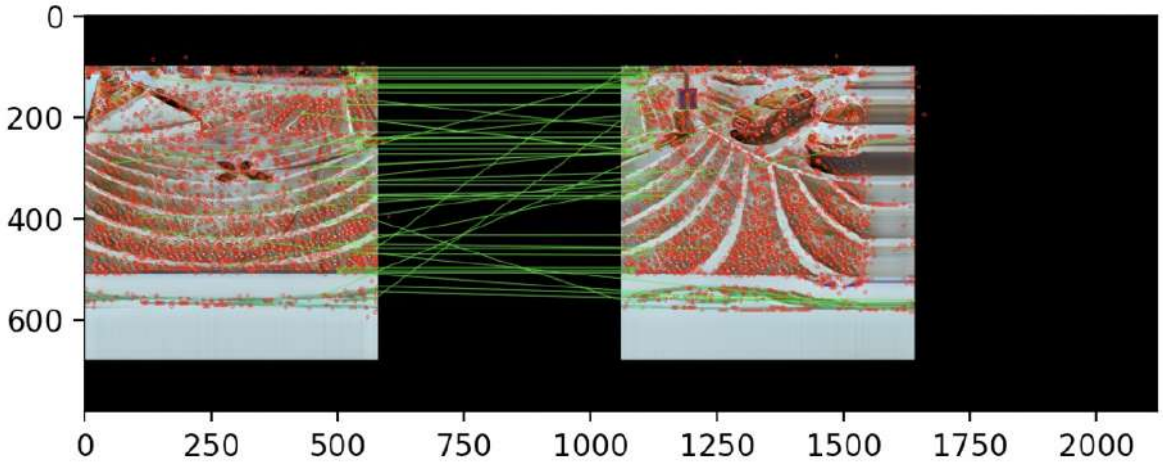


Figure 4.2: Feature points matching in SIFT

Among them, the red points are the similarity points found by SIFT algorithm, while the green line represents the similar points with higher reliability selected from all the similar points found since the similarity points found by the algorithm are not necessarily 100% correct.

After finding the matching feature points, the Homography Matrix can be calculated according to the selected similarity points. Afterwards one of the cropped images is transformed with the calculated Homography Matrix. The transformed image is intersected with the other image and the new pixel value of the overlapping area is recalculated. For details about calculating the pixels of the overlapped area, the simplest but effective way is using linear weighted algorithm.

One trick to note is that the right side of the rightmost crop of the panoramic pictures needs to be SIFT matched with the left side of the leftmost crop and remove the extra part, so as to ensure the consistency of the boundary. The detailed code is shown in the link in the appendix.

4.3.3 Border Artifacts

To stylize 360 images in this method[31][37], images must be present as Cubemap Projection with overlapping borders. Because spherical reality media is most commonly dispersed by means of 2D projection, a reprojection becomes necessary. Basically, we will translate each point from each face which is a plane in Cartesian coordinates into spherical coordinates and then gets the pixel value from the image. And vise versa, Equirectangular Reprojection can reestablish 360 images from six small crops.

With the purpose of achieving the consistency constraint and avoiding new gradients on the inner edges, I expanded the boundary 100 displacements when slicing, and experiment with filling mean values in missing regions. This fine-tuning technique can cover false edges.

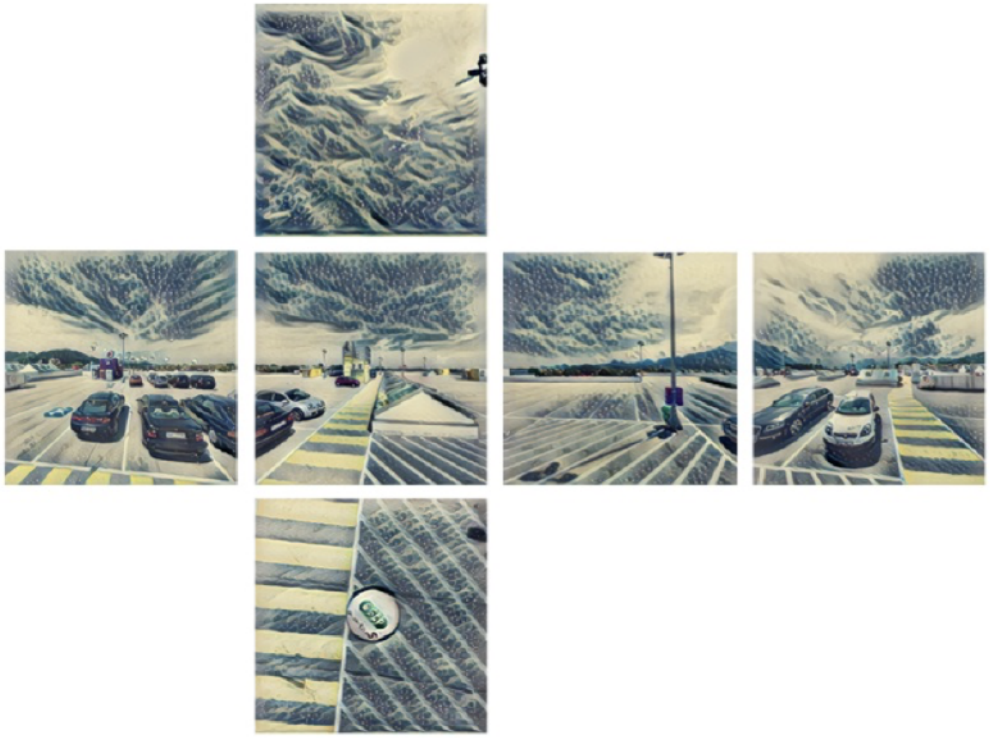


Figure 4.3: Cubemap Projection used for stylizing spherical images

4.4 User Study

There is some complexity to evaluate the style transfer for 360 images. So I chose to measure the perceptual attributes of interest to quantitatively appraise the methods. We will ask participants to sort a set of presented images and indicate the higher quality, which is also known as pairwise comparison that the users have to select the one that he thinks is better through observation. The investigation starts by collecting user details and some example stimuli are exhibited that the participants can familiarize themselves. Next, a few dummy images are displayed for which the results are not stored. Once the dummy images are done, the actual experiment starts.

Despite the fact that I have processed many different content images and style images using the above implementations, considering the experimental time spent by the tester throughout the user study, I only selected three kinds of content images with obvious structural differences in which some showed apparent distortion due to the

format while others did not, along with three styles with obvious different arts and the four methods mentioned above. The figures below show the content images and style images for user study respectively. In this case, there will be a total of $C_6^4 * 3 * 3 = 54$ pairs of comparisons. The experimenter will spend about 15 minutes and I have collected 15 samples from different ages and genders for my study.

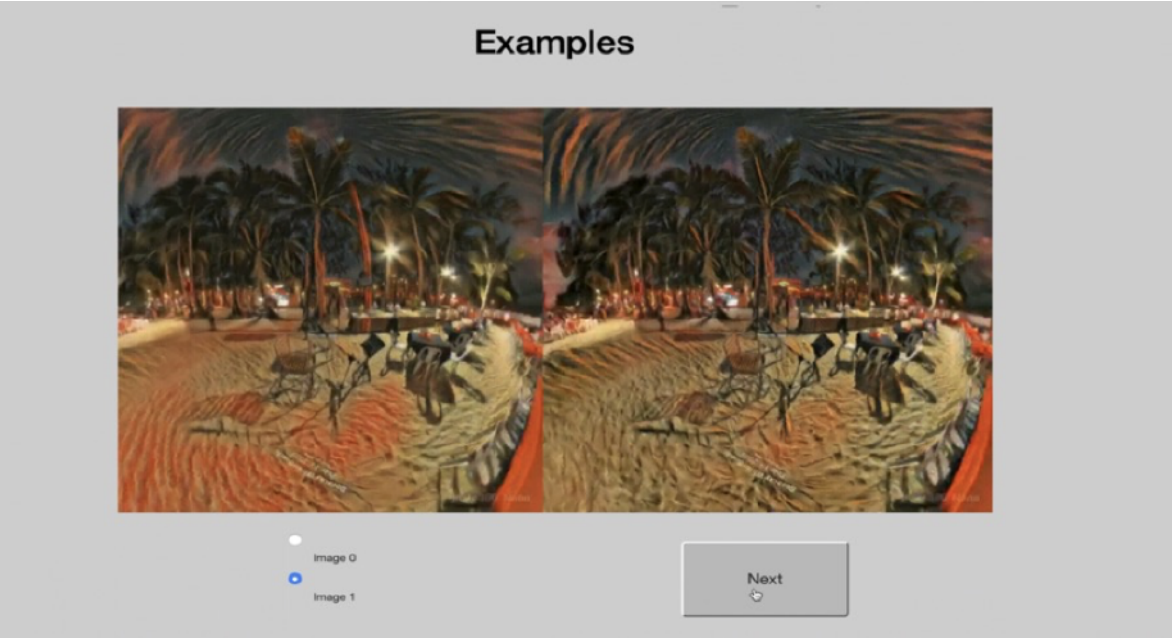


Figure 4.4: pairwise comparison user study interface

Data processing[32] is required after we have obtained the csv data from user study. The first step is to convert the data into a set of Comparison Matrices M. $[L, L\_dist] = pw\_outlier\_analysis(M)$ is performed to identify outlying data points that the observers may behave anomalously during processing. The result is a matrix of all possible viewer responses referenced by the probability of any one viewer observing a particular data. The inter-quartile-normalised score L_dist is given to each point based on this probability, which will identify the points that require analysis in greater detail. $compare\_probs\_observer(M, n\_obs)$ is used to determine the likelihoods of users selecting particular conditions. $[jod, stats] = pw\_scale\_bootstrp(M)$ is used for scaling the results for comparisons to determine confidence levels for each method. $pw\_plot\_ranking\_triangles(jod, stats)$ applies statistical analysis to permit a graphic representation of the statistically differences between pairs of conditions.

(a) Image 1

(b) Image 2

(c) Image 3

Figure 4.5: Content

(a) Wave

(b) Udnie

(c) Scream

Figure 4.6: Style

Chapter 5

Result

This chapter discusses the evaluation results of the project. We assess the subjective observation results of different methods for handling style transfer for 360 images and furthermore conduct professional analysis based on user study data to summarize and evaluate the performances of these different processing methods.

5.1 Perceptual Visual Performance

The agility of human vision allows us to clearly notice the nuances of images. The sensitivity characteristics of human eyes to errors are not absolute and persistent, and its perceptual results will be affected by many factors. In particular, people are more sensitive to differences with lower spatial frequency, more sensitive to luminance contrast differences than chroma, and the perception results of human eyes to a region are

(a) Content Image (b) Style Image

Figure 5.1: An example of content and style image for exhibition

likely to be affected by the surrounding areas.

By reason of the relatively large amount of images, I choose to use one content and one style to specifically explain the human eye's perceptual evaluations towards these methods.

Direct: In consequence of the randomness of the fast style transfer algorithm, users who use HMD or other equipments will obviously discern that the both sides of the 360 image cannot completely coincide when stitching, and there are obvious and unpleasant white seams if we stylize the 360 images directly as what marked in red box shown in Figure 5.2.

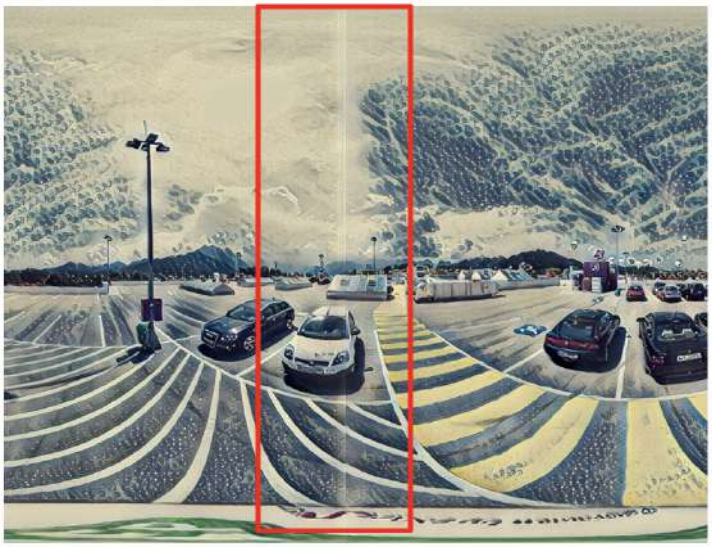


Figure 5.2: The boundary and problem of directly applying style transfer

Simple: By simply lengthening the two sides of the original image, feeding the entire extended image into the style transfer network, and finally cutting off the redundant boundaries, it is obvious to the naked eye that the gap is solved perfectly. However, due to the excessively large resolutions of the 360 image and the network architecture has certain restrictions on the size of the input images. At the same time, the panoramic image distributed via 2D distribution causes distortion of pixels. Therefore, it still cannot solve the problem of losing edge style pixels due to large pixels at the border regions.

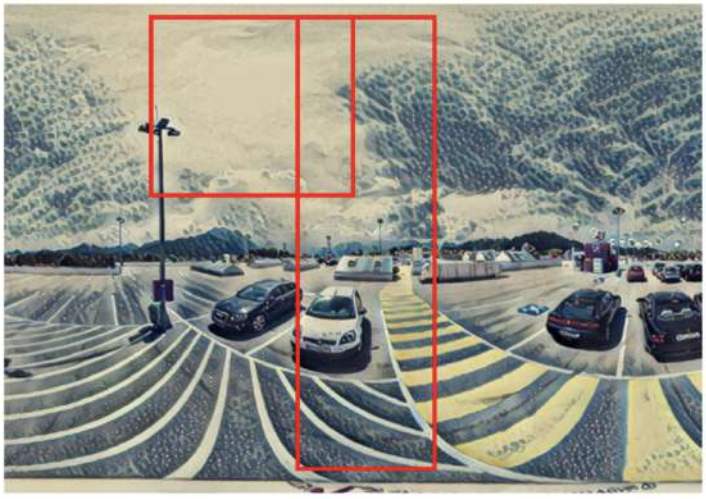


Figure 5.3: The boundary and problem of using simple method

SIFT: After applying SIFT, we can find that the left and right sides of the image can be stitched together well, but if we look at it carefully, there will still be a slight inconsistency in style.

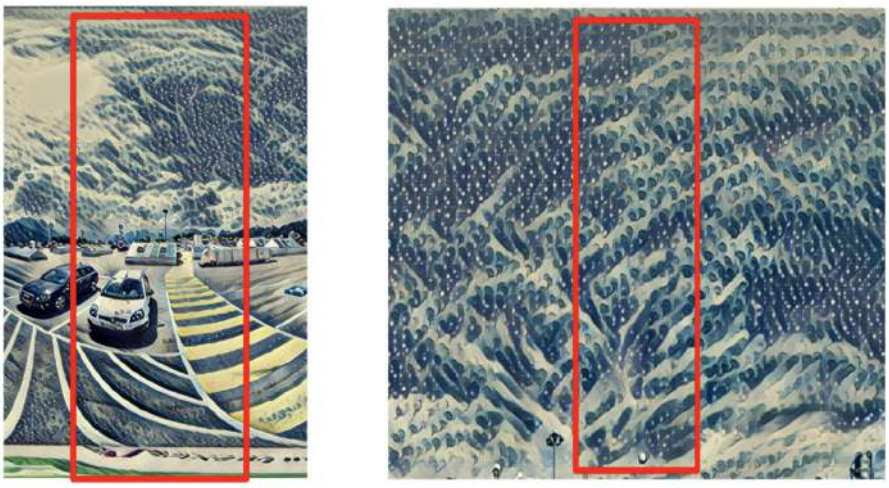


Figure 5.4: The boundary and problem of using SIFT

Mean: Restoring from six cubes into 360 images requires Equirectangular, we can find that the borders on the two sides can be well spliced. Nevertheless, it brings new gradients for the inner edge, especially in the areas that have little structure.

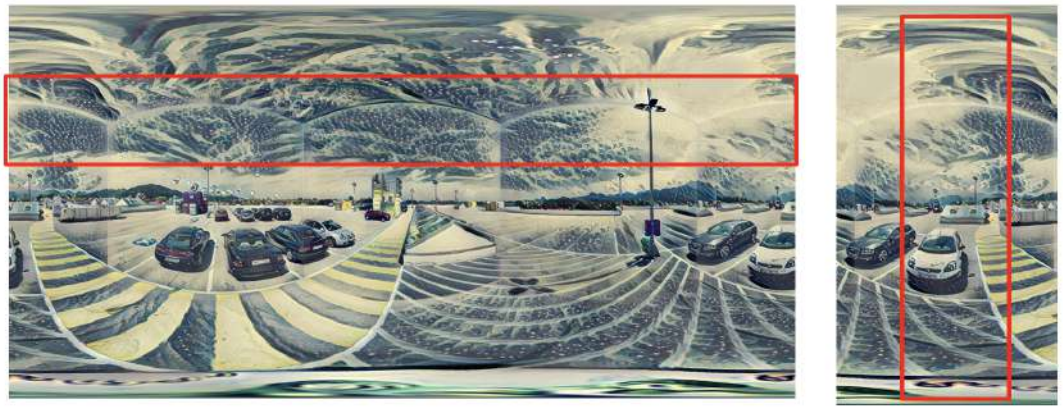
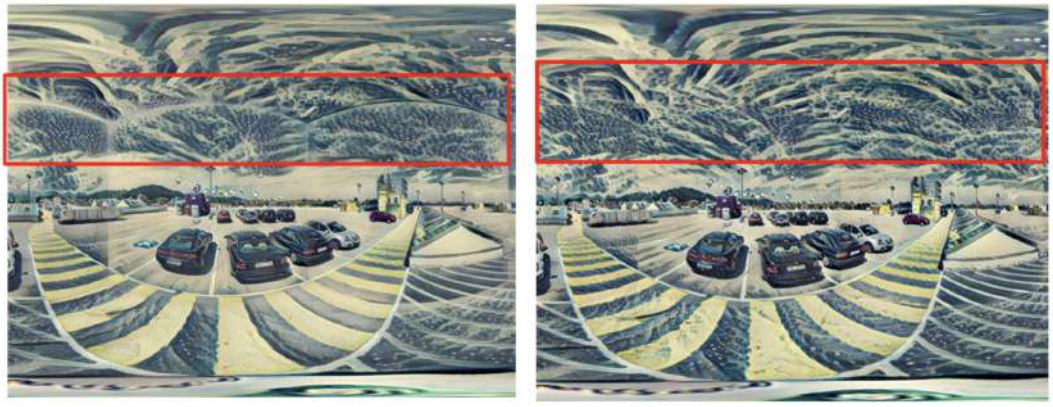


Figure 5.5: The boundary and problem of using cubemap and equirectangular

To further fix this problem, I experiment with filling mean values in missing regions. This fine-tuning technique can cover false edges. Even the gradient magnitude of quantitative measurement is still increased, the visual quality has improved and edges will be noticed only in non-textured parts, such as the blue sky at the top.



(a) None (b) Mean values

Figure 5.6: Comparison of the equirectangular with different ways to fill masked regions

5.2 User Study Data Analysis

In the csv data collected from user study, the first and the second column collect the experimenter's information, the 'scene' collects the current style or content information, the fourth and the fifth column are the different conditions of the methods, followed by the experimenter's choice and reaction time.

In order to collect the opinions of 15 different participants through pairwise comparison, data processing is required. We need to find the preference matrix and use pwcmp software to estimate the JOD scores. With the purpose of being able to analyze the performances of the processing methods from multiple angles, I categorized the collected data into different contents and different styles as two reference for further discussion about whether the differences in image content and style have a certain impact on the method itself on the basis of the overall conclusion.

observer	session_id	scene	condition_1	condition_2	selection	time
ZhangXin	1	Image2	mean	sift	1	5.05240893363953
ZhangXin	1	Image3	mean	direct	0	5.04004621505737
ZhangXin	1	Image1	mean	simple	1	9.84072184562683

Figure 5.7: Formatting pwc data example with content as reference

observer	session_id	scene	condition_1	condition_2	selection	time
ZhangXin	1	scream	mean	sift	1	5.05240893363953
ZhangXin	1	udnie	mean	direct	0	5.04004621505737
ZhangXin	1	wave	mean	simple	1	9.84072184562683

Figure 5.8: Formatting pwc data example with style as reference

Figure 5.9 show the distribution of the average perceptual quality, as well as the maximum, minimum, median and outlier of the data, which is convenient for subsequent data filtering. Figure 5.10 indicates the visualization of scaling results and confidence intervals for the dataset and the probabilities of selecting one condition over others. 1 JOD indicates that 75% onlookers have chose one condition as the better one.

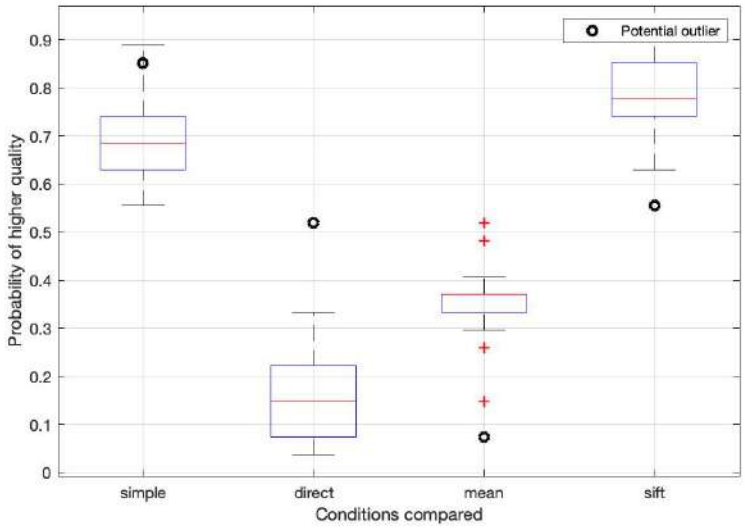


Figure 5.9: Distribution of the general perceived quality for each condition

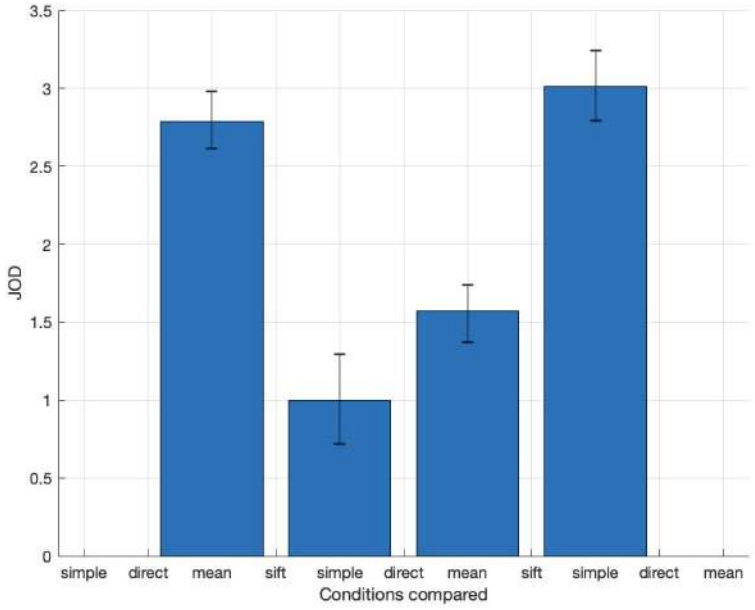


Figure 5.10: The probabilities of selecting one condition over all others

Figure 5.11 uncovers the graphical representation of the scaling. Red points speak to various conditions and are just associated with their neighbors. They are positioned in order, with the least favored condition on the left. Blue lines shows the statistically measurably differences, the statistic number indicates the probability that an average situation that the observer will choose the condition on the right as superior to the condition on the left. On the off chance that the red and dashed line connect two conditions, it show that there is no statistical difference between this pair of conditions.

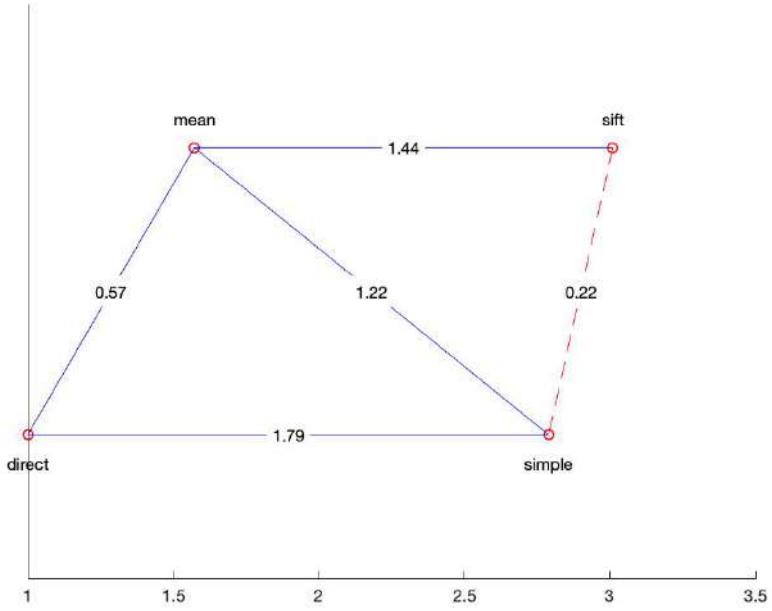
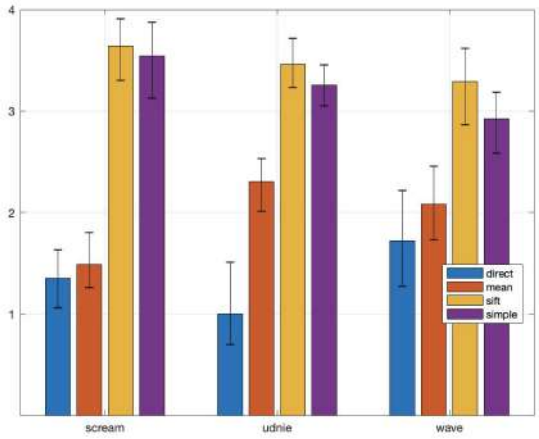
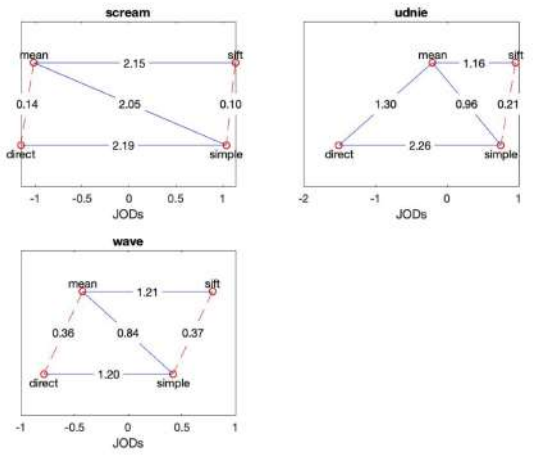


Figure 5.11: Graphical representation of the scaling

For different styles, we can observe the overall effect, but there are subtle differences between different methods. For various content images, the results show a greater discrepancy. This is considered that the structure of the image itself, texture and pixel distribution will have a certain impact during the operation of the style transfer for 360 images. For example, in the Figure 5.12 and Figure 5.13, since Image 2 and Image 3 have greater disorder via Equirectangular while Image 1 seems flatter, so it is better to use the method that fill with mean value than style transfer directly. But if there are more unstructured areas in the images like Image 1, then the method of filling the mean value is not so ideal.

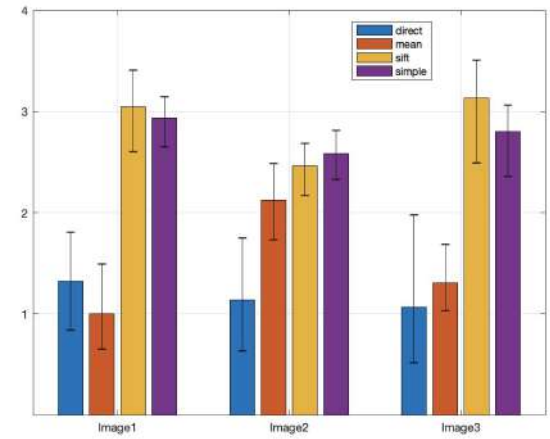


(a) Scaling and confidence intervals

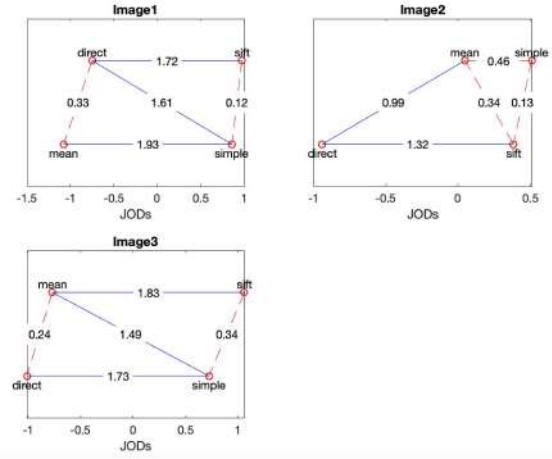


(b) Graphical representation

Figure 5.12: Different style conditions



(a) scaling and confidence intervals



(b) Graphical representation

Figure 5.13: Different content conditions

Chapter 6

Conclusion

The rise of style transfer based on CNNs has opened a new chapter in image processing technology. This technology can be utilized as an incredible assistant work for operators to create diverse masterful images. With the help of it, users do not need to have professional art skills, only need to provide relevant content and style to create arts. In today's fast-developing information age, the combination of 360-degree images and style transfer can quickly beautify, edit and render 360 VR images with colorful and diverse types, which will greatly meet the user's future visual needs, and it is of great significance to its commercial application.

Based on the framework of fast style transfer, this paper proposed different processing methods for 360 images, namely direct style transfer, simply extension and re-cutting, SIFT, and Cubemap Equirectangular and border artifacts. Those methods were performed qualitatively and quantitatively using user study and anatomized with standard statistical methods. The practice has proved that these four different methods have corresponding advantages and disadvantages. The overall order of effects is as follows: SIFT is preferable than simple, than mean, than direct, but there will be some differences between different images and styles.

Due to the limitation of time, the processing methods of style transfer for 360 images put forward in this thesis have some restraints, which need further theoretical and practical exploration.

6.1 Limitations

As mentioned above, the boundary of the 360 images formed by wearing the HMD will be magnified by presenting a cylindrical shape, and the pixels near the top and bottom of the images will be deformed. Moreover, the results of style transfer algorithm are random, which can not guarantee that the left and right sides of styles can completely overlap. In spite of the fact that the strategies I set forward could deal with the problems of inconsistent borders after the stitching caused by stereoscopic observation of 360 images, these processing methods will still bring or remain problems to a certain extent.

For simple extension, this approach disregards the problem of 360 images that the quantity of assigned increments towards the north and south pole. There are a lot more pixels "wasted" speaking to the sphere at the poles. At the same time, the resolution of one 360 image is relatively large, the structure around the picture is more likely to be ignored due to the functional limitations of style transfer network. So the degree of style effect at the border will be less or even be missing.

For SIFT, although the image combination is carried out to some extent, it still cannot ensure that the boundary of two cuts can maintain consistency of the style when feature points are matched in the areas with less structure. Chances are that the left border has more style features, and the right side has simpler features, which leads to some errors in matching the key points. At the same time, SIFT algorithm has the phenomenon of redundant feature points in image mosaic and error-prone matching, and the operation time is too long due to the uncertainty of the algorithm.

For Equirectangular, we can find that the border on both sides can be well spliced together, but it brings new gradients to the inner edge, especially in the area with few structures. Although the method of filling mean value was introduced, its impact is diverse when applying to different styles and different contents. Some image outputs can be ideal, while the cracks of others can be very obvious.

6.2 Future Works

In view of the above shortcomings, I have tried other methods to improve in the past few months, but due to the limited references and time, these works may continue to be studied as future interest after the dissertation. For example, on the basis of Cubemap and the Equirectangular, the convolution kernel may be considered to process cube boundary, so that they are forcibly consistent. The second approach is to combine spherical CNN with style transfer, such as using a new spherical data structure which can manipulate pixel using spherical coordinates, so as to address all distortion challenges while also being more computationally efficient for transfer on the whole 360 images. The third conjecture is to use distortion-aware convolution[38] for rectification as significant errors in depth prediction, especially along the y-axis, may occur when the spherical pixels are projected onto the plane. Part of the network used for encoding of the input content can be adjusted by supplanting standard convolutions with distortion-aware ones. The grid of sampling will rectify the receptive field in accordance with the coherence and contortion of 360 images. So the projected images, in this case, will not reveal these discontinuities and seem more natural. In addition, it can also be improved in terms of style transfer, combined with more advanced algorithms, making the effect of style more ideal and the calculation more productive.

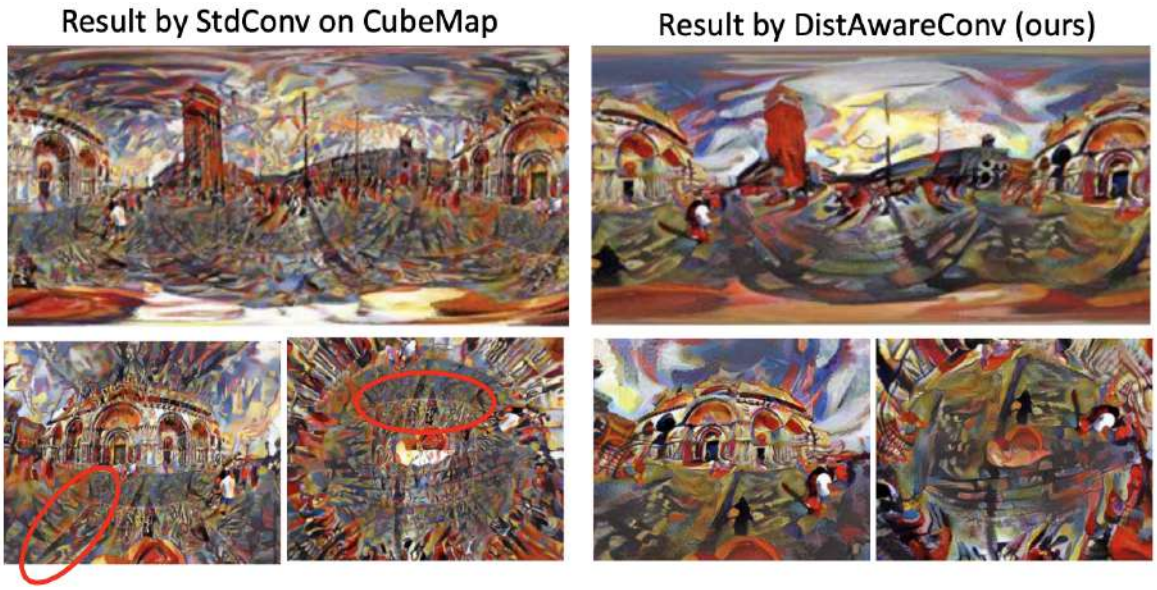


Figure 6.1: Distortion-aware convolution used to replace standard convolutions

Bibliography

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015.

[2] 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)*, pp. 2414–2423, 2016.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," 2015.

[4] C. Shuhuan, W. Y. Ke, X. Le, D. Xiaohua, and W. Kunzhe, "Overview of research on image style transfer based on deep learnings(in chinese)," *Application Research of Computers*, vol. 36, no. 8, 2018.

[5] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," 2017.

[6] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," 2016.

[7] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy," 2017.

[8] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016.

[9] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016.

[10] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," 2017.

[11] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysicss*, vol. 5, no. 4, 1943.

[12] B. W. White and F. Rosenblatt, "Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," *The American Journal of Psychology*, vol. 76, no. 4, pp. 705–707, 1963.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, 1986.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[16] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," vol. 2, pp. 1033 – 1038 vol.2, 02 1999.

[17] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," 2016.

[18] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin, "Image analogies," *Proceedings of ACM SIGGRAPH*, vol. 2001, 06 2001.

[19] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[20] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[21] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," 2016.

[22] D. Vaghela and K. Naina, "A review of image mosaicing techniques," *International Journal of Advance Research in Computer Science and Management Studies*, vol. volume 2, 05 2014.

[23] "About tensorflow." https://www.tensorflow.org/.

[24] M. C. Lisa Lei, "Style transfer for vr," *EE267 Virtual Reality, Spring 2018*, 2018.

[25] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–, 11 2004.

[26] T. Li, Z. Liu, and Z. Ma, "A new method of panoramic image mosaic based on sift," *Advanced Materials Research*, vol. 816-817, pp. 562–565, 09 2013.

[27] C. Yue, Z. Yan, and W. Shigang, "Fast image stitching method based on sift with adaptive local image feature," *Chinese Optics*, vol. 9, pp. 415–422, 08 2016.

[28] J. Liu and F. Bu, "Improved ransac features image-matching method based on surf," *The Journal of Engineering*, 03 2019.

[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, 1981.

[30] P. B, "Converting to/from cubemap." http://paulbourke.net/miscellaneous/cubemaps/, 2016.

[31] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos and spherical images," 2017.

[32] M. Perez-Ortiz and R. K. Mantiuk, "A practical guide and software for analysing pairwise comparison experiments," 2017.

[33] M. Pérez-Ortiz, A. Mikhailiuk, E. Zerman, V. Hulusic, G. Valenzise, and R. K. Mantiuk, "From pairwise comparisons and rating to a unified quality scale," *IEEE Transactions on Image Processing*, vol. 29, pp. 1139–1151, 2020.

[34] P. G. Engeldrum, "Psychometric scaling: a toolkit for imaging systems development," *Imcotek Press*, 2000.

[35] L. Engstrom, "Fast style transfer." https://github.com/lengstrom/fast-style-transfer/, 2016.

[36] J. Johnson, "Fast style transfer." ,https://github.com/jcjohnson/fast-neural-style, 2016.

[37] M. Ruder, "Fast style transfer." ,,https://github.com/manuelruder/fast-artistic-videos, 2018.

[38] K. Tateno, N. Navab, and F. Tombari, "Distortion-aware convolutional filters for dense prediction in panoramic images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Appendix

The codes and scripts implemented for my final year project can be found:

Style Transfer (Hyperlink)
https://drive.google.com/drive/folders/1J
ddFQAWxiMRUHuOrgj84ybv9Q6IkltQD?usp=sharing

The user study including subjective test and the data analysis can be found:

User Study (Hyperlink)
https://drive.google.com/drive/folders/1_O1
coAcDmqiFICaFXMFy6cCBScyO9wmY?usp=sharing

360 imaegs dataset can be found:

360 Images (Hyperlink)
https://www.dropbox.com/sh/4nfffwc8eb
u42p7/AADqxizsf0Nuf6Q7J14_jd7aa?dl=0

My latex dissetaion can be found:

Overleaf Thesis (Hyperlink)
https://www.overleaf.com/read/rnytnkynsgvm