BOURNEMOUTH UNIVERSITY

DOCTORAL THESIS

Neural Style Transfer for Images, Videos and Reliefs

Author:
Li WANG

Supervisor:
Dr. Xiaosong YANG
Prof. Jianjun ZHANG

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

National Centre for Computer Animation
Faculty of Media & Communication

November 12, 2020

Declaration of Authorship

I, Li WANG, declare that this thesis titled, "Neural Style Transfer for Images, Videos and Reliefs" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Abstract

For hundred years, artists engage into art creation to present their understanding of subjective and objective world, and their style representations can be inspired from other artworks and followed by other people. To grasp the spirit and styles from artworks, followers have to practice for years even for professional artists. In the past two decades, researchers in computer science have dedicated to propose automatic techniques to create paintings in different artistic styles, which gradually forms a research area **Artistic Style Transfer** (AST). The breakthrough on Convolutional Neural Network (CNN) recently drives the AST into a new era called **Neural Style Transfer** (NST).

Since born as a new technique, NST has been researched as a powerful tool to benefit other areas such as colour transfer, video temporal consistency and geometry detail transfer etc. However, applying NST directly into different research fields often causes unexpected artefacts. For example, the distortion artefacts in stylized results is inevitable especially when the content and style inputs are both photographic, the flickering artefacts existing in video style transfer methods, and the mismatching of geometric inputs in geometry detail transfer.

To address those challenges, this thesis aims to leverage NST to develop new techniques for the related research fields. A new photo style transfer method is proposed to prevent distortion artefacts and preserve style and photorealism simultaneously. To enhance the temporal consistency in consecutive frames, a stable video style transfer method is proposed to mitigate flickering artefacts by a set of masking techniques and multi-frame coherent losses. Furthermore, a semantic neural normal transfer network is proposed to match desired texture patterns from style reference input onto content inputs by an automatically attention-based mask technique.

Acknowledgements

The author would like to thank for the advice and instruction provided by his supervisors: Dr. Xiaosong Yang and Prof. Jianjun Zhang. These researches are kindly supported by Bournemouth University and China Scholarship Council. The author would also like to thank friends and people for their kindly support and help.

Contents

List of Figures

List of Tables

List of Abbreviations

AST	**A**rtistic **S**tyle **T**ransfer
CNN	**C**onvolutional **N**eural **N**etwork
MMD	**M**aximum **M**ean **D**iscrepancy
NST	**N**eural **S**tyle **T**ransfer
NAST	**N**eural **A**rtistic **S**tyle **T**ransfer
NGTS	**N**eural **G**eometry **T**exture **S**ynthesis
NPST	**N**eural **P**hoto **S**tyle **T**ransfer
NPR	**N**on-**P**hotorealistc **R**endering
NVST	**N**eural **V**ideo **S**tyle **T**ransfer
PSPN	**P**er-**S**tyle-**P**er-**N**etwork
MSPN	**M**ultiple-**S**tyle-**P**er-**N**etwork
ASPN	**A**rbitrary-**S**tyle-**P**er-**N**etwork
SFM	**S**tyle **F**usion **M**odel
RF	**R**ecursion **F**ilter
WCT	**W**hitening and **C**olouring **T**ransform

List of Publications

JOURNALS:

[1] **Li Wang**, Zhao Wang, Xiaosong Yang, Jianjun Zhang. "Photographic Style Transfer". *The Visual Computer*, 36.2 (2020), pp. 317-331.

[2] Meili Wang*, **Li Wang***, Tao Jiang, Nan Xiang, Mingqing Wei, Xiaosong Yang, Taku Komura, Jianjun Zhang. "Bas-relief Modelling from Enriched Detail and Geometry with Deep Normal Transfer". *Neurocomputing*, 2020.

[3] **Li Wang**, Xiaosong Yang, Weidong Min, Jianjun Zhang. "Fast Coherent Video Style Transfer via Flow Errors Reduction". Submitted to IEEE Access.

[4] Nan Xiang, Ruibin Wang, Tao Jiang, **Li Wang**, Yanran Li, Mingqiang Wei, Xiaosong Yang, Jianjun Zhang. "Sketch-based Modeling with a Differentiable Renderer". *CASA 2020* (CAVW Special Issue).

[5] Yanran Li, Linteng Qiu, **Li Wang**, Fangde Liu, Zhao Wang, Sebastian Iulian Poiana, Xiaosong Yang, Jianjun Zhang. "Densely Connected GCN Models for Motion Prediction". *CASA 2020* (CAVW Special Issue).

[6] Jaime Martin-Martin*, **Li Wang***, Adrian Escriche-Escuder; Irene De-Torres Garcia; Manuel González-Sánchez; Antonio Muro-Culebras; Cristina Roldán-Jiménez; María Ruíz-Muñoz; Fermin Mayoral-Cleries; Attila Biró; Wen Tang; Borjanka Nikolova; Alfredo Salvatore; Antonio I. Cuesta-Vargas. "OPEN SOURCE CODE for Energy Expenditure through Two Inertial Sensors". Submitted to PLoS One.

CONFERENCES:

[1] Nan Xiang, **Li Wang**, Tao Jiang, Yanran Li, Xiaosong Yang, Jianjun Zhang. "Single-image Mesh Reconstruction and Pose Estimation via Generative Normal Map". In:

Proceedings of the 32nd International Conference on Computer Animation and Social Agents, 2019, pp. 79-84.

[2] **Li Wang**, Nan Xiang, Xiaosong Yang, Jianjun Zhang. "Fast Photographic Style Transfer based on Convolutional Neural Networks". In: *Proceedings of Computer Graphics International*, 2018, pp. 67-76.

List of Symbols

$\widehat{x}$	stylized image
x_c	content image
x_s	style image
μ	mean
σ	standard deviation
ϕ	feature map
ψ	Gramian Matrix

To my family...

Chapter 1

Introduction

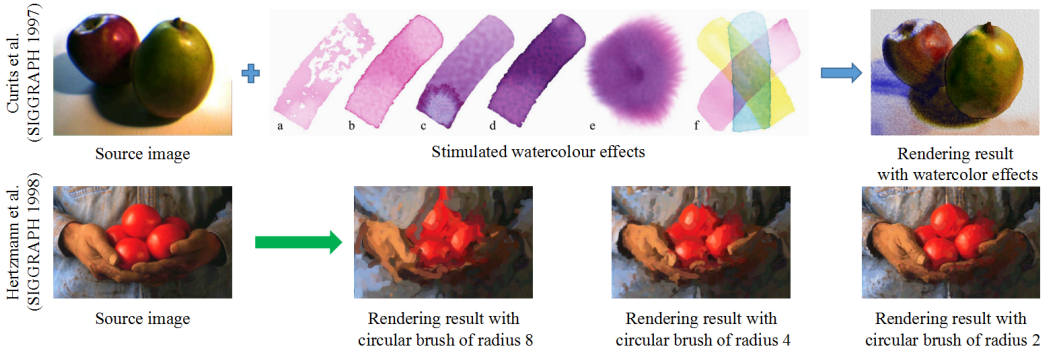


Figure 1.1: Examples of Non-photorealistic Rendering. The upper and lower examples are from Curtis et al. [23] and Hertzmann et al. [52], respectively.

1.1 Background

Painting has been an essential form of art to record the human perception of the world for thousands of years. There are dozens of representative painting styles around the world, such as "Impressionism", "Surrealism", and "Modernism" etc. And many other kinds of styles presented by special brush strokes or texture patterns in paintings. These stylistic presentations from either masterpieces or ordinary paintings have inspired generations to engage in the art creation. However, it takes really long time for followers to learn and grasp the spirit of one specific style, not even mention universal styles before the digital era. Therefore, researchers in computer science constantly propose algorithms or techniques to automate the learning process, help people easily create paintings in any desired styles. Hence, a research field **Artistic Style Transfer** comes into being.

Starting from the 1990s, algorithms like painterly renderings in *Non-Photorealistic Rendering* (NPR) [40, 123, 138] firstly succeed in transforming images with a few specific art styles [23, 52, 123] like watercolor and brush strokes. For example, Curtis [23] proposed a technique to repaint the source image with simulated watercolor style (e.g., the upper row in

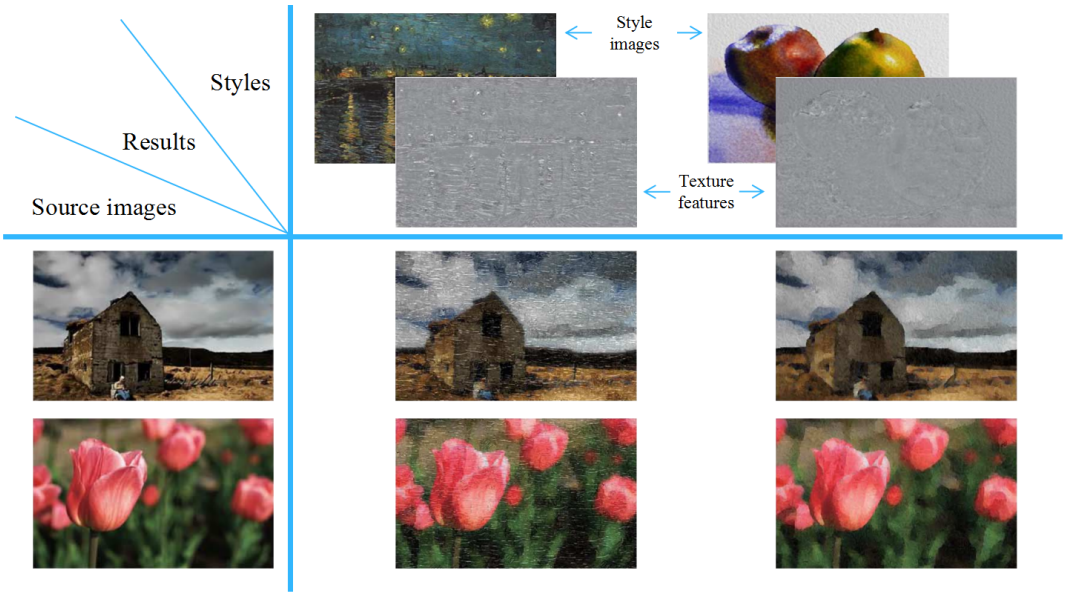


Figure 1.2: Examples of style transfer based on texture features extracted from gradient domain. All examples are from Zhang et al. [175]

Fig. 1.1). Hertzmann [52] proposed a painterly rendering method by simulating multi-size curved brush strokes to create "Impressionist" stylized images (e.g., the lower row in Fig. 1.1). These techniques are limited by certain simulated artistic styles, and lack the capability of transferring images into any other artistic styles. For computer vision researchers, the transformation process is studied as a generalization of texture synthesis, which seeks to extract texture representations from artistic paintings and transfer them to target images. For example, Zhang et al. [175] proposed an example-based image stylization method which extracts paint component in YIQ colour space to be texture representation (e.g., Fig. 1.2). Before the neural networks are applied, the above mentioned methods can only extract features in colour space to form certain style representations, which leads to unsatisfactory results and limitations on artistic style numbers that one method can learn.

Recently, *Convolutional Neural Network* (CNN) shows its power as a rich feature provider in visual perception, which boosts several computer vision research areas such as face and object recognition [24]. Inspired by the visual perception success [11, 18, 29, 81, 107] using CNN, Gatys et al. [36] initialize an artificial algorithm for the creation of artistic images, which separates and recombines content and style of images. Feeding images into the pre-trained CNN, they present the semantic content of a photo as neural responses of CNN, and refer to the artistic style of an artwork as its texture representations via the spatial summary statistic of neural responses. Their experiments visualize the capability of separating and re-organizing of content and style representations, which indicates the possibility of re-drawing

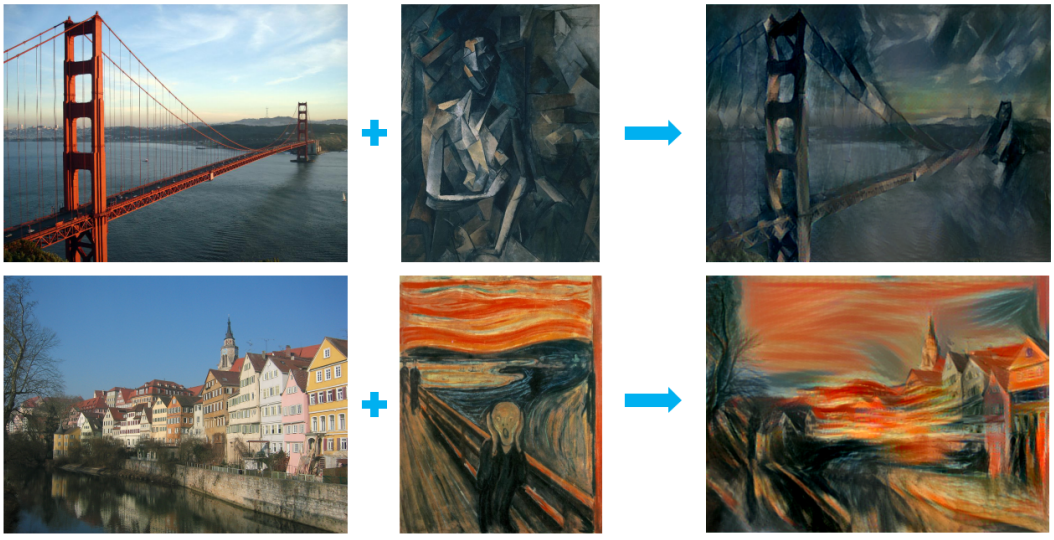


Figure 1.3: Examples of Neural Artistic Style Transfer in [37]. The artworks are "Seated Nude"–Pablo Picasso (top middle) and "The Scream"–Edvard Munch (bottom middle)

a new image preserving both spatial semantic content of a photo and artistic style of an artwork. Based on this finding, Gatys et al. [37] firstly propose a technique that is capable of automatically recreating images in as many artistic styles as possible, which creates a path to universal style transfer. Their proposed approach formulates style transformation as an optimization problem, which generates from white noise to a new image containing similar neural activations as the content image and similar stylistic correlations as the style image. Without explicit constraints on style types, their artificial algorithm produces visual appealing results in universal artistic styles. Two examples of artistic stylized result are shown in Fig. 1.3. The image style transformation method proposed by Gatys et al. [37] opens up a new era called *Neural Style Transfer* (NST) in computer vision community.

This *NST* field have been given wide attention in both academic and industrial areas. In academia, a surge of recent works [17, 19, 37–39, 59, 66–68, 77, 79, 91, 94, 98, 99, 115, 130, 134, 144, 148, 149, 155, 168] address the problem of style transfer using deep neural networks. These methods are proposed to either **improve** the quality [91, 93, 144] of the Gatys et al's algorithm [36] or **extend** its applications like style transfer for photos [97, 108] and videos [5, 16, 124, 126]. In this thesis, the **prior proposed improvement** methods are called *Neural Artistic Style Transfer* (NAST), and the **prior proposed extension** methods for photos and videos are called *Neural Photo/Video Style Transfer*. In industrial applications, many follow-up algorithms [17, 59, 68] have been adopted successfully in social Mobile Apps such as *Prisma*, *Ostagram*, *DeepArts* etc. In addition, *NST* has also been proved to be useful in other research areas: image super-resolution [68, 154], image-to-image translation [62, 87,

185], image synthesis [70, 71], normalization in neural networks [145], domain adaptation [13, 55], image inpainting [105], neural differentiable renderer [72], 3D motion transfer [56] etc. This thesis further extends the application of *NST* onto digital bas-relief generation by synthesizing novel textures on geometry surfaces. Therefore, I call this extension branch as *Neural Geometry Texture Synthesis* (NGTS).

In the academic extension branch of *NST*, for example *Neural Photo Style Transfer*, researchers pursue to obtain faithful and photorealistic stylized results when both the content and style images are photographic. In the extension branch like *Neural Video Style Transfer*, research studies focus on developing algorithms which not only process videos faster, but also produce coherent stylized results. As a further extension, *Neural Geometry Texture Synthesis* aims to synthesize new textures on 3D surfaces, which enriches geometry details for art design. *Neural Artistic Style Transfer* methods are capable of creating satisfactory artistic images, however, in extension research areas they often fail for certain purposes and need to be improved. For example, in NPST, NAST methods fail to preserve the photorealism because of the content-mismatching and distortions which produce the painting look of results. In NVST, NAST methods process video frame-by-frame and cause flickering artefacts. As for NGTS, there is few researches performing texture synthesis based on NST onto geometry surfaces. To address these challenges, this research develop new methods in the extension branches of *NST*: *Neural Photo Style Transfer* (NPST), *Neural Video Style Transfer* (NVST) and *Neural Geometry Texture Synthesis* (NGTS).

1.2 Motivation

As a popular social activity, photo sharing and exchanging in social media like Instagram, Twitter and Facebook have great impact and strong appeal in people's daily life. To enrich the photo sources for social media, photo editing tools especially photo style transfer makes it easy for people to create beautiful pictures with desired colour styles. Though softwares like Photoshop are powerful tools to do photo style transfer work, it still needs professional skills to create natural and wonderful result. To make it easier, researchers have proposed many methods from the perspectives of colour [47, 116, 121, 162] and texture [175]. However, these techniques are limited by either failing to transfer faithful colour between sophisticated images or highly depend on the similar scenes with different colours, views and illumination.

Gatys et al.'s method produces impressive artistic stylized results, but it fails to transfer the photorealistic style when both the content and style images are photographic due to

content-mismatching and distortions. To address these problems, Luan et al. [108] propose a two-stage *Deep-Photo-Style-Transfer* (DPST) method, which utilizes semantic segmentation to avoid the content-mismatching problem in its first stage, and add a photorealism regularization term based on locally affine colour transformations to prevent distortions in its second stage. However, in the first stage, semantic segmentation masks for both content and style images consume lots of computation time even for low resolution images. In the second stage, the content spatial structures are preserved in many situations, but details especially the exact edges are erased when semantic segmentation is inaccurate or contains overlapping areas. In addition, the added photorealism regularization term using Matting Lapacian matrix also causes posterization artefacts.

For *Neural Photo Style Transfer*, hence, the challenges are:

- **CHAL 1** The photorealism loss of stylized results.
- **CHAL 2** Slow execution time for effective neural photo style transfer.

Artistic video stylization by manually re-drawing requires a large amount of time and human labour. Benefit from the speedup of *Neural Artistic Style Transfer* methods [19, 68, 144], now it is possible that the whole video stylization processing can be done automatically. So far, the *Neural Artistic Style Transfer* methods can be divided into two groups: methods based on **gradient descent optimization networks** (e.g., [37, 91]) and methods based on **feed-forward networks** (e.g., [68, 146]). However, naively extending these techniques into video stylization may produce new issues. For example, processing a video sequence via per-frame stylization often leads to flickering and incoherence between adjacent frames. The cause of this problem is the unstable solution for style transfer task. For methods based on optimization process, the random initialization and non-convex nature leads to the instability and local minima of style transfer. For methods based on feed-forward networks, slight changes of illumination, view and movement in adjacent frames may cause large variations in stylized results. Hence, temporal consistency of consecutive frames should be considered for neural video style transfer.

Anderson et al. [5] and Ruder et al. [124] introduce a temporal consistency term into Gatys et al.'s optimization-based network, and apply optical flow estimation to constrain the initial image and consistency loss function. Their methods give stable stylized videos but their methods produce ghost arefacts and slows in run time (several minutes for per frame). Searching for a fast and stable solution for video stylization, Huang et al. [57] and Gupta et al. [46] train their feed-forward networks with temporal consistency loss function and give stable stylized videos without optical flow at test time. Chen et al. [16] present a fusion

stylization network, which also applies optical flow into temporal coherence loss. Ruder et al. [125] formulate video stylization as a learning problem. Their proposed network-based approach stylizes arbitrary-length videos in a stable and coherent way, which clearly outperforms independent per-frame processing baselines (e.g., [68]). Lai et al. [85] proposes a blind video processing technique which is capable of preserving temporal consistency. Their method processes the per-frame stylized results and achieves real-time execution time as no optical flow needed in testing time. However, the blurriness artefacts of Lai et al.'s method can accumulate along with processing time. Besides, the aforementioned stable video stylization methods above still suffer from some incoherency where **motions and occlusions are too large for the flow to track object correctly**. Li et al. [95] proposes a new linear transformation matrix to minimize the difference between covariance of content features and style features, which keeps temporal consistency by propagating the computed matrix from the beginning of frame to the rest of them. Their method sacrifices diverse stylistic outputs for temporal consistency. In addition, most of these fast methods need to be trained for only one style, which means they present a per-style-per-network solution for *Neural Video Style Transfer* task. And usually, the training time for each style needs around several hours, which makes it less practical in real applications.

For *Neural Video Style Transfer*, hence, the challenges are:

- **CHAL 3** Unstable artefacts produced by fast methods based on feed-forward networks.
- **CHAL 4** Slow performance of optimization-based networks.
- **CHAL 5** Less practical solution as per-style-per-network needs much more time for training.

Bas-relief, a special type of sculpture that figures are slightly emerged from a background, is a bridge between 2D drawings and 3D sculptures. Considerable attentions have been received in recent years since it can be viewed from many different angles without causing distortion of the figures.

Realistic bas-reliefs should present both detailed appearances and stereoscopic perception. For detail transfer, existing works ([63, 64]) tend to rely on straightforward image processing techniques like cut-and-paste and decompose-and-compose, which often lead to imperfect composition results. For example, the cut-and-paste operation cannot preserve the normals of the target. The decompose-and-compose [131, 132] operation requires larger detail patches than target patches while it cannot manipulate target surfaces or cause scaling issue for textures. In addition, these methods are only able to copy details from other

sources without creation ability. Thus they are less practical for applications as their limitations mentioned above. For geometry preservation, existing methods (e.g., [156]) decompose the normal field into a base layer and a detail layer by directly subtracting the base normal value from the original normal, which causes unexpected triangle distortions in the resultant bas-reliefs.

Traditional texture synthesis on geometry surfaces usually requires low-distortion mapping [90] between the source surface and the target surface via parameterization [129, 137]. Texture synthesis based on parameterization inevitably produces visible seams between boundaries as mapping topological discs with boundaries onto a manifold without boundaries forms boundary discontinuities. A few following works [2, 3, 15, 76] propose frameworks to deal with symmetrical textures and obtain seamless mapping between closed surfaces with compatible genus.

Generation and synthesis of irregular structures, particularly meshes, is still a open problem in computer graphics. To synthesize 3D textures of surfaces, Lefebvre et al. [88] encode colour neighbourhoods of a 2D texture examplar into appearance-space vector, which is able to synthesize 3D textures on surface by using radiance transfer texture. More recently, Liu et al. [106] leverage 2D image processing filters via multi-view rendering onto 3D surface by a gradient-based optimization. It constructs a differentiable renderer which can back propagate changes in the image domain to the 3D mesh vertex positions. In this way, their method is capable of synthesizing 3D textures on surfaces through the gradients computed from Neural Style Transfer methods. However, their method takes a few hours to complete the operations. Hertz et al. [51] propose to learn deep features of geometric texture statistics from local neighbourhoods on a single reference 3D model, which is used to generate offsets of mesh vertices to form desired textures on the target mesh. It completes the task without parameterization and facilitates texture transfer between shapes of different genus. However, none of aforementioned methods is capable of transferring texture features like non-stationary textures (large-scale structures and spatially variant textures) as these approaches are based on local neighbourhoods of either a reference model or examplar.

For *Neural Geometry Texture Synthesis*, hence, the challenges are:

- **CHAL 6** Lack of geometry detail creation for digital bas-relief design.
- **CHAL 7** Less practical detail transfer for bas-relief modelling.
- **CHAL 8** Lack of efficient technique to synthesize geometric textures with non-stationary features.

1.3 Research Questions

For *Neural Photo Style Transfer*, this research attempts to answer the following questions:

- **Q1** Without semantic segmentation, how to solve the content-mismatching problem?
- **Q2** How to improve the photorealism of stylized results by preventing distortions?
- **Q3** How to speedup of the proposed method for fast neural photo style transfer?

For *Neural Video Style Transfer*, this research attempts to answer the following questions:

- **Q4** How to improve the stability of the stylized videos produced by fast methods based on feed-forward networks?
- **Q5** How to propose a practical optimization-based method with arbitrary-style-per-network solution?

For *Neural Geometry Texture Synthesis*, this research attempts to answer the following questions:

- **Q6** How to enrich detail appearances for digital bas-relief design?
- **Q7** How to achieve efficient geometry detail transfer via NST?

1.4 Aims and Objectives

This thesis mainly aims at applying Neural Style Transfer scheme for images, videos and geometry data. To achieve this aim, the specific objectives are as followed:

- **OBJ1** Review current state-of-the-art methods on Neural Style Transfer. Compare and analyze the pros and cons of existing approaches and identify the factors that influence the performance on different extension branches, e.g., photorealism of NPST, stability of NVST and capability of NGTS.
- **OBJ2** Design the loss functions to solve the content-mismatching problem.
- **OBJ3** Design a feasible operator or post-processing step to prevent distortions.
- **OBJ4** Develop an efficient technique for faster photo style transfer.
- **OBJ5** Develop a stable technique for video style transfer even in large motions and strong occlusion cases.

- **OBJ6** Develop techniques to speed up optimization-based video style transfer method for an arbitrary-style-per-network fashion.
- **OBJ7** Develop a texture synthesis technique to enrich digital bas-relief generation.

1.5 Contributions

For Neural Photo Style Transfer, this research dedicates to improve the photorealism of stylized images. It focuses on solving content-mismatching and photorealism problems, and proposes a solution for speeding up the proposed method. Without semantic segmentation, this research introduces a similarity loss function to solve the content-mismatching problem, and a post-processing technique to reduce potential distortion and noise arefacts. The similarity loss function reconstructs finer details of content image and constrains the content match between reference style and content images. The post-processing refinement technique extracts the colour without the details from stylized result and combines it with the details of content input. Distortion and noise arefacts will be eliminated after the refinement step. Compared to the second stage of Luan et al. [108], the post-processing step is much faster, and it avoids unexpected posterization arefacts. Integrating the above mentioned methods into prior Neural Artistic Style Transfer methods, the proposed methods improve the photorealism of stylized results. And the proposed NPST methods also achieve almost real-time performance by integrating similarity loss function and post-processing step into NAST methods based on feed-forward networks.

Hence, the contributions for NPST are:

- **CTRB1** Compared to the semantic segmentation, the proposed similarity loss function is capable of solving content-mismatching problem without extra computation time.
- **CTRB2** Distortion and noise arefacts are eliminated effectively by the post-processing refinement step, and it successfully avoids the posterization effect of prior state-of-the-art methods.
- **CTRB3** Compared to previous NPST methods, the proposed method is capable of transforming representative prior Neural Artistic Style Transfer methods into Neural Photo Style Transfer methods.
- **CTRB4** The proposed NPST methods achieve almost real-time performance by integrating the proposed similarity loss and post-processing step into prior NAST methods based on feed-forward networks.

For Neural Video Style Transfer, this work makes attempts to reduce flow errors via a set of mask techniques and a new initialization for optimization-based video style transfer. These mask techniques reduce significantly both the flow untraceable errors and flow traceable errors (ghosting artefacts). The new initialization contributes to produce stable video stylization results even in large motions and strong occlusion cases, and it also speeds up the optimization process from minutes per frame to around seconds per frame. Multi-frame Coherent Losses are proposed to preserve the temporal consistency between consecutive frames, and Sharpness Losses proposed effectively mitigate the image blurriness artefacts during the entire video stabilization process.

Hence ,the contributions for NVST are:

- **CTRB5** Compared to feed-forward based NVST methods, the proposed method produces stable video style transfer results even for large motions and strong occlusion cases.
- **CTRB6** A speed-up optimization-based NVST is proposed for arbitrary styles in one network.

For Neural Geometry Texture Synthesis, this work proposes a semantic neural normal transfer network, which is capable of learning the texture patterns from the source normal images and transferring them onto the target normal images in arbitrary shapes and multiple scales. To preserve geometric properties, this research presents a normal decomposition scheme which contributes to the generation of bas-relief results free from artefacts. The contributions for NGTS are:

- **CTRB7** Prove that NST can be applied to geometry texture synthesis on 2.5D surfaces.
- **CTRB8** Normal images can be the intermediate representation of geometry surface for NST to synthesize new texture patterns on 2.5D surfaces.

1.6 Thesis Outline

This thesis mainly focuses of the research on NPST, NVST and NGTS .The proposed methods have been published on peer-reviewed international journals and conferences. For NGTS, this thesis will further discuss the texture synthesis on geometry mesh in Chapter 7 Future Work. This thesis is organized into 7 chapters with the research challenges, questions, objectives and contributions which are shown in Fig. 1.4.

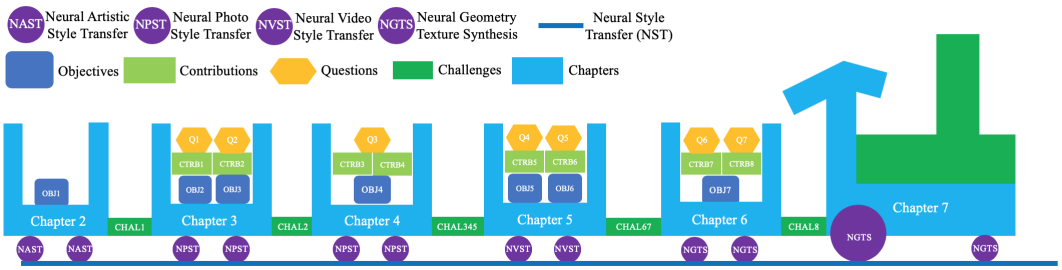


Figure 1.4: Thesis outline.

In Chapter 2, a literature review on Artistic Style Transfer and Neural Artistic Style Transfer, Colour Style Transfer and Neural Photo Style Transfer methods and their related computer vision researches have been made. It gives the overview of current state-of-the-art NAST and NPST methods on their advantages and drawbacks, and summarizes the key factors that influences the photorealism of stylized results [**OBJ1**]. In addition, the related computer vision researches are also detailed reviewed which include image reconstruction via CNN and edge-preserving image filter.

In Chapter 3, inspired by the ideas of other computer vision tasks, a new pipeline for solving content-mismatching [**OBJ2**] and distortion problems [**OBJ3**] have been proposed. It also gives the detailed system design and implementations. [Publications [1] of Journals].

In Chapter 4, as an important media tool in social life, a fast solution for Neural Photo Style Transfer has been proposed [**OBJ4**]. In addition, it also shows that the prior Neural Artistic Style Transfer methods are successfully transformed to Neural Photo Style Transfer methods. [Publications [2] of Conferences]

In Chapter 5, in NVST field, a novel framework is proposed for video style transfer, which is capable of generating stable video stylization results in large motions and strong occlusion cases [**OBJ5**], and reduces execution time of optimization-based NVST methods from minutes per frame to seconds per frames in an arbitrary-style-per-network fashion [**OBJ6**]. [Publications [3] of Journals]

In Chapter 6, a digital bas-relief modelling method is proposed which applies neural style transfer on geometry texture via a semantic normal transfer network. This network synthesizes new textures on normal images and reconstructs them on 2.5D surfaces in order to enrich geometry details for bas-relief design [**OBJ7**]. [Publications [2] of Journals]

In Chapter 7, a further step has been made to synthesize 3D textures on geometry mesh. And a potential solution with application of NST is discussed as future work.

Chapter 2

Related Works

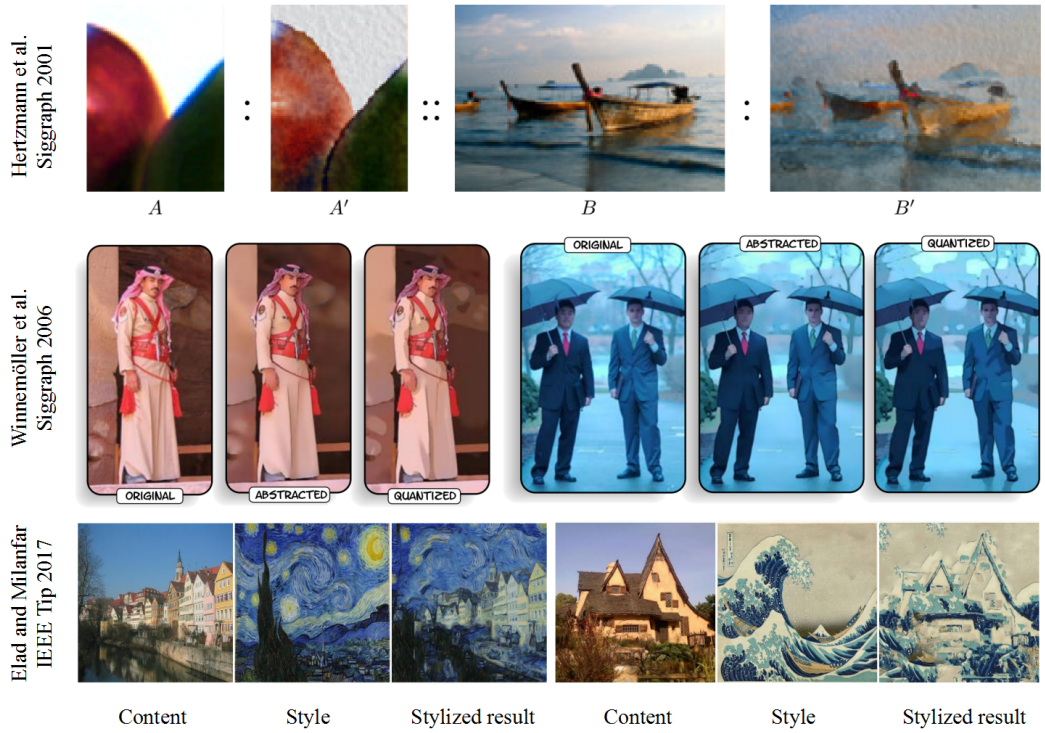


Figure 2.1: Artistic Style Transfer Before Neural Style Transfer Era. From top to bottom, the examples are from Hertzmann et al. [52], Winnemöller et al. [161], and Elad and Milanfa [30].

2.1 Artistic Style Transfer Before Neural Style Transfer Era

The desire for easy and practical media editing tools has motivated the development of artistic style transfer for over two decades. Before the appearance of Neural Style Transfer, the related works are called Non-Photorealistic Rendering (NPR) and Texture Synthesis in computer graphics and computer vision community, respectively. In this section, a brief review of these style transfer algorithms will be made.

2.1.1 Artistic Style Transfer using Computer-generated Styles

Previous Non-Photorealistic Rendering methods [23, 52] model several particular artistic strokes (e.g., brush strokes, tiles and stipples) to render an image with hand-painted appearance from a photograph. The process is generally starting from a given source image, then matches brush strokes to the colours in the source image, and finally composites the strokes to the photo and produces a non-photorealistic image. The final image contains the rough spatial contents of the source image but in an particular artistic style. This process is accomplished by iterative placement of strokes onto the source image (refer to Fig. 1.1). Such methods usually follow per-method-per-style pattern and need to pre-simulate a particular artistic style, for example, oil paintings, watercolour and sketches etc. The pre-simulated requirement of these methods limits the extension of style transfer.

2.1.2 Artistic Style Transfer using Image Analogy

Image analogy [53] is a technique proposed to learn an "analogous" filter from a pair of images which are one photograph and one image purported to be a stylized version of the other. The learned filter is then applied to a given target photo and create a new image with the similar style of the stylized image. Such technique computes a new "analogous" image B' that relates to B, in the upper row of Figure 2.1, which should be in "the same way" as A' relates to A. A, A' and B are input images and B' is the output image. The stylization process is under a supervised manner, and the training images are A and A'. In general, image analogy is capable of stylizing arbitrary artistic styles, and it also can be extended for portrait painting rendering (e.g., [183]). However, the training pairs like A and A' are usually unavailable in practice.

2.1.3 Artistic Style Transfer using Image Filtering

Based on the spatial content of the source photo, creating an artistic image is a process of abstracting and modifying the contrast of visually important features (e.g., luminance and colour). Based on this, Winnemöller et al. [161] initialize an automatic style transfer technique which adopts and combines image processing filters, such as bilateral filter [143] and difference of Gaussian edges [41], to produce cartoon-like effects (c.f. the middle row in Fig. 2.1). This algorithm is highly parallel and can achieve real-time performance when implemented using GPU. Compared to image analogy, style transfer methods based on image filtering are more efficient and practical in the real world. However, the limitation is also obvious that they lack the diversity of styles.

2.1.4 Artistic Style Transfer using Texture Synthesis

In computer vision community, style transfer is researched as a generalization problem of texture synthesis, which grows the similar visual texture patterns learned from a source image onto a target image. The texture pattern is learned from the texture instance x_s, for example, given the distribution $d(x_s)$, texture synthesis seeks to generate a new image $\widehat{x}$ which should have the similar texture distribution as $d(x_s)$:

$$\widehat{x} \sim d(\widehat{x}|x_s) \tag{2.1}$$

The style transfer process is very similar as algorithms transforming certain texture from a given source image to a target image when the style of the source image is considered to be a kind of texture. From this perspective, style transfer is regarded as a process of texture transfer. The generalized new image $\widehat{x}$ is constrained by both the content of one image x_c and the distribution of another image x_s, which means $\widehat{x}$ should meet the distributions of x_c and x_s:

$$\widehat{x} \sim d(\widehat{x}|x_c, x_s) \tag{2.2}$$

Efros and Leung [28] for the first time propose a texture synthesis method in this way, and many other works [25, 27] follow this route. For example, Efros and Freeman propose a simple image-based texture transfer method [27] which synthesizes a new image based on patch matching and quilting. More recently, Frigo et al. [31] propose an unsupervised style transfer method based on local texture transfer and global colour transfer. This algorithm first decomposes an image into an adaptive partition of the source image which divides image into suitable regions, and searches optimal maps from reference style image which are matched for the adaptive partition. Then it exploits bilinear blending and colour transfer to produce the final stylized result. Elad and Milanfar [30] propose a style transfer based on traditional texture synthesis theory which includes multiple steps to stylize images such as patch matching, patch aggregation, content fusion and colour transfer, etc. Such style transfer methods using texture synthesis can be performed in an unsupervised manner, however, only low-level image features (e.g., pixel intensity and gradient orientation in RGB domain) are considered during the process which lead to unsatisfactory stylized results (c.f. the bottom row of Fig. 2.1) to Neural Style Transfer.

This section briefly review the style transfer methods proposed in both computer graphics and vision community, respectively. For more detailed information, please refer to [83, 123, 128].

2.2 Colour Style Transfer before Neural Style Transfer Era

As an essential element of visual world, colour is one of the most essential features that are widely used in different visualization areas such as art, photography to relay information. Recolouring images is able to change the illumination conditions of a scene, or transfer totally different style effects between images. Therefore, Colour Style Transfer methods are proposed to recolour a given image by mapping the colour distribution from a reference style image. Due to the wide applications in social media, the Colour Style Transfer field receives constantly attention from both academic and industry. Based on region correspondence, the Colour Style Transfer methods can be divided into two different groups: Global Colour Style Transfer and Local Colour Style Transfer.

2.2.1 Global Colour Style Transfer Methods

In early years, colour style transfer methods [47, 117, 121] tend to explore a global colour transformation between images. A spatial-invariant transfer technique is applied to handle simple cases, such as global colour move (e.g., sepia) and tone curves (e.g., low or high contrast). Reinhard et al. [121] propose the earliest colour style transfer method to achieve the characteristic transfer between the input image and reference image. Since the RGB colour space is highly correlated, the proposed method chooses a colour space called $l\alpha\beta$ which is a less correlated space. Then they convert image values from RGB space to $l\alpha\beta$ space, and match the mean and standard deviation of one image to those of another image in the $l\alpha\beta$ space. Their method successfully alters the colour distributions from one image to the desired reference style image. However, it is limited to linear transformations. To handle non-linear colour mapping, Pitié et al. [117] propose a continuous transformation that is capable of mapping a N-dimensional distribution to another. Their method treats colour transfer as the transfer of the whole Probability Density Function (PDF) of the samples in both input images. And they iteratively use one-dimensional PDF transfer to achieve exact transfer of a PDF, which is proved to be effective and less computational for non-linear colour transfer.

The aforementioned methods are based on statistic properties transfer, another branch of colour style transfer is recently using the correlations of dense correspondence. One of the representative method is proposed in [47]. Their method is a global non-linear parameteric colour transformation, which uses Generalized PatchMatch algorithm [8] to extract reliable dense correspondence patches from two input images. The nearest-neighbour field computations are interleaved with aggregating consistent matching regions using locally adaptive

constraints. Their results are compelling but the method is limited to pairs of images depicting similar scenes under different illumination and views. In addition, there are three drawbacks: (1) their method can not find reliable correspondences in very large smooth regions, (2) their approach can not match accurate object regions when an object appears over a different background in the two input images, (3) their technique can not handle two or more different colour models.

2.2.2 Local Colour Style Transfer Methods

Local Colour Style Transfer Methods are capable of being more expressive and handling complex applications such as season and weather change[33, 84], and time-of-day hallucination [33, 135]. Taking season change as an example, these methods change the leaves of trees from green colour (spring) to red color (autumn) or snowwhite (winter).

Gardner et al. [33] investigate how to make meaningful distribution changes between source and target domain. Inspired by [9], they exploit a deep convolutional network to simplify the manifold of natural images to a linear feature space. Their method utilizes kernel Maximum Mean Discrepancy (MMD) [43] in convolutional feature space to match the distributions of source and target images. The proposed data-driven method is applied to change the appearance of faces, city skylines or nature scenes. However, the requirement of memory storage is large as holding thousands of 964x540 image feature extracted from the pre-trained VGG-19 networks requires over 128 GB of main memory.

Such spatial colour mapping based methods highly rely sparse correspondence guidance from either user input [4, 158] or image segmentation [6, 141, 162, 169]. Results of these algorithms are not precise enough because some pixels can be transformed into inaccurate colours.

For matching sophisticated colour appearances between image pairs, Shih et al. [135] propose a local colour style transfer technique based on additional pair of well-aligned images. For example, given a source image and a target image, their method first finds an additional reference image which is the same scene of the target image but with a well-aligned colour distribution to the source image. Then their method utilizes the reference image as the bridge for the transfer process. To be precise, they estimate a locally linear colour model and then apply it to perform linear colour transfer.

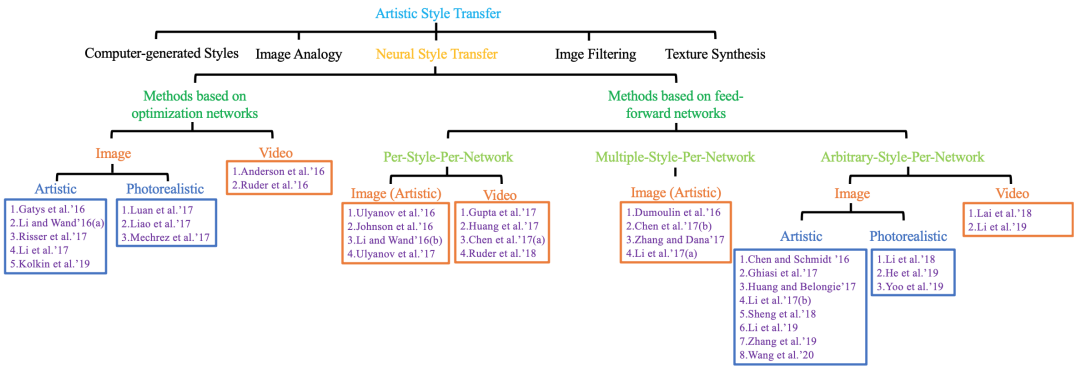


Figure 2.2: A taxonomy of NST. Columns from left to right and rows from top to bottom: Gatys et al.'16[37], Li and Wand'16 [91], Risser et al.'17[122], Li et al.'17[93], Kolkin et al.'19[78], Luan et al.'17[108], Liao et al.[103], Mechrez et al.'17[111], Anderson et al.'16[5], Ruder et al.'16[124], Ulyanov et al.'16[146], Johnson et al.'16[68], Li and Wand'16(b)[92], Ulyanov et al.'17[144], Gupta et al.'17[46], Huang et al.'17[57], Chen et al.'17(a)[16], Ruder et al.'18[125], Dumoulin et al.'16[26], Chen et al.'17(b)[17], Zhang and Dana'17[174], Li et al.'17(a)[98], Chen and Schmidt'16[19], Ghiasi et al.'17[39], Huang and Belongie'17[59], Li et al.'17(b)[99], Sheng et al.'18[134], Li et al.'19[95], Zhang et al.'19[181], Wang et al.'20[155], Li et al.'18[97], He et al.'19[49], Yoo et al.'19[170], Lai et al.'18[85], Li et al.'19[95]

2.3 Neural Artistic Style Transfer in Neural Style Transfer

For the style transfer task, there are generally three issues need to be considered: 1. how to build a model to present style information of a given reference style image; 2. how to build a model to present content information of a content image; 3. how to reconstruct a new image which should have spatial content information of the content image while preserving texture of the reference style image. As mentioned in Section 2.1.4, the style of an image can be regarded as a form of texture, a simple and effective way is to model texture by capturing image statistic from a sample texture and exploiting the summary statistic property. Julesz [69] first describes this idea via modeling textures as pixel-based N-th order statistic. After that, Heeger and Bergen [50] utilize filter responses to analyze textures rather than pixel-based measurement. Later work [119] further models textures based on multi-scaled orientated filter responses and uses gradient descent to improve the synthesis results.

2.3.1 Basis of Neural Artistic Style Transfer

Different from previous methods [50, 119], Gatys et al. [35] for the first time propose to measure summary statistics in the domain of CNN. They discover that the summary statistic, the correlations of multi-level neural responses of a texture image, can be used to model textures from low-level features to high-level features. And it succeeds to solve the first issue. To address the second issue, Gatys et al. find that hierarchy image information encoded in CNN layers can also be presented by the neural responses in different layers (before the softmax is

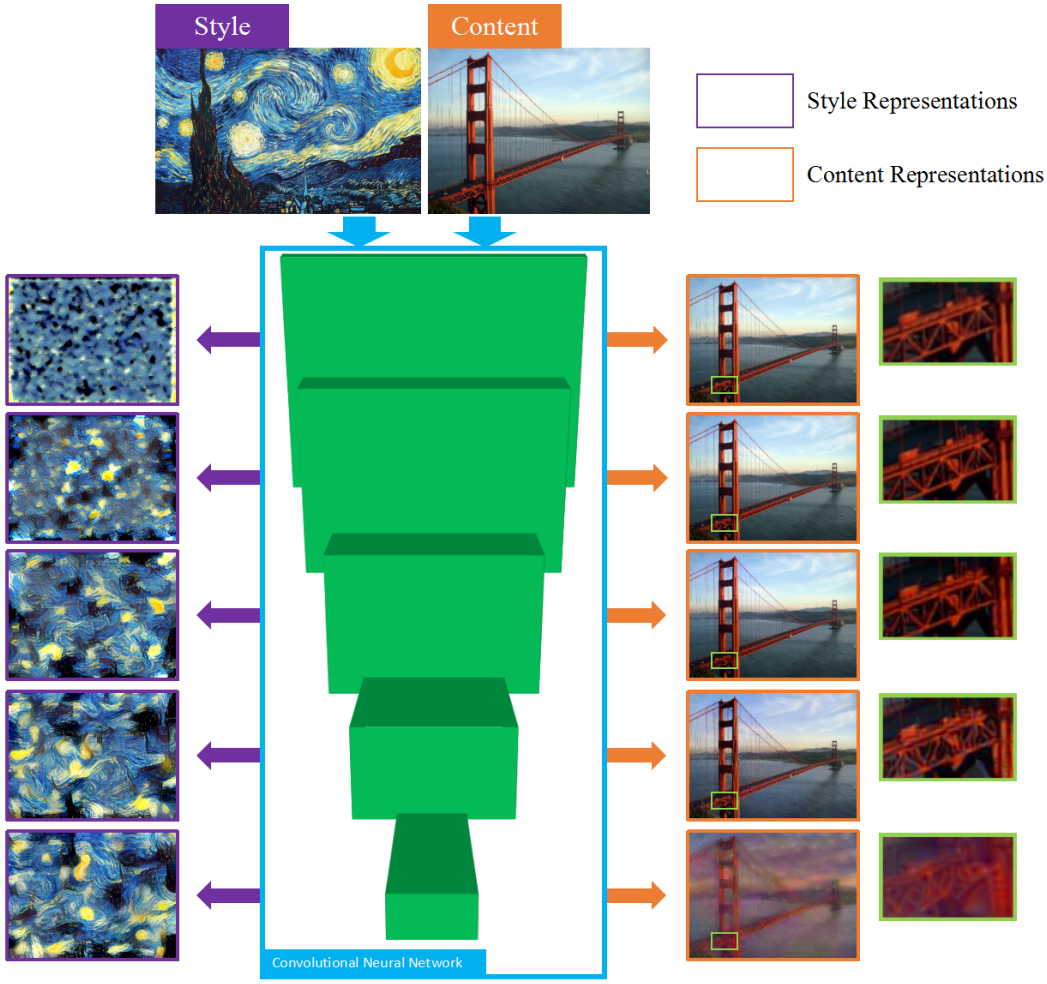


Figure 2.3: The representations of the content and style images in CNN.

applied) of a pre-trained image classification network (e.g., AlexNet [80]) [110]. As an example, Fig. 2.3 shows the representations of content and style images extracted from different level of a pre-trained VGG network. To address the third issue, there are two most common ways to reconstruct the new image, which exactly categorise Neural Style Transfer methods into two groups: gradient descent optimization networks and feed-forward networks. For clarity of these methods and their extensive branches, a taxonomy of NST algorithms is described in Fig. 2.2.

2.3.2 Methods based on gradient descent optimization networks

The first category of Neural Style Transfer methods is based on gradient descent optimization process which iteratively synthesizes a new image matching the content representation of one photograph and the style representation of one artistic image. Gatys et al. [37] observe that the semantic image content and some artistic appearance can be extracted from arbitrary content photograph and an artwork, respectively. And these representations are encoded in

the neural responses of a deep neural network. Based on this observation, they propose an artificial algorithm that generates a new stylized image, starting from white noise, by penalizing the difference of hierarchy content representations between the content image and stylized image, and hierarchy style representations between the reference style image and stylized image.

For the given content photograph x_c and artistic image x_s, the algorithm in [37] searches a new stylized image $\widehat{x}$ by minimizing the following loss function:

$$\mathcal{L}_{total}(\widehat{x}, x_c, x_s) = \alpha \mathcal{L}_{content}(\widehat{x}, x_c) + \beta \mathcal{L}_{style}(\widehat{x}, x_s) \tag{2.3}$$

where the content loss $\mathcal{L}_{content}$ penalizes the difference of content representations between the content image x_c and the stylized image $\widehat{x}$, and the style loss $\mathcal{L}_{style}$ penalizes the difference of style representations between the artistic image x_s and the stylized image $\widehat{x}$. α and β are the weights to balance the content component and style component of the stylized result $\widehat{x}$.

Let the matrix $\phi_j \in \mathbb{R}^{N_j \times M_j}$ denote the vectorized feature maps representing the neural responses in a layer j, then the content loss $\mathcal{L}_{content}$ is defined as the squared Euclidean distance between the two feature representations $\phi_j(x_c)$ and $\phi_j(\widehat{x})$:

$$\mathcal{L}_{content}(\widehat{x}, x_c) = \sum_{j \in J_c} \|\phi_j(x_c) - \phi_j(\widehat{x})\|^2 \tag{2.4}$$

where J_c denotes the set of layers in a pre-trained VGG network in which compute the content loss. The style loss $\mathcal{L}_{style}$ is denoted as the squared Euclidean distance between the two style representations of x_s and $\widehat{x}$:

$$\mathcal{L}_{style}(\widehat{x}, x_s) = \sum_{j \in J_s} \|\psi(\phi_j(x_s)) - \psi(\phi_j(\widehat{x}))\|^2 \tag{2.5}$$

where J_s denotes the set of layers in the pre-trained VGG network in which the style loss is computed, $\psi(\phi_j(\cdot)) = \phi_j(\cdot) \cdot \phi_j(\cdot)^T$ is the Gramian Matrix, which is used to represent the style information. According to [35], lower layers in a pre-trained image classification network tend to preserve the lower-level image features (e.g., colours and contrast), and higher layers tend to reserve high-level image features (e.g., semantic image spatial structures). Therefore, the lower layers usually are chosen for the style loss functions, and the higher layers usually compute the content loss functions. In [37], Gatys et al. choose the set of lower layers in {*relu*1_1, *relu*2_1, *relu*3_1, *relu*4_1, *relu*5_1} for the style reconstruction and the set of high layers in {*relu*4_1} for the content reconstruction. In general, the choice of

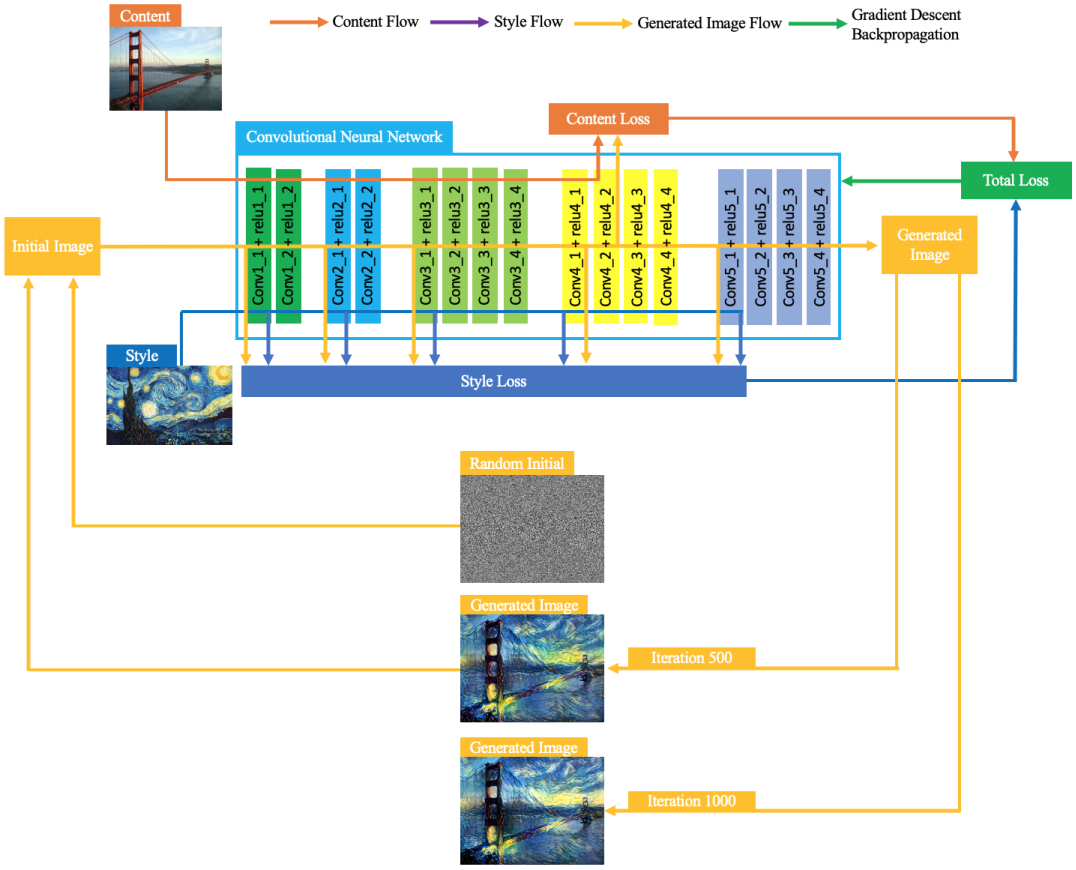


Figure 2.4: The Neural Style Transfer algorithm.

pre-trained image classification networks can be diverse, for example, the Resnet is used to perform the similar results in [68].

For implementing the style transfer method, Gatys et al. [37] propose an optimization process based on gradient descent to minimize the Equation 2.3. As the Equation 2.4 and 2.5 are differentiable, the proposed algorithm [37] starts with a random white noise as the initialization image $\hat{x}$, then iteratively generates a new image $\hat{x}$ by using gradient descent backpropagation to produce the final stylized result. The total process is shown in Fig. 2.4. In [68], Johnson et al. add a total variation term to encourage the smoothness of the stylized results in practical.

Inspired by the Gram-based representation, Li et al. [91] prove that the style transfer problem can be solved by separately matching the distribution of style representations and content representations. Hence, they consider style transfer as domain adaption, which seeks a method to match samples in the source domain to that in the target domain. In other word, the methods minimizing the distribution discrepancy of two domains are also suitable to style transfer. Inspired by [43], Li et al. choose the Maximum Mean Discrepancy

(MMD) with a quadratic polynomial kernel (e.g., linear kernel, polynomial kernel and Gaussian kernel) to measure the style distribution discrepancy of the stylized image and the style image. In addition, they also use mean and standard deviation of feature maps, which is referred to BN statistic representations in a pre-trained VGG layer to model the style:

$$\mathcal{L}_{style}(\widehat{x}, x_s) = \sum_{j \in J_s} \frac{1}{N_j} \sum_{i=1}^{N_j} (\|\mu(\phi_j^i(x_s)) - \mu(\phi_j^i(\widehat{x}))\|^2 + \|\sigma(\phi_j^i(x_s)) - \sigma(\phi_j^i(\widehat{x}))\|^2) \tag{2.6}$$

where J_s denotes the set of layers in VGG network computing the style loss, i denotes the i-th feature channels and N_j denotes the total number of channels in the j-th layer. $\mu(\phi_j^i \cdot)$ and $\sigma(\phi_j^i(\cdot))$ denote the mean and standard deviation of the corresponding feature maps.

Despite the visually appealing results, the methods based on Gram-based style representation are limited by the nature of instability, which are easily stuck at local minima of the style loss function. Moreover, the manually tuning of the hyperparameters is quite tedious. Risser et al. [160] observe that the Gramian Matrix of feature map activations is non-sensitive to their mean and variance. To be precise, if the Gramian matrix is constant (e.g., the style image is fixed), then the variance of the stylized results can be freely change with corresponding changes to the mean, and vice versa. In other words, there will be infinite number of potential stylized results have the same Gramian matrix of the style image, which causes the instability of style transfer process. Based on this finding, Risser et al. [160] propose to preserve the entire histogram of the feature maps by adding an extra histogram loss, which guarantees the mean or variance of the Gram Matrix is preserved during optimization process. In addition, they also provide an automatic tuning process which is designed for preventing extreme values of gradients during the optimization process.

For abstract styles, Gatys et al.'s method has already shown impressive results. For photorealistic styles, however, the limitation of the Gram-based methods [35] is obvious. For example, Gatys et al.'s method lack photorealistic details and strong smears in their results. Li and Wand [91] discover that Gram-based style representations are absent of the spatial content layout by only capturing the per-pixel feature correlations. Based on this finding, they introduce a new loss function with a patch-based MRF prior:

$$\mathcal{L}_{style}(\widehat{x}, x_s) = \sum_{j \in J_s} \sum_{i=1}^{m} \|\psi_i(\phi_j(x_s)) - \psi_{NN(i)}(\phi_j(\widehat{x}))\|^2 \tag{2.7}$$

where J_s denotes the set of layers computing style loss, $\psi(\phi(\cdot))$ denotes the list of all local patches extracted feature maps $\phi(\cdot)$. For each patch $\psi_i(\phi(\cdot))$, a best matching style patch $\psi_{NN(i)}(\phi(\cdot))$ is pursued by using normalized cross-correlation over all the style patches in

x_s. Due to the match of content and style on a patch-level, their method produces much better results than Gatys et al.'s [37] method especially for the photorealistic styles.

The aforementioned stylization networks mainly focus on the improvement of the style reconstruction aspect while the loss of content low-level details during the content reconstruction is barely noticed. In the traditional optimization objective, the content details like exact edges are lost as only high-level layers are used to constrain the content reconstruction process, which causes inconsistent and unexpected artefacts appear in the stylized results. To address this problem, Li et al. [93] introduce an extra loss function named Laplacian loss, which measures the difference of the Laplacians computed by Laplacian filter operating on the feature maps of the content image and the stylized result in VGG layers.

Kolkin et al. [77] propose a new optimization-based style transfer algorithm, which replaces second order summary statistic widely used in prior works with an efficient approximation of the Earth Movers Distance initially proposed in the Neural Language Processing community [82]. The proposed objective function is denoted as:

$$\mathcal{L}(\widehat{x}, x_c, x_s) = \frac{\alpha\mathcal{L}_c + \mathcal{L}_m + \mathcal{L}_r + \frac{1}{\alpha}\mathcal{L}_p}{2 + \alpha + \frac{1}{\alpha}} \tag{2.8}$$

where the content loss $\mathcal{L}_c$ aims to minimize the normalized cosine distance between feature vectors extracted from any pair of coordinates, the moment matching loss $\mathcal{L}_m$ aims to prevent over/under-saturation artefacts, the relaxed earth movers distance $\mathcal{L}_r$ aims to transfer the structural forms of the source image to the target, and the color matching loss $\mathcal{L}_p$ encourage output and the style image to have a similar palette. Their method also provides user control by region-to-region and point-to-point masks. The proposed approach preserves better stylistic patterns with respect to the spatial structures in content inputs, but the execution time is around 4 times than Gatys et al.'s method.

2.3.3 Methods based on Feed-forward networks

The second category of Neural Style Transfer methods is based on feed-forward networks, which is proposed to address the efficiency issue. ***Motivation***: As mentioned in Section 2.3.2, prior optimization process has one most concerned problem that they need several minutes to reconstruct the stylized result for single pair of still images even though the algorithms are implemented in high parallel platforms i.e., GPU-based Torch [22] and Tensorflow [1] with CUDA [113]. To speedup the reconstruction process, a feed-forward or generative network Ω is trained in advance by a large set of images x_c and one or multiple style images x_s, which seeks Weights to minimize the similar loss functions to [37]. The energy objective for

this process is formulated as:

$$W^* = \arg\min_{W^*} \mathcal{L}_{total}(\Omega_{W^*}(x_c), x_c, x_s) \tag{2.9}$$

So far, these feed-forward methods are categorised by the number of styles that Ω is able to produce. To be precise, they are named *Per-Style-Per-Network*, *Multiple-Style-Per-Network* and *Arbitrary-Style-Per-Network*.

Per-Style-Per-Network (PSPN) Methods Ulyanov et al. [146] and Johnson et al. [68] are the first to propose fast neural artistic style transfer methods with the similar idea, which utilizes a pre-trained feed-forward network with one artistic texture to stylize still images by a single forward pass at the test stage. The main difference of their methods is the architecture of feed-forward network. Ulyanov et al. [146] exploit a generator network with multi-scale architecture which is trained by the backpropagation of Gatys et al.'s [35] loss functions inside the pre-trained VGG network [136]. Inspired by the network architecture in [120], Johnson et al. [68] follow their network design but with residual blocks and fractionally strided convolutions.

Ulyanov et al. [144] further explore that a simple normalization over each single image rather than a batch of images during the training process is capable of significantly strengthening the stylization quality. The proposed normalization on each image is referred as ***Instance Normalization*** (IN). In fact, the IN is the special case of Batch Normalization when the batch size is equivalent to 1. For style transfer networks with IN, a fast converge and better visual quality is obviously shown. One of reasonable explanations for the improvement of IN is that the normalization enforces the style of each training image to the desired style [59]. Hence, the energy objective is much easier to learn as the network only needs to focuses on the learning of content reconstruction.

Li and Wand [92] propose another efficient network based on the patch-based MRF method mentioned in [91]. Their method pre-computes a feed-forward and strided convolutional network via adversarial training, which captures the style feature statistics of Markovian patches and is capable of stylizing image in nearly real-time performance. Compared to [68, 146], their method produces more coherent synthesis results. Their novel patch-based design directly inspires the first work in the subgroup: Arbitrary-Style-Per-Network Method.

Multiple-Style-Per-Network (MSPN) Methods Per-Style-Per-Network methods already produce stylized results in two orders of magnitude faster speed than the previous gradient-based methods, however, the limitation is that the feed-forward network can only be trained to stylize images in single artistic style, and the training procedure costs hours to learn one style. An early work studied by [26] demonstrates that many artistic paintings actually share one common style (e.g., 'Expressionism') with similar visual elements (e.g., brush strokes) but in different colour palette. Hence, a true stylizing network should be able to exploit and learn from such regularities, which means one single separate feed-forward network would be capable of stylizing multiple styles. Inspired by this discovery, many following methods [17, 26, 98, 174] are proposed for the purpose that embeds multiple style paintings into one single network. In general, there are two different paths to accomplish the MSPN purpose: **(1)** learn an affine transformation from a small number of parameters in a network to each style (e.g., [17, 26]) and **(2)** train a single network like [68] but with multiple styles and contents as inputs [98, 174].

(1) learn an affine transformation from a small number of parameters in a network to each style. Dumoulin et al. [26] observe that it is sufficient enough to specialize scaling and shifting parameters after Instance Normalization mentioned in PSPN to each specific style, and propose a conditional instance normalization for the training process. Hence, their feed-forward network is called Conditional Style Transfer (CST) network, which is denoted by:

$$CST(\phi(x_c), s) = \gamma_s(\frac{\phi(x_c) - \mu(\phi(x_c))}{\sigma(\phi(x_c))}) + \beta_s \tag{2.10}$$

where $\phi(\cdot)$ denotes the feature maps, and $\mu(\phi(\cdot))$ and $\sigma(\phi(\cdot))$ denote the mean and standard deviation of feature maps. For each style index s, the network learns an affine transformation from a combination of γ_s and β_s to single specific style, therefore, choosing different combination of γ_s and β_s leads to stylized results in corresponding desired style.

Chen et al. [17] also propose a MSPN method which follows the idea of [26]. Their main contributions are exploring an explicit representation of styles which can be stored in a particular mid-level layer called 'StyleBank' layer and each style is tied to a set of parameters in the 'StyleBank' layer. In addition, they also propose an encoder-decoder network which learns to reconstruct semantic content information. Their method further allows flexible training, for example, the encoder-decoder component can be fixed while only the 'StyleBank' layer is trained for new styles.

(2) train a single network with multiple style and content as inputs Though the first

path discovers that the affine transformation from parameters in a particular layer to the specific desired style, the network would become larger along with the increasement of learned styles. To address this issue, the second path MSPN methods propose to expand the ability of a single network by fusing both styles and contents for style identification.

Li et al. [98] are the first to propose a multi-texture synthesis network which aims to learn multiple textures in one single network. To achieve this goal, they propose an incremental training strategy which makes sure that the single network can learn a new texture without forgetting previous learned textures. Then they apply the learned multi-texture synthesis network for multi-style transfer task. They propose a selector network which establishes a one-to-one mapping between noise maps and correspondence styles. Then they build a style transfer network with encoder-decoder architecture. The selector network generates a corresponding noise map $\psi(x_s)$ and concatenates ($\oplus$) the map with encoded features extracted from the content $Enc(x_c)$, then the concatenated result is fed into the decoder to get the stylized result. The formula is put as :

$$\widehat{x} = Dec(\psi(x_s) \oplus Enc(x_c)) \tag{2.11}$$

where the encoder-decoder network is similar to the feed-forward proposed in [68].

Zhang and Dana [174] propose Inspiration Layers in a generator network in which two inputs (encoded content and style representations) are fed into these layers. The Inspiration Layers match multi-scale summary statistic of content and style representations to complete the style transfer task. During the training stage, the generator network seeks a direct solution $\widehat{x}$ in each layer $j \in J$, the formula is denoted as:

$$\widehat{x} = \arg\min_{\widehat{x}} \sum_{j \in J} (\|\phi(\widehat{x}^j) - \phi(x_c^j)\|_F^2 + \alpha \|\psi(\widehat{x}^j) - \psi(\phi(x_s^j))\|_F^2) \tag{2.12}$$

where J denotes the set of layers where Inspiration Layers located in a pre-trained 16-layer VGG network[136]. $\phi(\cdot)$ and $\psi(\cdot)$ denote separately the encoded content representations and Gram-based style representations as mentioned in Section 2.3.2. α is the interpolation weight for content-style trade-off.

Arbitrary-Style-Per-Network (ASPN) Methods The MSPN methods have already shown the potential capability of integrating multiple styles into one single network with nearly real-time performance, but the limited number of styles is still undesired. Motivated by this, the ASPN methods are proposed in more recent.

Chen and Schmidt [19] propose the first ASPN method, which follows the patch-based style transfer path in [91]. They propose an Encoder-Decoder network in which their algorithm seeks the closest match between the encoded content and style representations and then swaps them in the patch level. The Encoder part consists of several convolutional layers from a pre-trained VGG network [136], and the match and swap procedures are proceeded in the latent space. Then the swap result (activations in CNN) is passed through an Decoder to reconstruct the stylized result. The Decoder can be a gradient descent optimization process proposed in [37] or a trained feed-forward network proposed in [68].

The affine transformation discovered in [26] using a few parameters to corresponding styles already shows the potential ability for multiple style transfer. Based on this finding, Ghiasi et al. [39] learn a prediction network which is capable of tuning parameters (γ_s and β_s) through the affine transformation to the arbitrary desired styles. Huang and Belongie further expand this idea to suit arbitrary styles by proposing an adaptive instance normalization(AdaIN), the AdaIN transfers the mean and variance of content representations to those of style representations in the channel-wise way. The Adaptive Instance Normalization is formulated as:

$$AdaIN(\phi(x_c), \phi(x_s)) = \sigma(\phi(x_s))(\frac{\phi(x_c) - \mu(\phi(x_c))}{\sigma(\phi(x_c))}) + \mu(\phi(x_s)) \tag{2.13}$$

where $\phi(\cdot)$ denotes the extracted feature activations in an Encoder which consists of the first few layers in a pre-trained VGG network. Then a Decoder produces a stylized result from the reconstruction of AdaIN result. This Decoder is trained by a large set of style and content images so that it is capable of decoding precisely feature activations after AdaIN to the transferred result: $\hat{x} = Dec(AdaIN(\phi(x_c), \phi(x_s)))$. The method of Huang and Belongie [59] is data-driven and not able to transform content inputs into unseen styles. It's hard to synthesize the complicated texture patterns with rich details by simple adjustment of mean and variance of feature statistic.

Li et al. [99] propose another effective path to accomplish arbitrary style transfer tasks. They discovered that feature transforms such as whitening and colouring transform (WCT) are capable of removing style information from content representations and inversing the step of whitening. To be precise, the whitening step removes the style information from the content representations and obtains an intermediate result ϕ_c which only contains the content information, then the colouring step recombines the style representations ϕ_s and ϕ_c and obtains an intermediate result $\phi_{cs} = WCT(\phi_c, \phi_s)$ which has the desired correlations

between content and style representations. The stylized result is finally obtained by decoding ϕ_{cs}: $\hat{x} = Dec(\phi_{cs})$. The Encoder and Decoder are fixed when processing arbitrary style transfer. Also, they extend their single-level to multi-level stylization in order to match the statistic of the style at all levels.

Sheng et al. [134] propose a high quality arbitrary style transfer with real-time execution time which combines style-swap from [19] and whitening/coloring transformation from [99]. They investigate into the trade-off between the generalization and efficiency, and propose patch-based a style decorator that makes up the content features by semantically aligned style features from an arbitrary style image. The style decorator makes sure that it maximally aligns the distributions of $Enc(\hat{x})$ and $Enc(x_s)$ and the detailed style patterns are semantically perceptible in $Enc(\hat{x})$. The style decorator is achieved by three steps: 1. Projection. $Enc(\hat{x})$ and $Enc(x_s)$ are projected onto the same space by subtracting their mean features then convolution operation by whitening kernels; 2. Matching and Reassembling. This patch matching between the projected of $Enc(\hat{x})$ and $Enc(x_s)$ ensures they have the maximal overlap with each other via the help of normalized cross-correlation; 3. Reconstruction. The patch-matched result will be reconstructed into feature domain via coloring transformation.

Li et al. [94] propose a learnable linear transformation matrix which is capable of transferring arbitrary stylistic patterns from a reference style input onto a content input efficiently at 140 fps. The proposed approach consists of an encoder-decoder image reconstruction module and a transformation learning module. The transformation matrix is proved that it is only determined by the covariance of the content and style image feature vectors, which is learnable by a light-weighted CNN network. In addition, they also present a linear propagation module to correct distortion artefacts in contours and textures, which obtains photo-realistic stylization results. Moreover, the proposed linear transformation method is able to preserve feature affinity across content frames and finally outputs stable video style transfer results.

Zhang et al. [181] point out that prior works treat the semantic patterns of style image uniformly, which leads to unpleasing results on complex styles. To address this problem, they introduce a multimodal style transfer algorithm following patch-based methods (e.g., [45, 91, 134]) that the style image features are clustered into sub-style components, which are matched with local content features under a graph cut formulation. Specifically, their approach formulates style-content matching as energy minimization problem with a graph and solve it via graph cuts. Style clusters are adapted to content features regarding to the content spatial structures.

Wang et al. [155] propose a simple and effective arbitrary style transfer method that has more advantages like generalized, diverse and scalability than previous works. They introduce a *Deep Feature Perturbation* (DFP), an orthogonal random noise matrix, which perturbs the deep image feature maps while keeps the original style information unchanged. The proposed DFP operation is easily integrated into many existing WCT-based methods (e.g., [99, 134]), and empower them to output more diverse results with arbitrary styles.

2.4 Neural Photo Style Transfer in Neural Style Transfer

Neural Photo Style Transfer methods aim to transfer the style of colour distributions from photorealistic style images to content images. So far, the common path to do so is using segmentation masks to accurately match the regions from a photorealistic style image to a content image. To be precise, the NPST methods need two segmentation masks, one mask labels the desired regions of a photorealistic style image in specific colours (i.e., blue and black) and the other mask labels the expected corresponding regions of a natural content image also in corresponding colours. By the end of May 2020, there are in total two different NPST methods are proposed. The earliest method [108] is based on the first NAST algorithm proposed by Gatys et al.[37], and the recent approach [97] is based on the WCT algorithm proposed by [99] in two steps.

Gatys et al. [37] discover that the photorealistic style transfer between photos using their method produces unexpected distorted stylized image with characteristic noise artefacts. To address this problem, Luan et al. [108] propose the first NPST method which uses Mask-based style representations encoded in a pre-trained VGG network [136] for colour distribution match and adds a photorealism regularization based on Laplacian matrix to eliminate distortions as a post-processing step. Since their method is built upon Gatys et al.'s method [37], thus the proposed method has computational optimization burden and needs several minutes to generate a stylized result. Moreover, the stylized results processed by this approach also exists posterization artefacts, which harms the photorealism.

Li et al. [97] propose a fast NPST method which proceeds photorealistic style transfer within two steps: the stylizing step and smoothing step. Inspired by the success of the unpooling layers [114, 171] , Li et al. [97] propose a stylizing step by using the improved WCT algorithm [99] which replaces upsampling layers with unpooling layers. After first step, the stylized result still lacks of photorealism due to the stylized regions are not inconsistent. Hence, a smoothing step is applied to preserve the consistent stylized regions with the

balance of local pixel colour and global stylization effects. Except those two methods mentioned above, another work [111] propose to utilize Screened Poisson Equation as additional post-processing step to handle the posterization artefacts introduced by Luan et al. [108]. He et al. [49] propose a more accurate color transfer method which jointly optimize semantical matching between images and color transfer by a local linear model with satisfying both local and global constraints of deep features. The proposed method also can be extended from "one-to-one" to "many-to-many" color transfer. More recently, Yoo et al. [170] present a wavelet transforms (WCT^2) that allows features to preserve their structural information and statistical properties of VGG features space during stylization. Their method gives a pleasing photorealistic output without any post-processing and is able to stylize a 1024×1024 resolution image in 4.7 seconds.

The aforementioned methods follow directly from corresponding NAST methods (e.g., [37] and [99]). Different from above methods, Liao et al. [103] utilizes the maintained dense correspondence of feature representations between input images, which exploits the PatchMatch algorithm [8] to match the nearest neighbour field between two dense correspondence. Their method not only handles "photo to artistic" transfer well but also performs good photo style transfer results. Since their method is inspired by "Image Analogy" mentioned in Section 2.1.2, it requires strict size and similar semantic structures of the source and target image.

Based on the above analysis of NPST methods, the content mismatching and slow speed performance are the main concerns in this research field. To address these issues, Chapter 3 and 4 will illustrate the reasons behind them by a deeper investigation and propose new methods to better match spatial structures of input images and boost speed performance into near real-time.

2.5 Neural Video Style Transfer in Neural Style Transfer

Neural Video Style Transfer methods extends NAST to video applications, which transform an entire video into a specific artistic style. In the early stage of NVST, the proposed methods usually naively apply per-frame stylization methods such as Fast-Neural-Style [68] and AdaIN Style Transfer [59] to process all the frames of a video, however, the flickering artefacts causes unpleasant outputs, in which the stylistic texture appearances are not consistent between consecutive frames. By the end of May 2020, there are two branches of NVST methods in total. The earliest methods [5, 124] are developed on optimization-based network

which is the very first of Neural Style method [37], the following works [16, 46, 57, 85, 95, 125] are based on feed-forward networks.

Anderson et al. [5] and Ruder et al. [124] follow Gatys et al.'s optimization-based approach [37] and directly use it to perform artistic video stylization. Gatys et al's method is utilized to independently stylize video sequences frame-by-frame. However, the flickering artefacts between coherent frames lead to unappealing results. To improve the temporal consistency between adjacent frames, Anderson et al. [5] proposed a NVST method that integrates optical flow into Gatys et al.'s method, and a warped image using flow as an initialization is fed into the optimization-based network. They also incorporate flow explicitly into a temporal consistency loss function. However, the ghosting artefacts occur due to flow errors. To reduce the ghosting artefacts, Ruder et al. [124] introduce masks to filter out the low flow confidences via forward-backward check. Their approach gives stable and consistent stylized videos even in large motion and occlusion cases. However, the ghosting artefacts still exist in their methods. To remove these artefacts, simple strategy like adopting original content into occlusion regions may work in some cases, but this can also raise another problem that contents warped from a previous frame may not be consistent with the context in the current frame due to flow errors. In other words, some original contents from previous frames copied into wrong positions as flow errors cause masks to filter out wrong low flow confidence regions. Moreover, the heavy iterative optimization process still costs minutes to stylize one frame. It's less practical for video style transfer.

Johnson et al. [68] propose the very first feed-forward networks for video stylization task in real time. Their approach trains a feed-forward network via gradient computed by a perceptual loss in a loss network which approximates the optimum of Gatys et al.'s loss functions. And at test time, one forward pass is able to complete the style transformation. It is three orders of magnitude faster without optimization process, however, this per-style-per-network approach is still less practical for image and video processing. To extend feed-forward networks for arbitrary styles, Chen et al. [19] present an approach based on patch matching strategy, which replaces the content image patch-by-patch from the style image on a single style swap layer in deep neural networks. To further reduce the time consumption, Huang et al. [59] replace the style swap layer by an adaptive instance normalization layer, which aligns the means and variance of the content features with stylistic features. These methods mentioned above are fast enough for video style transfer, but the incoherence problem is not yet solved.

To address the incoherence problem, Gupta et al. [46] and Huang et al. [57] improve

Johnson et al.'s feed-forward networks by introducing the temporal consistency loss during training time, while still obtain stable outputs in real-time. Chen et al. [16] assemble a stylization sub-network, a flow sub-network and a mask sub-network to a fusion network, which considers propagates a short-term coherence to long-term by a recurrent convolutional network strategy. Recently, Ruder et al. [125] propose a network-based approach which stylizes arbitrary-length videos in a stable and coherent way. However, their network still follows the pattern of per-style-per-network. More recently, Lai et al. [85] proposes a blind video processing technique which is capable of preserving temporal consistency. Their method processes the per-frame stylized results and achieves real-time execution time as no optical flow needed at test time. Li et al. [95] proposes a new linear transformation matrix to minimize the difference between covariance of content features and style features, which keeps temporal consistency by propagating the computed matrix from the beginning of frame to the rest of them.

This section analysizes that NVST methods concern most the following issues for practical applications: the temporal consistency among consecutive stylized frames, speed performance, style numbers that a single method can transfer and diverse results. Chapter 5 proposes a new method with a set of mask technique and coherent losses to address these issues.

2.6 Digital Bas-relief Modelling

In the last two decades, generating digital bas-reliefs from 3D scenes or 2D images has been a thriving subject in computer graphics. A detailed review work can be found in [73, 180], which classify methods into direct modelling, image-based and shape-based modelling. Direct modelling involves experts' laborious work. Image-based 3D construction inherently has the ill-posed problem. Most 3D model-based works create bas-relief by either directly compressing the depth or by working in the gradient domain, where the final model is obtained by solving a Poisson equation. Most cases aim to generate a reproducible and manipulable mesh, which can later be used and enhanced with some more advanced graphical tools. As a result, specific colour and texture of the relief are usually not considered in these previous research. Our work is designed to solve this problem by using the learning scheme to transfer textures from 3D source models to 3D target models.

2.6.1 3D model-based methods

By applying feasible constraints on a given 3D shape or scene, Christian et al. [127] propose a novel view-dependent surface representation which allows us to cast the optimization as a quadratic program. Ji et al. [65] propose a highly efficient two-scale bas-relief modelling method on GPU, in which the input 3D scene is first rendered into two textures with depth information and normal information respectively. The depth map is then compressed to produce a base surface with level-of-depth, and the normal map is used to extract local details. Finally, the local feature details are added back to the base surface to produce the final result. Based on 3D models, Zhang et al. [177] propose a series of gradient-based algorithms which operates directly on a triangular mesh and ensures that the mesh topology remains unchanged during geometric processing. They also present two types of shape editing tools that allow the user to interactively modify the bas-relief and exhibit a desired shape. Given target shapes, viewpoints and space restrictions, Zhang et al. [176] find a global optimal surface that delivers the desired appearance when observed from the designated viewpoints, which could guarantee exact depth bounds of per-vertex. Zhang et al. [179] treat an input object as a continuous relief depth map and use mesh intersection to paste the relief on the target object based on empirical mode decomposition in multi-scale levels [173].

2.6.2 Image-based methods

Generating bas-relief from natural images and photographs are intuitive. However, we all know that there is an ill-posed problem to recover 3D shape from a single image, since colour, luminance and texture in an image could not reflect the geometric attributes of the objects properly, especially for objects with complex materials. To overcome this problem, some researches are restricted to some special types of bas-relief from certain images. For example, Wu et al.[163], Wu et al. [164], and B. Sohn [142] concentrate on bas-relief modelling from human face photographs. Zhang et al.[179] pay special attention to model Chinese calligraphy reliefs. Li et al. [100] aim at restoring brick and stone alike relief from single rubbing image in a visually plausible manner. Zhang et al. [178] concentrate on portrait relief modeling. Some researches are based on Shape From Shading (SFS) which requires human interaction ([172][42][140]).

Unlike generating bas-relief from natural images, some recent works start from normal images[109]. Ji et al. [63] present a novel framework to design bas-relief in normal image space instead of object space which is capable of producing different styles of bas-relief and allows intuitive style control. Their method generates high-quality bas-relief which enables

a variety of applications, such as the cut-and-paste operation and bas-relief modelling on curved surface. Recently, Ji et al. [64] extend their previous work with a layer-based editing approach for normal images to generate more diversified styles of results, and is capable of transferring details from one region to another. Similar to Ji et al.'s work [65] , Wei et al. [156] decompose image normal of an input 3D model into a smooth base layer and a detail layer in order to contribute to both features of structure-preserving and detail-preserving.

2.6.3 Detail transfer

With the growing availability of abundant 3D mesh collections, some research works attempt to transfer textures [133] or details to geometric shapes.

Mitra et al. [150] propose an unsupervised learning method to transfer texture information from images of real objects to 3D models of similar objects by tackling the reconstruction problem of a set of base texture. Huang et al. [58] present a novel user-assisted approach to extract a non-parametric appearance model from a single photograph of a reference object (whose geometric structure roughly approximates that of the target object). A novel alignment algorithm is proposed to enable accurate joint recovery of the geometric detail and reflection. Berkiten et al. [10] propose a method which transfers details (specifically, displacement maps) from existing high-quality 3D models to simple shapes. They adopt metric learning to find a combination of geometric features that successfully predict detail-map similarities on the source mesh; then they use the learned feature combination to drive the detail transfer in texture space.

This section reviews the literature works on bas-relief modelling, and analysizes that modelling based on normal images could be more promising than 3D-modelling as it solves the ill-posed problems that other image-based approaches suffer and removes the need of sophisticated depth compression algorithms. In addition, for detail transfer, previous methods transfer textures between pairs of images and 3D models sharing similar objects which limits their applications. Chapter 7 proposes a new texture transfer algorithm using normal images, which is capable of transferring high quality details between arbitrary pairs of inputs and enrich the detail design for bas-relief modelling.

Chapter 3

Neural Photo Style Transfer

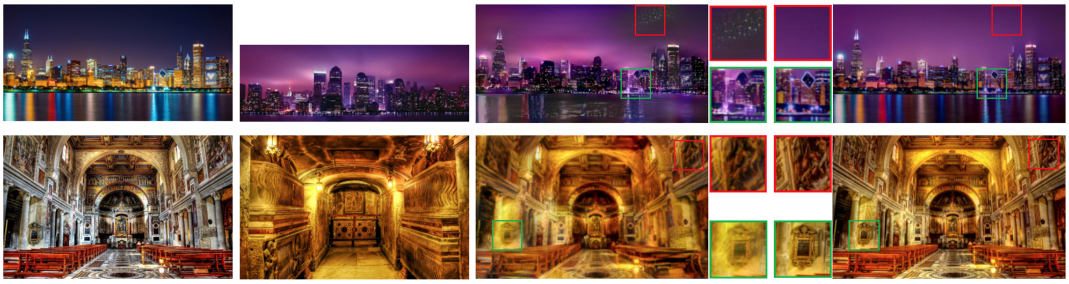


Figure 3.1: Given a reference style image and a content image as inputs, photographic style transfer seeks to generate output with photorealistic attribute, which should preserve both the context of content and style of reference. Gatys et al. [37] succeed in transferring style colour but introducing distortions to the context of output. In comparison, the proposed method transfers faithful style colour meanwhile also preserves the photorealistic attribute.

A neural artistic style transformation method (Neural-Style, NS) proposed by Gatys et al.[37] has achieved great success with Convolutional Neural Networks, which is followed by many works [17, 19, 39, 59, 91, 122, 146, 153, 174] recently. They produce convincing visual results by transferring artistic features from reference painting onto the content photograph. However, these artistic style transfer methods suffer from visual distortion problem, and make the results have a painting-like looking, especially when both of the content and reference style images are photographic.

To solve this problem, this chapter introduces a similarity layer with correspondence loss function to constrain both content preservation and style transformation processes. This similarity layer is added into several places of the Convolutional Neural Network to prevent distortions by minimizing a similarity loss function together with other loss functions proposed in ***Fast Neural Style*** algorithm [68]. To further enhance the photorealism, this chapter also introduces a **Style Fusion Model** (SFM) as a post-process step. The model extracts the colour information from style transformation output and the detail information from content preservation process, then combine together these information to generate a new output.

Figure 3.2: Distortions occur at both content preserving and style transformation process. (c-i) contains the zoom-in insights of input content . (c-ii) shows that (a) introduces distortions into reconstructed content details, and (c-iii) shows that (b) distorts details of (a).

3.1 The photorealism loss of NAST

Luan et al. [108] point out that the distortions appear only at style transformation process, they thus propose a two-stage photo style transfer method. The first stage (Seg-NS) integrates semantic segmentation masks to Neural-Style method [37] to avoid the unexpected geometric matching problem, and the second stage (Mat-NS) uses a photorealism regularization term based on Lapacian Matting to reconstruct fine content details. Although the content spatial structures are preserved in many situations, details and exact shapes of structures are erased when semantic segmentation is inaccurate or contains overlapping areas. ***And the computation of matting laplacian matrix and semantic segmentation consumes much extra time for high quality output.*** Moreover, Luan et al.'s method also suffers from the posterization artefacts [103]. After investigating the style transformation procedure , this work discovers the distortions occur at two stages: the spatial structures of content image may loss during content preserving process and the unexpected geometric matching can be introduced during style transformation process. Fig. 3.2 illustrates the distortions occur at both content preserving and style transformation process. As shown in (c-ii), the buildings of content image are obviously distorted by content preserving process. Also as shown in (c-iii), the buildings are also distorted after style transformation process. Buildings of (c-iii) hold different shapes and edges from (c-ii) from content preserving process, which means the buildings are distorted twice.

3.2 The Proposed NPST Method

The entire style transfer pipeline consists of two stages: detail reconstruction process and style transfer process. The proposed framework has two key components: a dual-stream deep convolution network as Loss Network and edge-preserving filters as ***Style Fusion Model*** (SFM). The edge-preserving filter is used to extract details and colour information

of the outputs from the loss network, which means the proposed ***scheme*** combines the details without colour from content and the colour without details from reference style. During the optimization process, the content and style features are captured first by the additional layers in the Loss Network, then a random white noise image X is passed through both detail reconstruction and style transfer networks. The final output of SFM is the stylized result.

The main contributions of this chapter: 1. this work investigates the problem of Gatys et al.'s method (Neural-Style, NS), and find out that the lost photorealism of stylized result is caused by distortions occurring at both content preservation and style transformation stages; 2. this work proposes a neural photographic style transfer (NPST) method which is capable of improving the photorealism of stylized results. A similarity loss function using L1-norm is applied for reconstructing finer content details and preventing geometric mismatching problem (**CTRB1** Answering to **Q1**). To further enhance the photorealism, a *Style Fusion Model* using edge-preserving filter is proposed to reduce artefacts (**CTRB2** answering to **Q2**). 3. this work prevents the posterization artefacts of Luan et al.'s method by replacing Luan et al.'s second stage (Mat-NS) with the proposed SFM.

3.2.1 Architecture of Neural Photo Style Transfer

Gatys et al.[37] propose an image transformation network with convolutional neural networks to accomplish the task that an input image is transformed into an output image. The network architecture of Gatys et al. [37] includes a pre-trained VGG-19 network [136] and two loss layers. The layers learn feature representations of input images and compute the representation differences between a generated image and inputs. Their algorithm adds two additional layers: content layer and style layer, which capture and store feature representations of inputs. Then a random white noise image initialized as the same size of content input is fed into the network. The loss functions compute the distance of feature representations between the generated image with respect to content and reference style inputs separately. The derivatives of loss terms are propagated back to the loss network for next iteration until the maximum iteration number is reached. Similar to this optimization-based approach, the proposed work in this chapter also uses the pre-trained VGG-16 network [136] as the loss network, the content loss function and perceptual loss functions in [68]. In addition, this work adds another layer with pixel-level loss function into the network, and a ***Style Fusion Model*** as the post-processing step to reduce artefacts. The proposed method is an optimization-based approach which is designed for arbitrary style and content image pairs.

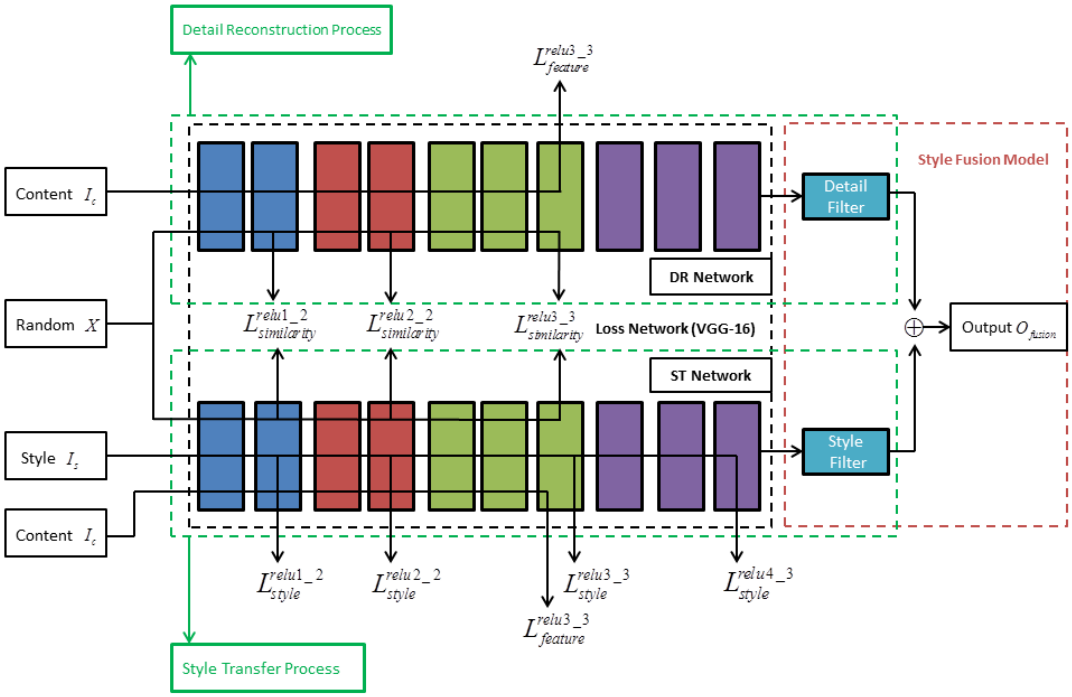


Figure 3.3: Framework Overview. This work uses the Loss Network to preserve content and transfer style from inputs to outputs. The loss functions are added into the pre-trained VGG-16 network [136], which are computed at certain layers and back propagated to the Loss Network during optimization process. For example, $L^{relu1\_2}_{style}$ computes the feature representation differences between random white noise image X and style image I_s, where relu1_2 denotes the placement for style layer in VGG-16 network. Then the deviation of $L^{relu1\_2}_{style}$ is propagated back to ST Network.

As shown in Fig. 3.3, the proposed framework consists of two components: ***a dual-stream convolution network consisting of a Loss Network and a Style Fusion Model***. The Loss Network is composed by two parallel deep convolution networks and several additional layers. A scalar value $Ł^i(y, y_t)$ of loss function at layer i is computed to measure the Euclidean distance between the output image y and target image y_t (y_t can be content image and reference style image). For the dual-stream loss network, this work refers the upper deep convolution network as *Detail Reconstruction network* (DR Network), which is designed for preserving the content details. Meanwhile, the lower convolution network is referred as *Style Transfer Network* (ST Network), which aims to transfer style information, mainly colour, from reference style image to content image. As shown on the right side of Fig. 3.3, the ***Style Fusion Model*** (SFM) also has two components: a detail filter and a style filter, which take the outputs of two parallel deep networks as their inputs separately.

Inputs and Outputs: For the DR Network, the inputs are one photograph as content image I_c and one random white noise image X_{DR} with the same size of I_c, and the output is one image O_c . For the ST Network, the inputs are one photograph as content image I_c

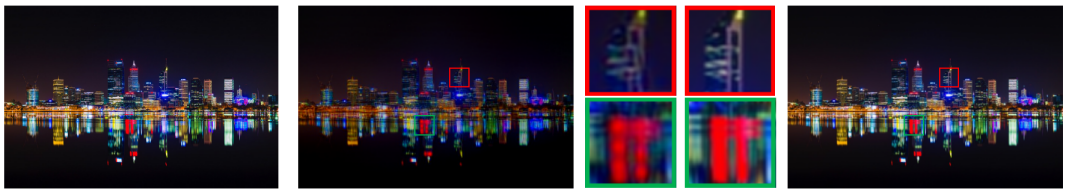


Figure 3.4: The similarity function for reconstructing finer content details. Left: the input content image. (a) and (c) are the reconstructed results through the DR Network without and with similarity loss respectively. (b) shows two insights of (a) and (c) (in that order) respectively. it is noticeable that (c) preserves more precise context of input than (a).

, one random white noise image X_{ST} with the same size of I_c and one photograph as style image I_s. The output is one image O_s. The X_{DR} and X_{ST} are initialized by random white noise image X. For the detail filter, the input is the output O_c of DR Network, and the input of style filter is the output O_s of ST Network. The output of entire SFM is one image O_{fusion}.

Additional Layers: There are three different layers in total: content layer, style layer and similarity layer. The content and similarity layers carry loss functions for the purpose of preserving content features from I_c onto O_c. And the style layers hold the loss functions to transfer stylistic features from I_s to O_s.

3.2.2 Loss Functions

In general, this work defines three different loss terms for two purposes: 1. preserve the content feature information F as structure details and reconstruct them on X_{DR}; 2. learn the reference style features and correctly match them to X_{ST}.

Layers in Convolutional Neural Network define non-linear filter banks to encode input image. Hence, the representations of features in a neural network actually are the filter responses to input image [110]. We assume that a layer has D different filters, and each filter has a size M, where M is height times width. For the reconstruction of feature, let ϕ_i be the feature representations captured at ith activation layer of the DR Network when I_c is on processing. Then ϕ_i is a feature map with the size of $D_i \times M_i$. The feature reconstruction loss is the squared and normalized Euclidean distance between the feature representations of X_{DR} and target I_c:

$$\mathcal{L}_{feat}(X_{DR}, I_c) = \sum_{i \in L} \frac{1}{D_i \times M_i} \|\phi_i(X_{DR}) - \phi_i(I_c)\|_2^2 \tag{3.1}$$

where L denotes the set of activation layers containing feature loss. This term helps to minimize the visual distinguishability between the random image X_{DR} and target image I_c.

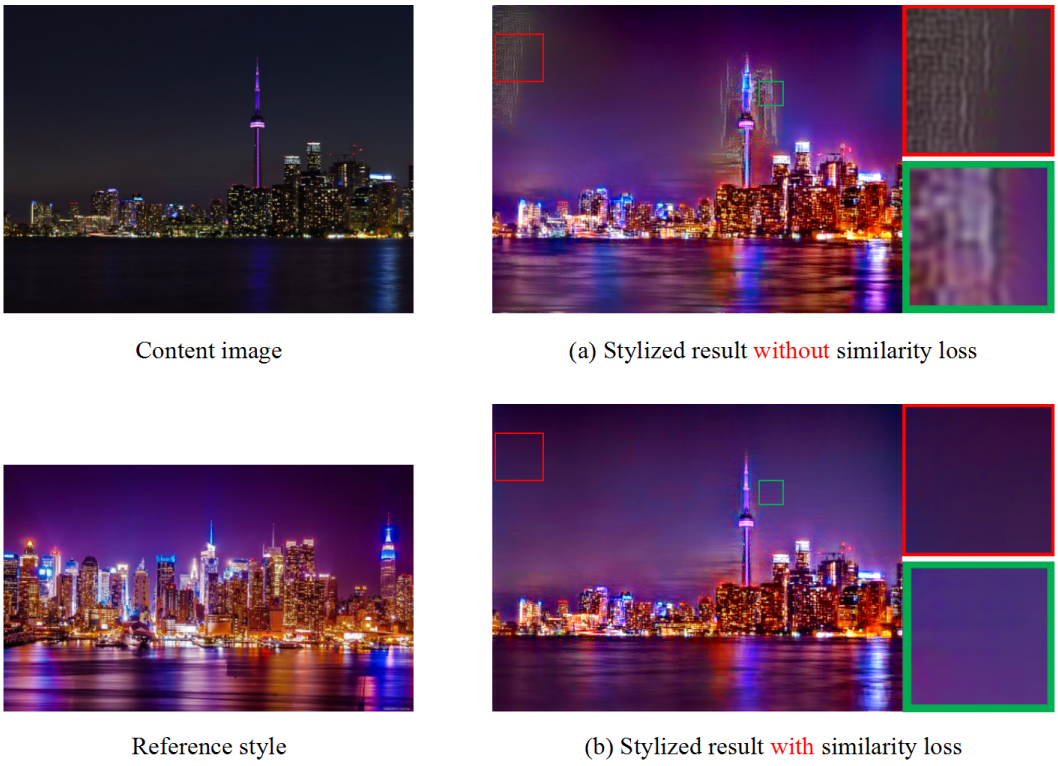


Figure 3.5: The similarity function for preventing geometric mismatching problem. (a) is the stylized result without similarity loss, and (b) is the stylized result with similarity loss. Note that the zoom-in regions show that the similarity loss effectively prevents the unexpected geometric matching.

However, as this reconstruction operates on high layers [110], the rough spatial structure of content image can be preserved but details especially exact shapes of the structure are lost.

For the same convolutional neural network architecture, Zhao et al. [182] demonstrate using L1-norm loss in the spatial constraint better preserves the spatial structures as compared to using L2-norm for image restoration task. Hence, another similarity loss $\mathcal{L}_{simi}$ is introduced based on mean absolute error (L1-norm) into the Loss Network. It is found that the L1-norm loss on RGB domain makes the style transformation output lose the colour information from style image as the loss also reconstructs colour of the content image to output. Hence, we add L1-norm loss in feature domain. Similar to the definition of $\mathcal{L}_{fea}$, $\mathcal{L}_{simi}$ is denoted as L1 loss of the feature representations of X_{DR} and I_c at jth activation layer of the Loss Network, then the similarity loss is defined as:

$$\mathcal{L}_{simi}(X_{DR}, I_c) = \sum_{j \in L} \frac{1}{D_i \times M_i} \|\phi_j(X_{DR}) - \phi_j(I_c)\|_1 \tag{3.2}$$

where L and $D_i \times M_i$ separately denote the set of activation layers and feature size. The purpose of this loss term is to measure how much information of target I_c is lost by X_{DR},

which contributes to reconstruct exact pixels of I_c into X_{DR} as many as possible by minimizing this term. As mentioned above, reconstructing content features with only $\mathcal{L}_{feat}$ is not enough to preserve precise details, especially the exact edges inside structures. Fig. 3.4 and Fig. 3.5 demonstrate the effect of $\mathcal{L}_{simi}$.

For the transformation of style, an effective representation of style in the reference image is needed. According to [36], the correlations of feature space is chosen to be the representation of style. And these feature correlations can be given by Gramian Matrix. Let ψ_k be the Gramian Matrix of vectorized feature map ϕ_k at kth activation layer of ST Network when the input X_{ST} is on processing, and the vectorized feature map ϕ_k is reshaped to $D_k \times H_k W_k$, then the Gramian Matrix is defined as:

$$\psi_k(X_{ST}) = \frac{1}{N}\phi_k(X_{ST}) \cdot \phi_k(X_{ST})^T \tag{3.3}$$

where N is the total number of pixels of $\phi_k(X_{ST})$. The Gramian Matrix is the dot product between feature maps at kth activation layer, which gives the feature correlations. Then the style loss is the squared Frobenius norm of the difference between the Gramian Matrices of the random image X_{ST} and the target I_s:

$$\mathcal{L}_{style}(X_{ST}, I_s) = \sum_{k \in L} \|\psi_k(X_{ST}) - \psi_k(I_s)\|_F^2 \tag{3.4}$$

where L denotes the set of activation layers holding style loss. The style loss is well-defined even for different sizes of X_{ST} and I_s since the $\psi_k(\cdot)$ always has the same $D_k \times D_k$ size. As demonstrated in [36], the generated output will only preserve the stylistic feature from style image, which means the spatial structure of target image can not be preserved by minimizing the style loss.

In this chapter, the $\mathcal{L}_{feat}$ and $\mathcal{L}_{simi}$ are used to constrain the detail reconstruction procedure, which produces output O_c with preservation of the spatial structures inside content image such as exact details like shapes and edges (shown as (a) in Fig. 3.6). These two loss terms forms $\mathcal{L}_{DR}$, the joint loss of DR Network. The $\mathcal{L}_{style}$, $\mathcal{L}_{feat}$ and L_{simi} constrain the style transformation procedure, which generates the output O_s with stylistic features mainly colour information from reference image and detailed features from content image. The combination of three loss terms forms L_{ST}, the joint loss of ST Network. Therefore, the two final joint loss terms are defined as:

$$\mathcal{L}_{DR} = \alpha_f \mathcal{L}_{feat} + \alpha_d \mathcal{L}_{simi} \tag{3.5}$$

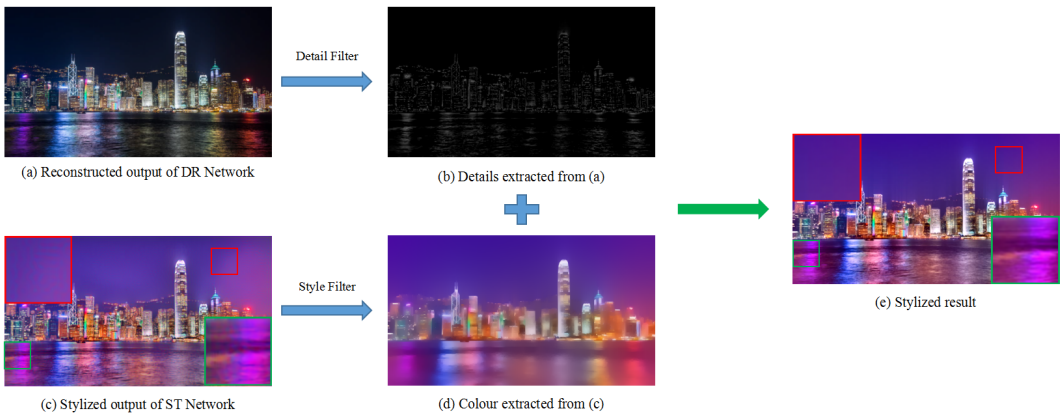


Figure 3.6: The ***Style Fusion Model*** for reducing noise artefacts and avoiding distortions. (a) is the reconstructed content output of the DR Network, and (b) is the extracted details (white points) of content without colour from (a). (c) is the stylized output of the ST Network, and (d) is the extracted colour without details from (c). (e) is the fusion stylized result from SFM. it is noticeable that (c) still exists noise (red rectangles) and distortion (green rectangles) arefacts due to content-style trade-off (please refer to Fig 3.8. However, the final stylized result (e) is free of noise and distortion artefacts.

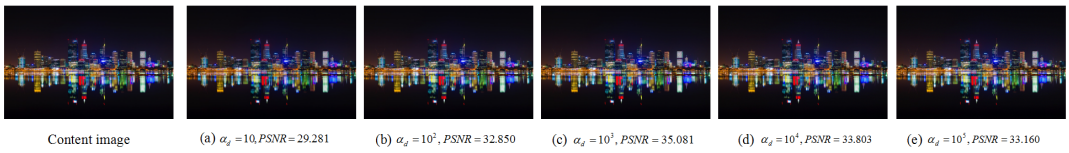


Figure 3.7: The effect of parameter α_d for the DR Network. Note that the reconstructed content result achieves the highest PSNR at $\alpha_d = 10^3$. The lower and larger values decrease the accuracy of reconstructed result. Hence, this work finds the best parameter $\alpha_d = 10^3$ for the DR network, and use it to produce all the other results in this chapter.

and

$$\mathcal{L}_{ST} = \beta_f \mathcal{L}_{feat} + \beta_d \mathcal{L}_{simi} + \beta_s \mathcal{L}_{style} \tag{3.6}$$

where α_f and α_d denote the weights of content layers and similarity layers in DR Network, and β_f, β_d and β_s denote the weights of three corresponding layers in ST Network. All the implementation details of these parameters are introduced in Section 3.3.

In previous researches [60, 117], the output of prior process contains stylistic features from reference style, and these features are distributed according to the semantic structures of content input. Hence, the style transformation procedure in the ST Network learns stylistic features and also distributes them into the semantic structures, which needs both style loss term and detail reconstruction loss terms.

3.2.3 Style Fusion Model

Section 3.1 mentions that the distortions are introduced by both detail preservation and style transformation procedures. This work uses $\mathcal{L}_{simi}$ to prevent geometric mismatching, however, the output of ST Network may still exist distortion and noise artefacts due to the

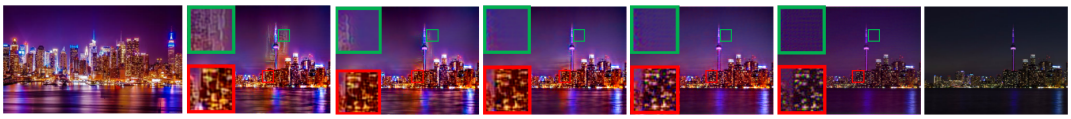


Figure 3.8: The effect of parameter β_d for content-style trade-off. A lower β_d value can not prevent unexpected geometric matching. For example, the regions of tower tops (green rectangles) in (a) and (b). A larger β_d value loses the style of reference image. For example, the buildings (red rectangles) in (d) and (e) have undesired dark colour style, which should be in the golden light style. Note that the stylized result at $\beta_d = 1 \times 10^1$ still exists some distortion and noise artefacts but they will be eliminated by SFM. This work thus chooses $\beta_d = 1 \times 10^1$ to produce the style transformation result of the ST Network and all the other results in this chapter.

Figure 3.9: The effect of parameter σ_r for SFM. Note that a lower σ_r value can not prevent noise artefacts, for example, red rectangles in (a) and (b), and a larger σ_r value suppresses the transferred style, for instance, green rectangles in (d) and (e). This work found the best parameter $\sigma_r = 1$ to produce the result and all the other results in this chapter.

content-style trade-off (shown in Fig. 3.8). To reduce the artefacts, this work applies a refinement technique ***Style Fusion Model*** (SFM) into the proposed approach. The edge preserving filter (Recursion Filter) proposed by Gastal et al.[34] is capable of effectively smoothing aways noise or textures while retaining sharp edges, which is a suitable technique for reducing artefacts. This work thus uses the edge preserving filter (Recursion Filter) [34] to smooth both output image O_c and O_s with joint image O_c. In this chapter, this work refers Detail Filter and Style Filter as the smooth process of O_c and O_s respectively. The final result O_{fusion} is defined as:

$$O_{fusion} = (O_c - RF(O_c, \sigma_s, \sigma_r, O_c)) + RF(O_s, \sigma_s, \sigma_r, O_c) \tag{3.7}$$

where σ_s denotes the spatial standard deviation and σ_r denotes the range standard deviation for the edge-preserving filter [34].As shown in Fig. 3.6 (e), the clear stylized result O_{fusion} obtained by the proposed SFM are free from the artefacts.

3.3 Implementation Details

This section describes the implementation details for the proposed approach. This work chooses the pre-trained VGG-16 network [136] as the basic architecture of the DR Network and ST Network. The content layer with L_{feat} is added into the activation layer of {$relu3\_3$}, and the style layers with L_{style} are added into {$relu1\_2$,$relu2\_2$,$relu3\_3$ $relu4\_3$} activation layers. The similarity layers are added into {$relu1\_2$, $relu2\_2$, $relu3\_3$} activation layers. For

Figure 3.10: Placements for similarity layers in DR Network. (a)-(d) show the reconstructed content results with similarity layers at different places in the DR Network. Note that the reconstructed result achieves the highest PSNR score at relu1_2, relu2_2, relu3_3. Hence, this work places similarity layers at relu1_2, relu2_2, relu3_3 in the DR Network for all the experiments in this chapter.

Figure 3.11: Placements for similarity layers in ST Network. (a)-(c) show the stylized results with similarity layers at different places in the ST Network. Note that (a) presents a worse stylized result than (b) and (c) as the centre area of blanket and walls upside are not in golden style colour. It is difficult to tell that either (b) or (c) produces better style transformation as they achieve a very similar style transfer result. This work thus chooses to place similarity layers at relu1_2,relu2_2,relu3_3 in the ST Network, which keeps the same placements as the DR Network.

the DR Network, this work adds content and similarity layers into the pre-trained VGG-16 network, and choose parameters $\{\alpha_f = 5, \alpha_d = 10^3\}$ for the detail reconstruction. For the ST Network, this work adds content, similarity and style layers into the pre-trained VGG-16 network, and choose $\{\beta_f = 5, \beta_d = 10, \beta_s = 100\}$ for the style transformation. This work uses $\sigma_s = 60$ (default in the public source code) and $\sigma_r = 1$ for the edge-preserving filter [34] in SFM. The effect of parameter α_d, β_d and σ_r is illustrated in Fig. 3.7, Fig. 3.8 and Fig. 3.9 respectively. In this chapter, PSNR (Peak Singal-to-Noise Ratio) is chosen as the criterion for setting parameter α_d as it is most commonly used to measure the quality of image reconstruction in literatures.

This work uses a random white noise image X (X_{DR} and X_{ST} represent X for DR Network and for ST Network respectively) with the same size of content image as the initialized input, and choose Adam [75] optimization algorithm with learning rate 1 and iteration 1000 in the optimization process for all the experiments in this chapter. All the inputs including I_c, I_s and X are scaled into 512 when width or height is over 512, otherwise they remain original resolution. The dual-stream convolution networks run the optimization process at the same time, and the optimization time is around 2.5 minutes by running on the GPU card (NVIDIA GeForce GTX 1060, 6G GDDR5). The whole optimization process only needs one content image and one reference style image without any limitation on resolution.

Table 3.1: Additional Layers in the pre-trained VGG-16 Network

Additional Layers	Layers of VGG-16	Size	Activation
	conv1_1	$64 \times 3 \times 3$	relu1_1
similarity,style	conv1_2	$64 \times 3 \times 3$	relu1_2
	Maxpooling	2×2	
	conv2_1	$128 \times 3 \times 3$	relu2_1
similarity,style	conv2_2	$128 \times 3 \times 3$	relu2_2
	Maxpooling	2×2	
	conv3_1	$256 \times 3 \times 3$	relu3_1
	conv3_2	$256 \times 3 \times 3$	relu3_2
content,similarity,style	conv3_3	$256 \times 3 \times 3$	relu3_3
	Maxpooling	2×2	
	conv4_1	$512 \times 3 \times 3$	relu4_1
	conv4_2	$512 \times 3 \times 3$	relu4_2
style	conv4_3	$512 \times 3 \times 3$	relu4_3

Table 3.2: Implementation details of DR Network

Loss	Parameters	Placements in VGG-16
L_{feat}	$\alpha_f = 5$	relu3_3
L_{simi}	$\alpha_d = 10^3$	relu1_2,relu2_2,relu3_3

3.4 Results

This section discusses the selection for hyperparameters, placement for similarity layer, comparisons between the proposed methods and state-of-the-art methods in terms of global and local colour transfer.

3.4.1 The effect of hyperparameters

Fig. 3.7 and Fig. 3.8 demonstrate the effect of parameters α_d and β_d respectively. As shown in Fig. 3.7, the content reconstructed result achieves the highest PSNR (peak signal-to-noise ratio) value when $\alpha_d = 10^3$. This work thus chooses $\alpha_d = 10^3$ to reconstruct content details in the DR Network. In Fig. 3.8, a lower β_d value still produces stylized result with geometric mismatching problem. Conversely, a larger β_d value produces less style result. Hence, this work finds the best value $\beta_d = 10$ to produce the stylized result and all the other results in this chapter. Fig. 3.10 and Fig. 3.11 illustrate the choices of similarity layers in the DR Network and ST Network respectively. For DR Network, this work chooses to place similarity layers at {$relu$1_2, $relu$2_2, $relu$3_3} as it achieves the highest PSNR score. For ST Network, the stylized results (b) and (c) have very similar style transformation appearance, this work thus chooses to place similarity layers at {$relu$1_2, $relu$2_2, $relu$3_3} in the ST Network, which keeps the same placements as the DR Network. The implementation details of the proposed network are described in Table 3.1 to Table 3.3.

Table 3.3: Implementation details of ST Network

Loss	Parameters	Placements in VGG-16
L_{feat}	$\beta_f = 5$	relu3_3
L_{simi}	$\beta_d = 10$	relu1_2,relu2_2,relu3_3
L_{style}	$\beta_s = 100$	relu1_2,relu2_2,relu3_3,relu4_3

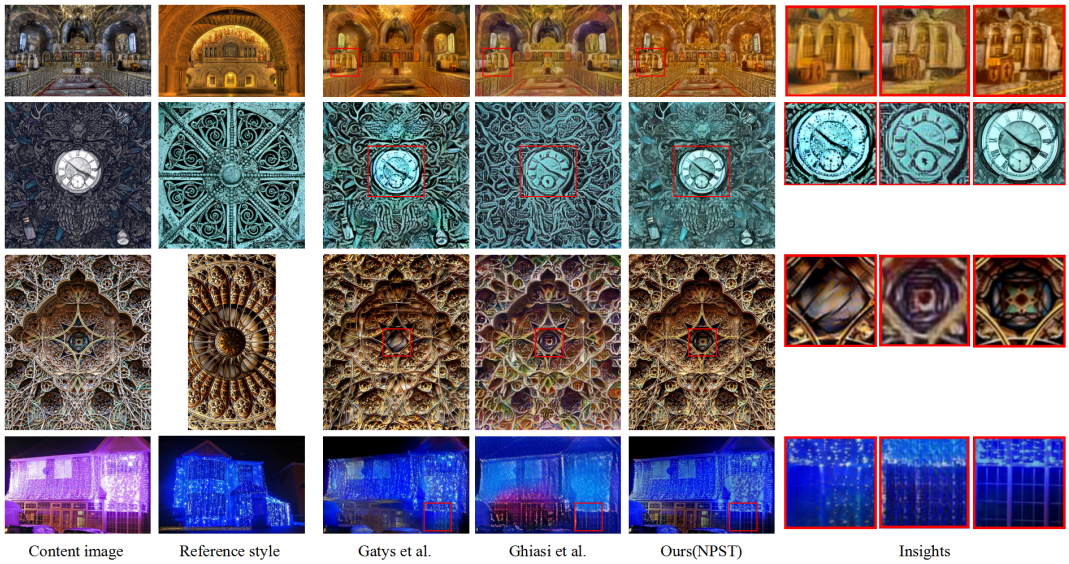


Figure 3.12: Comparison between Gatys et al.[37], Ghiasi et al. [39] and NPST. Gatys et al.[37] and Ghiasi et al. [39] produce a larger amount of distortions in their results while NPST results are free of distortions. The stylized results of Ghisai et al. [39] method use the interpolation weight of 0.8 and other default parameter values in their chapter.

3.4.2 Comparisons to State-of-the-art Works

Comparison between representative artistic style transfer methods and NPST. The compared methods are Gatys et al. [37], Ghiasi et al. [39] and NPST across great differences among content images in Fig. 3.12. NPST results preserve content structures with more precise details than other artistic prior methods. For example, NPST results contains all details of ceiling lamp, frescoes, carpets and railings which are not reconstructed well by Gatys et al. [37] and Ghiasi et al. [39]. To illustrate the ability of preserving precise details, this figure compares content and reference style image with great details to prior artistic style transfer methods in third row. NPST method reconstructs almost every detail in content image and transfer the colour style faithfully while Gayts et al. and Ghiasi et al. [39] lose great details. The detail representations on other examples also show the strong ability of the proposed method to reduce distortions and preserve content spatial structures as well.

Comparison between representative global colour transfer methods and NPST. Fig. 3.13 compares the proposed method with representative global colour transfer algorithms such as Reinhard et al.[121] and Pitié et al. [117]. A global colour mapping technique is

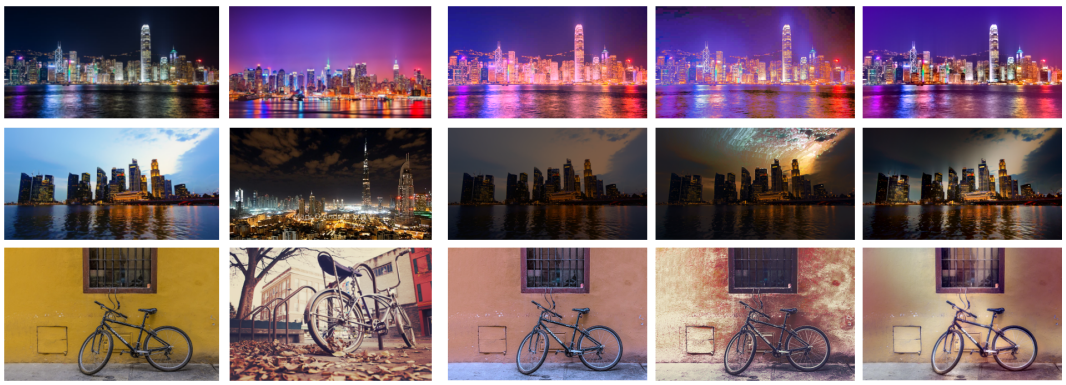


Figure 3.13: Comparison between representative global colour transfer methods Reinhard et al.[121], Pitié et al. [117] and NPST.

applied by both of them to match the colour statistics of content input and reference style image. However, they can not obtain faithful colour transformation results when the inputs contain spatially varying objects, which limits their applications. For example, in the second row of Fig. 3.13, Reinhard et al. and Pitié et al. methods can not transfer light style in reference style image to buildings.

Comparison between representative local photographic style transfer methods and the proposed methods. Fig. 3.14 compares NPST with the state-of-the-art methods, Luan et al. [108] and Liao et al.[103]. The approaches proposed by Luan et al. [108] and Liao et al. [103] are the latest methods which effectively avoids the distortion problem. The proposed method preserves more precise content details than Luan et al. For example, the plants in the first row, the texts on the postcard in the third row and the windows in the bottom row. The proposed method may not obtain better faithful transformation results but NPST method achieves the highest score on the photorealism. Please refer to user study for more details in section 3.4.3. All the stylized results (including user study) of Luan et al. [108] are their best results with manually semantic segmentation mask and parameter $\lambda = 10^4$ (default value in Luan et al.'s paper).

Luan et al. [108] propose a two-stage photo style transfer method which expands Gatys et al.'s artistic style transfer method. Their first stage integrates semantic segmentation into Neural-Style [37] method for object-to-object colour transfer, and their second stage applies a post-processing step using Lapacian Matting to improve the photorealism of stylized result obtained from the first stage. In this chapter, the first stage is called Seg-NS and the second stage is called (Mat-NS). In terms of local object-to-object colour transfer, the proposed similarity loss function may not transfer colour for object-to-object as faithful as manually semantic segmentation. However, the proposed SFM may help Luan et al.'s results avoid the

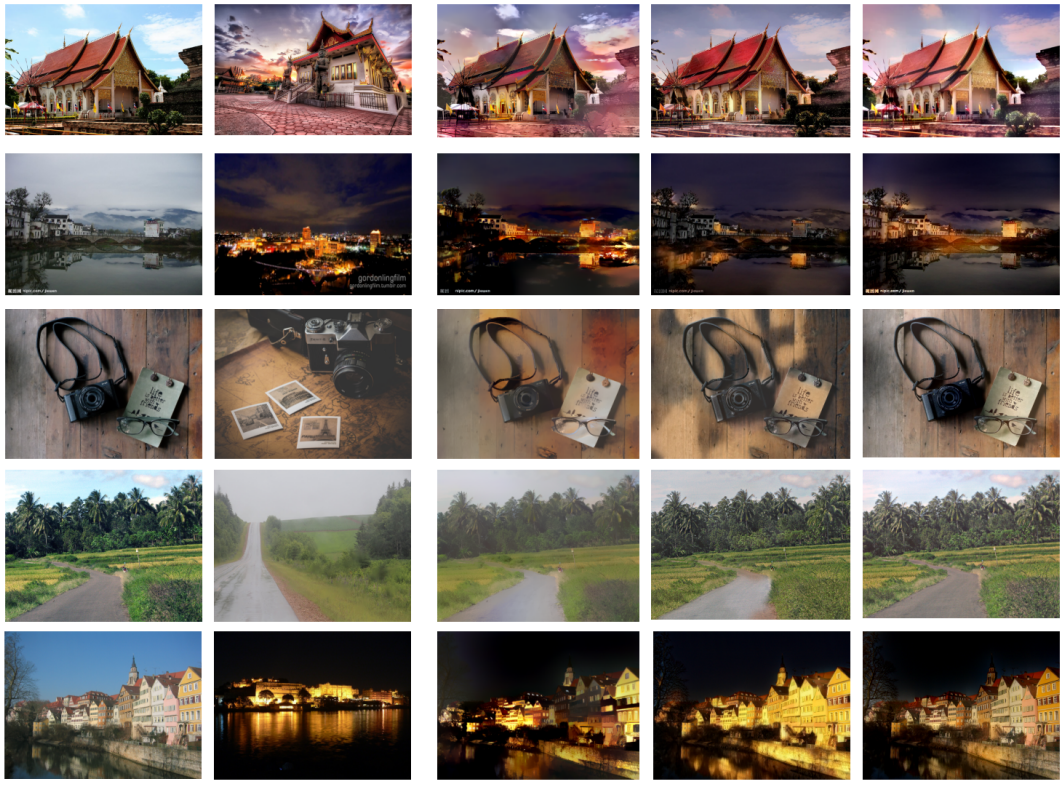


Figure 3.14: Comparison between Luan et al. [108], Liao et al. [103] and NPST. All examples from Luan et al.[108] dataset.

posterization artefacts. Fig. 3.15 shows the stylized results that the SFM is applied to process the results obtained from Luan et al.'s first stage. For example, the proposed method (Seg-NS+SFM) effectively prevents the posterization artefacts on buildings in the first row, water in the second row and forehead in the third row.

Fig. 3.16 compares the proposed method (Seg-NS+SFM) with state-of-the-art neural photographic style transfer methods Luan et al.[108] and Liao et al.[103]. Note that the proposed method (Seg-NS+SFM) preserves more precise content details than Luan et al. [108] while transferring style more faithfully than Liao et al. [103]..

Fig. 3.17 compares the proposed method (Seg-NS+SFM) with Mechrez et al. (Seg-NS+SPE) [111] which proposes to apply Screened Poisson Equation (SPE) [112] to improve the photorealism of result obtained from Luan et al.'s first stage. Note that Mechrez et al. [111] method can not remove the artefacts introduced by Luan et al.'s first stage. For example, the unexpected blue colour and inconsistent colour in the first and third row respectively.

Limitation The proposed method NPST is unable to transfer faithful colour between images which have semantic similarity for human observers but with much complex spatial-varying. Fig. 3.18 shows some failure cases. For example, the blanket and floor in first row fail to be transferred into brown and white colour style.

Figure 3.15: Comparison between Luan et al. [108] and the proposed method(Seg-NS+SFM). The proposed method effectively handles the posterization effect of Luan et al.[108]. All examples from Luan et al.[108] dataset.

3.4.3 User Study

This work conducts a user survey to verify several colour transfer methods on photorealism and style faithfulness. There are six different methods considered in this survey, which include Reinhard et al. [121], Pitié et al. [117] , Luan et al. [108], Liao et al. [103], the proposed methods(NPST, Seg-NS+SFM). This work asks 26 human participants to score stylized results on 1-to-4 scale. These participants are all young people whose age ranges from 20 to 30 years old. As young people use Instagram or Facebook more often than other age groups, thus they are the potential users. For the photorealism, the score ranges from “1: definitely not photorealistic" to “4:definitely photorealistic". For the style faithfulness, the score ranges from “1:definitely not style faithful to reference style" to “4:definitely style faithful to reference style". For each participant, he or she is asked to score the stylized results of 6 methods in a random order. There are totally 44 different scenes (excluding unrealistic and repeated scenes) selected from Luan et al.[108] dataset.

Fig.3.19 shows the average score and standard deviation of each method. For the photorealism, the proposed method (NPST) and Liao et al.[103] rank the 1st and 2nd respectively. Luan et al. [108] and Pitié et al. [117] have the worst performance regarding to the photorealism as their results exist some artefacts. For the style faithfulness, Luan et al. [108] and the proposed method (Seg-NS+SFM) rank 1st and 2nd respectively. The edge-preserving filter [34] used in SFM slightly declines the style faithfulness score of Luan et al. [108] but it still achieves a higher score than Liao et al.[103]. Moreover, it significantly improves the photorealism score of Luan et al.’s results. Reinhard et al. [121] and Pitié et al. [117] perform

Figure 3.16: Comparison between Luan et al. [108], Liao et al. [103] and the proposed method(Seg-NS+SFM). The proposed method preserves finer content details than Luan et al.[108] and transfer style more faithful than Liao et al.[103]. All examples from Luan et al.[108] dataset.

the worst in the style faithfulness as their limitations for sophisticated images.

3.5 Summary

The work in this chapter investigates the reason why the photorealism of stylized results is lost especially when the photographic images are input to Gatys et al.'s method [37]. And this work discovers that both content preservation and style transformation stages in Gatys et al.'s method distort images to lose the photorealistic attribute. Hence, a neural photographic style transfer method is proposed to constrain detail reconstruction and style transformation processes by introducing a similarity loss function. This similarity loss function not only preserves exact details and structures of content image but also mitigates the content-mismatching problem. The qualitative evaluation on Luan et al.'s [108] dataset shows that the proposed approach is capable of preventing the distortions effectively, and obtaining faithful stylized results as well.

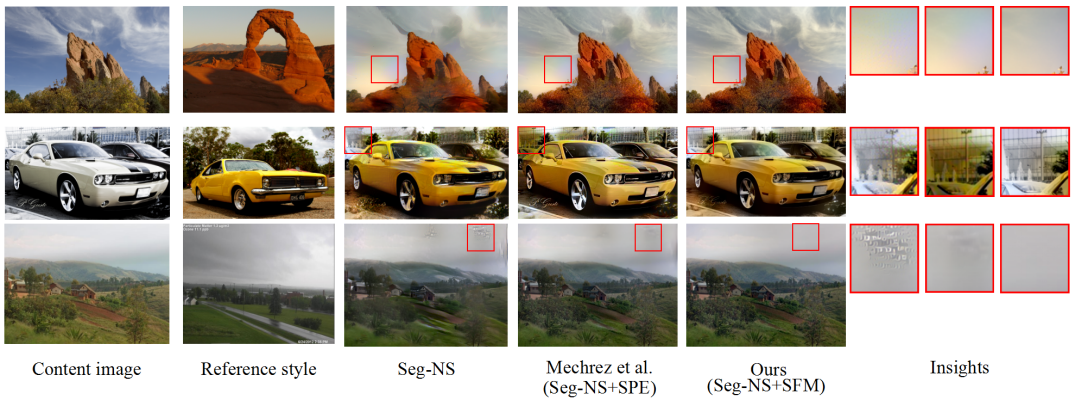


Figure 3.17: Comparison between Mechrez et al. [111] and the proposed method(Seg-NS+SFM). The zoom-ins show the insights of Luan et al.'s first stage output, Mechrez et al. [111] and the proposed method(Seg-NS+SFM) (in that order).

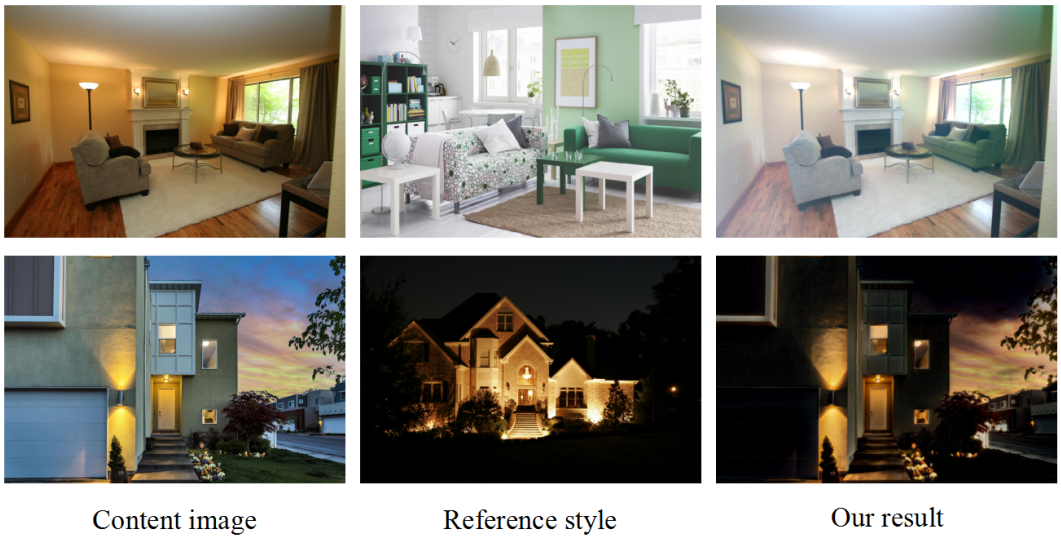
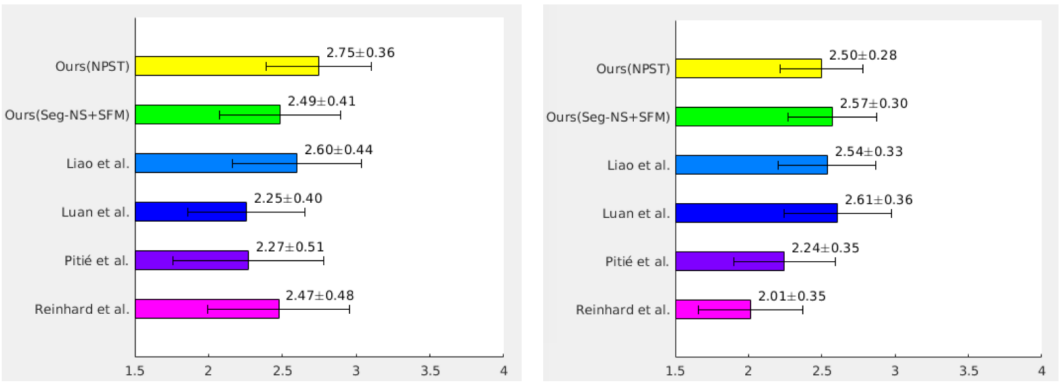


Figure 3.18: Some failure cases.



(a) Photorealism (average scores and standard deviation)

(b) Style faithfulness (average scores and standard deviation)

Figure 3.19: User study results for photorealism and style faithfulness.

Chapter 4

Fast Neural Photo Style Transfer

In this chapter, a new technology Fast Neural Photo Style Transfer (FNPST) is proposed to transform prior neural artistic style transfer methods into photographic style transfer, which improves the photorealism attribute of the stylized results. Without semantic segmentation, FNPST introduces a similarity loss function to solve the content-mismatching problem, and a post-processing technique to further reduce potential distortion and noise artefacts. The similarity loss function reconstructs finer details of content photographs and constrains the content match between reference style and content images. The post-processing refinement technique extracts the colour (without the details) from stylized result, and combines it with the details of content input. Distortion and noise artefacts will be eliminated after the refinement step. Integrating the mentioned above techniques into prior artistic style transformation networks, FNPST achieves nearly real-time performance. This advantage makes the proposed approach a good option for real-time application such as video style transfer.

There are TWO major contributions in this chapter:

1. a technique is proposed to transform representative NAST methods (e.g, Gatys et al. [37] and Johnson et al.[68]) to NPST methods (**CTRB3**).

2. a fast neural photo style transfer method is proposed with near real-time performance (**CTRB4**), which makes it a potential solution for social media apps and video style transfer.

There are two main differences between NPST (Chapter 3) and FNPST:

1. The NPST builds upon the slow NAST method [37] based on optimization network, but FNPST builds upon the fast NAST method [68], which is three orders of magnitude faster than NPST (Answering to **Q3**).

2. Through the extensive experiments, the similarity loss and edge-preserving filter are capable of transforming two representative NAST methods (e.g, Gatys et al. [37] and Johnson et al. [68]) into NPST methods, including slow methods based on optimization networks and fast methods based on feed-forward networks.

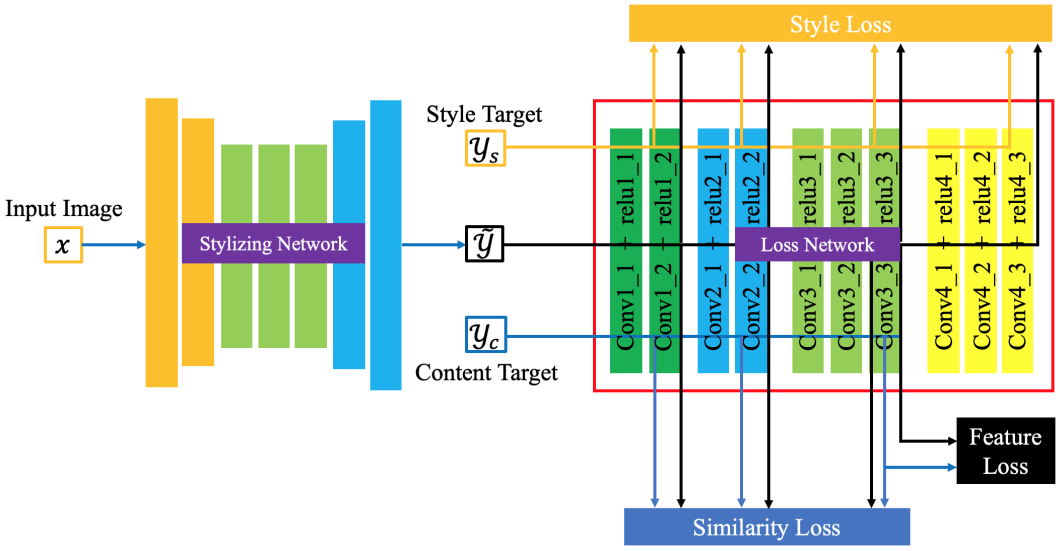
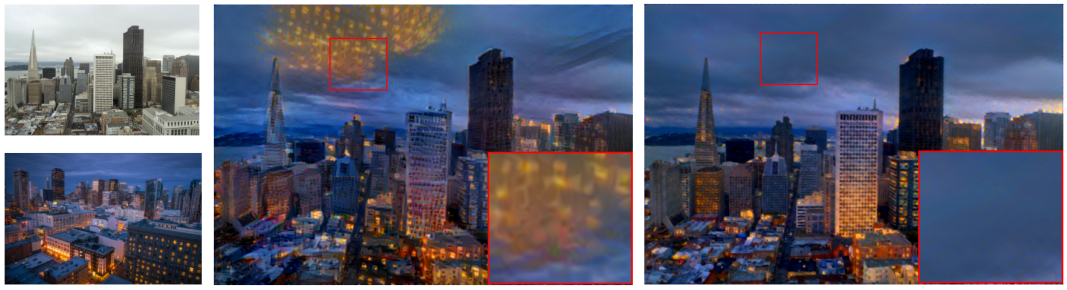


Figure 4.1: Framework overview. The system consists of two components: a Stylizing Network and a Loss Network. Orange, green, black and blue rectangles represent an input image, a style target, an output image and a content target, respectively. The style loss, feature loss and similarity loss are defined on the Loss Network. These losses are used to train the Stylizing Network.

4.1 The FNPST Method

The FNPST method builds upon the work of [68]. Training a feed-forward neural network with a per-pixel loss in a supervision manner is widely used for image transformation tasks such as super resolution [74] and segmentation [107]. Since the proposed method aims to speed up the transformation by a single forward pass, it is natural to use Gatys et al. losses (aka VGG loss below) as supervision to train the feed-forward network. As shown in Fig. 4.1, the basic architecture of th proposed framework consists of two components: an image stylizing feed-forward network $F_W(\cdot)$ and a loss network that is used to define several loss functions $\mathcal{L}_1, ..., \mathcal{L}_k$. Johnson et al. use a deep residual CNN as the image stylizing network, which is parameterized by weights W. The Loss Network is the pre-trained VGG-16 network [136]. The input images $\vec{x}$ are passed through the image stylizing network, and they are transformed into one output image $\tilde{y}$ via the mapping function $\tilde{y} = F_W(\vec{x})$. For each $\vec{x}$, it has a content target y_c and style target y_s. For the loss network, the content target y_c is $\vec{x}$. The training of the image stylizing network pursues weights W which minimizes a weighted total loss function:

$$W = \arg\min_W E_{\vec{x},\{y_c,y_s\}}[\Sigma_{i=1}^{k}\lambda_i\mathcal{L}_i(F_W(\vec{x}), y_c, y_s)] \tag{4.1}$$



Inputs (a) Stylized result with L_{VGG} (b) Stylized result with $L_{VGG} + L_{sim}$

Figure 4.2: The similarity loss function for preventing content-mismatching problem. (a) and (b) are the stylized results through the Loss Network without and with similarity loss respectively. Note that (b) indicates the similarity loss effectively prevents the content-mismatching problem in the stylized result.

4.1.1 Loss Functions for the Loss Network

To clarify the background and improvement, the loss functions in the Loss Network are : $\mathcal{L}_{VGG}$ loss and $\mathcal{L}_{sim}$ loss described in Chapter 3. The Fast-Neural-Style algorithm minimizes the following objective function:

$$\mathcal{L}_{VGG} = \alpha \mathcal{L}_{fea}(\tilde{y}, y_c) + \gamma \mathcal{L}_{style}(\tilde{y}, y_s) \tag{4.2}$$

with:

$$\mathcal{L}_{fea}(\tilde{y}, y_c) = \sum_{j \in J_{fea}} \frac{1}{N_j \times M_j} \|\phi_j(\tilde{y}) - \phi_j(y_c)\|_2^2 \tag{4.3}$$

$$\mathcal{L}_{style}(\tilde{y}, y_s) = \sum_{j \in J_{sty}} \frac{1}{N_j^2} \|\psi_j(\tilde{y}) - \psi_j(y_s)\|_F^2 \tag{4.4}$$

where J_{fea} and J_{sty} denote the set of activation layers in the Loss Network for $\mathcal{L}_{fea}$ and $\mathcal{L}_{style}$, respectively. In each layer, the feature maps have N channels and M size where M is width times height. $\phi_j[\cdot] \in \mathbb{R}^{N_j \times M_j}$ denotes the feature matrix at j-th layer. $\psi_j[\cdot] = \phi_j[\cdot]\phi_j[\cdot]^T \in \mathbb{R}^{N_j \times N_j}$ denotes the Gramian Matrix, which is the inner product between the vectorized feature maps. α and γ denote the weights to feature loss $\mathcal{L}_{fea}$ and style loss $\mathcal{L}_{style}$, respectively. y_c and y_s represent the content target and style target separately.

Overall, the total loss of the Loss Network is given by:

$$\mathcal{L}(\tilde{y}, y_c, y_s) = \alpha \mathcal{L}_{fea}(\tilde{y}, y_c) + \beta \mathcal{L}_{sim}(\tilde{y}, y_c) + \gamma \mathcal{L}_{style}(\tilde{y}, y_s) \tag{4.5}$$

where β denotes the weight of similarity loss $\mathcal{L}_{sim}$, and the effect of $\mathcal{L}_{sim}$ is demonstrated in Fig. 4.2.

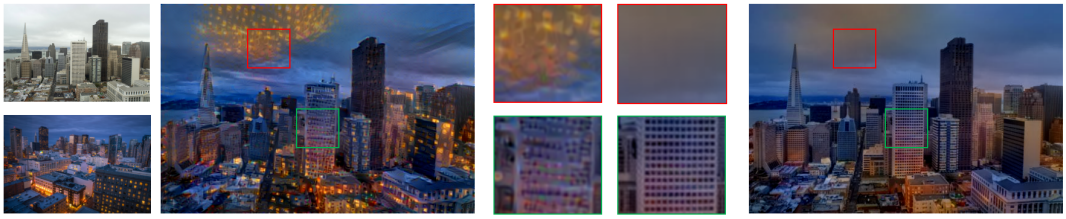


Figure 4.3: The post-processing step can not prevent the content-mismatching problem. In the middle (b), it shows 2 insights of (a) and (c) (in that order). Zoom in to compare results. Note that the stylized result (c) preserves well the spatial structures of building (green rectangle), but it can not prevent the unexpected yellow colour regions (red rectangle) caused by content-mismatching.

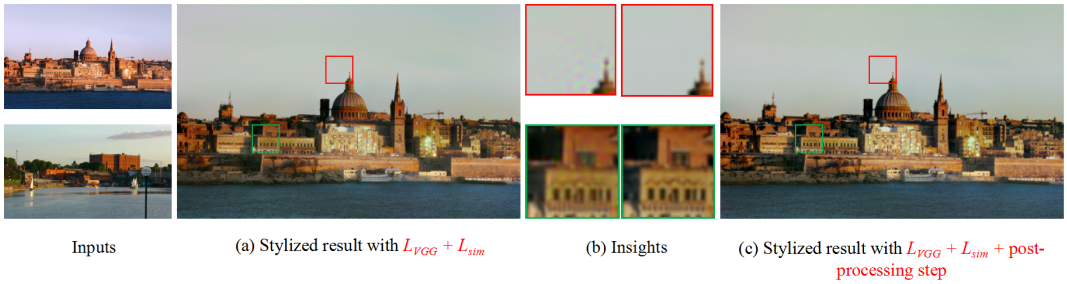


Figure 4.4: The post-processing step reduces the potential distortion and noise artefacts. In the middle (b), it shows 2 insights of (a) and (c) (in that order). Zoom in to compare results. Note that the stylized result (c) prevents the distortion (green rectangle) and noise artefacts (red rectangle), thus exhibits finer details than (a). Examples are from Shih et al. [135]

4.1.2 Post-processing Step

This chapter uses $\mathcal{L}_{sim}$ to avoid the content-mismatching problem, however, the output result may still show distortion and noise artefacts (c.f. Fig. 4.4a).To further reduce the artefacts, a refinement technique SFM described in Chapter 3 is used. To demonstrate the effect of refinement step, Fig. 4.3 shows that post-processing step is not able to prevent the content-mismatching problem without $\mathcal{L}_{sim}$, and Fig. 4.4c shows that the refined result produced by the post-processing step finally reduces the distortion and noise artefacts, and exhibits fine content details.

4.2 Implementation Details

The proposed FNPST method is based on the feed-forward network, which means loss functions are only applied in the training stage, and post-processing refinement step is only applied in the test stage. For the training process, the stylizing network is trained on the MS-COCO dataset [104]. The 80k training images are all resized to 256×256, and the style image is resized to $width = 384$ for 40k iterations using a batch size of 4. The training process has 2 epochs as the dataset contains more than 80,000 images and 2 epochs are enough.

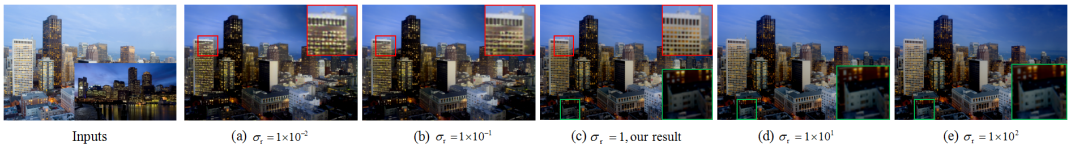


Figure 4.5: Effect of parameter σ_r for Recursion Filter [34] in the post-processing step. The inputs in the left contains content image and style image (bottom right). Note that a small σ_r value does not reduce noise artefacts (red rectangles) in (a) and (b). In contrast, a too large σ_r value does not keep buildings in dark colour (green rectangles) of (d) and (e), while (c) does. Hence, this work uses $\sigma_r = 1$ to produce the result and all the other results in this chapter.

Figure 4.6: The effect of similarity weight β for content-style trade-off. The transformation result $\tilde{y}$ (b) using parameter $\beta = 10$ preserves finer context of content than smaller β value, for example, the left trees (red rectangle) in (b) are reconstructed with finer details than (a). Moreover, (b) remains the white colour gradient style of house (green rectangle) better than (c) and (d). This work conducts a series of experiments with the parameter $\beta = 10$, and obtain almost the same content-style trade-off effect on other images. Hence, this work uses similarity weight $\beta = 10$ to produce the stylized result $\tilde{y}$ and all the other stylized results in this chapter.

This work uses Adam [75] with learning rate 1×10^{-3}, and a total variation regularization with the strength weight 1×10^{-6}. No weight decay or dropout is used because of the model does not overfit within 2 epochs. For all the image stylization experiments, this work adds the similarity layers into {$relu1\_2$,$relu2\_2$. $relu3\_3$} activation layers of the Loss Network. The feature layers and style layers use the default settings of the Fast-Neural-Transfer [68], which are {$relu3\_3$} and {$relu1\_2, relu2\_2, relu3\_3, relu4\_3$} activation layers of the Loss Network respectively. The hyperparameters of the Loss Network are set as $\alpha = 1.0$ and $\beta = 10.0$ for content reconstruction, and $\gamma = 5.0$ for style transformation. The training takes roughly 2 hours on a single NVIDIA GTX 1080 Ti GPU in the implementation of Torch [22] and cuDNN [20]. For the post-processing refinement step, this work uses $\sigma_s = 60$ (default in its open source code) and $\sigma_r = 1$ for the Recursion Filter [34]. The effect of σ_r is illustrated in Fig. 4.5.

4.3 Results

4.3.1 The content-style Trade-off

As shown in Fig. 4.6, different values of parameter β directly affect the content-style trade-off. A small β (*e.g.*, (a)) value reconstructs content details worse than bigger β values. Conversely, a large β value suppresses the style transfer. For example, the bigger β value tends

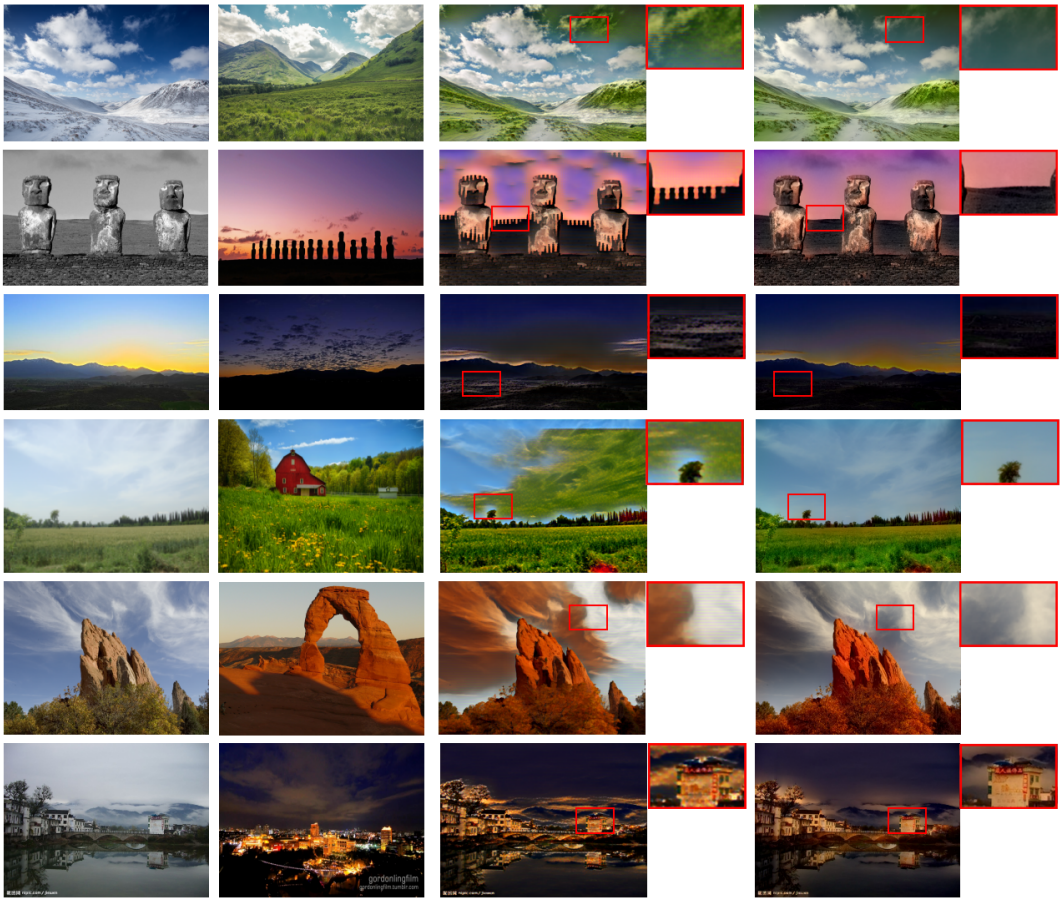


Figure 4.7: Comparison between representative artistic style transfer method [68] and FNPST. All examples are from Luan et al. [108]

to remain the colour of house in (c) and (d) as content image does, which actually should be in white colour just like the style image. Hence, the parameter $\beta = 10$ is used to produce the result and all the other results in this chapter.

4.3.2 Comparison to State-of-the-art Works

This work introduces the similarity loss function and post-processing refinement step into the representative artistic style transfer method [68], and transfer the colour of the style image while improving the photorealism of stylized results.

Comparison with representative artistic style transfer method. Fig. 4.7 compares FNPST to prior representative artistic style transfer network Johnson et al. [68]. The stylized results obtained from [68] method still suffers from the content-mismatching problem, for example, the sky in the first three rows. The proposed method also reconstructs finer content details than the previous works (e.g., the fourth and fifth row).

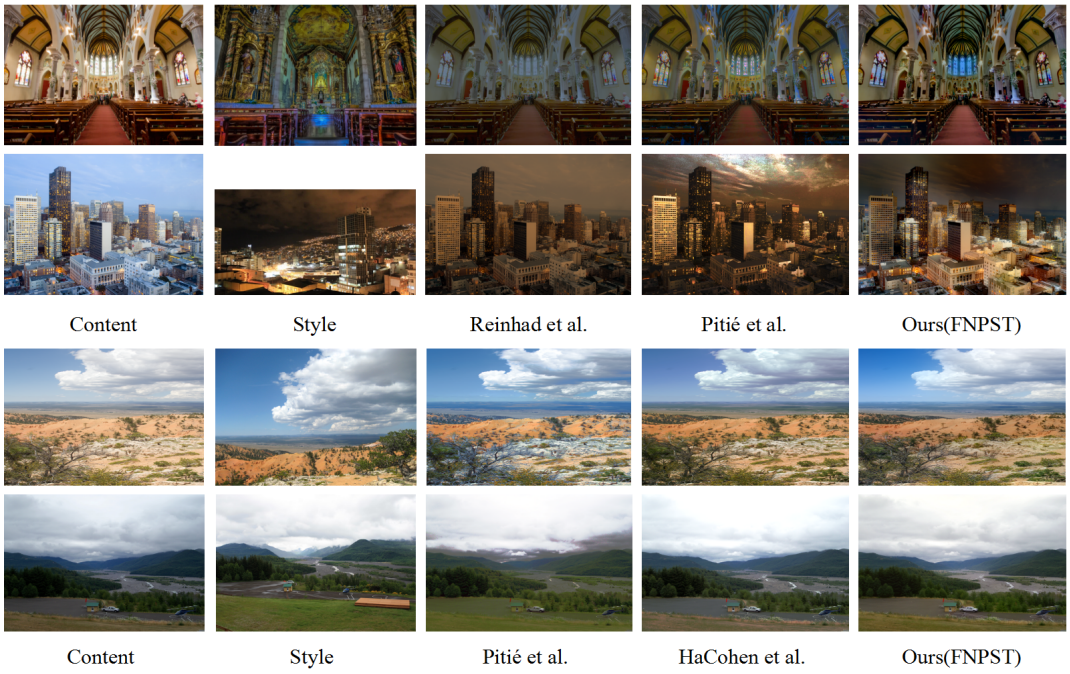


Figure 4.8: Comparison between global colour transfer methods [121], [117], [47] and FNPST. Top two examples are from Luan et al. [108], and bottom two examples are from HaCohen et al. [47].

Comparison with global colour transfer methods. Reinhard et al. [121] and Pitié. et al. [117] are based on the global colour statistics of inputs, which limits their ability to transfer colour between more sophisticated images. For example, in the second row of Fig. 4.8, Reinhard et al. and Pitié et al. fail to render the sky in black to match the colour of sky in the style image. On the contrary, the proposed method is local and capable of handling more semantic colour transfer.

HaCohen et al. [47] propose a NRDC method which relies on a small number of matchable points to estimate the global colour transfer between inputs. Due to this, their method obtains better results than Pitié et al.'s (e.g., branches of trees in the third row in Fig. 4.8). However, their method fails to conduct colour transfer between two different scenes (e.g., top two rows in Fig. 4.8). FNPST method matches colour statistic in different levels of deep feature maps (matching Gramian Matrix at several layers of Loss Network), therefore FNPST results are more accurate than HaCohen et al.'s method. For example, in the fourth row of Fig. 4.8, the sky in FNPST's result preserves the style of reference image better than HaCohen et al.'s, which is too bright in HaCohen et al.'s result. Besides, FNPST's region of grassland covered in green is more accurate than HaCohen et al.'s result.

Comparison of the integration of the proposed method with local photographic style transfer frameworks. Luan et al. [108] propose a two-stage photo style transfer method. Its first stage (Seg-NS) integrates Neural-Style algorithm [37] with semantic segmentation to

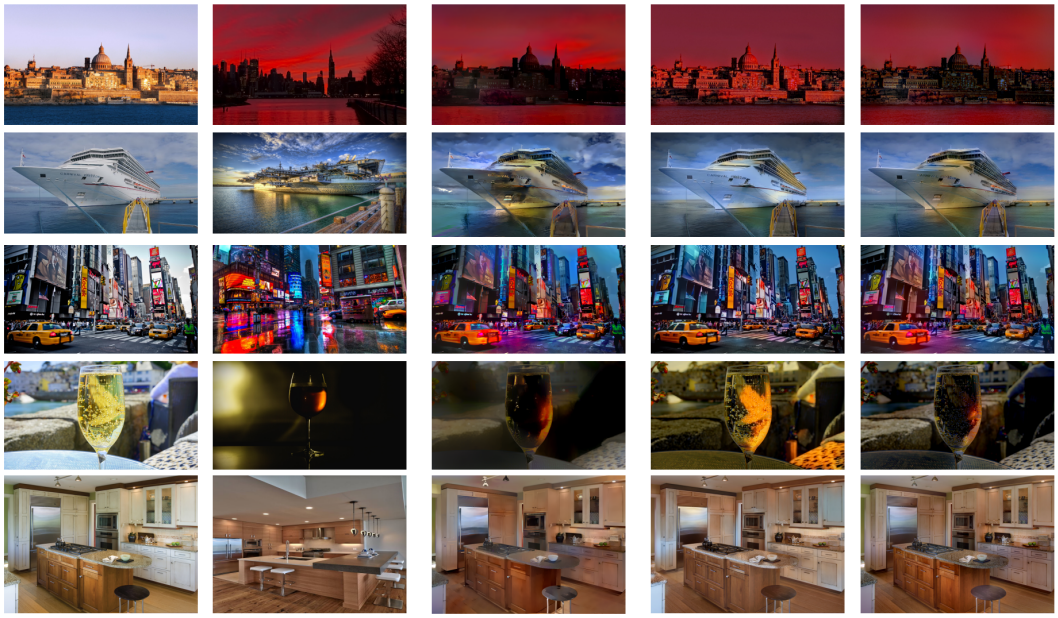


Figure 4.9: Comparison between state-of-the-art style transfer methods based on deep features [108], [103] and the refined results. All examples are from Luan et al. [108].

achieve local object-object colour transfer. The second stage (Mat-NS) attempts to improve the photorealism of stylized results via a post-processing step, which is based on the Laplacian Matting of [89]. Compared to semantic segmentation, the proposed similarity loss function can not achieve such sophisticated object-to-object style transfer as Luan et al.'s method does. However, the proposed post-processing refinement technique further improves the photorealism of stylized result obtained by Luan et al.'s first stage. This work uses the Recursion Filter [34] rather than [89] to refine the stylized results obtained by Neural-Style with semantic segmentation. In Fig. 4.9, it is noticeable that results produced by FNPST obtain finer details than Luan et al.'s results while preserving the style transfer performance. For example, the refined result preserves finer details of the buildings in the first row and bubbles inside the glass in the fourth row. Moreover, the refined results maintains clearly even the characters on bottle bottom in the third row and better boundaries of cupboards in the bottom row. Compared to Liao et al. [103], the refined results achieve more faithful style transfer results. For instance, the FNPST refined results remain the dark colour gradient of style image in the first and fourth row.

Fig. 4.10 compares FNPST to [108] and [103]. The proposed method may not achieve better style transformation performance than them, but it is three orders of magnitude faster than theirs while obtaining similar visual transfer appearance. The detailed comparison of speed performance is described in Section 4.3.3.

Failure Cases Fig. 4.11 shows some examples of failure. Note that the content-mismatching

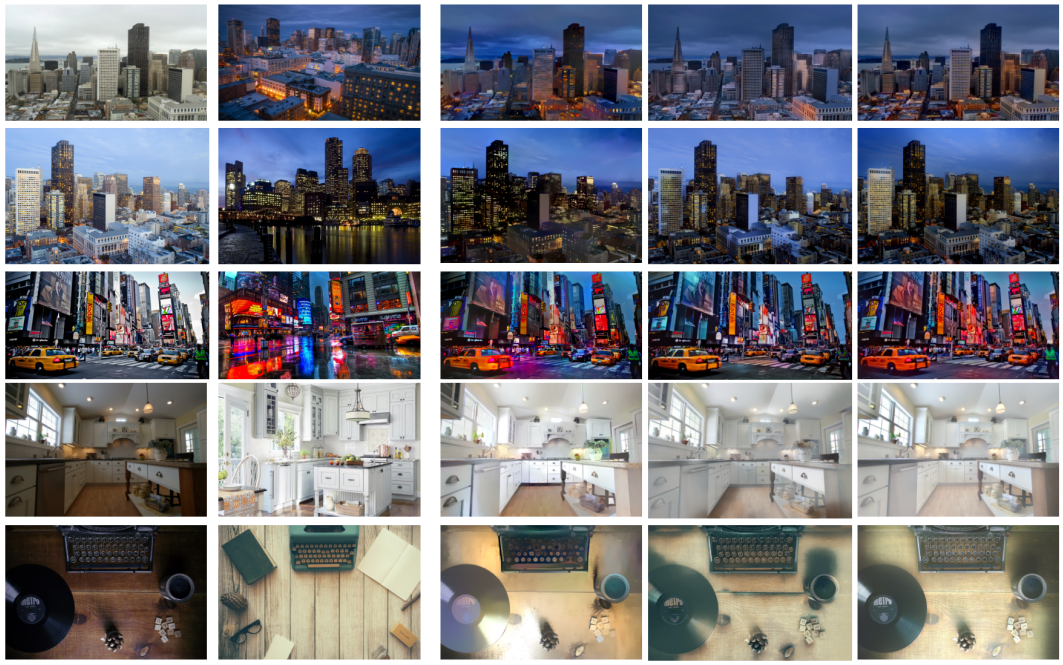


Figure 4.10: Comparison between state-of-the-art style transfer methods Luan et al. [108], Liao et al. [103] and ours (FNPST). All examples are from Luan et al. [108].

Figure 4.11: Some failure cases.

problem may still occur when inputs have very poor content semantic similarity. For example, the kitchen and the nightscape images in the left have big content differences. Additionally, the proposed refinement step may produce stylized results with colour floating artefacts (e.g. apple in Fig. 4.11). This can be fixed by fine-tuning of parameters σ_s and σ_r in the post-processing refinement step.

4.3.3 Speed Performance

Table 4.1 compares the runtime of state-of-the-art methods and FNPST for 256×256 and 512×512 image resolution. The compared methods include Luan et al. [108] and Liao et al. [103]. FNPST uses the Recursion Filter proposed in [34] as post-processing refinement step, and the code provided by the authors is implemented in MATLAB with CPU E5 (3.50GHz). All the runtimes exclude the I/O operation (e.g. write the file into the disk). The runtime

Table 4.1: Speed (in seconds) for the state-of-the-art literature approaches and our method.

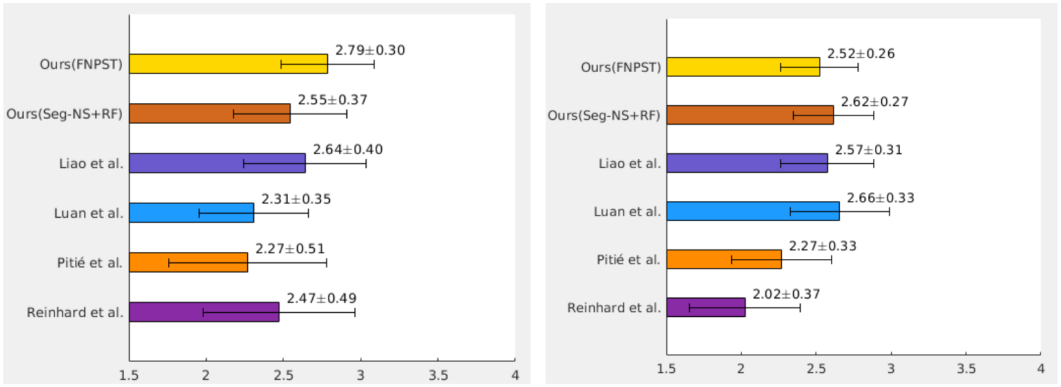
Image size	Literature approaches		Our method	Speedup	
	Luan [108]	Liao [103]	FNPST	Luan [108]	Liao [103]
256×256	108.510	72.358	0.023	4717x	3118x
512×512	342.723	449.833	0.059	5808x	7624x

of FNPST contains two parts: 1. Fast-Neural-Style: around 0.015s and 0.05s for 256×256 and 512×512 resolution, respectively; 2. Post-processing step: around 0.007s and 0.009s for 256×256 and 512×512 resolution, respectively. As listed in Table 4.1, for 256×256 resolution, the proposed method achieves a speed up of approximately 4717 and 3118 compared to Luan et al. [108] and Liao et al. [103], respectively. For 512×512 resolution, the proposed method achieves a speed up of 5808x and 7624x, compared to them, respectively. FNPST processes 512×512 image at approximately 16 FPS, which makes it feasible to run in near real-time or on video.

4.3.4 User Study

A successful photographic stylized image should look natural to a human observer. Therefore, a user survey is conducted to verify FNPST and other four methods. The user survey assesses the photorealism of results and the style faithfulness. There are six methods in total considered in the survey: Reinhard et al. [121], Pitié et al. [117], Luan et al. [108], Liao et al. [103] and the proposed methods (FNPST, Seg-NS+RF). Each result image has been shown to human participants who were asked to score the image from 1 to 4. There were only two simple questions: “Does the picture look photorealistic? " and “Do you think the colour looks like the reference style image". For the first question on photorealism, the score on a 1-to-4 scale ranging from ‘definitely not photorealistic’ to ‘definitely photorealistic’, and only the stylized results were presented to people. For the second question on style faithfulness, the score on a 1-to-4 scale ranging from ‘definitely not ’ to ‘definitely yes’ and only corresponding pairs of the stylized results and style images were presented to people. This work used 40 images from the dataset of [108] excluding unrealistic inputs. This work showed the stylized results to 26 human observers in the survey. This work uses manually semantic segmentation masks provided by Luan et al. for all the results of [108] in this chapter.

The average score and standard deviation of each method is shown in Fig. 4.12. For the photorealism, the proposed FNPST method and Liao et al. [103] rank 1 and 2 respectively regarding to photorealism. Pitié et al.’s method [117] and Luan et al. [108] perform the worst in photorealism, due to some artefacts. For the style faithfulness, Luan et al. [108] achieves



(a) Photorealism scores (b) Style faithfulness scores

Figure 4.12: User study results for photorealism and style faithfulness.

the highest score among the six methods, because this work uses manually semantic segmentation masks for [108]. The proposed refinement step ([34]) slightly reduces the style faithfulness of Luan et al. [108] but still obtains a higher faithfulness score than Liao et al. [103]. Moreover, it significantly improves the photorealism of Luan et al.'s [108] results by avoiding the posterization artefacts. Reinhard et al. [121] and Pitié et al. [117] are the worst in style faithfulness as they are limited to transfer colour for sophisticated images.

4.4 Summary

To improve the photorealism of style transformation results, a similarity loss function and post-processing refinement step is introduced into the existing Neural Artistic Style Transfer networks. The similarity loss function effectively avoids the content-mismatching problem while reconstructing finer content details, and the refinement step reduces the potential distortion and noise artefacts. The introduced techniques can transform prior Neural Artistic Style Transfer methods (e.g., Gatys et al.[37] and Johnson et al.[68]) into Neural Photo Style Transfer approaches. The extensive experiments show that the proposed method obtains finer content details and less artefacts than state-of-the-art methods, and transfers style faithfully. In addition, the proposed approach is capable of processing photographic style transfer in nearly real-time, which makes it a potential solution for video style transfer.

Chapter 5

Fast Coherent Video Style Transfer via Flow Errors Reduction

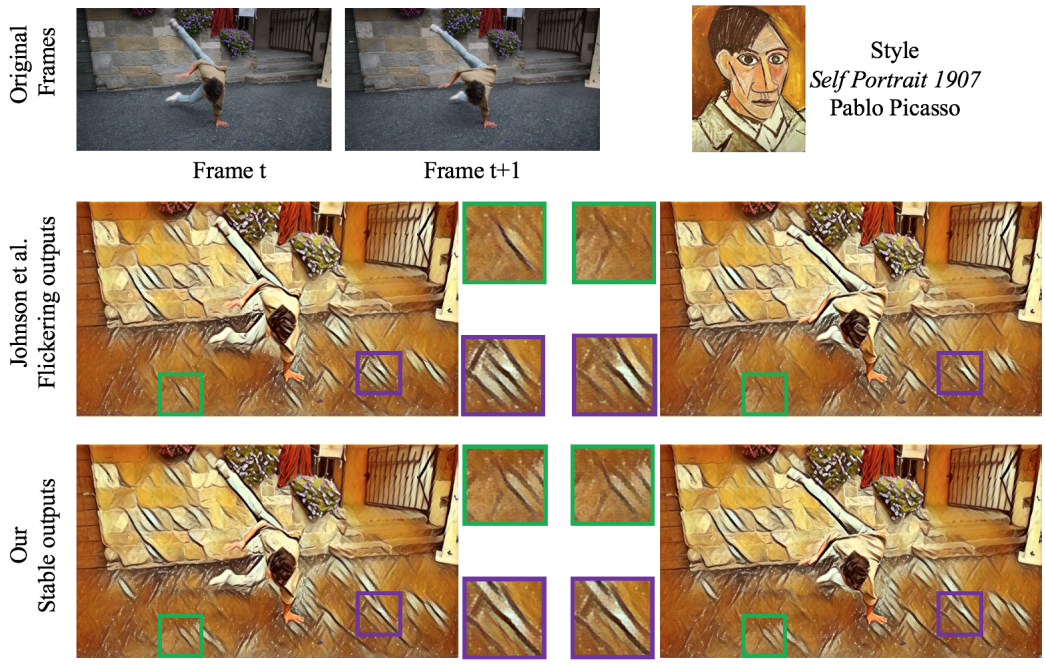


Figure 5.1: Flickering artefacts in video style transfer. The first row shows two original consecutive video frames (left) and the style image (right). The second row shows the flickering stylized results by Johnson et al. [68]. The green rectangles indicate the different appearances (texture and colour) between these two stylized outputs, which exhibit flickering artefacts. The third row shows the stable results by the proposed method, where the outputs preserve the consistent texture appearances.

Chapter 3 and 4 have answered the Questions 1-3, and achieved the Objectives 1-4. To further explore the extension of Neural Style Transfer, this chapter turns its focus onto the NVST field.

Recently, the success of artistic style transfer for still images [37] has inspired a surge of works ([17, 19, 59, 68, 98, 144]) to tackle the style transfer problem and style classification ([21, 54, 166]) task based on the deep correlation features. In the seminal work of artistic style

transfer, Gatys et al. [37] seek to transfer the artistic style of a painting to another photorealistic image by formulating the task into a gradient-based optimization problem. Starting with random white noise, a new image is evolved to present similar spatial structures of a content image and stylistic feature correlations of a painting image. The stylized results are impressive but the heavy optimization process is very slow in run time. To address this issue, Johnson et al. [68] present a speed up solution by introducing an offline feed-forward network. Recently, Chen et al. [19] propose another feed-forward network which swaps arbitrary styles to content images and also gives pleasing results. Chen et al.'s method introduces a patch-based matching technique that replaces the content image patch-by-patch by the style image on neural activations. More recently, Huang et al. [59] propose to replace Chen et al.'s style swap layer with an adaptive instance normalization layer, which is capable of transferring arbitrary artistic style in real-time.

Directly extending these methods to video stylization produces new issues. For example, processing a video sequence via per-frame stylization often leads to flickering and incoherence between adjacent outputs. For optimization-based methods (e.g., [37]), the random initialization and non-convex nature leads to the local minima of the style loss, which causes unstable texture appearances in consecutive frames. For methods based on feed-forward networks ([19, 59, 68, 98, 144]), slight changes of illumination and movements in coherent frames cause large variations in stylized results. Therefore, temporal consistency of consecutive frames in video processing techniques (e.g.,[7, 184]) should be considered for video stylization.

The main contributions of this chapter are: 1. a stable video style transfer method is proposed which can handle large motions and strong occlusions compared to previous feed-forward based methods.(**CTRB5**, achieving **OBJ5**) 2. the proposed technique speeds up the optimization-based video style transfer and is capable of handling arbitrary styles in one network. (**CTRB6**, achieving **OBJ6**)

This framework deals with video style transfer from a new perspective with a new design of initialization strategy which contains novel mask techniques and initialization for the optimization-based network. The new mask techniques significantly reduce flow errors even for large motion or strong occlusion cases (answering to **Q4**), and the new initialization boosts the optimization process (answering to **Q5**) which handles arbitrary styles in one network (answering to **Q6**). To be specific, the proposed approach proposes a set of new mask techniques such as multi-scale scheme, incremental mask and multi-frame mask fusion to prevent the ghosting artefacts in previous optimization-based methods ([5, 124]). The initialization obtained via the proposed mask techniques needs much less iterations to keep

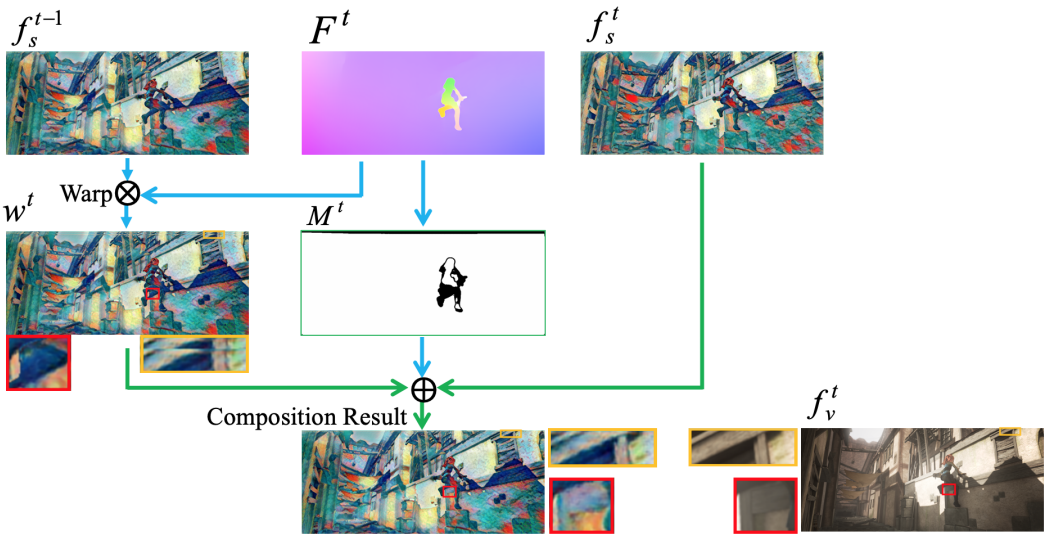


Figure 5.2: Prerequisite. We test the straight-forward idea by recomposing pasted stylized content at flow untraceable regions (see zoom-in rectangles), which preserves well the content consistency. $\otimes$ denotes the warp operation which warps f_s^{t-1} into w^t with F^t here, and $\oplus$ denotes element-wise addition in this chapter.

temporal consistency and image quality. To enhance the temporal consistency, the proposed approach takes both multi-frame RGB-level and Feature-level Coherent Loss into account which outperforms single one of them. To retain the image quality, the Sharpness Losses are proposed to deal with the image blurriness artefacts. In this way, the proposed approach produces coherent video outputs even for large motion or strong occlusion cases and boosts the optimization-based network by two orders of magnitude faster speed.

5.1 Method

5.1.1 Motivation

For artistic video stylization, the unexpected flickering problem causes unsatisfactory results when still image style transfer methods (e.g., [59, 68]) are applied to process frames independently. As shown in Fig. 5.1, the adjacent frames exhibit some colour and texture incoherence (e.g., middle columns in zoom-ins). To preserve the coherency for video style transfer, this work starts with a straight-forward idea.

To simplify, let's start with two consecutive original video frames f_v^{t-1} and f_v^t, and their corresponding per-frame stylized results f_s^{t-1} and f_s^t, and their corresponding optical flow F^t from f_v^t to f_v^{t-1}, then a warped image $w^t = \mathcal{W}(f_s^{t-1}, F^t)$ is produced by warping f_s^{t-1} with F^t and a mask M^t containing per-pixel flow traceable (e.g., values tend to be 1) and untraceable regions (e.g., values tend to be 0). To obtain a stable consecutive stylized result, a straight-forward idea coming-up is to compose the warped image w^t and f_s^t into the flow

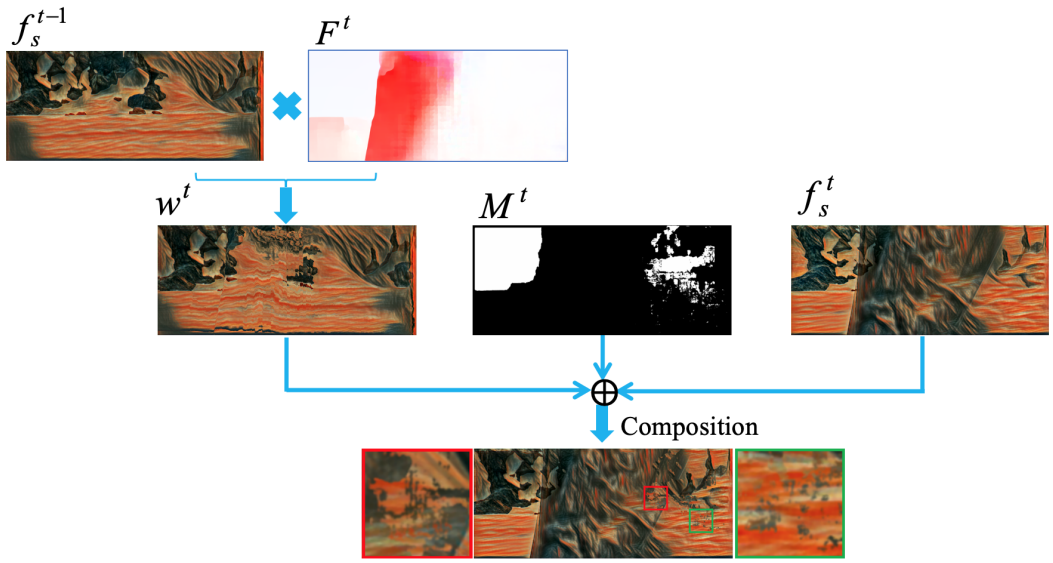


Figure 5.3: Texture Discontinuity Problem. Naively combining w^t and f_s^t via M^t into flow regions causes texture discontinuity problem. For example, in the green rectangle, the gray colours preserved from w^t lose the consistency of texture context (in red and yellow colours) which look like noise artefacts.

Figure 5.4: Image Blurriness Artefacts. Images in the upper rows are original video frames. Along with time step, the blurriness artefacts become more obvious.

traceable and untraceable regions, respectively. In this way, the composition result is capable of preserving coherency as much as possible. However, there is one prerequisite that the pasted contents at untraceable regions must have the exact original content details especially at occlusion regions. Otherwise, a heavy image optimization process is needed during video stabilization. Fortunately, artistic style transfer methods for still images (e.g., [68]) satisfy this prerequisite as they may change the textures or colours on the occluded regions but they indeed do not damage the consistency of original content details. For example, in Fig. 5.2, compared to the original frame, the content details in red and orange rectangles of composition result (both belong to black regions in mask) preserve well the consistency. Hence, this straight-forward idea is worthy of carefully investigating to obtain stabilized video outputs.

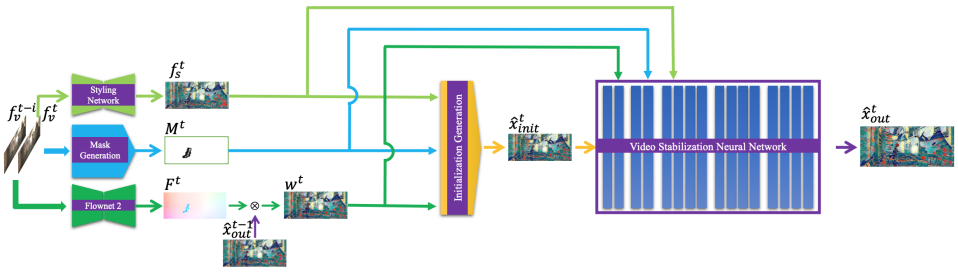


Figure 5.5: System Overview. Starting from three consecutive frames, the proposed system takes corresponding per-frame stylized f_s^t, mask M^t and warped image w^t as inputs, then computes initialization $\hat{x}_{init}^t$ for optimization-based video stabilization network.

Based on this observation, a further investigation is taken and it is found that naively applying this straight-forward composition may produce two new issues. Firstly, it obviously may produce discontinuous transferred textures in the composition result. In fact, this discontinuous textures happen a lot in large motion or strong occlusion cases as there are many small and irregular boundary lines between flow traceable and untraceable regions. For example, in Fig. 5.3, the errors caused by the optical flow method lead to unexpected flow errors in the mask, which directly cause discontinuous textures or colours. Secondly, naively copying and pasting pixels from a warped image w^t into corresponding flow traceable regions, it degenerates image quality and produces blurriness artefacts. For example, in Fig. 5.4, the copied and pasted results will accumulate degeneration errors and produce image blurriness artefacts in eye regions (red rectangles) over a long period.

To address the ***texture discontinuity problem***, a set of new mask techniques are proposed which include multi-scale mask fusion, incremental mask and multi-frame mask fusion. The multi-scale mask fusion is capable of reducing flow untraceable errors, and the incremental mask and multi-frame mask fusion deal with the flow traceable errors. In this way, we obtain a mask with much less errors and compose the warped image w^t and per-frame stylized result f_s^t . The composition image will be the new initialization for optimization-based network to preserve consistency. To reduce the ***image blurriness artefacts***, Perceptual Losses in [68] and a Pixel Loss are adopted as Sharpness Losses to update pixel values iteratively.

The aforementioned techniques and losses only preserve two-frames' coherency. To ensure the coherency in entire video level, both multi-frame RGB-level and Feature-level Coherent Losses are introduced to produce results with averagely lower stability errors than single of them independently [32]. In addition, a recurrent convolutional network strategy [167] is adopted which means that the proposed network takes the current stabilized warped frame $w^t = \mathcal{W}(\hat{x}_{out}^{t-1}, F^t)$ and the current per-frame stylized frame f_s^t as inputs, then produces a stabilized output $\hat{x}_{out}^t$. During the optimization process, Coherent Losses and Sharpness

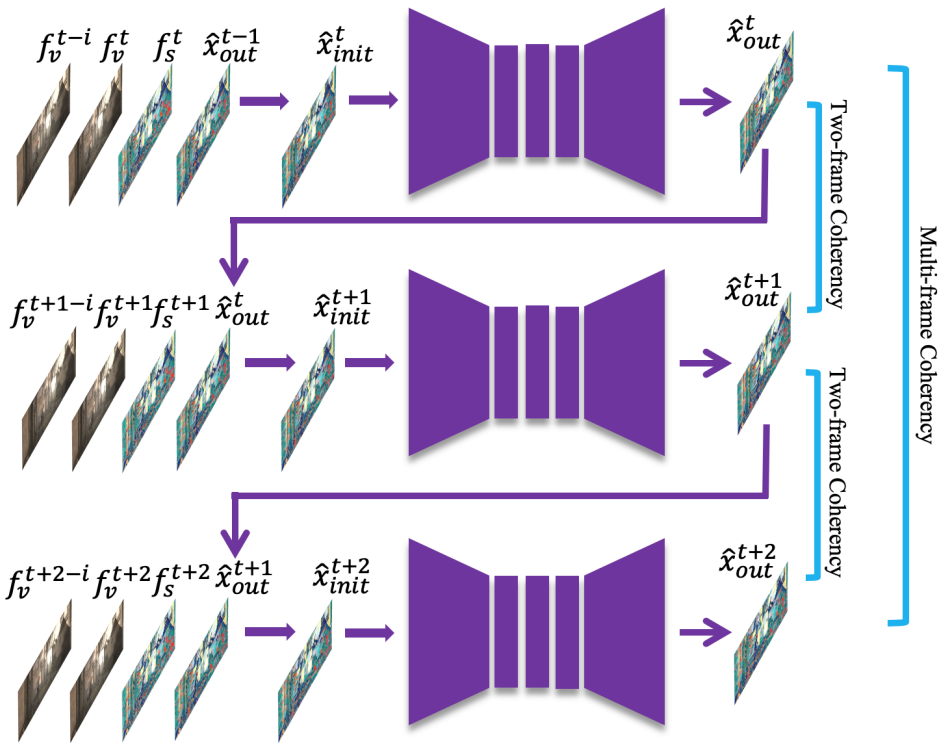


Figure 5.6: Recurrent strategy for video style transfer problem.

Losses are forced to ensure the coherency and image quality between the generated image $\hat{x}^t$ and previous output image $\hat{x}_{out}^{t-1}$. In this manner, the proposed method propagates all the flow traceable points as far as possible during the entire video style transfer process.

5.1.2 Fast Coherent Video Style Transfer

System Outline

Fig. 5.5 shows the overview of the proposed framework. Our method takes original video frames f_v^{t-i} where $i \in \mathcal{T}$ and $\mathcal{T}$ denotes a set of frame indices, f_v^t and per-frame stylized results f_s^t and previous output $\hat{x}_{out}^{t-1}$ as inputs, and produces coherent output video frames $\hat{x}_{out}^t$ where $t \in \{1, ..., N\}$ and N denotes the total number of frames. A mask generation method is developed which consists of a set of techniques (mentioned in Section 5.1.1) to reduce the flow errors, and an initialization generation method is proposed to output a new initial image much closer to final coherent result which speeds up the optimization-based network. Specifically, starting with original video frames f_v^{t-i} ($i \in \mathcal{T}$) and f_v^t in time step t, the proposed method generates a backward flow F^t from time t to $t-1$ using Flownet2 [61], an image w^t that warps previous output image $\hat{x}_{out}^{t-1}$ and F^t, and a mask M^t. Then these three images are warped to obtain an initial image $\hat{x}_{init}^t$ which is fed into the network along with three images above. The optimization process needs much less iterations than

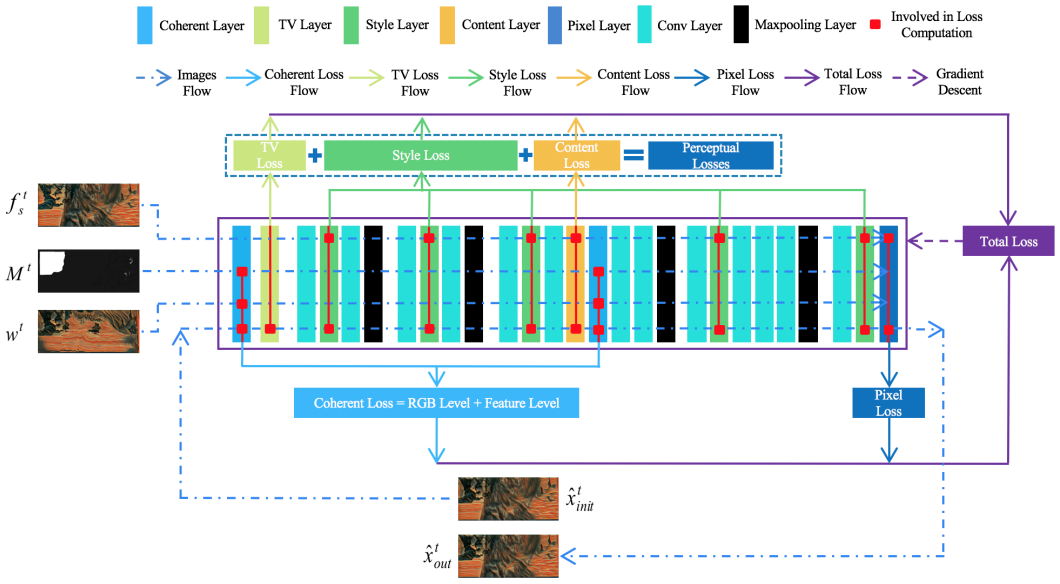


Figure 5.7: Network Architecture Overview. During optimization, the network takes $\widehat{x}^t_{init}$ obtained from Initial Generation, current per-frame stylized result f^t_s, mask M^t and a warped image w^t as inputs, gradually optimizes initial image $\widehat{x}^t_{init}$ into $\widehat{x}^t_{out}$ based on gradients computed from losses. The Perceptual Losses and Pixel Loss is described in Section 5.2.4, and Coherent Losses are described in Section 5.2.5.

previous methods (e.g., [5, 124]) as flow errors have been reduced significantly in the initial image. In order to obtain a long-term coherency, a recurrent strategy is adopted which means the output result $\widehat{x}^t_{out}$ will be fed as input into next time step. Fig. 5.6 shows the recurrent strategy. The short-term coherency between adjacent outputs is propagated into a long-term temporal consistency during the entire video style transfer process. In this way, the proposed method is capable of propagating all the flow traceable points as far as possible. The details of mask generation, initialization generation and optimization-based network will be discussed in following sections.

Network Architecture Overview

Fig. 5.7 shows the details of the proposed optimization-based network. In time step t, there are four total images as inputs passed into the network which are per-frame stylized result f^t_s, mask M^t, warped image w^t and initial image $\widehat{x}^t_{init}$. Coherent Losses force the temporal consistency between adjacent outputs, and Perceptual Losses and Pixel Loss ensures to reduce the image blurriness artefacts. The Coherent Losses contain a RGB-level loss and a Feature-level loss where the first one constrains the mean square error between RGB values of $\widehat{x}^t$ and w^t and second one restricts the mean square error between feature representations of them. The Perceptual Losses force to reduce the differences between $\widehat{x}^t$ and f^t_s in feature domain, and the Pixel Loss intends to minimizes the differences of the RGB values between

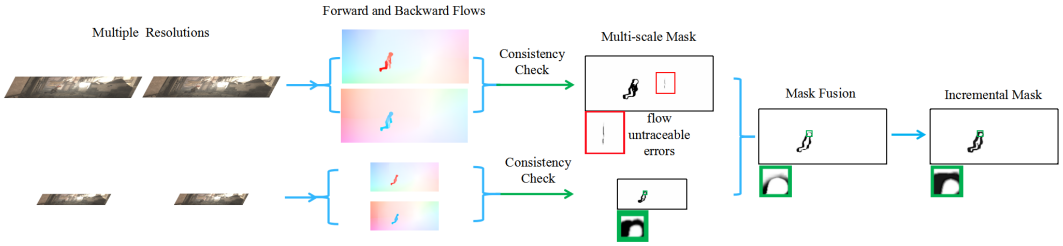


Figure 5.8: The process of multi-scale mask fusion and incremental mask. The unexpected flow untraceable errors are fixed in this step. The fused Mask after multi-scale scheme may cause worse ghosting artefacts as the flow untraceable regions become thinner than before, thus the incremental mask is proposed to thicken the boundaries (see green rectangles).

$\widehat{x}^t$ and f_s^t. During each iteration, the generated image $\widehat{x}^t$ gradually compensates discontinuous texture points and updates features into the entire image. Specifically, the gradients computed from Total Loss are back propagated into the network, and the updated weights and biases inside each CNN layer push $\widehat{x}^t$ to grow into an image with more similarity to inputs. In addition, the proposed optimization-based network is much faster than previous literature methods [5, 124] by using a new initial image $\widehat{x}^t_{init}$. The reason is that the proposed optimization process needs much less iterations than previous methods [5, 124] using w^t, since the initial image $\widehat{x}^t_{init}$ has reduced significantly the flow errors while w^t does not.

5.1.3 A New Initialization for Optimization-based Network

Based on the observation in Section 5.1.1, the most important part of the initialization generation is to create a reliable flow mask reducing flow errors. To this end, a mask generation method is proposed to deal with it. At the beginning, we start with the following items: original adjacent video frames f_v^{t-i} ($i \in \mathcal{T}$ and $\mathcal{T}$ denotes the set of frame indices) and f_v^t, per-frame stylized results f_s^{t-i} and f_s^t. Then the original video frames are rescaled into multiple resolutions. For MPI Sintel dataset [14], let's consider two scales $r \in \mathcal{R}$ where $\mathcal{R} = \{\sigma, \frac{1}{2}\sigma\}$ denotes the set of resolutions and σ denotes the original video resolution. And optical flow methods (e.g., [61]) are utilized to compute the corresponding forward flow $F_{r_f}^t$ and backward flow $F_{r_b}^t$. At this time, a warped image $w^t = \mathcal{W}(f_s^{t-1}, F_{\sigma^b}^t)$ is obtained at original video resolution by warping previous per-frame stylized result f_s^{t-1} and flow $F_{\sigma^b}^t$. Next, the multi-scale per-pixel flow masks are given by a forward-backward consistency check. The values at points of flow masks tend to be 1 at flow traceable regions where both forward and backward direction estimation agrees. On the contrary, the values at positions of flow masks tend to be 0 at disagreed points. Then the flow masks are scaled into original video resolution and composed into one mask in a value maximum manner which remains the maximum values

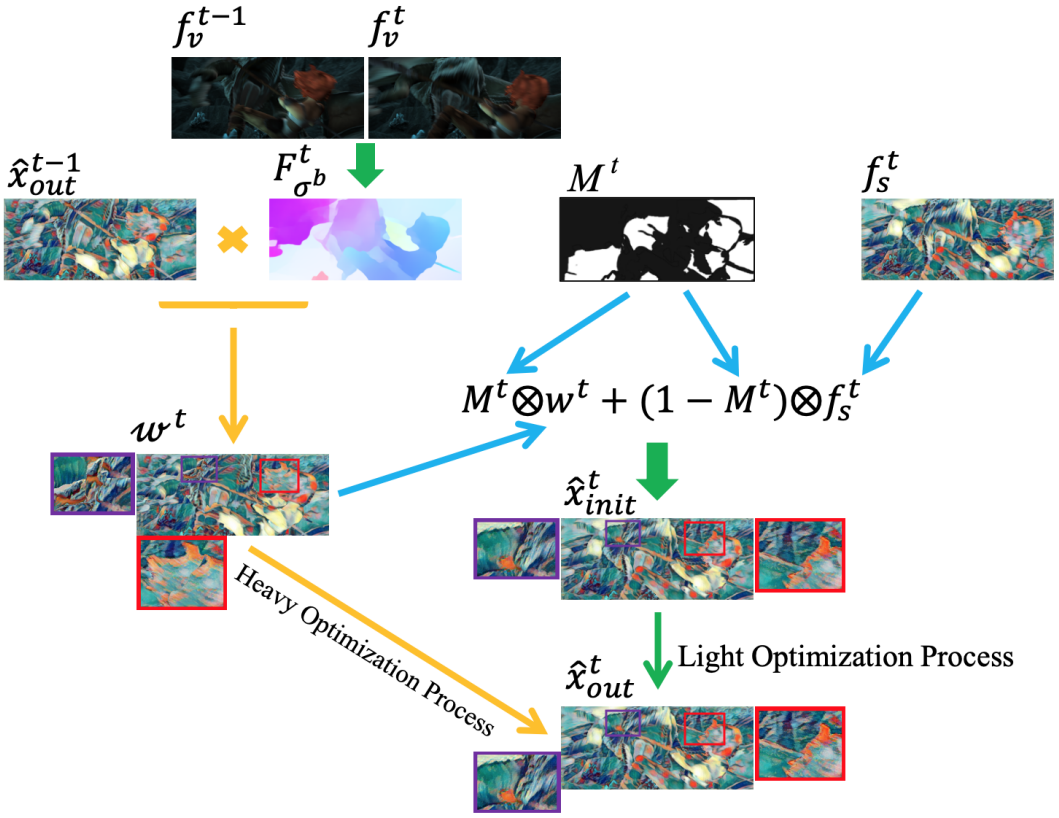


Figure 5.9: Initialization Generation. M^t is a single channel per-pixel mask which is obtained from Mask Generation. Note that the generated $\hat{x}_{init}^t$ contains much less errors than the warped image w^t in purple and red rectangles, which leads to much less iterations to compensate correct pixel values.

from those flow masks at each pixel location. This step is able to fix unexpected flow untraceable errors . It is because untraceable errors in higher resolution, caused by small detail differences (including illumination perturbation) among adjacent frames, will be ignored in lower resolution, and these errors are eliminated by mask fusion using maximum operation. For example, the flow untraceable errors (black regions) in the red rectangle are removed by multi-scale fusion in Fig. 5.8. In this NVST method, it is found that two scales above are enough for removing flow untraceable errors, as a larger scale like $\frac{3}{2}\sigma$ could introduce more untraceable errors due to enlarged illumination differences, and a smaller scale like $\frac{1}{4}\sigma$ could ignore too much traceable points due to lost details. In addition, it is found that copying the current per-frame stylized results into corresponding flow untraceable regions may cause worse ghosting artefacts as the flow untraceable regions become much thinner than before multi-scale fusion. Hence, an incremental mask $M_\theta^{t=>t-1}$ is proposed to generate an incremental circle along with flow untraceable regions. Specifically, the points in the circle closer to untraceable regions has lower values, and $\theta = \{w, g\}$ in $M_\theta^{t=>t-1}$ where w denotes the circle width and g denotes the gradient. In this work, the circle width is set by default to 3 pixels and the gradient is 0.2. To further reduce flow traceable errors where large motions

occur, multiple incremental masks $M_\theta^{t=>t-i}$ $(i \in \mathcal{T})$ are combined together where $\mathcal{T}$ denotes the set of indices of adjacent video frames. The $M_\theta^{t=>t-i}$ $(i \in \mathcal{T})$ are combined in a value minimum manner to correct errors. In general, the fusion in a maximum manner reduces flow untraceable errors (black regions, see Fig. 5.8) and the fusion in a minimum manner reduces flow traceable errors (white regions). The entire mask generation process is shown in Fig. 5.10 in Section 5.3.1.

Eventually, a flow mask $M^t = \min(M_\theta^{t=>t-i}), (i \in \mathcal{T})$ is used for initialization generation by composing the warped image w^t and per-frame stylized result f_s^t. The generation of initial image is shown in Fig. 5.9. The proposed initial image is defined as:

$$\widehat{x}_{init}^t = M^t \otimes w^t + (1 - M^t) \otimes f_s^t \tag{5.1}$$

where $\otimes$ denotes element-wise multiplication. M^t is a single channel mask. $\widehat{x}_{init}^1$ is the first per-frame stylized result f_s^1 when $t = 1$.

5.1.4 Loss Functions for Image Sharpness

Over a long period video processing, especially for time-lapse videos, some points in frames are propagated from the beginning to the end and the copied pixel values gradually lose their quality, which results in the loss of image quality. To prevent the image degeneration, the Perceptual Losses [68] and a Pixel Loss are adopted into the proposed network. The Perceptual Losses constrains the differences between high-level feature representations of the generated image $\widehat{x}^t$ and the current per-frame stylized result f_s^t. The Pixel Loss preserves the pixel values between the generated image $\widehat{x}^t$ and f_s^t in RGB domain.

The Perceptual Losses contain a Content Loss $\mathcal{L}_{con}(\widehat{x}^t, f_s^t)$, Style Loss $\mathcal{L}_{sty}(\widehat{x}^t, f_s^t)$ and Total Variation Regularization $\mathcal{L}_{tv}(\widehat{x}^t)$, which can be formulated as following:

$$\mathcal{L}_{perce}(\widehat{x}^t, f_s^t) = \alpha \mathcal{L}_{con}(\widehat{x}^t, f_s^t) + \beta \mathcal{L}_{sty}(\widehat{x}^t, f_s^t) + \gamma \mathcal{L}_{tv} \tag{5.2}$$

where α, β and γ are the weights of three loss terms, respectively. In experiments, the ratio of α/β close to 0.3 produces better image quality, and $\gamma = 1e-3$ (default in [68]) is set for all experiments. The Pixel Loss is defined as the mean square error between the generated image $\widehat{x}^t$ and current stylized result f_s^t:

$$\mathcal{L}_{pixel}(\widehat{x}^t, f_s^t) = \frac{1}{D} \sum_{i,j}^{D} (\widehat{x}_{(i,j)}^t - f_{s(i,j)}^t)^2 \tag{5.3}$$

where $D = N \times H \times W$ denotes the total pixel number of the input image and $H \times W$ denotes the height times width. In this chapter, the Perceptual Losses and Pixel Loss are referred as Sharpness Losses which ensure the image sharpness during the entire video style transfer process. The Sharpness Losses are the combination of Perceptual Losses and Pixel Loss, which is defined as :

$$\mathcal{L}_{sharpness} = \mathcal{L}_{perce} + \kappa \mathcal{L}_{pixel} \tag{5.4}$$

where κ denotes the weight for Pixel Loss.

5.1.5 Loss Functions for Temporal Consistency

RGB-level Coherent Loss

The flickering artefact is actually presented by texture and colour discontinuities at RGB-level regions between consecutive frames, such as disoccluded regions and motion boundaries. Pixel values in these areas change in adjacent frames, and the optimizer [37] or feed-forward network [68] transforms them differently in a particular style as well. To detect these disoccluded regions and motion boundaries, optical flow methods (e.g., Flownet2 [61]) are applied to estimate these flow traceable areas between coherent frames. Let f_s^{t-1} and f_s^t denote two adjacent per-frame stylized results, $\widehat{x}_{out}^{t-1}$ denote the previous output, $\mathcal{W}(\cdot)$ denote the function to warp image, and w^t denote the warped image using previous output image $\widehat{x}_{out}^{t-1}$ and the optical flow F^t from f_s^t to f_s^{t-1} (backward direction). The warped image w^t is then given by:

$$w^t = \mathcal{W}(\widehat{x}_{out}^{t-1}, F^t) \tag{5.5}$$

In [5, 124], the coherent loss function is supposed to preserve pixel values of the flow traceable regions in the stabilized outputs, and the flow errors in w^t are then rebuilt by style transfer process. The straight-forward two-frame temporal coherency loss considers the consistency between two adjacent frames, thus the ***two-frame RGB-level Coherent Loss*** is denoted as the mean squared error between generated image $\widehat{x}^t$ and w^t:

$$\mathcal{L}_{two}^{RGB}(\widehat{x}^t, w^t, M^t) = \frac{1}{D} \sum_{i=1}^{D} M_i^t \cdot (\widehat{x}_i^t - w_i^t)^2 \tag{5.6}$$

where $D = N \times H \times W$ denotes the dimensionality of $\widehat{x}^t$ and w^t. N denotes the number of image channel and $H \times W$ is height times width, and M^t denotes the per-pixel flow mask with weights of the coherent loss. This $\mathcal{L}_{two}^{RGB}$ considers consistency between only two adjacent frames, which causes small errors as the proposed initial image utilizes a mask operating on multiple consecutive frames. To further enhance the coherency, the consistency

between more adjacent frames is taken into account. Let us consider a multi-frame coherency between several adjacent frames, and let $\mathcal{T}$ (same as Section 5.1.3) denote the set of indices of video frames which are considered as relative frames. For instance, $\mathcal{T} = \{1, 2, 3\}$ denotes that processing frame $\widehat{x}^t$ considers coherency between frame f_s^t and frame f_s^{t-1}, frame f_s^t and frame f_s^{t-2}, frame f_s^t and frame f_s^{t-3}. Then the ***multi-frame RGB-level Coherent Loss*** is defined as the combination of three two-frame RGB-level coherent loss $\mathcal{L}_{wo}$:

$$\mathcal{L}_{mul}^{RGB}(\widehat{x}^t, w^{t-\mathcal{T}}) = \sum_{i \in \mathcal{T}: t>i} \mathcal{L}_{two}^{RGB}(\widehat{x}^t, w^{t-i}, M^{t-i}) \tag{5.7}$$

Feature-level Coherent Loss

In previous methods ([46, 57, 125]), RGB-level coherent loss is considered to constrain the consistency between pixel values of the warped image w^t and generated $\widehat{x}^t$. However, it may not be accurate to preserve stylized texture consistency since their methods do not concern the temporal consistency of feature representations in CNN layers. Hence, a feature-level coherent loss [32] is adopted for preserving texture consistency which is capable of constraining the feature consistency in high-level CNN layers. Let $\psi^l(\widehat{x}^t) \in \mathbb{R}^{N_l \times H_l \times W_l}$ and $\psi^l(w^t) \in \mathbb{R}^{N_l \times H_l \times W_l}$ denote the feature representations of generated image $\widehat{x}^t$ and warped image w^t at layer l respectively, and M^t denote the per-pixel mask, then the ***two-frame Feature-level Coherent Loss*** for two frames is defined as the mean squared error between $\psi^l(\widehat{x}^t)$ and $\psi^l(w^t)$:

$$\mathcal{L}_{two}^{fea}(\widehat{x}^t, w^t, M^t) = \sum_{l \in L_{coh}^{fea}} \frac{1}{N_l} \sum_{i}^{N_l} M_i^t \cdot (\psi_i^l(\widehat{x}^t) - \psi_i^l(w^t))^2 \tag{5.8}$$

where L_{coh}^{fea} denotes the set of layers computing Feature-level Coherent Loss and N_l denotes the dimensionality of feature representations $\psi^l(\cdot)$. Similar to RGB-level Coherent Loss, the feature-level coherent loss between more adjacent frames (same $\mathcal{T}$ in $\mathcal{L}_{mul}^{RGB}$) is also considered, and the ***multi-frame Feature-level Coherent Loss*** term is denoted as:

$$\mathcal{L}_{mul}^{fea}(\widehat{x}^t, w^{t-\mathcal{T}}) = \sum_{i \in \mathcal{T}: t>i} \mathcal{L}_{two}^{fea}(\widehat{x}^t, w^{t-i}, M^{t-i}) \tag{5.9}$$

The total multi-frame Coherent Losses are defined as the combination of $\mathcal{L}_{mul}^{RGB}$ and $\mathcal{L}_{mul}^{fea}$:

$$\mathcal{L}_{coherent} = \lambda_{coh}^{RGB} \mathcal{L}_{mul}^{RGB} + \lambda_{coh}^{fea} \mathcal{L}_{mul}^{fea} \tag{5.10}$$

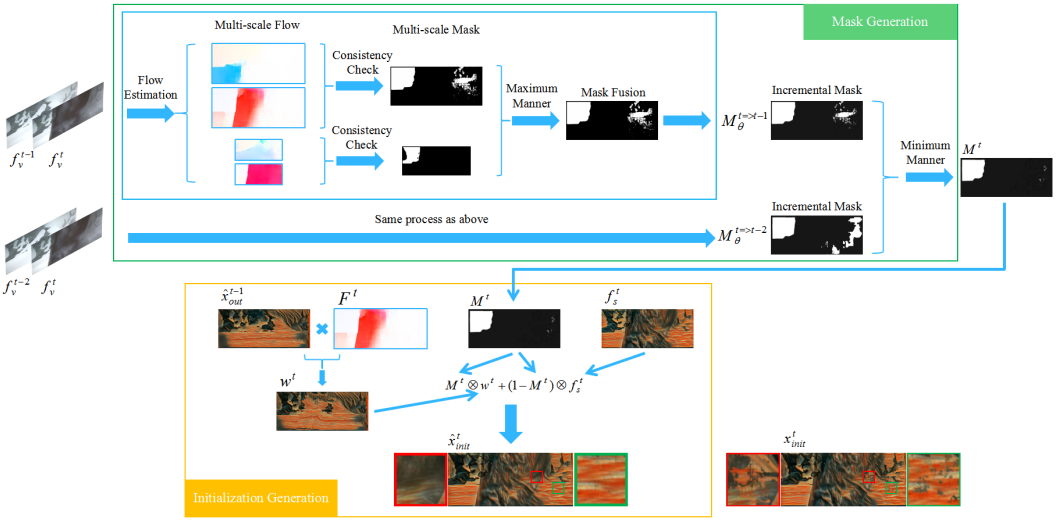


Figure 5.10: The decrease of flow traceable errors (white regions in the right side) by using the proposed initialization. The rectangles indicate the error difference between the initialization $\hat{x}^t_{init}$ and x^t_{init} without the mask generation. The fusion in a maximum/minimum value manner indicates that the maximum/minimum values from those masks are remained at each pixel location.

where λ^{RGB}_{coh} and λ^{fea}_{coh} are the weights to corresponding terms. It is found that the ratio of $\lambda^{RGB}_{coh}/\lambda^{fea}_{coh}$ close to 2.5 makes a better temporal consistency preservation.

Overall Loss The overall loss term for optimization process in each time step is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{sharpness} + \mathcal{L}_{coherent} \tag{5.11}$$

5.2 Implementation Details

The proposed network is developed on a Torch implementation called artistic video style transfer [124]. The chosen layers for losses are: $\{relu1\_1, relu2\_1, relu3\_1, relu4\_1, relu5\_1\}$ layers for Style Loss and $\{relu3\_2\}$ for Content Loss in Perceptual Losses, and $\{relu3\_2\}$ for Feature-Level Coherent Loss. The optimization algorithm for iterations is L-BFGS. The following inputs are fed into the proposed network: per-frame stylized image f^t_s as feature and style target, warped image w^t_{t-i} $(i \in \mathcal{T})$ as temporal consistency target, mask M^t as per-pixel flow weight and the generated $\hat{x}^t_{init}$ (Equation 5.1) as initial image. The stopping criterion: the optimization is considered to be converged when the total loss does not change by more than 1 during 10 iterations. For videos at 1024×436 (MPI Sintel dataset) and 854×480 (Davis 2017 dataset) resolution, the hyperparameters are chosen as followings: $\alpha = 3e1$, $\beta = 9e1$, $\kappa = 9e-7$, $\lambda^{RGB}_{coh} = 5e1$, $\lambda^{fea}_{coh} = 2e1$, $\mathcal{T} = \{1,2\}$. In experiments, the ratio of α/β close to 0.3 preserves better stylistic texture appearance of per-frame stylized image f_s onto outputs. A higher ratio will generate results with more sharpness but less stylistic textures,

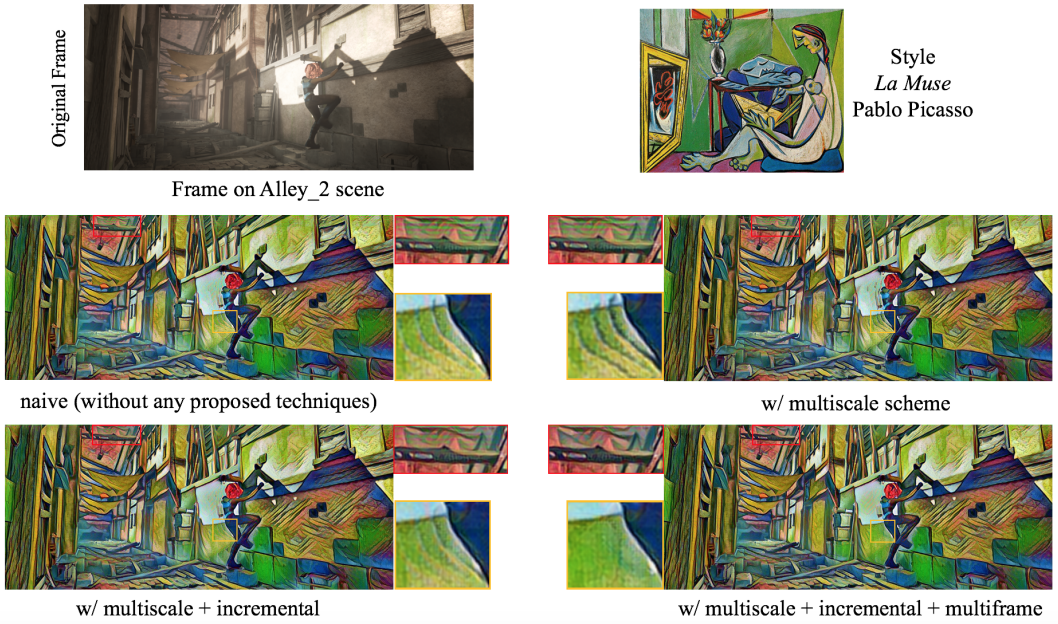


Figure 5.11: Qualitative ablation study on proposed mask techniques of Alley_2 scene from MPI Sintel dataset [14]. The naive method using the flow mask [124] causes ghosting artefacts (see unexpected grid and curved lines in red and orange rectangles). The multi-scale scheme causes worse ghosting artefacts (more obvious grid and curved lines). By gradually adding the incremental mask and multi-frame mask fusion techniques, the unexpected grids and curves are effectively mitigated and produces better visual quality without ghosting artefacts.

in contrary, a lower ratio degenerates image quality due to excessive stylization. The coherent ratio of $\lambda_{coh}^{RGB}/\lambda_{coh}^{fea}$ close to 2.5 preserves better balance between execution time and temporal consistency. A higher coherent ratio warps f_s^t on RGB-level may make pixel errors accumulate along with propagation, which costs more time of network to correct them. While a lower coherent ratio tends to lose the stylization consistency on the same object among consecutive frames. In addition, κ close to $9e-7$ keeps a better balance between image quality and temporal consistency. For example, larger κ values tend to produce outputs with better image quality (e.g., pixels updated more frequently) but poor temporal consistency as pixels propagated from the beginning are lost along with time. Smaller κ values, to some extent, fail effectively mitigate blurriness artefacts but preserve better temporal consistency. In this chapter, the aforementioned hyperparameters values are chosen for all the testing videos. The optical flow used in this chapter is Flownet2 [61], but Deepflow2 [157] are also supported.

Speed Compared to optimization-based methods [5, 124] (3-5 minutes per frame), the proposed optimization process takes around **1.8 seconds** per frame for resolution 1024×436 and around **1.6 seconds** per frame for resolution 854×480 on a single NVIDIA GTX 1080 Ti graphics card. The reason of fast speed is that the proposed network needs much less iterations which are already enough for temporal consistency and image sharpness.

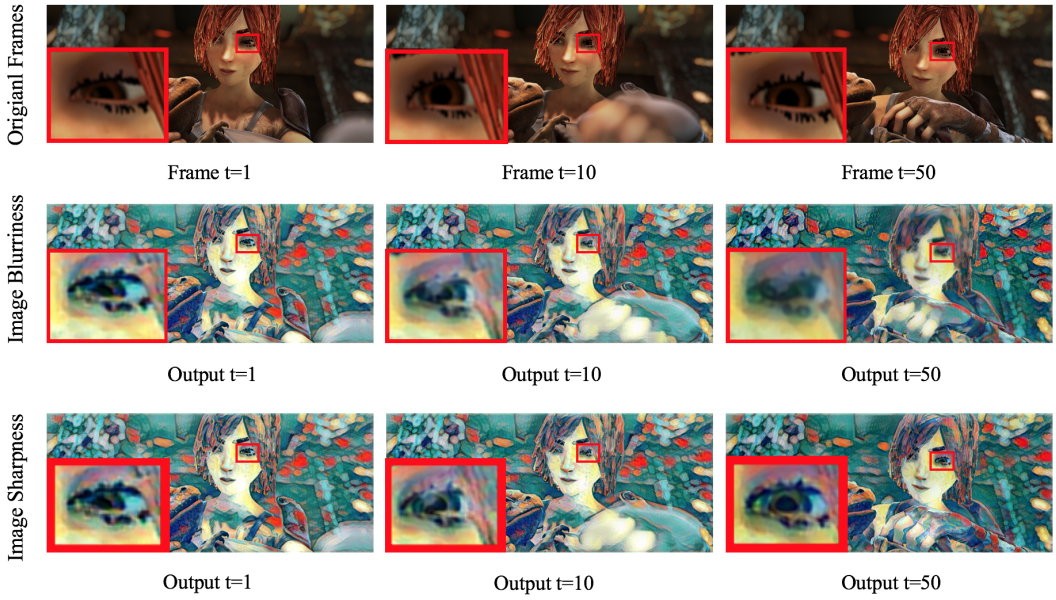


Figure 5.12: The effect of image sharpness. The top rows are original video frames, the middle rows are outputs without Sharpness Losses, and the bottom rows are outputs with Sharpness Losses. The red rectangles indicate the difference of image sharpness.

5.3 Experiments

5.3.1 Qualitative Evaluation

Analysis of Initialization

Fig. 5.8 in Section 5.1.3 shows the mask generation is capable of reducing flow untraceable errors (see rectangle). In this section, the capability of reducing flow traceable errors (ghosting artefacts) is analyzed. Fig. 5.10 shows the effect of the proposed initialization on reduction of flow traceable errors. Starting from three adjacent original video frames, the Mask Generation outputs a mask with much less flow traceable errors (white regions in the right side) than previous single scale mask. This directly helps the composed initial image contain more consistent textures (see rectangles) than x^t_{init} in Fig. 5.3 in Section 5.1.1. The left small discontinuous textures are then fixed by the optimization-based network with Sharpness Losses.

Ablation study on proposed mask techniques. As mentioned in Section 5.1.3, the multi-scale scheme, incremental mask and multi-frame mask fusion are proposed for initialization generation. To analyze these techniques fairly, the proposed mask techniques are categorised into four different groups: naive method (without any proposed techniques), with multi-scale scheme, with multi-scale + incremental and with multi-scale + incremental + multi-frame. The outputs of four groups are shown in Fig. 5.11. As can be seen, the zoom-in rectangles of naive method indicate that the general flow mask proposed by [124] causes

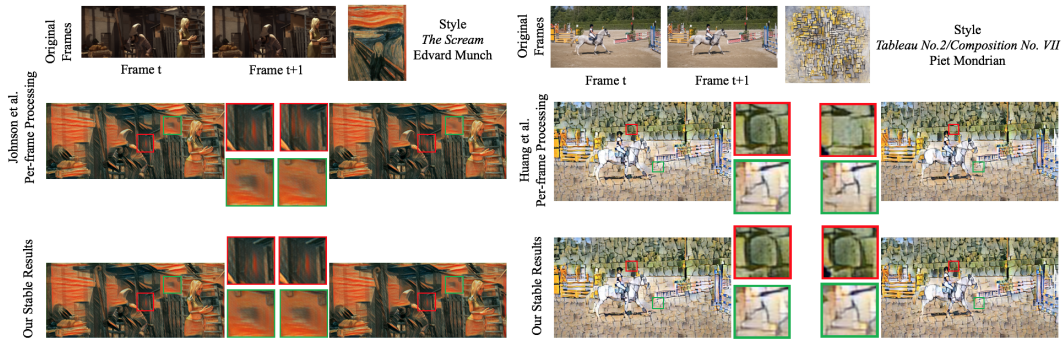


Figure 5.13: The effect of temporal consistency. The per-frame processing methods are Johnson et al. [68] and Huang et al. [59]. The red and green rectangles indicate the discontinuous texture appearances.

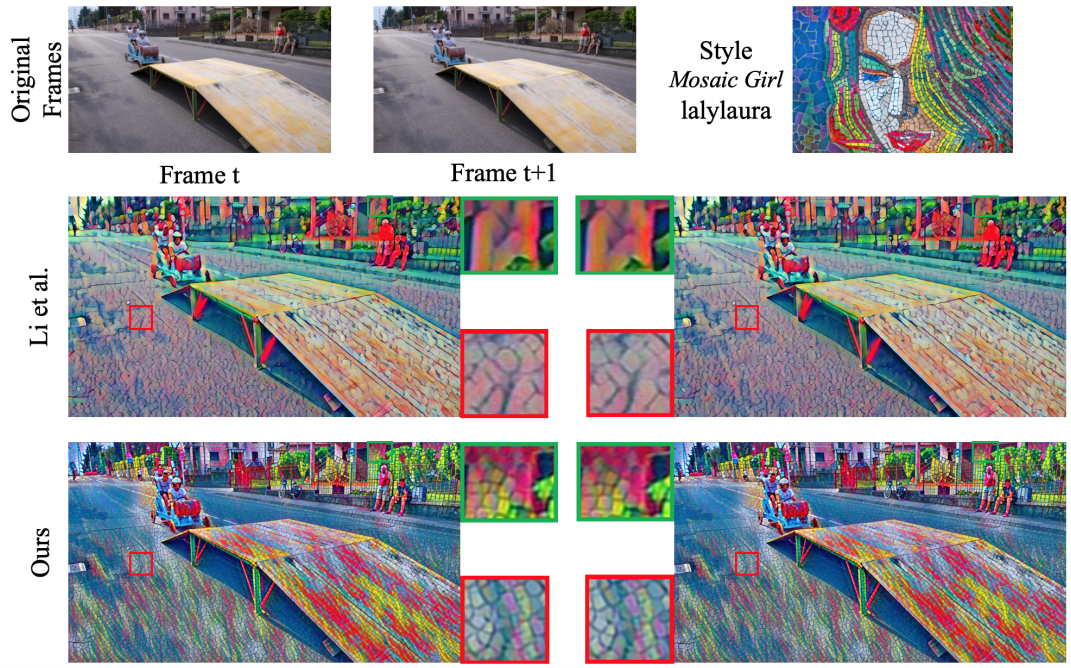


Figure 5.14: Comparison to Li et al. [95] on Soapbox scene from DAVIS 2017 dataset [14]. The rectangles indicate the difference in two adjacent stabilization results. Please view the supplementary video for better observation.

the ghosting artefacts. Adding multi-scale scheme into naive method is able to reduce flow untraceable errors (c.f. Fig. 5.8) while causes worse ghosting artefacts as well. Then the incremental mask (bottom-left) and multi-frame mask fusion (bottom-right) techniques are gradually added into w/ multi-scale scheme method, which finally produces results without ghosting artefacts.

Analysis of Loss Functions

Sharpness Losses. Fig. 5.12 shows the effect of Sharpness Losses in the proposed approach. Without the Sharpness Losses, the straight-forward idea that copying and pasting pixels from per-frame stylized result f_s^t and warped image w^t into corresponding regions through

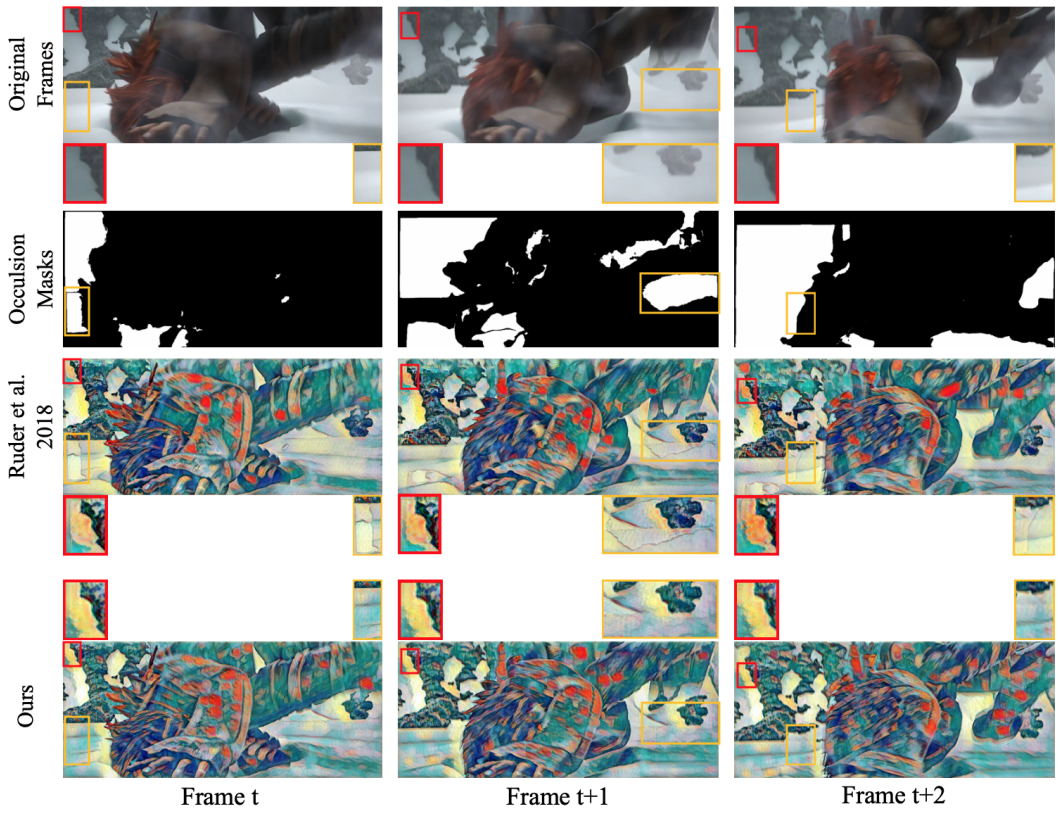


Figure 5.15: Comparison to Ruder et al. [125] on Ambush_4 scene from MPI Sintel dataset [14]. The per-frame processing method for both methods is Johnson et al. [68]. Red rectangles demonstrate that temporal consistency among adjacent frames, and yellow rectangles illustrate texture (spatial) consistency in a single frame. For temporal consistency, the proposed method achieves more consistent textures than Ruder et al.'s, which darkens the colours and adds texture patterns among adjacent frames (see red rectangles). For spatial consistency, the proposed method also obtains more consistent textures than Ruder et al.'s around boundaries of flow (see yellow rectangles).

mask M^t causes the pixel loss, and accumulates this loss along with the entire video process, which results in great image blurriness artefacts. By adding the Sharpness Losses, the proposed approach ensures that the pixel values are compensated from the beginning to the end which prevents the image blurriness artefacts in video outputs.

Coherent Losses. Fig. 5.13 shows the effect of Coherent Losses in the proposed method. Per-frame processing methods like [68] and [59] produce flickering artefacts in adjacent frames (see the zoom-ins) without any consideration of temporal consistency. The proposed method takes the per-frame stylized frames as inputs and produces the texture consistent consecutive outputs, for example, the rectangle areas.

Comparisons to State-of-the-art Methods

Fig. 5.14 shows the comparison between the proposed approach and Li et al. [95]. The method proposed by Li et al. [95] learns a linear transformation matrix to minimize the difference between covariance of transformed content features and style features, which serves

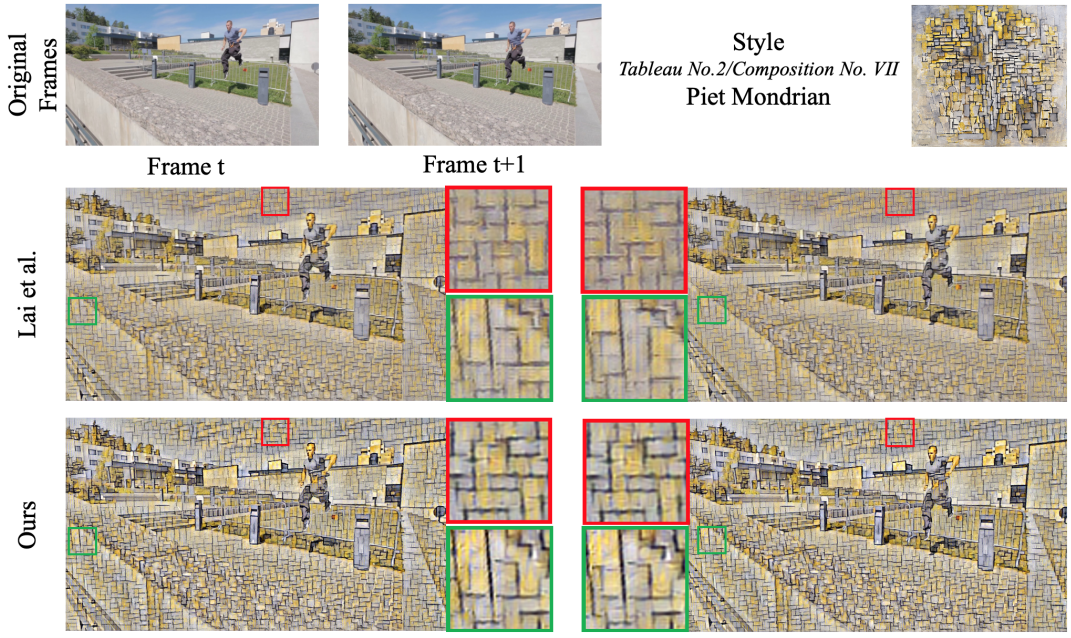


Figure 5.16: Comparison to Lai et al. [85] on Parkour scene from DAVIS 2017 dataset [14]. The per-frame processing method for both methods is Johnson et al. [68]. The rectangles indicate the difference in two adjacent stabilization results. Please view the supplementary video for better observation.

as second order statistics transformation from reference image onto content image in prior methods [37, 68]. The linear transformation is highly efficient but causes less stylistic texture presentation in transformed videos. For instance, the mosaic texture patterns of style reference image in Fig. 5.14 do not appear in their transferred video frames, which increases the temporal consistency among adjacent frames but less artistic texture appearance. However, the proposed method preserves better temporal consistency (c.f. Table 5.5) and texture patterns than theirs (c.f. zoom-ins).

Fig. 5.15 shows the comparison between the proposed approach and Ruder et al. [125] in a large motion and strong occlusion case. As mentioned in Section 5.1.1, texture discontinuity is a common problem in large motion or strong occlusion cases, thus a better video style transfer method should produce results with both high temporal consistency among adjacent frames and texture (spatial) consistency in a single frame. For temporal consistency, red rectangles of adjacent frames show that Ruder et al.'s method darkens the colours and inserts additional texture patterns which do not exist in original frames. In contrast, the proposed method preserves better colours and texture patterns than Ruder et al.'s. For spatial consistency in one single frame, Ruder et al.'s method produces discontinuous textures with context while the proposed approach does not (see yellow rectangles).

Fig. 5.16 shows the comparison between the proposed approach and Lai et al. [85]. Their

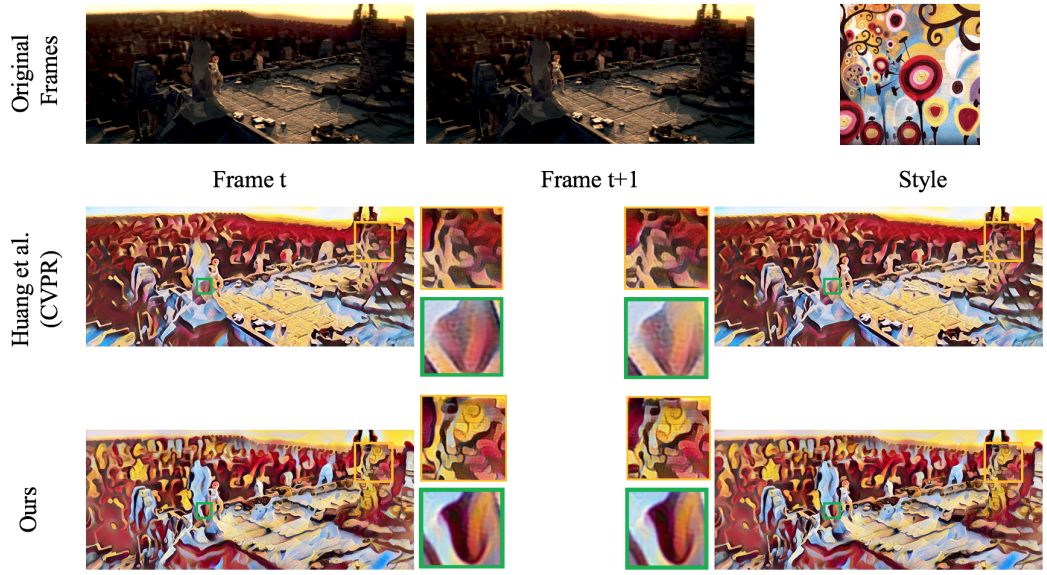


Figure 5.17: Comparison to Huang et al. [57] on Temple_2 scene from MPI Sintel dataset [14]. The per-frame processing method for both methods is Johnson et al. [68]. The rectangles indicate the zoom-ins in two adjacent stabilization results. As can be seen in rectangles, results by the proposed method obtain more diverse styles (orange) and better temporal consistency (green) than Huang et al.'s . Please view the supplementary video for better observation.

method is similar to ours which also deals with stylized videos obtained by per-frame processing methods (i.e.,[68]). However, their method needs to sacrifice temporal consistency for better perceptual quality. For example, the zoom-in boxes indicate that the proposed method preserves temporal consistency better than Lai et al.

Fig. 5.17 shows the comparison between the proposed approach and Huang et al. [57]. The method proposed by Huang et al. [57] does not need optical flow estimation in their test time thus it achieves real-time performance. Our method may not compete with their speed, but ours is capable of obtaining more coherent texture and diverse outputs than theirs. For example, the green boxes indicate that the proposed method preserves temporal consistency better than Huang et al., and the orange boxes illustrate that the proposed method produces outputs with more rich style features than their results.

Fig. 5.18 shows the comparison between the proposed approach and Chen et al. [16]. Note that the outputs of the proposed method show more rich features than those of [16] (see red rectangles). And the yellow rectangles indicate that the proposed approach produces more stable outputs than [16].

Fig. 5.19 shows the comparison between the proposed approach and Ruder et al.[124]. The method proposed by Ruder et al. produces the most stable outputs among the literature methods so far, but it suffers from great ghosting artefacts as their method warps wrong contents into flow untraceable regions (occlusions). However, their method needs heavy

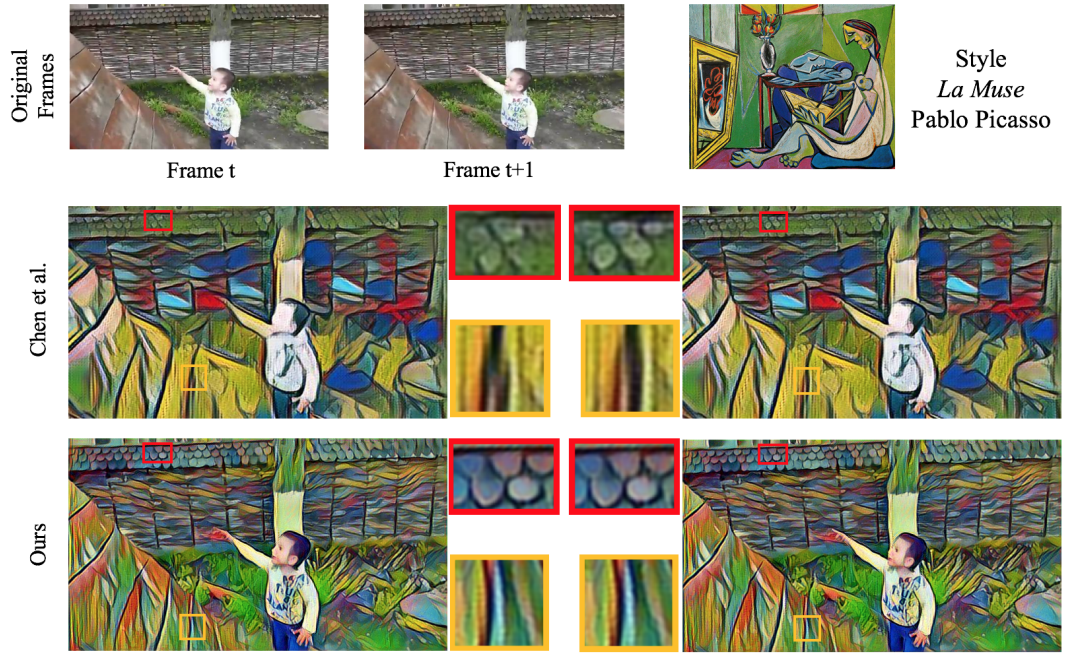


Figure 5.18: Comparison to Chen et al. [16] on Child scene from [16]. The rectangles indicate the difference in two adjacent stabilization results.

optimization process even though inserting original content images into occlusion regions when these movements are large, which still costs more time than ours. For example, the yellow rectangles show that their method leaves a texture representing arm behind in the next output, and it is especially obvious in the entire video style transfer process. Our method effectively mitigates this artefact by introducing the new mask techniques and initialization described in Section 5.1.2. In addition, the proposed approach also produces a more diverse output than Ruder et al. [124].

5.3.2 Quantitative Evaluation

The proposed method is verified on MPI Sintel dataset [14] and Davis 2017 dataset [118], and it is tested on more than 40 videos including animation and real-world videos. In this section, the ablation study is given on image sharpness, temporal consistency and proposed mask techniques in initialization to verify the proposed sharpness losses, coherent losses and initialization. Then the quantitative comparison of the proposed approach with state-of-the-art methods ([16, 57, 59, 68, 124, 125]) is given by using a term ***stability error*** e_{stab} which calculates the temporal errors between pairs of adjacent frames in an output video. The ***stability error*** e_{stab} is defined as :

$$e_{stab} = \sqrt{\frac{1}{(N-1)\times D}\sum_{t=2}^{N}\sum_{i=1}^{D} m_i^t (x_i^t - w_i^t)^2} \tag{5.12}$$

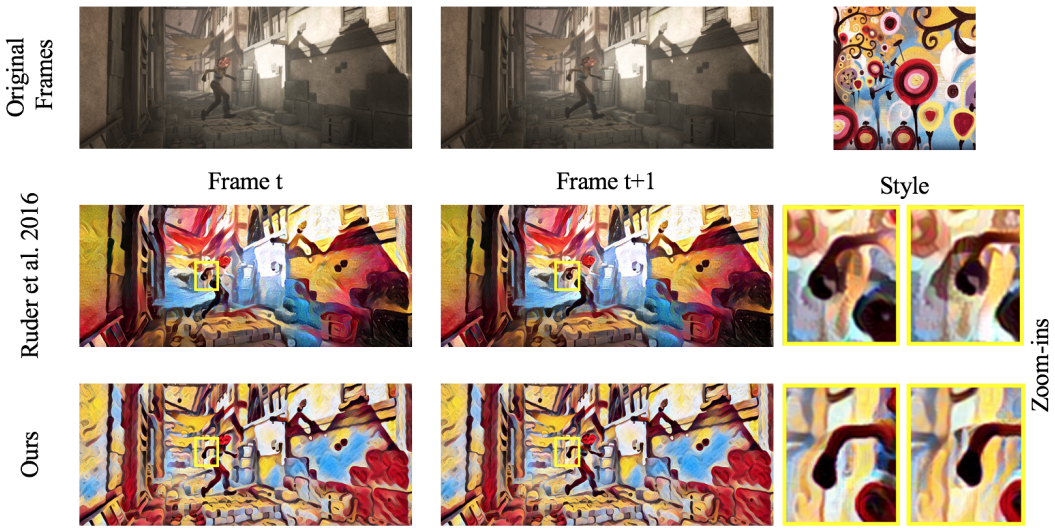


Figure 5.19: Comparison to Ruder et al. [124] on Alley_2 scene from MPI Sintel dataset [14]. The rectangles indicate the difference in two adjacent stabilization results. Please view the supplementary video for better observation.

where N denotes the total frame number of a video output, D denotes the total number of pixels in one video frame and m^t denotes the per-pixel flow weight. This formulation is similar to Equation (5.6), except that all the temporal loss are summed up for pairs of consecutive frames in a video. The warped image w^t warps the x_{out}^{t-1} at $t-1$ time forward to t. The ground truth of optical flow and flow weight can be from Flownet2 [61] or MPI Sintel dataset [14].

Ablation Study on Loss Functions

Sharpness Losses. To quantitatively verify the Sharpness Losses, an autoregressive(AR)-based Image Sharpness Metric (ARISM) without reference is chosen to assess image sharpness [44]. The ARISM is established on the hypothesis that AR model parameters estimated from 8-connected neighborhood of one image pixel tend to be very close to each other when this pixel locates in a comparatively smooth region, otherwise, these parameters are obviously distinct when this pixel is in a sharp region. The ARISM sharpness score is formulated as :

$$\rho = \sum_{k \in \Omega} \theta_k \rho_k \tag{5.13}$$

where $\Omega = \{E, C, E^{bb}, C^{bb}\}$ and θ_k are weights to each component. E and C are two classical metrics to define the difference between maximum and minimum values of AR parameters at point (i, j) of the input image. E^{bb} and C^{bb} are block-based pooling [147] of E and C, respectively.

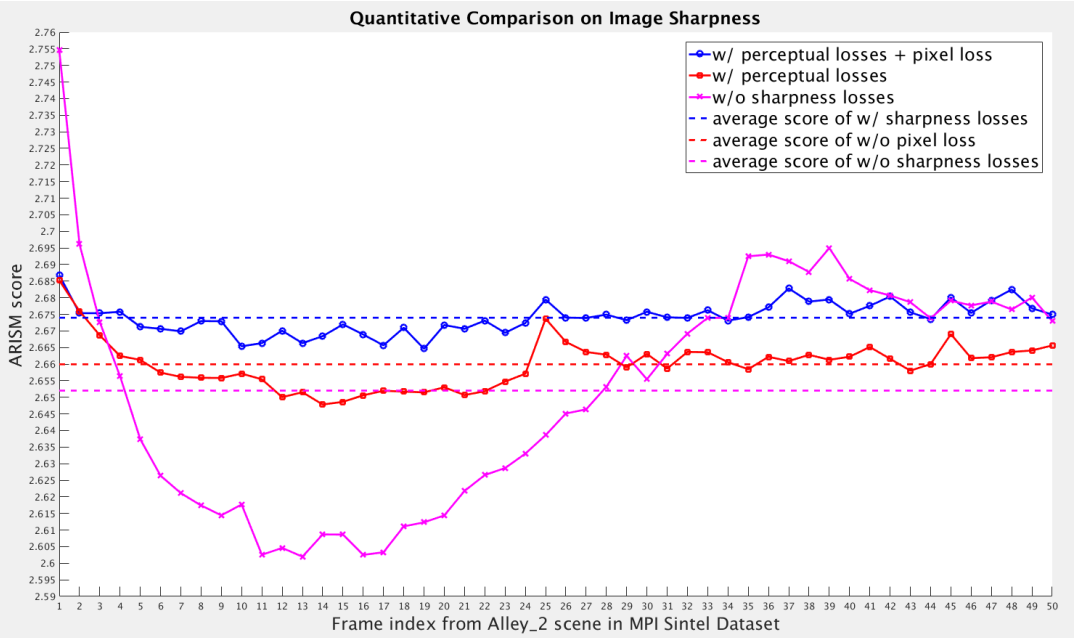


Figure 5.20: Ablation study on sharpness losses of Alley_2 scene from MPI Sintel dataset [14]. *The higher ARISM score is better*. The outputs with sharpness losses achieve highest ARISM scores than those without pixel loss and sharpness losses, which indicates that perceptual losses and pixel loss in proposed sharpness losses both contribute to reduce blurriness artefacts.

The ARISM has been proved to be robust to assess colour images with no reference which is suitable to our case since the outputs of video style transfer are colourful and also have no reference images. Fig. 5.20 shows the scores of each frame in Alley_2 scene (MPI Sintel dataset) which are obtained from w/ Perceptual Losses + Pixel Loss (aka,Sharpness Losses blue line) and w/ Perceptual Losses (red line) and w/o Sharpness Losses (magenta line), and **the higher ARISM score is better**. The scores of w/ Perceptual Losses + Pixel Loss are steady around the average score 2.674, while the average score of w/ Perceptual Losses tend to be decreased from 2.674 to 2.66, and that of w/o Sharpness Losses is close to 2.651. Note that the scores of w/ Sharpness Losses are always higher than those of w/o Pixel Loss, which indicates the outputs with both Sharpness Losses contain much less blurriness artefacts than those without Pixel Loss.

Coherent Losses. The quantitative ablation study on Coherent Losses is given in five testing scenes, which compares stability errors of two groups of temporal losses: multi-frame RGB-level only and both levels. Table 5.1 shows the detailed stability errors of the baseline method [68] and two groups. It is noticeable that multi-frame RGB-level only Coherent Loss contributes to 58.8% improvement compared to the baseline method [68], while multi-frame Feature-level Coherent Loss contributes further approximate 2.2% improvement which finally leads to 61.0% improvement in total.

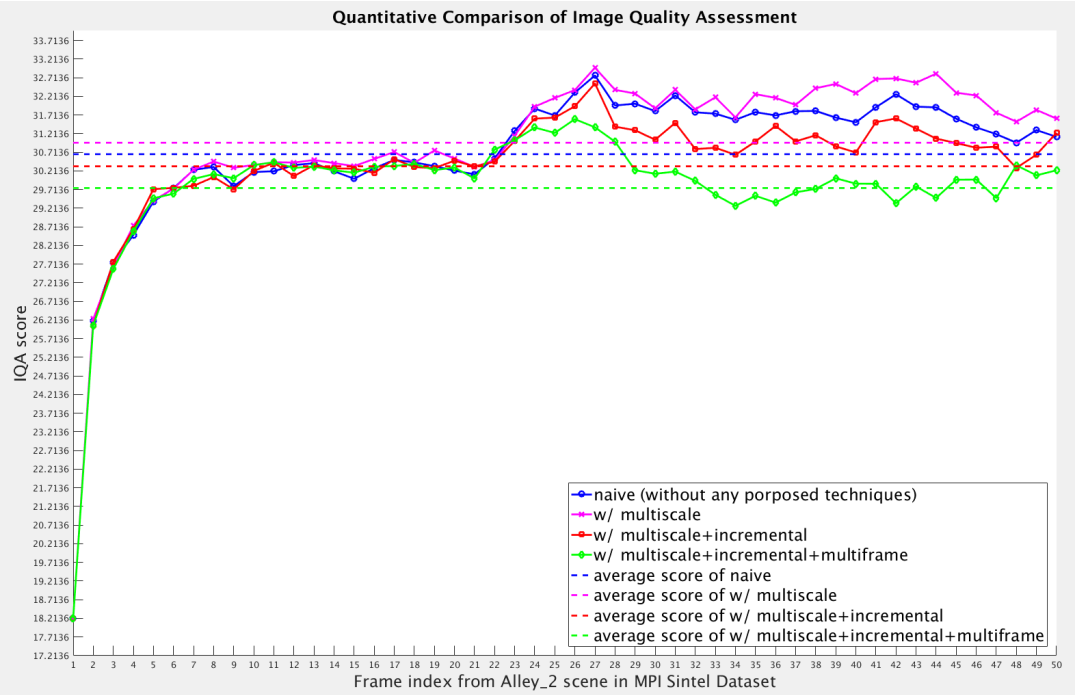


Figure 5.21: Ablation study on image quality assessment of Alley_2 scene from MPI Sintel dataset [14]. *A lower score indicates better visual image quality*. Note that adding multi-scale scheme (magenta line) causes image quality loss (higher score) compared to naive method (blue line), while adding incremental mask and multi-frame fusion (red line and green line) contributes to achieve lower scores than naive method (blue line).

Ablation Study on Initialization

As mentioned in Section 5.1.1, a set of new mask techniques are proposed to address the texture discontinuity problem. The detailed analysis is now given on image quality of the proposed techniques including multi-scale scheme, incremental mask and multi-frame mask fusion. To fairly compare these techniques, this ablation study follows Section 5.3.1 and it is categorised into four groups with general Image Quality Assessment (IQA) scores by the following term proposed in [12]:

$$Q = \frac{1}{N_p} \sum_{i}^{N_p} y_i \tag{5.14}$$

where N_p denotes the number of patches which are chosen from the given image, and y_i denotes the estimated visual qualities of patch i. This IQA score (refer to DIQaM-NR method in [12]) is chosen as our general image quality metric because it is capable of assessing image quality by coping with several distortion types such as luminance and contrast changes, compression, Gaussian noise and Rayleigh fading channel. Fig. 5.21 shows the detailed ablation study on proposed mask techniques, and **a lower score indicates better visual image quality**. Note that multi-scale scheme only (magenta line) reduces flow untraceable errors (see Fig. 5.8) but it causes higher scores than naive method (blue line). The incremental mask technique (red line) helps to achieve lower average scores than naive method (blue line) and

Table 5.1: Ablation study on Coherent Losses of five testing videos in MPI Sintel dataset.

Method	MPI Sintel Dataset + Woman with A Hat (style)						
	alley_2	bamboo_2	bandage_1	cave_4	market_2	average	improvement
Baseline [68]	0.1055	0.0721	0.0634	0.1206	0.0634	0.0850	*
Multi-frame RGB-Level only	0.0271	0.0317	0.0279	0.0593	0.0290	0.0350	58.8%
Both Levels	0.0243	0.0300	0.0259	0.0574	0.0275	0.0331	**61.0%**

Table 5.2: Stability errors of per-frame processing methods and the proposed approach on five testing videos in each dataset.

Method	MPI Sintel Dataset + Candy (style)					Davis 2017 Dataset + Mondrian (style)				
	alley_1	bamboo_2	cave_4	market_5	temple_3	dance-flare	car-turn	parkour	soapbox	stroller
Johnson et al. [68]	0.0815	0.0791	0.1196	0.1306	0.1450	0.1273	0.1263	0.1301	0.1362	0.1411
Ours	**0.0313**	**0.0356**	**0.0682**	**0.0788**	**0.0825**	**0.0408**	**0.0442**	**0.0485**	**0.0492**	**0.0513**
Huang et al. [59]	0.0916	0.1011	0.1451	0.1432	0.1444	0.1550	0.1452	0.1543	0.1577	0.1628
Ours	**0.0346**	**0.0439**	**0.0766**	**0.0828**	**0.0838**	**0.0500**	**0.0499**	**0.0586**	**0.0590**	**0.0614**

w/ multi-scale scheme method (magenta line), and multi-frame mask fusion (green line) further improves the image quality by decreasing the average score from 30.69 (dashed blue line) to 29.71 (dashed green line). This observation basically follows the qualitative evaluation in Fig. 5.11.

Quantitative Evaluation in Literatures

The stylized perceptual strokes or patterns in results of different methods are obviously distinct even for one particular style, which may make the IQA comparison unfair. Thus the detailed comparisons are given on stability error which is invariant to diverse strokes/patterns. Table 5.2 lists the stability errors of three different approaches at MPI Sintel dataset [14] and Davis 2017 dataset [118]. The proposed approach takes per-frame stylized results of per-frame processing methods as inputs and produces the stabilized outputs, thus two comparisons are listed in Table 5.2. There are five different scenes chosen from each dataset and combined with two representative styles. It is noticed that, for all testing videos, the proposed method significantly reduces the stability errors compared to per-frame processing methods [59, 68].

Table 5.3 lists the stability errors of state-of-the-art method [16] and the proposed approach. Four representative testing videos in [16] are chosen to compare. All the testing videos follow the default resolution 640×360 in [16]. Using the same per-frame method [68], the proposed method achieves much lower stability errors than that of Chen et al.'s in each case. And on the average improvement, the presented method (69.8%) is more competitive than Chen et al.'s method (40.3%).

Table 5.4 lists the stability errors of state-of-the-art methods [125],[95] and the proposed approach. The testing videos are from 26 scenes in MPI Sintel dataset [14] and 19 scenes in Davis 2017 dataset [118]. Compared to per-frame processing method in MPI Sintel dataset,

Table 5.3: Stability errors of state-of-the-art method [16] and ours on four testing videos. The groundtruth flow and occlusion masks are provided by Flownet2 [61]. *The results are provided by the authors. All the image resolutions are 640 × 360.

Method	Candy (style)		La_muse (style)		Statistics	
	alley_1	ice-age	alley_2	child	average	improvement
per-frame [68]	0.1160	0.0999	0.1367	0.1572	0.12745	*
Chen et al.* [16]	0.0648	0.0549	0.0881	0.0971	0.07622	40.3%
ours	0.0369	0.0408	0.0371	0.0390	0.03845	**69.8%**

Table 5.4: Average stability errors of state-of-the-art method [125] and the proposed approach on each dataset.

Method	MPI Sintel Dataset + Woman with A Hat (style)		Davis 2017 Dataset + Self Portrait 1907 (style)	
	average stability error	improvement	average stability error	improvement
Johnson et al. [68]	0.1027	*	0.0782	*
Ruder et al.[125]	0.0489	52.38%	0.0430	45.02%
Ours	**0.0401**	**60.98%**	**0.0358**	**54.24%**

Ruder et al. [125] improves the stability by 52.38% averagely while the proposed method obtains 60.98% improvement in terms of temporal consistency. For Davis 2017 dataset, Ruder et al. improve the stability errors by 45.02% compared to per-frame processing method [68] while the proposed approach achieves a higher improvement 54.24%.

Table 5.5 lists the stability errors of state-of-the-art methods and the proposed approach. The compared methods are verified at 5 different scenes in MPI Sintel dataset [14], which are used in [57]. All the testing videos are using 1024 × 436 resolution, and the groundtruth optical flow with corresponding masks are all provided from MPI Sintel dataset [14]. The stability errors are re-calculated. As the dataset only provides the forward direction of optical flow, the image w^t in Equation (5.12) warps per-frame stylized result f_s^{t+1} at time $t+1$ back to t. Our method may achieve higher average stability errors than Ruder et al. [124] but the outputs of the proposed approach obtain high quality results by mitigating effectively ghosting artefacts and it is much faster (1.8 seconds per frame) than theirs (3-5 minutes per frame or dozens of seconds even using loose constraints). Moreover, the proposed method achieves higher improvement (28.1%) than Lai et al. [85] (2.66%), Huang et al. [57] (21.3%) and Ruder et al. [125] (22.6%). Our method also achieves average lower stability errors than Li et al. [95].

5.4 Summary

This chapter proposes a novel framework to reduce the flow errors which includes multi-scale mask fusion, incremental mask, multi-frame mask fusion and a new initialization for optimization-based network. These mask techniques reduce significantly both the flow untraceable errors and flow traceable errors (ghosting artefacts). The new initialization ensures

Table 5.5: Stability errors of state-of-the-art methods and ours on five testing videos in MPI Sintel dataset. The groundtruth flow and occlusion masks are provided by MPI Sintel dataset. *The results are provided by the authors. All the image resolutions are 1024 × 436. †The baseline of Li et al. [95] and Ruder et al. [124] is not [68] thus it does not have improvement.

Method	MPI Sintel Dataset + Candy (style)						
	alley_2	ambush_5	bandage_2	market_6	temple_2	average	improvement
Johnson et al. [68]	0.0987	0.1487	0.0862	0.1291	0.1119	0.11492	*
Lai et al. * [85]	0.0959	0.1467	0.0839	0.1249	0.1079	0.11186	2.66%
Huang et al.* [57]	0.0694	0.1171	0.0695	0.1093	0.0867	0.0904	21.3%
Ruder et al.* [125]	0.0697	0.1142	0.0657	0.1076	0.0873	0.0889	22.6%
Li et al.* †[95]	0.0692	0.1061	0.0661	0.0941	0.0820	0.0835	*
Ours	0.0589	0.1097	0.0610	0.1043	0.0794	0.08266	**28.1%**
Ruder et al.†[124]	0.0572	0.1099	0.0471	0.0908	0.0726	0.07552	*

that the proposed approach produces stable video outputs even in large motion and occlusion cases, and it also speeds up the optimization process from minutes per frame to around seconds per frame. The proposed multi-frame Coherent Losses ensure the temporal consistency between consecutive outputs, and Sharpness Losses effectively mitigate the image blurriness artefacts during the entire video stabilization process.

Chapter 6

Deep Normal Transfer for Bas-relief Modelling with Enriched Detail and Geometry

Figure 6.1: The upper row shows artistic bas-relief works. The lower row shows digital bas-relief results produced by the proposed method. Readers are recommended to view the electronic version for details.

Previous chapters have answered the Questions 1-6, and also achieved the Objectives 1-6. Now this chapter turns its direction onto NGTS field, especially applying neural style transfer on geometry surfaces for digital bas-relief generation.

Bas-relief, a special type of sculpture that figures are slightly emerged from a background, is a bridge between 2D drawings and 3D sculptures. Bas-relief has received considerable

attentions in recent years since it can be viewed from many different angles without causing distortion of the figures. Due to this desirable intrinsic nature, bas-reliefs as an art form has been very popular since prehistorical time [86]. They are now treated as either a single piece of artwork or decorations for walls, monuments, furniture, medals, potteries etc. In recent years, more and more researchers in Computer Graphics have developed approaches to fulfill the stylistic design purpose of bas-reliefs. Fig. 6.1 shows a couple of real bas-relief examples designed by artists and digital ones generated by the framework described in this chapter. Please note the fine details of the bas-relief examples in Fig. 6.1.

Detail and geometry richness is an essential key to artistic creations. Missing any of them, the generated bas-relief will be impossible to convey the ideas from the artists. For example, the relief in the right of Fig. 6.1, the scaly textures and petal details present realistic fish and lotus flowers. The display of complex geometry shapes and capture of such realistic fish in motion make them stand out artistically.

In general, digital bas-reliefs come from two types of sources: 2D images and 3D models. Methods based on natural images have ill-posed problems ([172][42]) in nature, and the approaches based on 3D models mainly focus on designing sophisticated non-linear depth compression algorithms which typically incur a high computational cost. Some recent works (e.g.,[63, 64, 156]) attempt to simplify the bas-relief modelling problem by working on the normal images which contain both pixel-level detailed appearance (2D information) and the normal information of the geometry leading to stereoscopic perception (3D look). The bas-reliefs are modelled in normal image space rather than in object space, which solves the ill-posed problems that other image-based approaches suffer and removes the need of sophisticated depth compression algorithms.

Realistic bas-reliefs should present both detailed appearances and stereoscopic perception. For detail transfer, existing works ([63, 64]) tend to rely on straightforward image processing techniques like cut-and-paste and decompose-and-compose, which often lead to imperfect composition results. For example, the cut-and-paste operation cannot preserve the normals of the target. The decompose-and-compose [131, 132] operation requires larger detail patches than target patches while it cannot manipulate target surfaces or cause scaling issue for textures. For geometry preservation, existing methods (e.g., [156]) decompose the normal field into a base layer and a detail layer by directly subtracting the base normal value from the original normal, which causes unexpected triangle distortions in the resultant bas-reliefs.

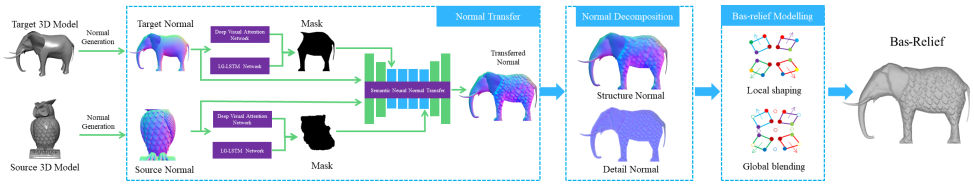


Figure 6.2: The overview of the proposed approach. The proposed method contains three stages: normal transfer, normal decomposition and bas-relief modelling. Normal transfer completes the task that transfers the fine details from source normal to target normal; Normal decomposition creates structure normal and detail normal with enhanced geometry properties; Bas-relief modelling constructs bas-relief from structure and detail normal obtained from normal decomposition.

In this chapter, a novel bas-relief modelling method is proposed to overcome the above-mentioned issues, and produces bas-reliefs with rich details as well as preserving the geometry intact. Inspired by image style transfer [37], a semantic neural network of normal transfer is proposed that treats detail transfer as a style transfer problem. The proposed network is capable of transferring the fine details from a source normal image to the target normals in arbitrary shapes and scales. In addition, to generate transferred details on desired areas of the normal images, a visual attention mechanism [151, 152] and object parsing [101] are adopted to predict the corresponding masks. Rich texture areas and desired target areas are extracted from input normals into masks, and these masks are used in the proposed semantic normal transfer network to produce the transferred normals. For geometry preservation, a normal decomposition scheme based on Domain Transfer Recursive Filter (DTRF) is proposed to enhance the geometry properties. The local shaping and global blending steps from [156] are adopted to construct the mesh of a bas-relief from detail and structure layers obtained by normal decomposition.

The overview of the proposed framework is shown in Fig. 6.2. The digital bas-relief modelling pipeline which takes 3D models as inputs, manipulates normal fields and generates diverse visual effects of texture transfer, structure and detail preservation. The proposed semantic neural network of normal transfer facilitates the design and texture transfer of bas-reliefs and makes the generation process in an efficient and intelligent way. The main contributions of this chapter are:

- **Semantic neural network of normal transfer**. The proposed semantic neural network of normal transfer learns the texture and structure representations, and then recombines them to generate a new normal image which shows the similar texture patterns of the source normal image and similar structure surface of the target normal image. It is capable of taking arbitrary sizes and shapes of normal images as inputs, and then synthesizing them into a new texture (**CTRB7-8**, answering to **Q6-7**, achieving **OBJ7**).

- **Geometry preservation without introducing artefacts.** By considering the orientation

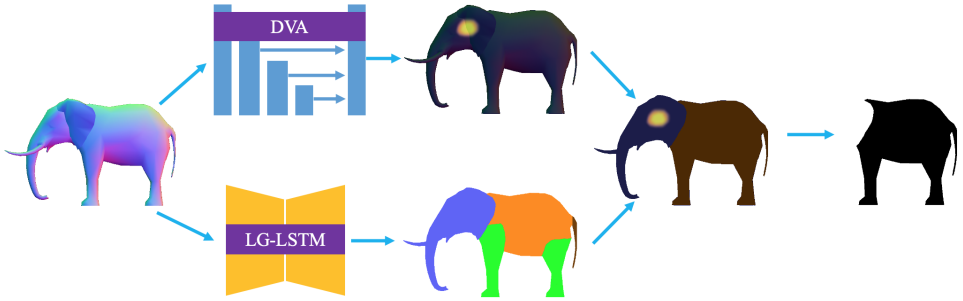


Figure 6.3: The diagram of mask area extraction. The DVA network generates human eye attention area on elephant's head which is an identity region with richer details than its other parts, and the LG-LSTM network segments the elephant into various regions based on object parts. Then regions exclude head are chosen as target mask. In general, areas without eye attention are chosen for target normals while highlight regions are chosen for source normals. For target normals, areas with less details are more desired as the transferred textures are better presented on these areas. For source normals, however, areas with highlight attention are more desired.

of vector rotations, the proposed normal decomposition scheme obtains a structure layer and a detail layer with continuous and more natural edges and shapes, which contribute to produce artefacts free bas-relief modellings.

6.1 Semantic Neural Normal Transfer

A point in a normal image indicates a normal vector, thus spatial structures and texture patterns are actually the orientation differences between normal vectors. As far as known, deep neural networks ([80, 102]) are capable of learning these structures and patterns from images, which indicates that the orientation differences of normal vectors can be captured and stored as neural responses in networks. Based on this, a semantic deep neural network of normal transfer is proposed to accomplish the detail transfer task by learning and recombining spatial structures and texture patterns from input normal images. In addition, Deep Visual Attention Network (DVA) [151] and LG-LSTM Network [101] are adopted to generate mask images.

Fig. 6.3 illustrates the basic procedure of extracting areas for masks. The DVA network produces human eye attention areas for the input normal map, and LG-LSTM network generates parsing segmentations based on object parts (e.g., head, body and legs etc). Since the attention areas usually lie on heads or faces [151] which are identity areas with rich details, thus this feature is able to guide us to extract masks. For example, DVA network produces highlight area in the elephant normal on its head, while LG-LSTM network generates part segmentations. For target normals, areas with less details are more desired as transferred textures are better presented on these areas, thus segmented areas without attention areas (i.e., regions exclude head in Fig. 6.3) are chosen to be final target mask areas. For source

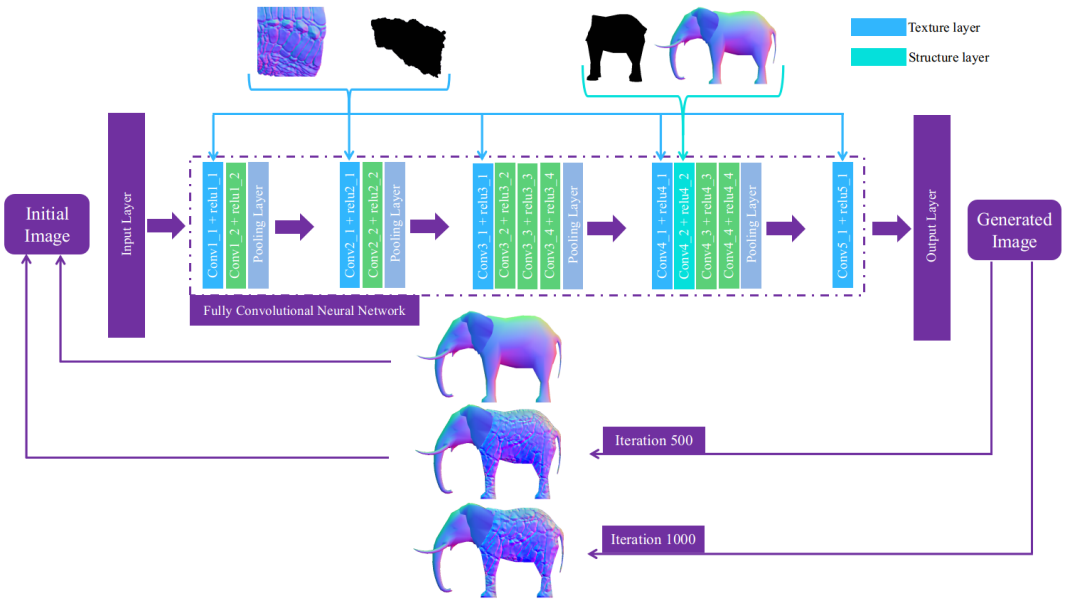


Figure 6.4: The overview of the proposed semantic neural normal transfer network. The network takes source/target normal images and their corresponding masks as inputs, then computes a new generated normal image with transferred details from source normal via an optimization process.

normals, in contrast, areas with rich details are more desired, thus highlight areas are segmented as mask areas. To increase the user control on design, the masks can also be manually segmented by artists.

Fig. 6.4 shows the architecture of the proposed semantic neural network of normal transfer. In this work, the source normal image provides the texture patterns, the target normal image provides the spatial structures and the pixel-level binary mask images indicate the regions that are valid for texture transfer. At the beginning of the optimization process, the target normal image and its corresponding mask, the source normal image and its corresponding mask are passed into the network and their features are learnt in the network. Then the proposed network starts from the target normal image and gradually synthesizes it into a new normal result via optimization iterations. This optimization process minimizes the Euclidean distance between texture and structure representations. For the given source normal image x_{tex}, target normal image x_{str} and masks m_{tex} and m_{str}, the proposed network searches a new stylized normal image $\hat{x}$ by minimizing the following loss term:

$$\mathcal{L}(\hat{x}, x_{str}, x_{tex}, m_{str}, m_{tex}) = \alpha \mathcal{L}_{str}(\hat{x}, x_{str}, m_{str}) + \beta \mathcal{L}_{tex}(\hat{x}, \kappa, x_{tex}, m_{tex}) + \delta \mathcal{R}_{tv} \tag{6.1}$$

with:

$$\mathcal{R}_{tv} = \sum_{i,j} ((\hat{x}_{(i,j+1)} - \hat{x}_{(i,j)})^2 + (\hat{x}_{(i+1,j)} - \hat{x}_{(i,j)})^2) \tag{6.2}$$

where the **structure loss** $\mathcal{L}_{str}$ penalizes the difference of valid structure representations between x_{str} and $\widehat{x}$; the **texture loss** $\mathcal{L}_{tex}$ penalizes the difference of valid texture representations between x_{tex} and $\widehat{x}$. To encourage the spatial smoothness in the generated image $\widehat{x}$, a **total variation regularization** is added in the proposed network. α and β denote the weights to balance the structure component and texture component of the stylized result $\widehat{x}$ while κ and δ respectively denote the weights of texture scales and smoothness.

Let the matrix $\phi_j(\cdot) \in \mathbb{R}^{N_j \times M_j}$ denotes the vectorized feature maps representing the neural responses in a layer j where N_j is the number of channels and M_j is $Height \times Width$ of the corresponding feature maps, and the mask m_{str} indicates the valid regions for structure preservation. The structure loss $\mathcal{L}_{str}$ is defined as the mean square error between the two valid feature representations $\phi_j(x_{str})$ and $\phi_j(\widehat{x})$ in the masked area:

$$\mathcal{L}_{str}(\widehat{x}, x_{str}, m_{str}) = \sum_{j \in J_s} m_{str} \cdot (\phi_j(x_{str}) - \phi_j(\widehat{x}))^2 \tag{6.3}$$

where J_s denotes the set of layers in a pre-trained VGG-19 network [136] in which the structure loss is computed. Gatys et al. [35] have discovered that the Gram-based correlations of neural responses can be exploited as the texture representations. Hence, the texture loss $\mathcal{L}_{tex}$ is denoted as the squared Euclidean distance between the scaled texture normal representation $\kappa \cdot x_{tex}$ and the generated new normal representation $\widehat{x}$:

$$\mathcal{L}_{tex}(\widehat{x}, \kappa, x_{tex}, m_{tex}) = \sum_{j \in J_t} \kappa \cdot m_{tex} \cdot (\psi(\phi_j(\kappa \cdot x_{tex})) - \psi(\phi_j(\widehat{x})))^2 \tag{6.4}$$

where J_t denotes the set of layers in the pre-trained VGG network in which the texture loss is computed, $\psi(\phi_j(\cdot)) = \phi_j(\cdot) \cdot \phi_j(\cdot)^T \in \mathbb{R}^{N_j \times N_j}$ is the Gramian Matrix [35], which is used to represent the texture information.

6.2 Image-based Normal Decomposition

Normal decomposition aims to extract a structure layer L_s and a detail layer L_d from the original normal field L_o. Generally, the structure layer L_s is achieved by applying a normal filtering to smooth L_o, and L_d is obtained by subtracting L_s from L_o [156]. However, the detail layer actually presents the orientation differences between normal vectors in L_s and L_o. Thus the idea [64] considering orientation on normal subtraction is adopted in the proposed normal decomposition scheme, and it utilizes an edge-preserving technique Domain Transfer Recursive Filter [34] to extract the structure layer. Let $P_{(i,j)}$ and $P_{(i,j-1)}$ denote two

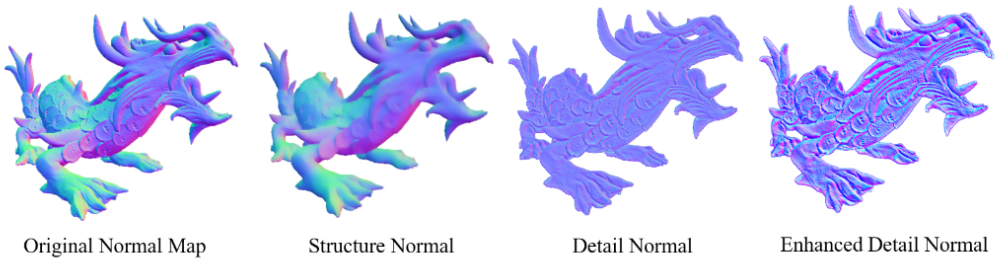


Figure 6.5: One example of the proposed normal decomposition results $\{\sigma_s = 10, \sigma_r = 1.0\}$ and the detail normal is enhanced by increasing brightness and contrast.

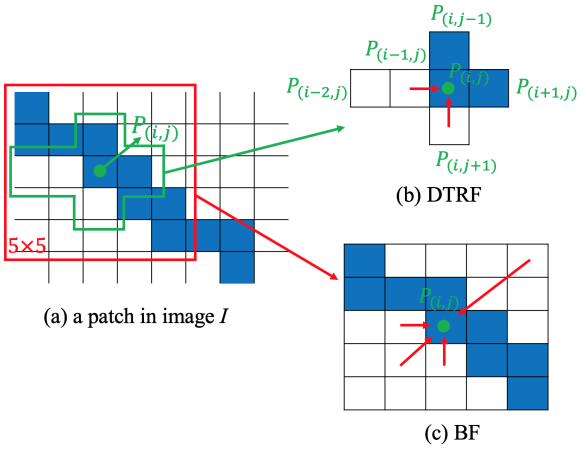


Figure 6.6: Demonstration of DTRF and BF in computing values of edge points. The blue points are assumed as edge points while other points are non-edge points. The point $P_{(i,j)}$ is the current point computed by Equation (6.5).

adjacent points in a normal image I, the result $J(P_{(i,j)})$ of DTRF in [34] is defined as:

$$J(P_{(i,j)}) = \frac{1}{K_p} \sum_{k=1}^{c} (1 - a^d) \cdot I^k(P_{(i,j)}) + a^d \cdot J^k(P_{(i-1,j)}) \tag{6.5}$$

where $a = exp(-\sqrt{2}/\sigma_s)$ and $d = 1 + \frac{\sigma_s}{\sigma_r}|I(P_{(i,j)}) - I(P_{(i-1,j)})|$. σ_s and σ_r are respectively standard deviation and deviation range. K_p denotes the scaling factor that normalizes $J(P_{(i,j)})$ to a unit vector and c denotes the channel index (e.g., RGB). To achieve a symmetric response of Equation (6.5), the DTRF filter is applied twice: for a normal image I, Equation (6.5) is performed left-to-right (top-to-bottom) and then bottom-to-top (right-to-left). To simplify the user-specified parameters, σ_r is set to 1.0 which works well in all experiments.

Fig. 6.5 shows one example using the proposed normal decomposition. The normal decomposition filter is defined on an edge-preserving technique Domain Transform Recursive Filter (DTRF) [34] which is capable of working on colour images at arbitrary scales without the need of resorting to subsampling or quantization. The following paragraph now gives the reason why DTRF is better than BF [143] for normal smoothing. For DTRF, the output of Equation (6.5) can only be affected by previous points (e.g., $I(P_{(i-1,j)})$ and $I(P_{(i,j+1)})$ for

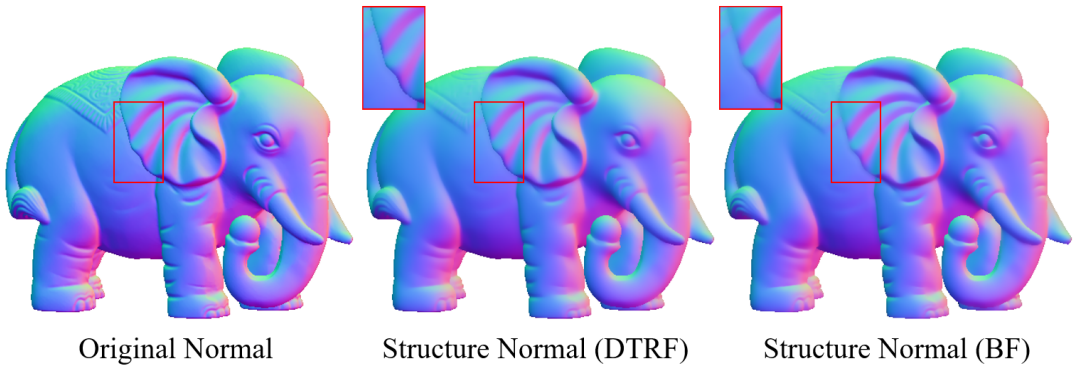


Figure 6.7: Structure Normals obtained by DTRF $\{\sigma_s = 4, \sigma_r = 0.7\}$ and BF $\{\sigma_s = 3, \sigma_r = 0.3\}$. The zoom-ins of the red rectangle areas indicate that DTRF preserves better edges than BF in normal smoothing application.

horizontal and vertical directions, respectively) and their own values. We will demonstrate in horizontal direction as that of vertical direction is similar. For example, in Fig. 6.6 (b), point $P_{(i-1,j)}$ is a non-edge point and $P_{(i,j)}$ is an edge point, then d in Equation (6.5) increases compared to that of $P_{(i-1,j)}$ and $P_{(i-2,j)}$ which leads a^d to be zero, thus the value $I(P_{(i,j)})$ is preserved well and barely affected by its neighbouring non-edge point. In contrast, values of points on edges computed by BF can still be affected by non-edge points (i.e., points in white color in 5x5 window size (c)), and this influence could be enhanced after a few iterations, especially when the neighbouring points of an edge point share similar values. For example, in Fig. 6.7, the edge of top-right corner in the zoom-in of structure normal (BF) is almost wiped out and updated into the similar color of non-edge neighbouring points.

Detail enhancement. To demonstrate the effectiveness compared to value subtraction, the comparison of the detail enhancement results obtained by Wei et al. [156] and the proposed normal decomposition is shown in Fig. 6.8. To achieve detail enhancement effect, the detail layer is simply enhanced via increasing brightness and contrast. Note that the detail layer looks much more natural than the result of Wei et al. [156] in structure surface and boundaries between edge and surfaces.

This paragraph now gives the reason why the proposed normal decomposition achieves more natural bas-relief modellings than Wei et al. [156]. The orientation for the normal subtraction is taken into account. Each point in the normal image indicates a normal vector. Thus the normal differences between two points in base and original normal are actually orientation differences which should follow the vector rotation. As the orientations of the corresponding points in base and original normal field are continuous, the orientations of new vectors computed have the property of continuity as well. Therefore, the detail layer containing those orientation differences gives continuous and reasonable edges and surfaces in image space. Wei et al. [156] perform normal decomposition on mesh level via a vector

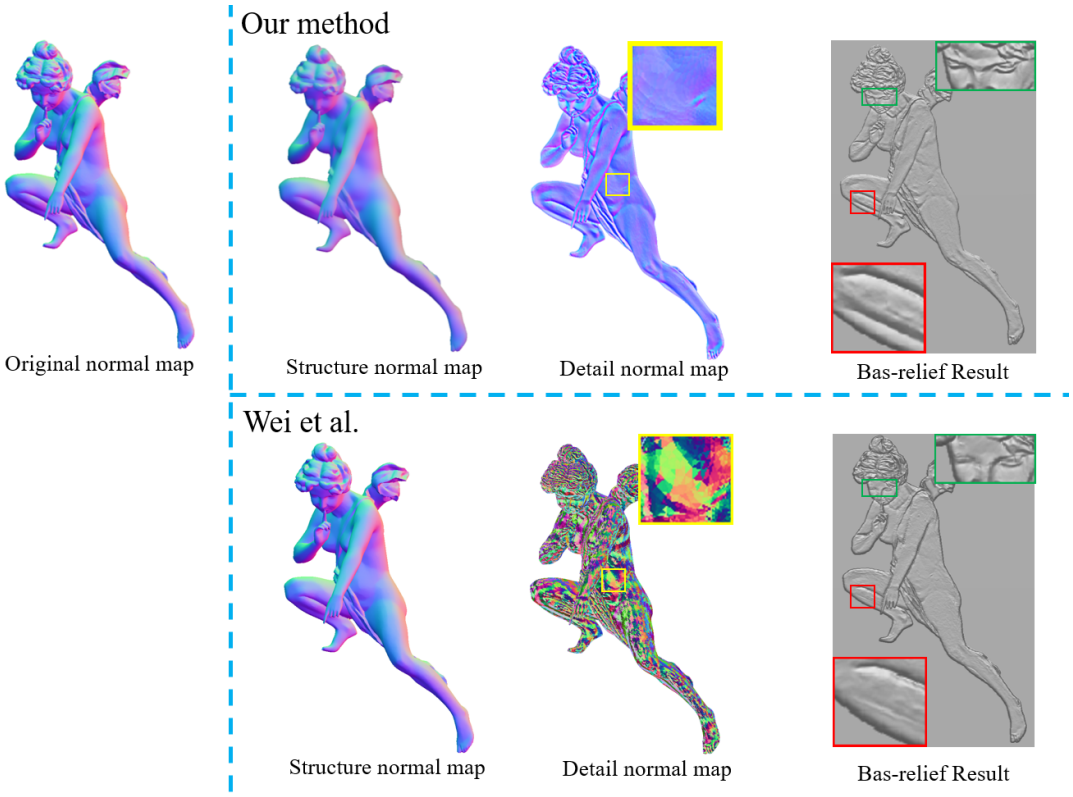


Figure 6.8: Comparison of normal decomposition between Wei et al. [156] and the proposed method. The proposed normal decomposition $\{\sigma_s = 10, \sigma_r = 1.0\}$ extracts details based on normal orientations and structure surface information as well, while Wei et al. [156] only obtain details which may damage the structure surface when detail enhancement is applied (see the zoom-ins of the red rectangle areas).

length threshold θ which follows a certain Gaussian distribution, then the small triangles with points having shorter vector length are remained in the detail layer (c.f. the orange rectangle shown in the bottom-middle of Fig. 6.8). However, the orientations between remained neighbour triangles are not continuous which leads to unexpected triangle distortions and damages the structure surface in bas-relief modelling results when detail enhancement is applied (c.f. the red and green rectangles in Fig. 6.8). The proposed normal decomposition has no such damage.

6.3 Bas-relief Modelling

In this section, the bas-relief model is constructed on the decomposed structure normal map L_s and detail normal map L_d. To generate the surface from decomposed normals, the idea from [156] is adopted, which regards the mesh construction as an optimization problem. During each iteration, the generalized SfG [165] technique firstly splits current mesh into quadrangular faces according to the normal orientation of L_s and L_d, then stitches all the disconnected faces together to form a complete surface. The entire bas-relief modelling is

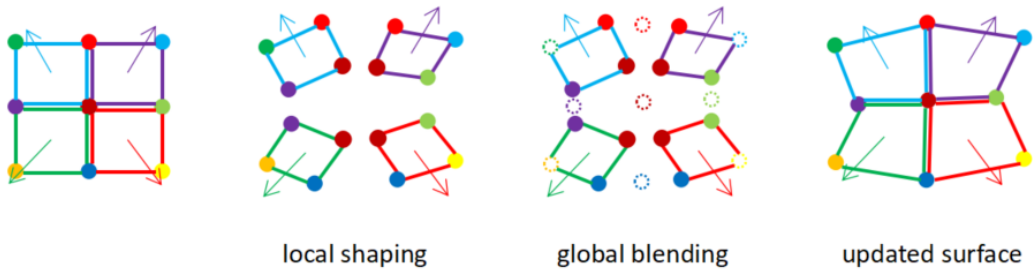


Figure 6.9: The overflow of bas-relief modelling. In the local shaping step, each face (represented with the same colour edges and a normal vector) is projected according to its transferred normal vector. A vertex may be spit into two or four vertices, which are represented in the same colour. In the global blending step, new vertex positions (marked as hollow circles) are calculated by minimizing the total energy (Equation (6.6)). These vertices are re-organized in originally connected way to form an updated surface. Then, the iteration on the updated surface is repeated until it is converged.

proceeded in two steps: local shaping (split) and global blending (stitch) (shown in Fig. 6.9).

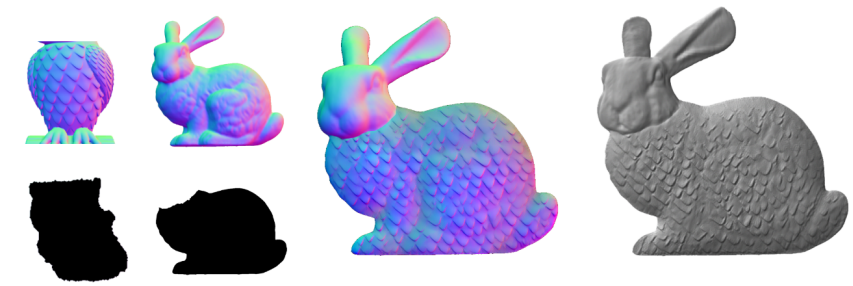
Let f_o denotes the expected output mesh, f_s and f_d denote the desired structure and detail, and h_f denotes the expected fixed overall height, then the total energy of bas-relief modelling is formed as:

$$E(f_o, f_s, f_d, h_u, h_f) = E_s(f_o, f_s) + \lambda_a E_d(f_o, f_d) + \lambda_b E_f(f_o, h_f) \tag{6.6}$$

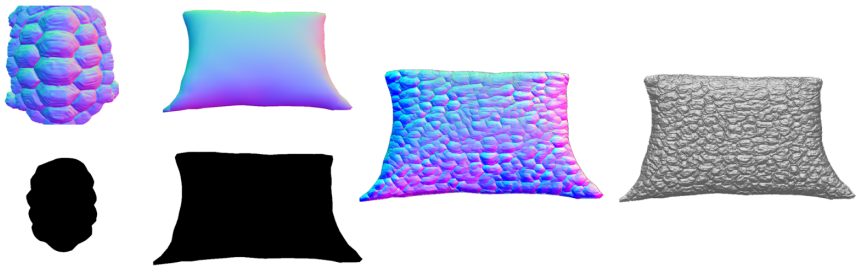
where $E_s(f_o, f_s) = \|f_o - f_s\|^2$ is the energy function that minimizes the difference between output faces f_o and structure faces f_s which aims to preserve the structure of transferred normal. $E_s(f_o, f_d) = \|f_o - f_d\|^2$ aims to preserve the details of transferred normal. $E_f(f_o, h_f) = \left\|h_{f_o} - h_f\right\|^2$ is to control the overall fixed height of relief. λ_a is the weight to recover the geometry details, and λ_b affects the style of the resulting bas-relief (roundness or flatness), and larger value of λ_b means flatter style.

6.4 Implementation Details

The normal computation and bas-relief modelling is implemented using C++ and OpenGL, and the normal transfer is implemented using Pytorch 1.3.1 with CUDA 10.0. All the experiments are performed on a desktop PC with two 2.10GHz Intel(R) Xeon(R) Platinum 8160T CPU, 256 GB RAM and two NVIDIA TITAN RTX graphics card. A user-friendly GUI is created. The proposed method is tested on a set of models with detail enhancement, and transfers texture details to various models in different scales and arbitrary shapes, which demonstrates its capability and effectiveness of diverse styles.



(a) Owl transfer on Bunny.



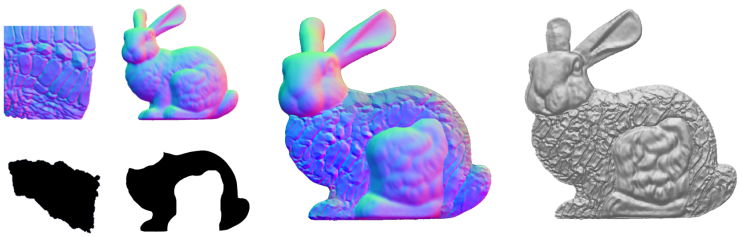
(b) Turtle transfer on Pillow.

Figure 6.10: Examples of transfer results and their corresponding generated bas-reliefs based on regular shapes of source and target normal images.

Parameters. For mask generation, the pre-trained models of [101, 151] provided by authors are utilized in mask generation. Since the pre-trained model of [101] is only trained on images with entire animal bodies, the segmented results for the source normals are not clear as target normals and could be arbitrary shapes like owl shown in Fig. 6.2. For normal transfer, the proposed network has four parameters: $\{\alpha, \beta, \sigma, \kappa\}$ where they respectively control the structure preservation, texture preservation, texture scales and smoothness. Specifically, the ratio α/β presents the emphasis on either reconstructing the structures or the texture patterns. A larger ratio α/β indicates the structure identity of target normal in the synthesized result is strongly preserved, and a smaller ratio α/β indicates the texture patterns of source normal are effectively presented in the synthesized result. For a specific pair of source and target normals, user can adjust the trade-off between structure identity and texture patterns to create visually desired styles. For normal decomposition, the proposed method has one user-specified parameter σ_s which is the weight of structure preservation.

For bas-relief modelling, the proposed method has three parameters: $\{\lambda_a, \lambda_b, h_f\}$ where they are the weights of geometry preservation, flatness and fixed relief height, respectively. The normal image resolution for normal transfer and bas-relief modelling is fixed to 700×700. To reduce the number of user-specified parameters, the settings are followings: $\{\alpha = 1e0, \beta = 1e6, \delta = 1e-3\}$ and $\{relu4\_2\}$ for structure layers, and $\{relu1\_1, relu2\_1, relu3\_1, relu4\_1$ and $relu5\_1\}$ for texture layers which are default settings([37]) in normal transfer

(a) Turtle transfer on Buddha.



(b) Dragon transfer on Bunny.

Figure 6.11: Example of transfer results and their corresponding generated bas-reliefs based on arbitrary shapes of source and target normal images.

stage.

6.5 Results and Analysis

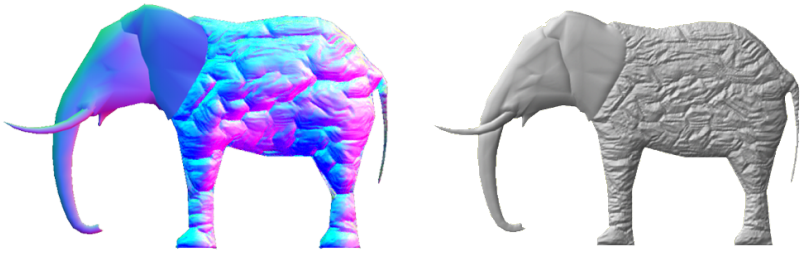
6.5.1 Detail transfer of Bas-relief Modelling

One advantage of the proposed method is the detail transfer on normal field, which is capable of transferring vivid texture patterns from one normal image to another. Unlike existing cut-and-paste technology in [63], the proposed approach learns the texture patterns from target normal images and transfers them onto source normal images in arbitrary shapes and multiple scales.

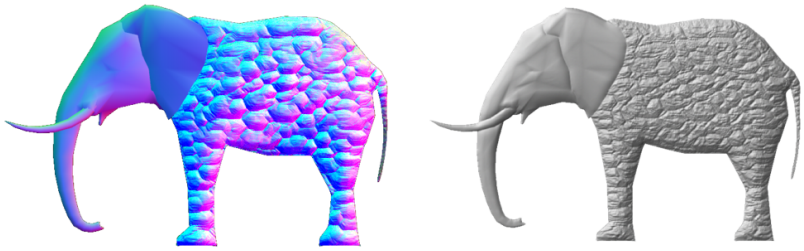
Regular shapes. Fig. 6.10 demonstrates that the proposed method transfers the partial texture details of Turtle shell onto the round belly shape of Buddha and a rectangle-like shape of Pillow. The parameters are $\{\kappa = 0.4, \lambda_a = 2, \lambda_b = 0.05, h_f = 0.5\}$.

Arbitrary shapes. Besides the regular shape, the proposed method still deals with arbitrary shapes as shown in Fig. 6.11. The texture details of the Owl and the Dragon are transferred to the body of the Bunny without any distortion. The parameters are $\{\lambda_a = 2$, $\lambda_b = 0.05$, $h_f = 0.5\}$. The masks of buddha and bunny are extracted in photoshop.

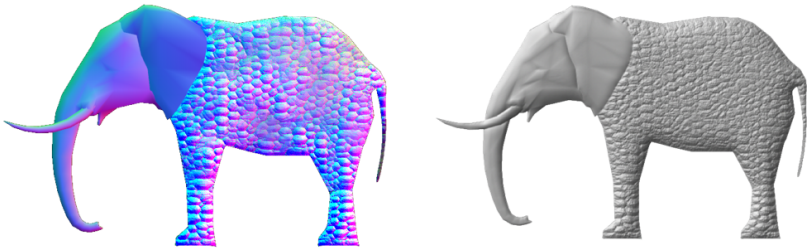
(a) Source normal and target normal.



(b) Turtle transfer on Elephant (κ = 1.0).



(c) Turtle transfer on Elephant (κ = 0.5).



(d) Turtle transfer on Elephant (κ = 0.25).

Figure 6.12: An example of detail transfer results with multi-scale texture features.

Multiscales. The proposed work is able to transfer different scales of details to the same target model by simply tuning one hyperparameter κ for texture scales during optimization process as shown in Fig. 6.12. The parameters $\{\lambda_a = 2, \lambda_b = 0.05, h_f = 0.5\}$.

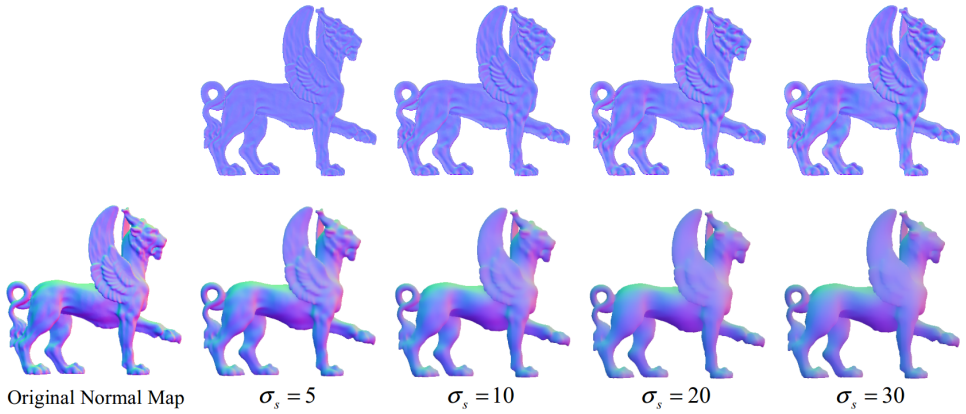


Figure 6.13: The proposed normal decomposition results with different σ_s values. The first column is the original normal map as input. Results from the second column show the structure normals (bottom row) and corresponding detail normals (top row).

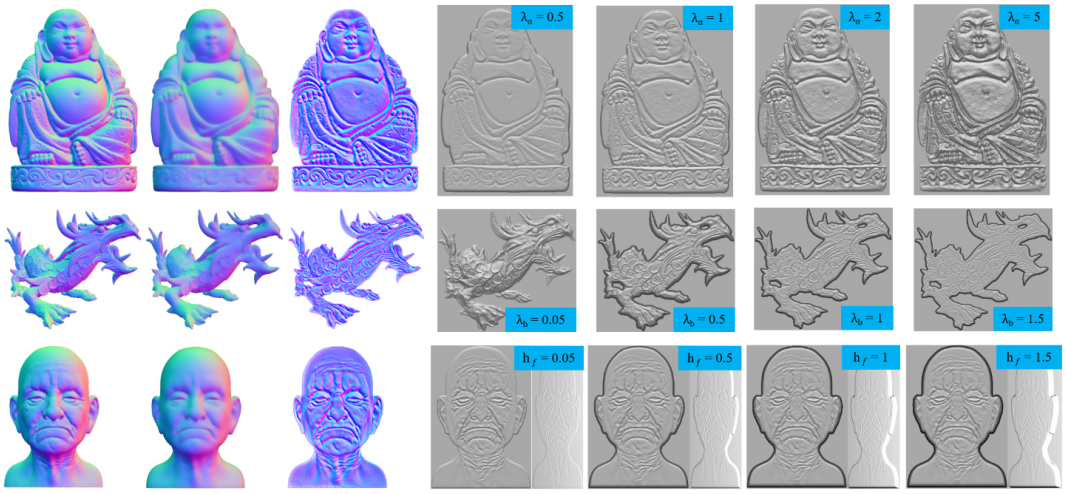


Figure 6.14: Analysis of parameters λ_a, λ_b and h_f in Bas-relief modelling. In the left, the images are original normal map, structure normal map and detail normal map enhanced in each column. The first row shows that a larger λ_a value would enhance the detail preservation in relief result $\{\lambda_b = 0.5, h_f = 0.1\}$. The second row shows that a larger λ_b value would produce flatter relief results $\{\lambda_a = 1.5, h_f = 0.3\}$. The bottom row shows that a larger h_f value would increase the height of relief results $\{\lambda_a = 2, \lambda_b = 1.0\}$.

6.5.2 Parameter σ_s for Normal Decomposition

Fig. 6.13 shows the influence of parameter σ_s on the proposed normal decomposition. As can be seen, the structure normal becomes smoother along with the increase of σ_s which preserves clearer details in detail normal. Thus, a larger σ_s value captures more details while preserving less structure information, and a lower σ_s value ignores some details while preserving more structure information. The σ_s values are chosen between 10 and 20 in all experiments.

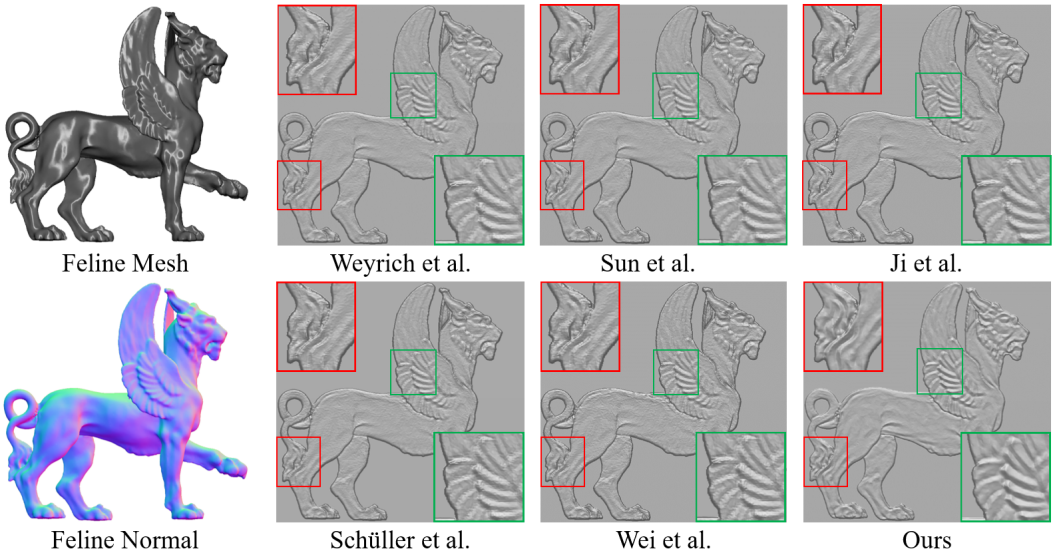


Figure 6.15: Comparison to state-of-the-art methods with detail enhancement on standard thickness.The images in the first column is the original Feline mesh and Feline normal field, then the bas-relief results on the right columns follow the order: Weyrich et al. [159], Sun et al. [139], Ji et al. [63], Schüller et al. [127], Wei et al. [156] and ours $\{\lambda_a = 5, \lambda_b = 1.5, h_f = 0.1\}$. Readers are recommended to view the electronic version for more clear details.

6.5.3 Hyperparameters for Bas-relief Modelling

Fig. 6.14 shows how to tune the hyperparameters in the bas-relief modelling to get desired visual effects. For geometry preservation, a larger λ_a preserves surface details more clearly and even enhances it in an over-compressed case. A larger λ_b value produces a flatter bas-relief while a smaller one generates a round style of a model. Parameter h_f determines the overall height of the produced bas-reliefs.

6.5.4 Comparisons to previous literature methods

Comparison to state-of-the-art methods with detail enhancement. For fair comparison of the bas-relief results, a linear scaling as post-processing step is adopted from [156], which aims to make sure the generated bas-reliefs share same height since these approaches usually do not control their depth exactly. Additionally, hyper-parameters for each method are carefully fine-tuned to show their best visual appearance under the same lighting environment and height compression. The proposed approach is compared with five state-of-the-art bas-relief modelling methods in a flatten style. All the produced results share some similarities, and details on Feline model are preserved well, which are shown in Fig. 6.15. However, the proposed approach $\{\lambda_a = 5, \lambda_b = 1.5, h_f = 0.1\}$ produces more continuous and obvious edges (c.f. red rectangles) and natural shape surface (c.f. green rectangles) than others.

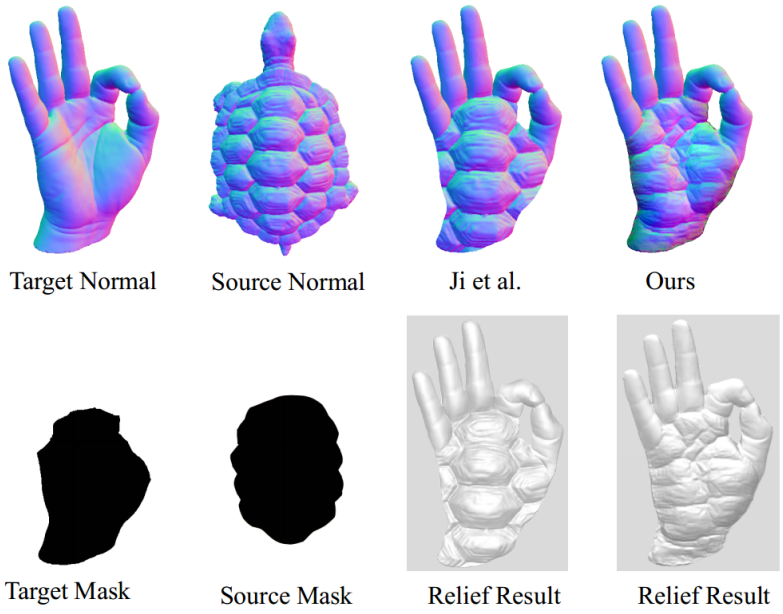


Figure 6.16: Comparison on detail transfer between Ji et al. and ours $\{\lambda_a = 5, \lambda_b = 0.05, h_f = 3\}$.

Comparison on detail transfer between Ji et al. [63] and the proposed method. Fig. 6.16 shows the detail transfer comparison. Ji et al. [63] proposed a normal-based model method which constructed reliefs from normal images as well. However, their method utilized the cut-and-paste operation on the image domain to achieve detail transfer results which inevitably covered the original surface structures and details of target normal images. The proposed approach transfers the texture patterns to target normals while preserving original geometric property. As can be seen in Fig. 6.16, our result not only preserves well in the surface structures but also produces turtle textures spreading along with the detailed lines and surface of a human hand.

6.5.5 Time Consumption

In general, the proposed approach contains three stages which are normal transfer, normal decomposition and bas-relief modelling. The normal maps of 3D input meshes are fed into the proposed semantic neural network in normal transfer stage. This stage generates synthesized normal images by running on two NVIDIA TITAN RTX graphics card. Next, the transferred normal result is decomposed into structure normal and detail normal via the proposed normal decomposition operator. Finally, the bas-relief modelling stage—surfaces are generated based on the normal images produced by previous stage. For typical models in this work, the time performance of three stages are recorded, where stage 2 and 3 are implemented on CPU while stage 1 on GPU. Time cost of experiments in figure 6.10, 6.11 and

Table 6.1: Time consumption (seconds). Here shows the time occupation for typical bas-relief results.

Models	Stage 1	Stage 2	Stage 3	Total
	normal transfer	normal decomposition	bas-relief modelling	
Buddha	110.78	16.36	47.98	175.12
Pillow	110.28	15.88	35.70	161.86
Bunny(Owl)	110.56	16.07	43.83	170.46
Bunny(Dragon)	109.08	16.15	43.86	109.09
Elephant(κ=1.0)	109.00	24.37	32.58	165.95
Elephant(κ=0.5)	109.57	19.58	36.25	165.40
Elephant(κ=0.25)	109.39	21.31	36.03	166.73

6.12 can be found in Table 6.1.

Limitation. The proposed current semantic normal transfer method is a CNN network which regards the texture transformation as an online optimization problem (without any training process). It runs around 110 seconds in execution time for two 700×700 normal images which is slow in practice. To speed up the process, the optimization process is recommended to be replaced with feed-forward networks which may achieve real-time performance. The details on feed-forward networks can be found in these related works ([17, 68, 146]).

6.6 Summary

This chapter presents a normal based bas-relief modelling method. To enrich the detailed features, a semantic neural network of normal transfer is developed, which learns distributions of texture patterns and structure details from both source and target normal images respectively. Then a new normal image combining these distributions is generated by an optimization process. Unlike previous normal editing methods, the proposed work is capable of learning the texture patterns from the source normal images and transferring them onto the target normal images in arbitrary shapes and multiple scales. To preserve geometric properties, a normal decomposition scheme is presented to generate bas-relief results free from artefacts. A number of experimental results show that the proposed method produces reasonable and pleasant bas-reliefs with enriched details and preserved geometry. Our future work will focus on speeding up the pipeline as the proposed current semantic normal transfer network uses a slow optimization process. A promising solution is to use feed-forward networks instead of optimization process, which will save time for texture transformation.

Chapter 7

Conclusion and Future Works

7.1 Conclusion

This thesis has focused on applying Neural Artistic Style Transfer on a few related computer vision research fields such as Neural Photo Style Transfer, Neural Video Style Transfer and Neural Geometry Texture Synthesis. It starts from Chapter 2 Literature Reviews, which reviews state-of-the-art methods on NST, analyses the pros/cons of existing approaches and summarizes the factors that influence the performance on different research areas, e.g., photorealism of NPST, stability of NVST and capability of NGTS (**OBJ1**).

In Chapter 3, the content-mismatching and the photorealism problems of NPST (**Q1-2**) have been solved by proposing a similarity loss function and a Style Fusion Model (SFM). The similarity loss terms in style transformation network prevents geometric mismatching (**OBJ2, CTRB1**) between content and style reference inputs, and the SFM utilizes edge-preserving filter to reduce distortion artefacts (**OBJ3, CTRB2**).

In Chapter 4, a fast solution of NPST (**OBJ4**) is proposed by integrating prior similarity loss term and SFM into a feed-forward network based NAST method (**Q3**), which proves that representative NAST methods can be transformed into NPST methods (**CTRB3-4**).

In Chapter 5, the ghosting and flickering artefacts of NVST are solved by a novel optimization-based framework in an arbitrary-style-per-network fashion (**OBJ6, Q4-5**), which is capable of reducing flow errors and generating stable video stylization results in large motions and strong occlusion cases (**OBJ5, CTRB5**). The flow errors are filtered out by a set of mask techniques and the temporal consistency are enhanced by mutli-frame RGB-level and Feature-level Coherent Losses. The proposed approach also speeds up the optimization-based NVST from minutes per frame to seconds per frame via a new initialization (**CTRB6**).

In Chapter 6, the extension of NST on geometry texture synthesis has been proved to be successful on digital bas-relief modeling (**Q6, CTRB7**). The proposed semantic neural normal transfer network is able to enrich details in bas-relief design (**OBJ7**) by synthesizing

new texture patterns from a style reference normal image (**Q7, CTRB8**) onto desired content inputs via an automatically attention-based mask technique.

7.2 Future Works

In the previous chapters, this thesis has investigated into NPST, NVST and NGTS, and proposes a few novel methods to achieve **OBJ1-7** with **CTRB1-8**, which proves related research fields benefits from integrating NST into their frameworks. In this section, a further possible step for NST extension has been made which considers to synthesize non-stationary textures on 3D mesh surfaces in an efficient way (**CHAL8**).

7.2.1 Motivation

Since the convolutional neural networks succeed in image classification task [80], it has been applied into vast research fields including geometry processing which inspires a new way to approach geometry problems. The challenges of geometry processing using deep neural networks lie in the irregular and unordered 3D representations. To tackle these challenges, Li et al. [96] and Hancoka et al. [48] have developed deep networks on manipulating point clouds and meshes for classification and segmentation tasks. However, as a fundamental topic, mesh generation and surface synthesis through deep networks have not attracted much attention in computer graphics.

7.2.2 Potential Solution

Based on the brief literature reviews in Section 1.2 in Introduction, a few factors should be considered: 1. parameterization should be avoided as much as possible as the distortion is inevitable if arbitrary textures are considered for synthesis on surfaces; 2. non-stationary texture synthesis still remains a challenge even using deep neural networks; 3. arbitrary-texture-per-network is more practical for geometry texture synthesis.

Texture synthesis on surfaces via multi-view rendering seems a potential solution as no parameterization is part of solution. In addition, non-stationary texture synthesis requires global structures of reference inputs to be learned and transferred to target mesh surfaces, thus coarse-to-fine rendering scheme could be a key to solve this problem. For example, coarse level is responsible to synthesize the global structure information of the reference texture patterns captured by deep features to target mesh, and fine level takes care of local texture synthesis on target mesh, thus the camera numbers of coarse level is less than next finer level. The inputs to neural networks could be a target mesh and reference geometry

images like normal image since Chapter 6 has proved that normal images can be applied for texture synthesis using NST. As for arbitrary-texture-per-network, an encoder-decoder network plus loss network (e.g., loss network in [68]) architecture could be useful when the coarse-to-fine mutli-view rendering images of target mesh are fed into loss network as content representations, and an arbitrary texture image can be regarded as conditional input into latent vector of target mesh and their deep second order statistics (e.g., Gram Matrix) captured by loss network can be style representations. During training, the decoder aims to update the offsets of target mesh vertices, which will be added into original vertex positions to form desired textures on surfaces by minimizing the Gram Loss computed in loss network.

In summary, a potential system overview could be described as followings:

1. The inputs are a target mesh without details on surfaces and a reference texture normal image;

2. During training, the pipeline consists of two encoders (e.g., one mesh encoder and one image encoder), one decoder and one loss network (e.g., a pre-trained VGG network) where the mesh vertex positions and reference images are separately encoded into latent vectors, decoder is responsible to generate offsets for each vertices, thus the output of decoder plus original vertex positions forms the synthesized textures. While at test time, only two encoders and one decoder are utilized to synthesize textures;

3. Before training, the camera position and scale of each view of mesh are pre-computed. Thus during training, these rendered views by a differentiable renderer are fed into loss network along with the random cropped regions of the reference normal image, where the coarse-level views are paired with the full reference image, mid-level views are paired with random cropped large regions of the reference image, and fine-level views are paired with random cropped small regions of the reference image;

4. The loss terms could consist of a KL loss and regularization for the mesh encoder, a Gram loss for the texture synthesis in loss network, and a normal loss to ensure the vertex won't be changed too much;

5. The image encoder could be the encoder of AdaIN [59] which scales the mesh input with standard deviation and shifts it with mean;

6. After training, the potential network could synthesize texture all over the target mesh in real-time.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv: 1603.04467* (2016).

[2] Noam Aigerman and Yaron Lipman. "Orbifold Tutte embeddings." In: *ACM Trans. Graph.* 34.6 (2015), pp. 190–1.

[3] Noam Aigerman, Roi Poranne, and Yaron Lipman. "Seamless surface mappings". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–13.

[4] Xiaobo An and Fabio Pellacini. "User-Controllable Color Transfer". In: *Computer Graphics Forum*. Vol. 29. 2. Wiley Online Library. 2010, pp. 263–271.

[5] Alexander G Anderson, Cory P Berg, Daniel P Mossing, and Bruno A Olshausen. "DeepMovie: Using Optical Flow and Deep Neural Networks to Stylize Movies". In: *arXiv preprint arXiv: 1605.08153* (2016).

[6] Benoit Arbelot, Romain Vergne, Thomas Hurtut, and Joëlle Thollot. "Local texture-based color transfer and colorization". In: *Computers & Graphics* 62 (2017), pp. 15–27.

[7] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. "Spatio-temporal saliency networks for dynamic saliency prediction". In: *IEEE Transactions on Multimedia* 20.7 (2018), pp. 1688–1698.

[8] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. "The generalized patchmatch correspondence algorithm". In: *European Conference on Computer Vision*. Springer. 2010, pp. 29–43.

[9] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. "Better mixing via deep representations". In: *International Conference on Machine Learning*. 2013, pp. 552–560.

[10] Sema Berkiten, Maciej Halber, Justin Solomon, Chongyang Ma, Hao Li, and Szymon Rusinkiewicz. "Learning detail transfer based on geometric features". In: *Computer Graphics Forum*. Vol. 36. 2. Wiley Online Library. 2017, pp. 361–373.

[11] Manuel Berning, Kevin M Boergens, and Moritz Helmstaedter. "SegEM: efficient image analysis for high-resolution connectomics". In: *Neuron* 87.6 (2015), pp. 1193–1206.

[12] Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. "Deep neural networks for no-reference and full-reference image quality assessment". In: *IEEE Transactions on Image Processing* 27.1 (2017), pp. 206–219.

[13] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. "Unsupervised pixel-level domain adaptation with generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3722–3731.

[14] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. "A naturalistic open source movie for optical flow evaluation". In: *European Conference on Computer Vision*. Springer. 2012, pp. 611–625.

[15] Marcel Campen, Hanxiao Shen, Jiaran Zhou, and Denis Zorin. "Seamless Parametrization with Arbitrarily Prescribed Cones". In: *arXiv preprint arXiv: 1810.02460* (2018).

[16] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. "Coherent Online Video Style Transfer". In: *arXiv preprint arXiv: 1703.09211* (2017).

[17] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. "Stylebank: An explicit representation for neural image style transfer". In: *arXiv preprint arXiv: 1703.09210* (2017).

[18] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2018), pp. 834–848.

[19] Tian Qi Chen and Mark Schmidt. "Fast patch-based style transfer of arbitrary style". In: *arXiv preprint arXiv: 1612.04337* (2016).

[20] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. "cudnn: Efficient primitives for deep learning". In: *arXiv preprint arXiv: 1410.0759* (2014).

[21] Wei-Ta Chu and Yi-Ling Wu. "Image Style Classification based on Learnt Deep Correlation Features". In: *IEEE Transactions on Multimedia* (2018).

[22] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. "Torch7: A matlab-like environment for machine learning". In: *BigLearn, NIPS Workshop*. EPFL-CONF-192376. 2011.

[23] Cassidy J Curtis, Sean E Anderson, Joshua E Seims, Kurt W Fleischer, and David H Salesin. "Computer-generated watercolor". In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 1997, pp. 421–430.

[24] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition". In: *International conference on machine learning*. 2014, pp. 647–655.

[25] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. "Example-based style synthesis". In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2003, pp. II–143.

[26] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style". In: *CoRR, abs/1610.07629* 2.4 (2016), p. 5.

[27] Alexei A Efros and William T Freeman. "Image quilting for texture synthesis and transfer". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM. 2001, pp. 341–346.

[28] Alexei A Efros and Thomas K Leung. "Texture synthesis by non-parametric sampling". In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. IEEE. 1999, pp. 1033–1038.

[29] David Eigen and Rob Fergus. "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2650–2658.

[30] Michael Elad and Peyman Milanfar. "Style transfer via texture synthesis". In: *IEEE Transactions on Image Processing* 26.5 (2017), pp. 2338–2351.

[31] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. "Split and match: Example-based adaptive patch sampling for unsupervised style transfer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 553–561.

[32] Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. "ReCoNet: Real-time Coherent Video Style Transfer Network". In: *arXiv preprint arXiv: 1807.01197* (2018).

[33] Jacob R Gardner, Paul Upchurch, Matt J Kusner, Yixuan Li, Kilian Q Weinberger, Kavita Bala, and John E Hopcroft. "Deep manifold traversal: Changing labels with convolutional features". In: *arXiv preprint arXiv: 1511.06421* (2015).

[34] Eduardo SL Gastal and Manuel M Oliveira. "Domain transform for edge-aware image and video processing". In: *ACM Transactions on Graphics (ToG)*. Vol. 30. 4. ACM. 2011, p. 69.

[35] Leon Gatys, Alexander S Ecker, and Matthias Bethge. "Texture synthesis using convolutional neural networks". In: *Advances in Neural Information Processing Systems*. 2015, pp. 262–270.

[36] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A neural algorithm of artistic style". In: *arXiv preprint arXiv: 1508.06576* (2015).

[37] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2414–2423.

[38] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. "Controlling perceptual factors in neural style transfer". In: *arXiv preprint arXiv: 1611.07865* (2016).

[39] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. "Exploring the structure of a real-time, arbitrary neural artistic stylization network". In: *arXiv preprint arXiv: 1705.06830* (2017).

[40] Bruce Gooch and Amy Gooch. *Non-photorealistic rendering*. AK Peters/CRC Press, 2001.

[41] Bruce Gooch, Erik Reinhard, and Amy Gooch. "Human facial illustrations: Creation and psychophysical evaluation". In: *ACM Transactions on Graphics (TOG)* 23.1 (2004), pp. 27–44.

[42] Lapo Governi, Monica Carfagni, Rocco Furferi, Luca Puggelli, and Yary Volpe. "Digital bas-relief design: A novel shape from shading-based method". In: *Computer-Aided Design and Applications* 11.2 (2014), pp. 153–164.

[43] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. "A kernel two-sample test". In: *Journal of Machine Learning Research* 13.Mar (2012), pp. 723–773.

[44] Ke Gu, Guangtao Zhai, Weisi Lin, Xiaokang Yang, and Wenjun Zhang. "No-reference image sharpness assessment in autoregressive parameter space". In: *IEEE Transactions on Image Processing* 24.10 (2015), pp. 3218–3231.

[45] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. "Arbitrary style transfer with deep feature reshuffle". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8222–8231.

[46] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Characterizing and Improving Stability in Neural Style Transfer". In: *arXiv preprint arXiv: 1705.02092* (2017).

[47] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. "Non-rigid dense correspondence with applications for image enhancement". In: *ACM transactions on graphics (TOG)* 30.4 (2011), p. 70.

[48] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. "MeshCNN: a network with an edge". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.

[49] Mingming He, Jing Liao, Dongdong Chen, Lu Yuan, and Pedro V Sander. "Progressive color transfer with dense semantic correspondences". In: *ACM Transactions on Graphics (TOG)* 38.2 (2019), pp. 1–18.

[50] David J Heeger and James R Bergen. "Pyramid-based texture analysis/synthesis". In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM. 1995, pp. 229–238.

[51] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. "Deep Geometric Texture Synthesis". In: *arXiv preprint arXiv: 2007.00074* (2020).

[52] Aaron Hertzmann. "Painterly rendering with curved brush strokes of multiple sizes". In: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM. 1998, pp. 453–460.

[53] 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*. ACM. 2001, pp. 327–340.

[54] Samet Hicsonmez, Nermin Samet, Fadime Sener, and Pinar Duygulu. "DRAW: Deep networks for Recognizing styles of Artists Who illustrate children's books". In: *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM. 2017, pp. 338–346.

[55] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. "Cycada: Cycle-consistent adversarial domain adaptation". In: *International conference on machine learning*. 2018, pp. 1989–1998.

[56] Daniel Holden, Jun Saito, and Taku Komura. "A deep learning framework for character motion synthesis and editing". In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), pp. 1–11.

[57] 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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 783–791.

[58] Hui Huang, Ke Xie, Lin Ma, Dani Lischinski, Minglun Gong, Xin Tong, and Daniel Cohen-Or. "Appearance Modeling via Proxy-to-Image Alignment". In: *ACM Transactions on Graphics (TOG)* 37.1 (2018), pp. 1–15.

[59] Xun Huang and Serge Belongie. "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization". In: *arXiv preprint arXiv: 1703.06868* (2017).

[60] Youngbae Hwang, Joon-Young Lee, In So Kweon, and Seon Joo Kim. "Color transfer using probabilistic moving least squares". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3342–3349.

[61] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. URL: `http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17`.

[62] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[63] Zhongping Ji, Weiyin Ma, and Xianfang Sun. "Bas-relief modeling from normal images with intuitive styles". In: *IEEE transactions on visualization and computer graphics* 20.5 (2013), pp. 675–685.

[64] Zhongping Ji, Xianfang Sun, and Weiyin Ma. "Normal image manipulation for bas-relief generation with hybrid styles". In: *arXiv preprint arXiv: 1804.06092* (2018).

[65] Zhongping Ji, Xianfang Sun, Shi Li, and Yigang Wang. "Real-time bas-relief generation from depth-and-normal maps on GPU". In: *Computer Graphics Forum*. Vol. 33. 5. Wiley Online Library. 2014, pp. 75–83.

[66] Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. "Dynamic instance normalization for arbitrary style transfer". In: *arXiv preprint arXiv: 1911.06953* (2019).

[67] Yongcheng Jing, Yang Liu, Yezhou Yang, Zunlei Feng, Yizhou Yu, Dacheng Tao, and Mingli Song. "Stroke controllable fast style transfer with adaptive receptive fields". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 238–254.

[68] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference on Computer Vision*. Springer. 2016, pp. 694–711.

[69] Bela Julesz. "Visual pattern discrimination". In: *IRE transactions on Information Theory* 8.2 (1962), pp. 84–92.

[70] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[71] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.

[72] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. "Neural 3d mesh renderer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3907–3916.

[73] Jens Kerber, Meili Wang, Jian Chang, Jian J Zhang, Alexander Belyaev, and H-P Seidel. "Computer assisted relief generation—A survey". In: *Computer Graphics Forum*. Vol. 31. 8. Wiley Online Library. 2012, pp. 2363–2377.

[74] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1646–1654.

[75] Diederik Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv: 1412.6980* (2014).

[76] Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. "Stripe patterns on surfaces". In: *ACM Transactions on Graphics (TOG)* 34.4 (2015), pp. 1–11.

[77] 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)*. 2019.

[78] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. "Style transfer by relaxed optimal transport and self-similarity". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10051–10060.

[79] 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)*. 2019.

[80] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[81] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. "Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet". In: *arXiv preprint arXiv: 1411.1045* (2014).

[82] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. "From word embeddings to document distances". In: *International conference on machine learning*. 2015, pp. 957–966.

[83] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. "State of the" Art": A Taxonomy of Artistic Stylization Techniques for Images and Video". In: *IEEE transactions on visualization and computer graphics* 19.5 (2013), pp. 866–885.

[84] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. "Transient attributes for high-level understanding and editing of outdoor scenes". In: *ACM Transactions on Graphics (TOG)* 33.4 (2014), p. 149.

[85] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. "Learning blind video temporal consistency". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 170–185.

[86] Trudy Lawrence. "Relief Sculpture." In: *School Arts the Art Education Magazine for Teachers* 104 (2005), p. 3.

[87] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et

al. “Photo-realistic single image super-resolution using a generative adversarial network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[88] Sylvain Lefebvre and Hugues Hoppe. “Appearance-space texture synthesis”. In: *ACM Transactions on Graphics (TOG)* 25.3 (2006), pp. 541–548.

[89] Anat Levin, Dani Lischinski, and Yair Weiss. “A closed-form solution to natural image matting”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 228–242.

[90] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. “Least squares conformal maps for automatic texture atlas generation”. In: *ACM transactions on graphics (TOG)* 21.3 (2002), pp. 362–371.

[91] Chuan Li and Michael Wand. “Combining markov random fields and convolutional neural networks for image synthesis”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2479–2486.

[92] Chuan Li and Michael Wand. “Precomputed real-time texture synthesis with markovian generative adversarial networks”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 702–716.

[93] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. “Laplacian-Steered Neural Style Transfer”. In: *arXiv preprint arXiv: 1707.01253* (2017).

[94] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. “Learning linear transformations for fast arbitrary style transfer”. In: *arXiv preprint arXiv: 1808.04537* (2018).

[95] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. “Learning linear transformations for fast image and video style transfer”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2019, pp. 3809–3817.

[96] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. “Pointcnn: Convolution on x-transformed points”. In: *Advances in neural information processing systems*. 2018, pp. 820–830.

[97] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. “A Closed-form Solution to Photorealistic Image Stylization”. In: *arXiv preprint arXiv: 1802.06474* (2018).

[98] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. “Diversified texture synthesis with feed-forward networks”. In: *arXiv preprint arXiv: 1703.01664* (2017).

[99] 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, pp. 385–395.

[100] Zhuwen Li, Song Wang, Jinhui Yu, and Kwan-Liu Ma. "Restoration of brick and stone relief from single rubbing images". In: *IEEE Transactions on Visualization and Computer Graphics* 18.2 (2011), pp. 177–187.

[101] Xiaodan Liang, Xiaohui Shen, Donglai Xiang, Jiashi Feng, Liang Lin, and Shuicheng Yan. "Semantic object parsing with local-global long short-term memory". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3185–3193.

[102] Zhiyuan Liang and Jianbing Shen. "Local Semantic Siamese Networks for Fast Tracking". In: *IEEE Transactions on Image Processing* (2019).

[103] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. "Visual Attribute Transfer through Deep Image Analogy". In: *arXiv preprint arXiv: 1705.01088* (2017).

[104] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[105] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. "Image inpainting for irregular holes using partial convolutions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 85–100.

[106] Hsueh-Ti Derek Liu, Michael Tao, and Alec Jacobson. "Paparazzi: surface editing by way of multi-view image processing." In: *ACM Trans. Graph.* 37.6 (2018), pp. 221–1.

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[108] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. "Deep Photo Style Transfer". In: *arXiv preprint arXiv: 1703.07511* (2017).

[109] Tao Luo, Jianbing Shen, and Xuelong Li. "Accurate normal and reflectance recovery using energy optimization". In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.2 (2014), pp. 212–224.

[110] Aravindh Mahendran and Andrea Vedaldi. "Understanding deep image representations by inverting them". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 5188–5196.

[111] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. "Photorealistic Style Transfer with Screened Poisson Equation". In: *arXiv preprint arXiv: 1709.09828* (2017).

[112] Jean-Michel Morel, Ana-Belen Petro, and Catalina Sbert. "Screened Poisson equation for image contrast enhancement". In: *Image Processing On Line* 4 (2014), pp. 16–29.

[113] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. "Scalable parallel programming with CUDA". In: *ACM SIGGRAPH 2008 classes*. ACM. 2008, p. 16.

[114] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning deconvolution network for semantic segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1520–1528.

[115] Dae Young Park and Kwang Hee Lee. "Arbitrary Style Transfer With Style-Attentional Networks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[116] François Pitié, Anil C Kokaram, and Rozenn Dahyot. "Automated colour grading using colour distribution transfer". In: *Computer Vision and Image Understanding* 107.1 (2007), pp. 123–137.

[117] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. "N-dimensional probability density function transfer and its application to color transfer". In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 2. IEEE. 2005, pp. 1434–1439.

[118] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. "The 2017 davis challenge on video object segmentation". In: *arXiv preprint arXiv: 1704.00675* (2017).

[119] Javier Portilla and Eero P Simoncelli. "A parametric texture model based on joint statistics of complex wavelet coefficients". In: *International journal of computer vision* 40.1 (2000), pp. 49–70.

[120] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv: 1511.06434* (2015).

[121] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. "Color transfer between images". In: *IEEE Computer graphics and applications* 21.5 (2001), pp. 34–41.

[122] Eric Risser, Pierre Wilmot, and Connelly Barnes. "Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses". In: *arXiv preprint arXiv: 1701.08893* (2017).

[123] Paul Rosin and John Collomosse. *Image and Video-Based Artistic Stylisation*. Vol. 42. Springer Science & Business Media, 2012.

[124] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. "Artistic style transfer for videos". In: *German Conference on Pattern Recognition*. Springer. 2016, pp. 26–36.

[125] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. "Artistic style transfer for videos and spherical images". In: *International Journal of Computer Vision* (), pp. 1–21.

[126] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. "Artistic style transfer for videos and spherical images". In: *arXiv preprint arXiv: 1708.04538* (2017).

[127] Christian Schüller, Daniele Panozzo, and Olga Sorkine-Hornung. "Appearance-mimicking surfaces". In: *ACM Transactions on Graphics (TOG)* 33.6 (2014), pp. 1–10.

[128] Amir Semmo, Tobias Isenberg, and Jürgen Döllner. "Neural style transfer: a paradigm shift for image-based artistic rendering?" In: *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*. ACM. 2017, p. 5.

[129] Alla Sheffer, Emil Praun, and Kenneth Rose. "Mesh parameterization methods and their applications". In: *Foundations and Trends® in Computer Graphics and Vision* 2.2 (2006), pp. 105–171.

[130] Falong Shen, Shuicheng Yan, and Gang Zeng. "Neural style transfer via meta networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8061–8069.

[131] Jianbing Shen, Xiaoshan Yang, Xuelong Li, and Yunde Jia. "Intrinsic image decomposition using optimization and user scribbles". In: *IEEE transactions on cybernetics* 43.2 (2013), pp. 425–436.

[132] Jianbing Shen, Xiaoshan Yang, Yunde Jia, and Xuelong Li. "Intrinsic images using optimization". In: *CVPR 2011*. IEEE. 2011, pp. 3481–3487.

[133] Jianbing Shen, Xing Yan, Lin Chen, Hanqiu Sun, and Xuelong Li. "Re-texturing by intrinsic video". In: *Information Sciences* 281 (2014), pp. 726–735.

[134] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. "Avatar-net: Multi-scale zero-shot style transfer by feature decoration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8242–8250.

[135] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. "Data-driven hallucination of different times of day from a single outdoor photo". In: *ACM Transactions on Graphics (TOG)* 32.6 (2013), p. 200.

[136] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv: 1409.1556* (2014).

[137] Olga Sorkine, Daniel Cohen-Or, Rony Goldenthal, and Dani Lischinski. "Bounded-distortion piecewise mesh parameterization". In: *IEEE Visualization, 2002. VIS 2002.* IEEE. 2002, pp. 355–362.

[138] Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann, 2002.

[139] Xianfang Sun, Paul L Rosin, Ralph R Martin, and Frank C Langbein. "Bas-relief generation using adaptive histogram equalization". In: *IEEE transactions on visualization and computer graphics* 15.4 (2009), pp. 642–653.

[140] Daniel Sỳkora, Ladislav Kavan, Martin Čadík, Ondřej Jamriška, Alec Jacobson, Brian Whited, Maryann Simmons, and Olga Sorkine-Hornung. "Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters". In: *ACM Transactions on Graphics (TOG)* 33.2 (2014), pp. 1–15.

[141] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. "Local color transfer via probabilistic segmentation by expectation-maximization". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 747–754.

[142] Hai Thien To and Bong-Soo Sohn. "Bas-relief generation from face photograph based on facial feature enhancement". In: *Multimedia Tools and Applications* 76.8 (2017), pp. 10407–10423.

[143] Carlo Tomasi and Roberto Manduchi. "Bilateral filtering for gray and color images". In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 839–846.

[144] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis". In: *arXiv preprint arXiv: 1701.02096* (2017).

[145] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Instance normalization: The missing ingredient for fast stylization". In: *arXiv preprint arXiv: 1607.08022* (2016).

[146] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. "Multi-style Generative Network for Real-time Transfer." In: *ICML*. 2016, pp. 1349–1357.

[147] Phong V Vu and Damon M Chandler. "A fast wavelet-based algorithm for global and local image sharpness estimation". In: *IEEE Signal Processing Letters* 19.7 (2012), pp. 423–426.

[148] Hao Wang, Xiaodan Liang, Hao Zhang, Dit-Yan Yeung, and Eric P Xing. "Zm-net: Real-time zero-shot image manipulation network". In: *arXiv preprint arXiv: 1703.07255* (2017).

[149] Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. "Collaborative Distillation for Ultra-Resolution Universal Style Transfer". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[150] Tuanfeng Y Wang, Hao Su, Qixing Huang, Jingwei Huang, Leonidas J Guibas, and Niloy J Mitra. "Unsupervised texture transfer from images to model collections." In: *ACM Trans. Graph.* 35.6 (2016), pp. 177–1.

[151] Wenguan Wang and Jianbing Shen. "Deep visual attention prediction". In: *IEEE Transactions on Image Processing* 27.5 (2017), pp. 2368–2378.

[152] Wenguan Wang, Jianbing Shen, and Haibin Ling. "A deep network solution for attention and aesthetics aware photo cropping". In: *IEEE transactions on pattern analysis and machine intelligence* 41.7 (2018), pp. 1531–1544.

[153] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. "Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer". In: *arXiv preprint arXiv: 1612.01895* (2016).

[154] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. "Esrgan: Enhanced super-resolution generative adversarial networks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 0–0.

[155] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. "Diversified Arbitrary Style Transfer via Deep Feature Perturbation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 7789–7798.

[156] Mingqiang Wei, Yang Tian, Wai-Man Pang, Charlie CL Wang, Ming-Yong Pang, Jun Wang, Jing Qin, and Pheng-Ann Heng. "Bas-relief modeling from normal layers". In: *IEEE transactions on visualization and computer graphics* 25.4 (2018), pp. 1651–1665.

[157] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. "DeepFlow: Large displacement optical flow with deep matching". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1385–1392.

[158] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. "Transferring color to greyscale images". In: *ACM Transactions on Graphics (TOG)*. Vol. 21. 3. ACM. 2002, pp. 277–280.

[159] Tim Weyrich, Jia Deng, Connelly Barnes, Szymon Rusinkiewicz, and Adam Finkelstein. "Digital bas-relief from 3D scenes". In: *ACM transactions on graphics (TOG)* 26.3 (2007), 32–es.

[160] Pierre Wilmot, Eric Risser, and Connelly Barnes. "Stable and controllable neural texture synthesis and style transfer using histogram losses". In: *arXiv preprint arXiv: 1701.08893* (2017).

[161] Holger Winnemöller, Sven C Olsen, and Bruce Gooch. "Real-time video abstraction". In: *ACM Transactions On Graphics (TOG)*. Vol. 25. 3. ACM. 2006, pp. 1221–1226.

[162] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. "Content-Based Colour Transfer". In: *Computer Graphics Forum*. Vol. 32. 1. Wiley Online Library. 2013, pp. 190–203.

[163] Jing Wu, Ralph R Martin, Paul L Rosin, X-F Sun, Frank C Langbein, Y-K Lai, A David Marshall, and Y-H Liu. "Making bas-reliefs from photographs of human faces". In: *Computer-Aided Design* 45.3 (2013), pp. 671–682.

[164] Jing Wu, Ralph R Martin, Paul L Rosin, X-F Sun, Y-K Lai, Y-H Liu, and Christian Wallraven. "Use of non-photorealistic rendering and photometric stereo in making bas-reliefs from photographs". In: *Graphical Models* 76.4 (2014), pp. 202–213.

[165] Wuyuan Xie, Yunbo Zhang, Charlie CL Wang, and Ronald C-K Chung. "Surface-from-gradients: An approach based on discrete geometry processing". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2195–2202.

[166] Jufeng Yang, Liyi Chen, Le Zhang, Xiaoxiao Sun, Dongyu She, Shao-Ping Lu, and Ming-Ming Cheng. "Historical Context-based Style Classification of Painting Images via Label Distribution Learning". In: *2018 ACM Multimedia Conference on Multimedia Conference*. ACM. 2018, pp. 1154–1162.

[167] Xiaoshan Yang, Tianzhu Zhang, and Changsheng Xu. "Text2Video: An End-to-end Learning Framework for Expressing Text with Videos". In: *IEEE Transactions on Multimedia* (2018).

[168] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. "Attention-Aware Multi-Stroke Style Transfer". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[169] Jae-Doug Yoo, Min-Ki Park, Ji-Ho Cho, and Kwan H Lee. "Local color transfer between images using dominant colors". In: *Journal of Electronic Imaging* 22.3 (2013), pp. 033003–033003.

[170] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. "Photorealistic style transfer via wavelet transforms". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 9036–9045.

[171] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[172] Qiong Zeng, Ralph R Martin, Lu Wang, Jonathan A Quinn, Yuhong Sun, and Changhe Tu. "Region-based bas-relief generation from a single image". In: *Graphical models* 76.3 (2014), pp. 140–151.

[173] Dongbo Zhang, Xiaochao Wang, Jianping Hu, and Hong Qin. "Interactive modeling of complex geometric details based on empirical mode decomposition for multi-scale 3D shapes". In: *Computer-Aided Design* 87 (2017), pp. 1–10.

[174] Hang Zhang and Kristin Dana. "Multi-style Generative Network for Real-time Transfer". In: *arXiv preprint arXiv: 1703.06953* (2017).

[175] Wei Zhang, Chen Cao, Shifeng Chen, Jianzhuang Liu, and Xiaoou Tang. "Style transfer via image component analysis". In: *IEEE Transactions on multimedia* 15.7 (2013), pp. 1594–1601.

[176] Yu-Wei Zhang, Caiming Zhang, Wenping Wang, and Yanzhao Chen. "Adaptive bas-relief generation from 3D object under illumination". In: *Computer Graphics Forum*. Vol. 35. 7. Wiley Online Library. 2016, pp. 311–321.

[177] Yu-Wei Zhang, Yi-Qi Zhou, Xue-Lin Li, Hui Liu, and Li-Li Zhang. "Bas-relief generation and shape editing through gradient-based mesh deformation". In: *IEEE transactions on visualization and computer graphics* 21.3 (2014), pp. 328–338.

[178] Yu-Wei Zhang, Jing Wu, Zhongping Ji, Mingqiang Wei, and Caiming Zhang. "Computer-assisted Relief Modelling: A Comprehensive Survey". In: *Computer Graphics Forum*. Vol. 38. 2. Wiley Online Library. 2019, pp. 521–534.

[179] Yu-Wei Zhang, Yanzhao Chen, Hui Liu, Zhongping Ji, and Caiming Zhang. "Modeling chinese calligraphy reliefs from one image". In: *Computers & Graphics* 70 (2018), pp. 300–306.

[180] Yu-Wei Zhang, Caiming Zhang, Wenping Wang, Yanzhao Chen, Zhongping Ji, and Liu Hui. "Portrait Relief Modeling from a Single Image". In: *IEEE transactions on visualization and computer graphics* (2019).

[181] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. "Multimodal style transfer via graph cuts". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 5943–5951.

[182] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. "Is L2 a Good Loss Function for Neural Networks for Image Processing?" In: *ArXiv e-prints* 1511 (2015).

[183] Mingtian Zhao and Song-Chun Zhu. "Portrait painting using active templates". In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. ACM. 2011, pp. 117–124.

[184] Xiaofei Zhou, Zhi Liu, Chen Gong, and Wei Liu. "Improving Video Saliency Detection via Localized Estimation and Spatiotemporal Refinement". In: *IEEE Transactions on Multimedia* (2018).

[185] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.