Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Graphics and Interaction

Towards Interactive, Robust, and Stereoscopic Style Transfer

Ing. Michal Kučera

A dissertation submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy.

Ph.D. programme: Computer Science

Supervisor: prof. Ing. Daniel Sýkora, Ph.D.

April 2024

Towards Interactive, Robust, and Stereoscopic Style Transfer

Ing. Michal Kučera
kucerm22@fel.cvut.cz
Department of Computer Graphics and Interaction
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo nam. 13, 121 35 Prague 2, CZ

Abstract

Since its inception in the early 2000s, the research field of style-transfer and automatic stylization has seen a steady rise in popularity up to a point where its algorithms are being employed by professional digital artists in their creation process, allowing them to quickly and conveniently stylize images or video sequences based on either a hand-made or a generated example. Even though this research field has seen major strides in recent years, there are still substential issues and limitations preventing larger-scale utilization of such algorithms: limitations such as real-time or interactive stylization of either static images or video sequences, significant quality degradation in cases where example and target keyframes differ too much, temporal coherency of stylized video sequences, infeasible requirements for learning an image-to-image network, as well as stereoscopic applications of style-transfer algorithms remaining uncertain.

In this dissertation thesis we describe the current state-of-the-art in the field of example-based style transfer. Along with that, we propose a set of algorithms that allow interactive production of high quality real-time stylizations of video sequences, both based on semantically meaningful automatic style transfer and keyframe-based learning approaches, on which we introduce new methods to solve the difficult requirement of large paired datasets or domain-specific datasets. We also propose a new method that enables style transfer to still be possible when applied to a stereoscopic scenario.

In particular, we propose: (1) a neural method approximating results of a patch-based style transfer method in real time, (2) an interactive method for real-time style transfer of video sequences, (3) a computationally inexpensive method for real-time stylization of facial videos even on low-end devices, (4) a video style-transfer method greatly improving the output quality and long-term coherence, and finally (5) a method able to achieve stereo-consistent style transfer of video sequences.

Combined together, this thesis makes important steps forward to high-quality, real-time, interactive, temporally and stereoscopically consistent style transfer.

Keywords

computer graphics, machine learning, artistic style transfer, stylization, example-based synthesis, neural style transfer, virtual reality, augmented reality, stereoscopic rendering, nonphotorealistic rendering, digital art, video style transfer

Acknowledgements

Throughout my Ph.D. journey I have met many people that helped me grow both as a researcher and as a person, be it my research coworkers, fellow Ph.D. students, work colleagues and others. While I would like to extend my limitless gratitude to each of them, there are some people I would like to take the time to thank specifically. First and foremost, my Ph.D. supervisor Daniel Sýkora, who initially started me on this journey dating back to my early Master's study years. We have worked together for many years and achieved great progress in the field we are passionate about, for which I am extremely grateful. I'd like to thank both him and prof. Jiří Žára for going above and beyond to create a safe and supportive surroundings for our research and my personal life. Special thanks also belong to my internship supervisors, Menglei Chai and Dominik Kaeser, who have done the same during my stays in the USA and helped me thrive in an environment I would have considered foreign before. Dominik has also greatly helped in repatriating me from the USA during the outbreak of the global COVID-19 pandemic. And last, but certainly not least, my friends and family, who have been there to support me during hard times. For that, a very special thanks goes to my partner, Jana Kyllerová, with whom I have built a life together during this time, and who has been there for me whenever she knew I needed it.

The research presented in this thesis was conducted in collaboration with, and supported by, Adobe Research, Snap Inc. and has further been supported by the Technology Agency of the Czech Republic under research program TE01020415 (V3C – Visual Computing Competence Center), by the Grant Agency of the Czech Technical University in Prague, grants No. SGS13/214/OHK3/3T/13 (Research of Progressive Computer Graphics Methods) and No. SGS16/237/OHK3/3T/13 (Research of Modern Computer Graphics Methods), and by Research Center for Informatics (RCI) No. CZ.02.1.01/0.0/0.0/16_019/0000765.

Směrem k interaktivnímu, robustnímu a stereoskopickému přenosu výtvarného stylu

Ing. Michal Kučera
kucerm22@fel.cvut.cz
Katedra počítačové grafiky a interakce
Fakulta elektrotechnická
České vysoké učení technické v Praze
Karlovo náměstí 13, 121 35 Praha 2

Abstrakt

Výzkum v oblastí automatické přenosu výtvarného stylu se těší rostoucí popularitě od svého zrodu na počátku 21. století. Oceňují jej zejména výtvarníci a animátoři, kterým významným způsobem pomáhá snížit objem repetitivní ruční práce. Na základě jedné ručně kreslené či generované předlohy dokáží efektivně stylizovat řadu dalších obrazů či celé videosekvence. I přes velký pokrok v této oblasti, existuje stále řada problémů, jež brání většímu rozšíření metod pro přenosu výtvarného stylu. Hlavní potíží jsou poměrně vysoké nároky na výpočetní výkon, jež omezují možnost interaktivní práce v reálném čase. Často navíc dochází k významnému úbytku kvality v případech, kdy se předloha a cílové snímky významně liší. Je také obtížné dosáhnout časové koherence a v neposlední řadě není zřejmé, jakým způsobem provádět přenos stylu v případě, kdy se očekává stereoskopické zobrazení.

V této disertační práci nejprve nastíníme současný stav poznání na poli přenosu výtvarného stylu a na jeho základě navrhneme sadu několika nových postupů, které se pokusí překonat výše zmíněná omezení. Konkrétně představíme: (1) metodu založenou na použití neuronové sítě, která umožní stylizovat vstupní video lidské tváře v reálném čase s využitím trénovací sady dat generované pomocí výpočetně náročnějšího postupu, (2) tuto metodu dále zobecníme pro případ libovolné videosekvence, pro kterou existuje jen velmi omezená sada trénovacích dat. Představíme také (3) efektivní aproximaci výpočetně náročnějšího algoritmu pro stylizaci lidských tváří, která umožní provést stylizaci v reálném čase i na méně výkonných zařízeních a (4) metodu pro přenos výtvarného stylu na videosekvence, která významným způsobem zvýší kvalitu výstupu v případech, kdy se snímek na vstupu výrazně liší od klíčového snímku. V závěru popíšeme (5) metodu pro konzistentní přenos výtvarného stylu do stereoskopické videosekvence.

Na základě předložených výsledků srovnání s předchozími přístupy lze konstatovat, že tato práce posunuje současný stav poznání v několika aspektech zkoumané problematiky, ať už se jedná o zvýšením kvality stylizované sekvence, dosažení interaktivní odezvy při stylizaci v reálném čase nebo zajištění konzistence ve stereoskopickém scénáři.

Klíčová slova

počítačová grafika, strojové učení, přenos výtvarného stylu, stylizace, styl podle předlohy, neuronové sítě, nefotorealistické vykreslování, digitální tvorba, přenos stylu na video, virtuální realita, rozšířená realita, stereoskopické vykreslování

Contents

List of Figures

Chapter 1

Introduction

Similarly to any other form of art, digital art, such as movies and video games, has a lot of subtle yet very important nuances that is important to properly understand the evolution that got it into the present day state, and to understand the direction it might be heading in future. Directors never put something in their creation just for the sake of it being there, it always has to fulfill a purpose that is in line with their vision. The chosen camera and lens properties, color schemes, light intensity and direction, composition of a shot and level of detail on objects are all important aspects that help skilled creators convey a specific feeling to the viewer, without it being explicitly said, which is a compelling tool if used properly. Focal length of a lense can make a shot feel either more open and safe, or claustrophobic and tense. Unusual Z-axis angle of a camera can convey a feeling of something being wrong (commonly referred to as the *Dutch angle*). Level of detail and saturation of an object can attract the viewers attention to it, helping guide the viewer to the true experience intended by the creator. And this is exactly what these kinds of media are: an experience.

The world of cinematography has evolved massively since its inception in late 1800s. What first started as a series of moving pictures, which could be somewhat considered a proof-of-concept in today's terms, has evolved into a medium able to convey a story, providing an alternative to books, but allowing the authors to leave less to the reader's imagination and leaving more of the experience in the hands of the author. This led to creation of many movies: comedies, dramas, crime stories, romances, many of them rooted in reality and only capturing a raw, acted scene happening in front of the camera, but a branching genre of movies sought to provide a look into something unnatural, something that may or may not have happened, using special effect. An example of these would be the movies such as *The Execution of Mary, Queen of Scots*, or *A Trip to the Moon*, which used methods such as stop motion, or miniatures with forced perspective, to create something "fake", yet believable. Movies such as *Star Wars Episode IV: A New Hope*, the original *Jurassic Park*, or the *Flight of the Navigator* all sought to tell a fantastic story in the most photorealistic way possible, making it easier for the viewer to connect with the story being told. They achieved that by using many in-camera tricks, scaled miniatures of objects, or realistic life-sized suits, which made scenes look very photorealistic because the object were in fact real.

Photorealism is a crucial aspect in many of today's media, both movies and videogames. As mentioned before, it makes it easier for the viewer to connect with the events happening in front of them, because the environment in which the story is happening is similar

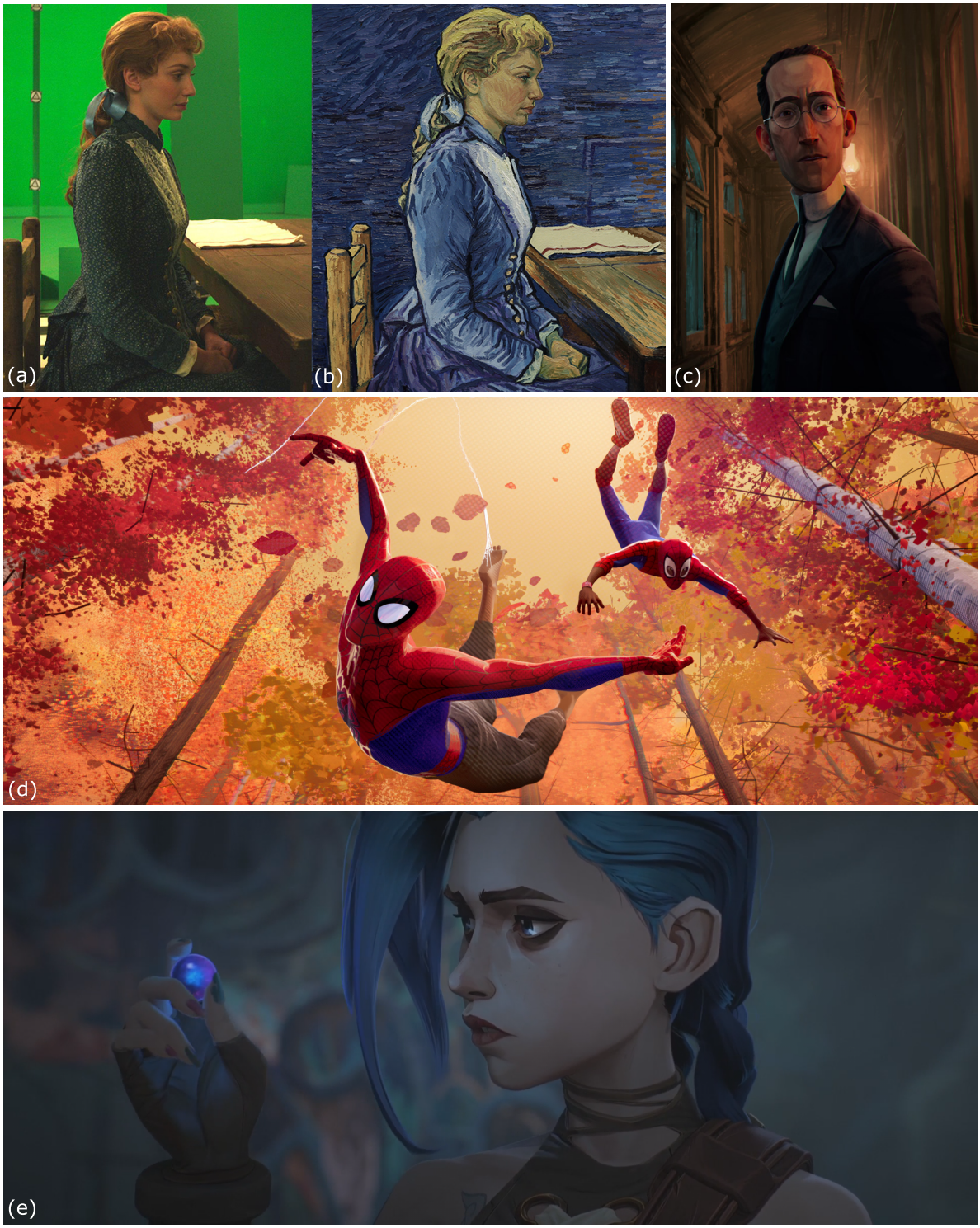


Figure 1.1: *Examples of recent stylized movies. (a) Loving Vincent unstylized captured frame, (b) final stylized frame. (c) Love, Death and Robots. (d) Spider-Man: Across the Spider-Verse. (e) Arcane.*

to the environment they know and everything seems plausible, and while the early 1900's special effect were not particularly convincing for today's standards, the techniques have been significantly refined over time. However, while the common early techniques, such

as prop and model making for movies, are definitely a good way to achieve photorealism, they are not perfect; otherwise we would have no reason to not use them anymore. Making realistic, scaled or life-sized, props and models is an expensive, time-consuming process, allowing very little in terms of flexibility once a prop or a model is done. This forces the artists to make compromises, even though their vision is not perfectly compatible with the assets they have available, they need to make it work somehow, sacrificing bits from the final presented experience and trying to stray as little as possible from the original ideas, either due to props not being up to the quality standard, or simple because the artist changed their mind about certain aspects, which is certainly not uncommon. It is important to understand that artists are often constrained by time and, more importantly, budget: very few artists release their works just from the altruistic desire to share an experience, instead it is considered a product with the goal of generating revenue.

This is where computer generated imagery (CGI) and our work comes in: it provides an alternative approach to practical effects, and gives the artist another tool which they can use. We can insert objects in the scene that were not really there during filming, or composit entire shots that never really happened. This gives the artists a lot of flexibility, allowing to change certain aspects of a shot almost until the last moment, making the resulting work much more faithful to their vision. The last couple of decades in both cinematography and video game industry has been thus defined by the chase of computer generated photorealism. At the beginning, CGI effects could be recognized very easily, but nowadays its hard to recognize what is real and what is CGI, allowing artist to make a true mirror of reality and express their visions freely, and while some visual effects in recent movies or games do not look as convincing as they could, it is often an issue of the production scheduling rather than the limits of the technology. The progress has been so great in fact, you can now recreate some of the CGI scenes from many older movies in a matter of minutes on your own home PC, rather than needing months of time and millions of dollars to do so.

However, pure photorealism can only offer you so much in terms of conveying a feeling and we have been investing so much time in it that it is hard to create a new, exciting experience that is just purely photorealistic. And the same way the original filmmakers in the early days of cinematography sought to create something new that the viewers have not seen before in real life, rather than just a video of a train going by, there is an increasing number of artists trying to not limit themselves by the mainstream photorealism, and instead use a more of an artistic spin. An example would be the *Loving Vincent*, a movie that is entirely painted by hand in the style of Vincent van Gogh. This movie has been quite a visual achievement, providing something fresh and new, but also has shown the infeasibility of hand-paiting of an entire movie, being dreadfully expensive and also extremely time-consuming to make. It became obvious that artistic tools, allowing the creation of such stylized content, were needed, to revolutionize the production process, similar to what CGI did to movie production, as mentioned previously. The demand for a stylized content has, fortunately, not stopped after the release of *Loving Vincent*, and continued with more movies and TV series, such as *Love, Death and Robots*, *Spider-Man: Into the Spider-Verse* or *Apollo 10 1/2: A Space Age Childhood*, which further fueled the work done in the research domain specializing on this exact sort of task: style-transfer. This domain aims to tackle problems and provide solutions to automatic stylization of some graphical content, often based on a hand-made or generated example based on which the target content, an image or a video sequence, should be "reimagined" to look

as if it was created using the particular style of the example. This research subdomain is called example-based style-transfer and it will be the field that we will be focusing on in this thesis. And while example-based style-transfer is not the only subdomain available, with text or description-based style-transfer also being quite prominent in recent years, it is a very popular subdomain since it provides a lot of control over the final look of the output that is difficult to precisely describe otherwise.

Another area relevant to our research is the that of video games, which have been following a similar trend to movies, chasing uncompromising realism for maximal immersion. There are some aspects and priorites with the video-game industry that are different to the movie industry, all stemming from the fact that video games are interactive experiences, they react to the user's actions and decisions. The priority is that the experience is smooth, performance-wise, on most consumer-grade hardware, and less on the raw visual quality. Video games give users a different way to experience a story that is very unique compared to movies, including a fairly modern way: through virtual reality. VR allows for an absolutely different level of immersion in the experience, if used properly. Examples of such an experience would be the games *Half-Life: Alyx*, *Beat Saber* or *A Fisherman's Tale*. While VR and games made for it were successful, it is hardly a success that would rival that of the traditional video games. There are many reasons for this, starting with the price and availability of such VR devices, but also due to other factors such as the *VR sickness*, which is a variation of motion sickness. Even with that, the potential of VR can be observed by the sheer amount of media created for it even now, and the amount of supporting modifications that make the usage easier. And similarly to how movies and traditional video games can benefit from an artistic tool for stylization - giving creators more freedom - the same can be applied to VR. In there, however, the parameters defining a good stylization become a little more difficult due to the binocular disparity and depth perception, when stylizing anything that is not a trivial geometrical plane. While, for example, the details of used strokes, the difference between consecutive frames (temporal coherency), or the amount of identity presevation vs. style reproduction, is up to the artist's choice, the stereoscopic consistency is not, since the lack of it creates very unpleasant artifacts. Whatever the future of VR devices might be, it is worth it investing time into the research of stereoscopic tools for stylizations, because it will remain relevant for as long as we use two eyes for depth perception, regardless of the shape or form of the VR devices which we will end up using.

While the field of non-photorealistic rendering (NPR) and style transfer has been around for some time, it is still an exciting and relatively fresh area for research, with many possibilities still being largely unexplored and many severe issues unaddressed, which is where we hope to contribute with this thesis. For example, with VR, the strong emphasis on interactivity, performance, freedom of artistic expression and strict stereo consistency requirement make it fairly niche. Current real world professional uses work offline, which means that it is not too much of an issue if the algorithm takes its time to produce a result, which is almost entirely incompatible with VR usage and is also a deterring factor preventing wider applications outside of VR. Our goal is to propose algorithms and solution to these individual tasks: real-time performance, interactivity, stereo consistency and output quality with emphasis on freedom of artistic expression, which in itself can serve as a good platform for further research.

In this chapter, we first introduce the field of example-based style-transfer, both using traditional optimization schemes as well as neural networks, and describe the algorithms

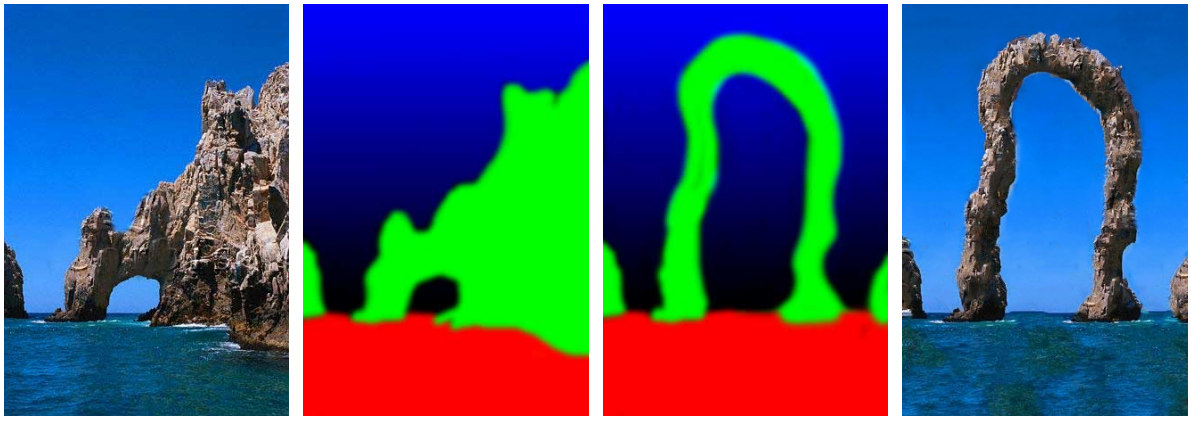
used for solving these tasks. Following that, we define the scope of the contributions of our thesis, which will then be properly described in more detail in the following chapters.

1.1 Introduction to Example-based Style Transfer

(a) *Input content* **(b)** *Input style* **(c)** *Output*

Figure 1.2: *Style transfer example, where (a) is the source content, (b) the source style and (c) the final output. Algorithm used is Gatys et al. [2016].*

The field of style transfer has been a hot topic in research circles for a couple of decades now with many significant advances made in this area. Along with the rapid increase of performance of many widely available devices, both computers and mobile devices, algorithms performing some kind of style transfer are steadily finding their way into commercial applications, giving artists, for example, the ability to quickly prototype artwork sketches and allowing them to sift through many conceptual ideas in a much shorter timeframe than before.

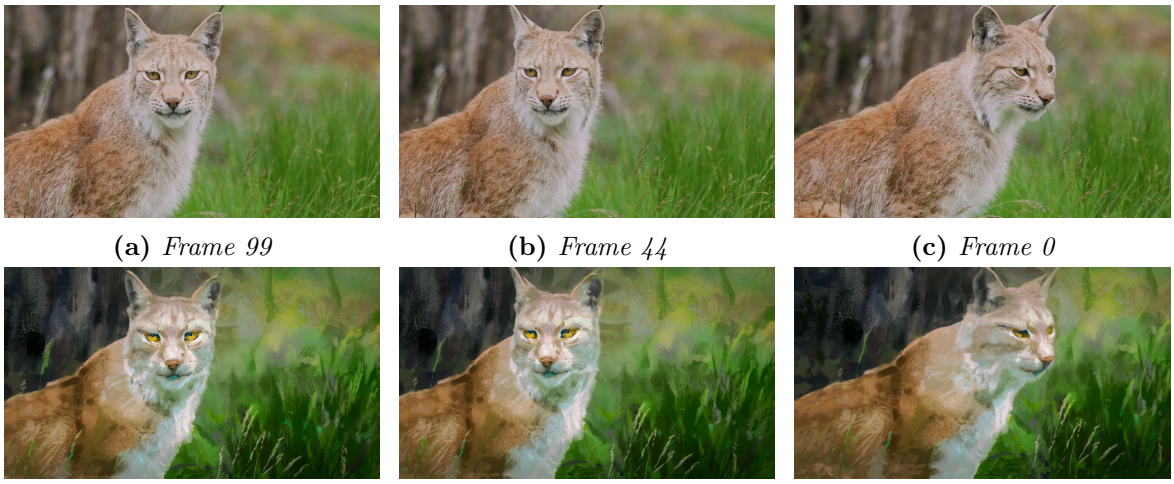


(a) *Source style* **(b)** *Source content* **(c)** *Target content* **(d)** *Output*

Figure 1.3: *In guided style transfer, we are trying the synthesize the output (d) based on the input source style (a), description of source content (b) and description of target content (c).*

The underlying problem of style transfer, specifically example-based, unfortunately lacks an unambiguous definition. What we are generally trying to achieve is given two input images, called input content and input style, to find an image which has the content of the first image, but is drawn in the artistic style of the second image. What is artistic style, however, can be left to interpretation. While we generally understand the style to

be a set of used colors, strokes and other artifacts left by the used medium, the artistic style could in reality also relate to geometric deformations of the captured content, or by the composition itself. While there exists research regarding these particular aspects of style transfer, such as Yaniv et al. [2019], in our research we will limit ourselves only to a subset of aforementioned aspects: color, strokes and artifacts left by the artistic medium. An example of this sort of example-based style transfer can be seen in Figure 1.2.



(a) *Frame 99* **(b)** *Frame 44* **(c)** *Frame 0*

(d) *Input stylized frame 99* **(e)** *Synthesized styled frame 44* **(f)** *Synthesized styled frame 0*

Figure 1.4: *An example of video stylization using the EbSynth algorithm of Jamriška et al. [2019]. Given a target sequence (a-c), we provide a stylized version of one or more of these frames (in this case frame 99 (d)) and the algorithm synthesizes the rest of the sequence using this style (f).*

Based on the nature of the input data we can divide the problems of style transfer into several categories, which can be placed on a spectrum between two extremes. One of these extremes would be arbitrary style transfer, where we do not place any requirements or limitations on the input data. An example of this approach would be the algorithm from Figure 1.2, which is the algorithm of Gatys et al. [2016]. In this approach any arbitrary style can be used to stylize any input content and some kind of output will always be generated. Another examples of the same category would be the approaches of Johnson et al. [2016], Frigo et al. [2016; 2019], and Kolkin et al. [2019]. A very similar approach can be seen in the work by Sanakoyeu et al. [2018] and the follow-up research by Kotovenko et al. [2019b], which allow multiple input style images from which the style is reproduced.

On the other side of the spectrum would be guided style transfer, where we place strict requirements on the input data to extract important guidance information. This allows users to steer the stylization and have greater control over many details of the final product, which is often required in professional commercial usage. In the guided workflow, we provide explicit guidance information by adding a source content input as well, which describes the structure of the input style. The stylization solution then attempts to synthesize a result based on the correlations between the content images. An example of this workflow can be seen in Figure 1.3, showing the output of the method of Jamriška et al. [2019], which can also be used for keyframe-based stylization of video sequences, where we draw over one of the frames of the sequence, making it a paired input with content and style and making other frames in the sequence target contents

(a) *Input content* **(b)** *Input style* **(c)** *Output*

Figure 1.5: *An example of style transfer using the Liao et al. [2017] hybrid approach. Note that the semantical content of both input matches, in this case both contain sailing ships.*

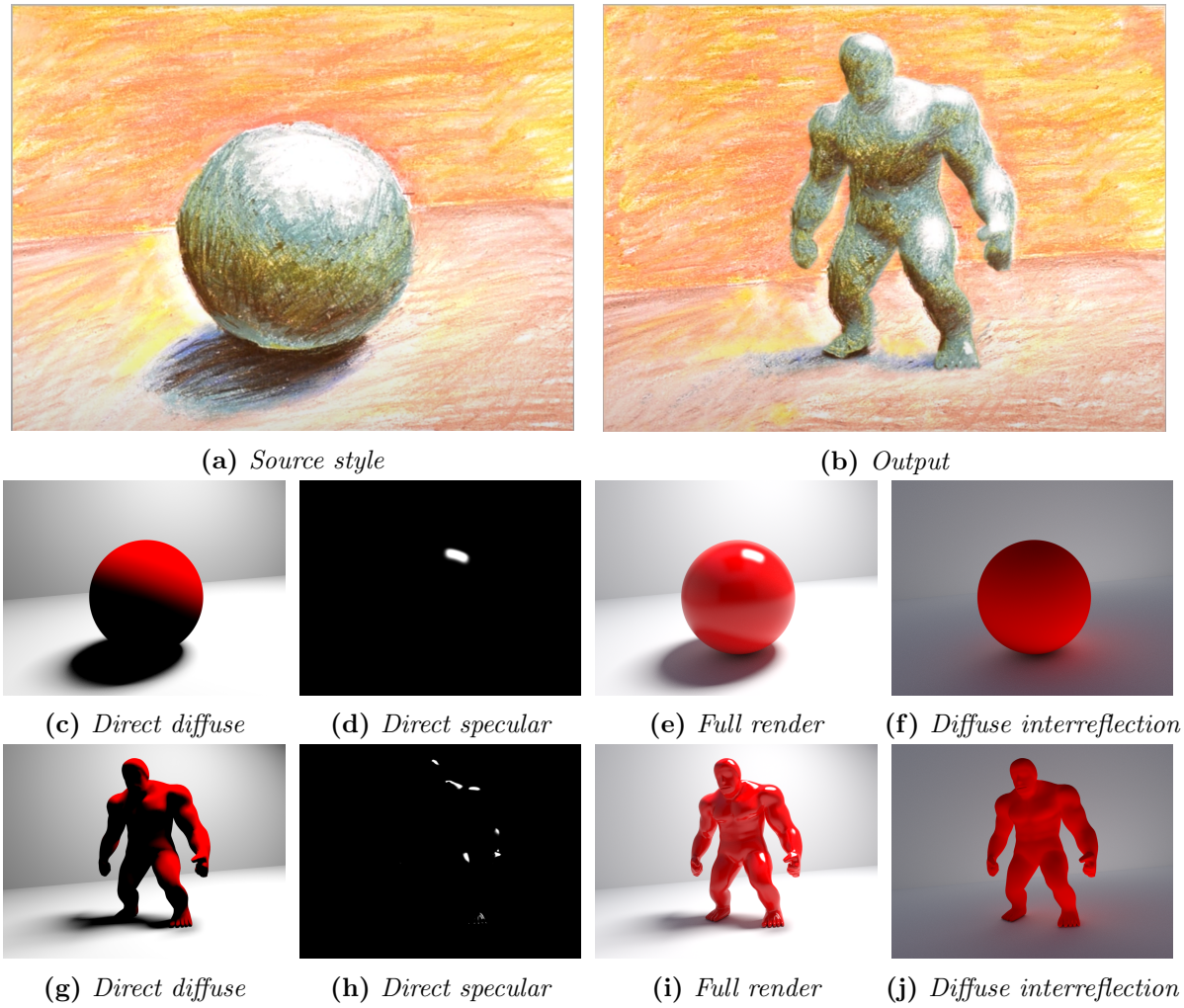
(Figure 1.4). Important approach from the category of guided style transfer would also be the method of Hertzmann et al. [2001], creating a framework that was further utilized in many other guided approaches, such as the work of Jamriška et al. [2015], the work by Fišer et al. [2016] (Figure 1.6) or the work of Fišer et al. [2017] (Figure 1.7).

As a sort of a middle ground between the two extremes, there are methods that impose relaxed requirements on the input data, making them not strictly guided, but also not completely arbitrary. This requirements generally takes the shape of limiting the target domain. Common limitation is to stylize only human faces, where the correspondences are easier to generate implicitly. An example of such an approach would be the algorithm of Selim et al. [2016]. Another common limitation is to require that both the source and target content contain similar semantical information, allowing visual attribute transfer between corresponding areas, as was done by the method of Liao et al. [2017] (Figure 1.5).

1.1.1 Algorithms for Style Transfer

There are many solutions to the aforementioned problems of style transfer, which we can categorize into two groups: patch-based approaches and neural approaches.

One of the first patch-based approaches was introduced in the work of Hertzmann et al. [2001], who proposed the framework named *Image Analogies* where small patches of the source style were transferred to the output following the provided guides. This approach has since been expanded with optimization-based techniques [Kwatra et al. 2005; Wexler et al. 2007; Jamriška et al. 2015; Kaspar et al. 2015] to stylize 3D renders [Bénard et al. 2013; Fišer et al. 2016] (shown in Figure 1.6), facial photos [Fišer et al. 2017] (shown in Figure 1.7), or video frames [Jamriška et al. 2019] (shown in Figure 1.4). However, these approaches require explicit guidance in order to work, which heavily limits their range of uses, and are often based on expensive optimization schemes, which makes usage in real-time scenarios very difficult. Further worth noting, the mentioned video stylization focused paper of Jamriška et al. [2019] is sequential in nature, making random access to stylized frames of the sequence expensive since all the frames from the last keyframe need to be processed and stylized first. This issue is partially addressed in the research done by Sýkora et al. [2019] that allows for real-time generation of outputs approximating those of Fišer et al. [2016] by leveraging specific structures of the input guidance, which however only works for stylization of 3D models.



(a) *Source style* **(b)** *Output*

(c) *Direct diffuse* **(d)** *Direct specular* **(e)** *Full render* **(f)** *Diffuse interreflection*

(g) *Direct diffuse* **(h)** *Direct specular* **(i)** *Full render* **(j)** *Diffuse interreflection*

Figure 1.6: *Example of the guided StyLit algorithm of Fišer et al. [2016]. When provided with the Source style (a), Source guidance (c-f) and Target guidance (g-j), the algorithm is able to generate the stylized output (b), retaining the content of (g-j) and reproducing the style of (a). The guidance data used in this case is based on the light propagation throughout the scene, which is one of the key contributions of Fišer et al. [2016].*

In the neural domain, the general approach is to synthesize an image matching statistics or features extracted from a neural network. This approach has been popularized by the seminal work of Gatys et al. [2016], who used a pre-trained VGG network of Simonyan et al. [2014] to optimize the output image until its VGG responses match the target content and style representations. A feed-forward adaptations of Gatys' algorithm has been introduced by [Johnson et al. 2016; Ulyanov et al. 2016b; Wang et al. 2017; Wilmot et al. 2017], allowing fast and inexpensive style transfer even on mobile devices, albeit for the cost of lengthy precomputation. Another possibility for neural style transfer would be image-to-image translation networks [Isola et al. 2017; Zhu et al. 2017c] which preserve textural details notably better and therefore produce higher quality stylizations, although these approaches require large datasets in order to be trained properly. Common network architecture used for these methods is U-net of Ronneberger et al. [2015]. While the mentioned algorithms work well to mitigate certain limitations of the patch-based approaches, they also suffer from limitations of their own. One of the

most prominent ones is the requirement for a large dataset from which the network can be trained, which can often be very hard to obtain, if not downright impossible, since artists often change and evolve their style and technique between their works.

There are also methods which combine the patch-based and neural approaches to create the output stylization. An example of this approach would be the work of Texler et al. [2020a], running a patch-based synthesis on top of the output of a neural style transfer algorithm, allowing high-fidelity arbitrary stylization even on extremely large images. Work of Liao et al. [2017] would also fall in this category. In this approach, the authors propose a framework called *Deep Image Analogies*, in which they adapt the concept of Hertzmann et al. [2001], but instead of using explicit guides this approach tries to find correspondences in feature representation of both the style and content images, generated using an object-recognition network of Simonyan et al. [2014], which however requires both of the input images to contain semantically similar information for to the output to be semantically meaningful (Figure 1.5).

1.1.2 Our Contribution

While the state-of-the-art methods of example-based style transfer has moved far in the last couple of years, there are still issues that remain to be solved. With patch-based approaches, the most common problem is the performance. As already mentioned, these approaches rely on expensive optimization schemes and take a significant amount of time to render the output, making a real-time application difficult to achieve. Approximative algorithms of these approaches can achieve better performance, but focus themselves only on a small subdomain of the problem. Neural algorithms can also solve the problem of performance but have their own downsides, mostly pertaining to the learning process where they require either a large paired training dataset, which can be difficult to obtain, or a large domain-specific dataset, which only allows stylization to work within that specific domain. Another issue common in both of these categories is the output quality and consistency when applied to a video sequence, where important fine details may degrade over time. To help solve these issues, we present new approaches, which are further described in their respective chapters.

In Chapter 3 we propose *FaceStyleGAN* [Futschik et al. 2019], a neural algorithm reproducing the output quality of the work of Fišer et al. [2017], even in cases where the original algorithm fails. Moreover, since our proposed network can be evaluated quickly it allows for real-time stylization, whereas the original algorithm took a long time to generate a single output due to its costly texture optimization process. In this approach, we utilize a U-net-based network architecture of Ronneberger et al. [2015], using a combination of $\mathcal{L}_1$, adversarial [Mao et al. 2017] and perceptual losses [Simonyan and Zisserman 2014] to train a translation network, creating paired data using the approach of Fišer et al. [2017].

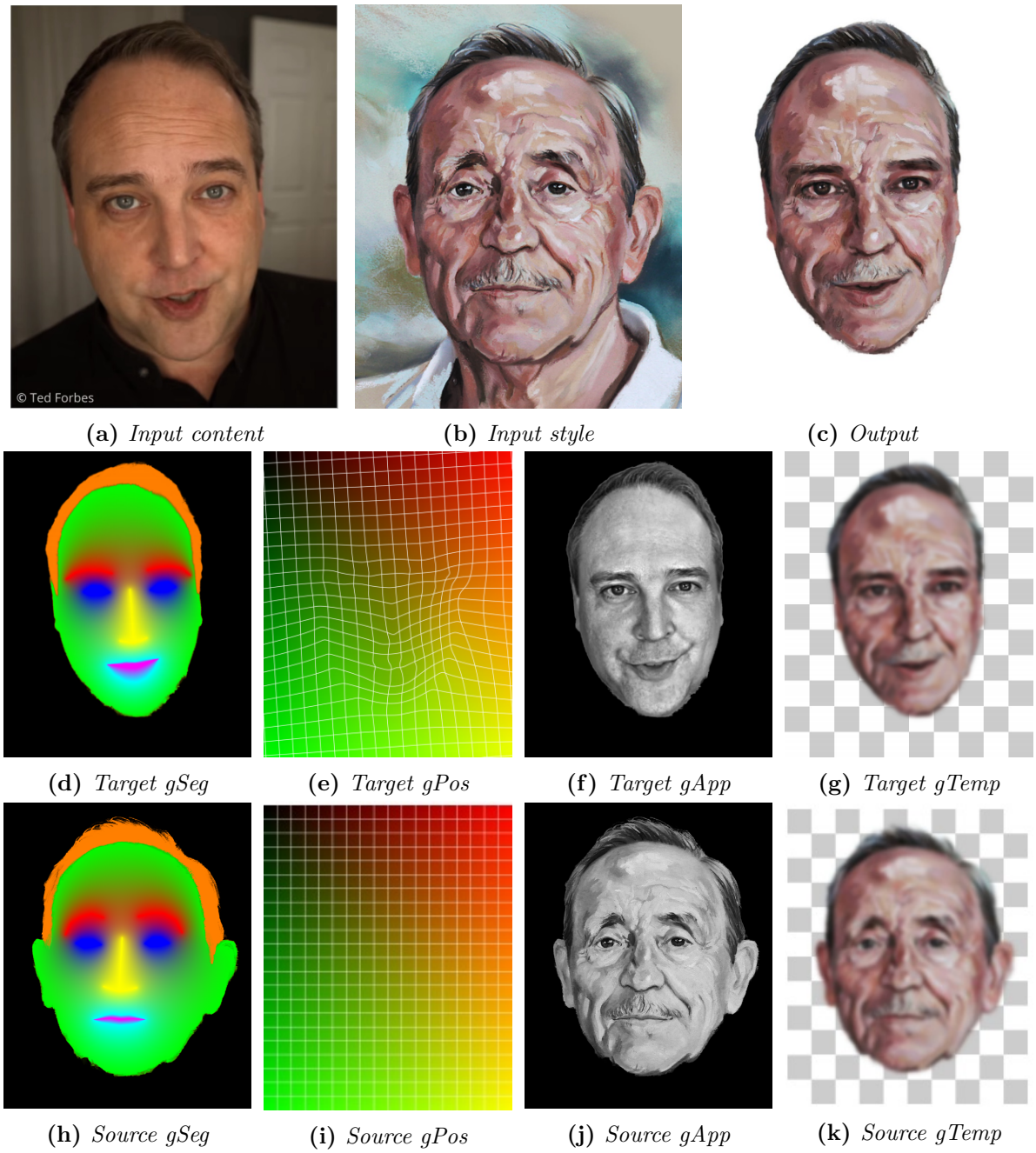
In the same Chapter, we also introduce *FaceBlit* [Texler et al. 2021], a guided patch-based approach to face stylization similar to that of Fišer et al. [2017]. We employ StyleBlit of Sýkora et al. [2019], allowing real-time response even on low-end devices without any lengthy pre-calculation.

In Chapter 4, we propose an approach to interactive video stylization based on few-shot patch-based training strategy [Texler et al. 2020b], where we introduce the usage of deliberately small batches of cropped patches as a means of overfitting prevention as one

of our key contributions, somewhat similar to the limited receptive field training seen in the works of Li et al. [2016a] or Ulyanov et al. [2016b]. Doing that, we can achieve real-time interactive style transfer that does not require restrictive sequential stylization within the sequence to propagate the style such as state-of-the-art method of Jamriška et al. [2019]. This upgrade allows parallel processing and random access to stylized frames, and also omits the need for large paired datasets or a domain-specific dataset.

In Chapter 4 we also propose *STALP* [Futschik et al. 2021], that explores a different way to prevent training overfitting with a small training dataset. Instead of restricting the learning to only a handful of patches at a time, we give the network the entire image at once and also measure style loss between individual stylized frames in a sequence using a VGG network of Simonyan et al. [2014]. This allows for higher fidelity stylization and prevents many artifacts where hand-drawn details get lost over time.

And finally, in Chapter 5, we present our contribution to the domain of stereoscopic style-transfer called *StyleBin* [Kučera et al. 2022]. In this work we propose a patch-based optimization method able to synthesize stylized sterescopically-consistent video sequences from an input video, one or more stylized keyframes, and one or more input disparity keyframes. This allows style-transfer scenarios and workflows to also be applicable when targeting a stereoscopic device, such as a VR headset or a 3D monitor, since the method explicitly eliminates stereoscopic inconsistencies, which prevented existing style-transfer methods to be applied in such a way, while also achieving higher output quality compared to neural solutions.



(a) *Input content* **(b)** *Input style* **(c)** *Output*

(d) *Target gSeg* **(e)** *Target gPos* **(f)** *Target gApp* **(g)** *Target gTemp*

(h) *Source gSeg* **(i)** *Source gPos* **(j)** *Source gApp* **(k)** *Source gTemp*

Figure 1.7: *Example of the guided FaceStyle algorithm of Fišer et al. [2017]. Given the Input content (a) and Input style (b), the algorithm recreates the identity of the person in image (a) in the style of (b). To guide the transfer, guides are created for both the target (d-g) and source (h-k) images. These are: gSeg, segmentation guide; gPos, positional guide from warped facial landmarks; gApp, grayscale image with histogram matching applied; and gTemp, blurred previous frame shifted with optical flow for temporal coherence.*

Chapter 2

Related Work and State-of-the-Art

The first attempts to perform non-photorealistic rendering [Kyprianidis et al. 2013], i.e., recreating an input image or a video with a specific artistic style, use hand-crafted algorithmic solutions. Some methods compose the final result using a library of predefined assets, e.g., pen and ink strokes [Salisbury et al. 1997; Praun et al. 2001; Snavely et al. 2006], hatching [Breslav et al. 2007], or brush strokes [Litwinowicz 1997; Hays and Essa 2004; Schmid et al. 2011; Zhao and Zhu 2011]. Others try to mimic the given artistic medium by employing physical simulation [Curtis et al. 1997; Haevre et al. 2007; Lu et al. 2012], or using hand-crafted shaders on the GPU [Bousseau et al. 2006; 2007; Bénard et al. 2010; Montesdeoca et al. 2018]. While these techniques are able to produce faithful stylization to some extent, their use is limited to a certain look given by the predefined visual style.

In Chapter 1 we have already outlined the main approaches providing solutions to the challenges of style-transfer. In this chapter we will reiterate some of the key work that we base our research off of and describe in more depth the already existing research pertaining to style-transfer and texture synthesis, starting with the more traditional patch-based approaches which laid the foundation for the field of style-transfer as we know it now, and then following with neural-based approaches which have gained massive popularity in recent years. We also include a description of the related work specifically in the stereoscopic domain, which is pertinent to our own research.

(a) *A* **(b)** *A‘* **(c)** *B* **(d)** *B‘*

Figure 2.1: *Image analogy as defined by Hertzmann et al.: A‘ relates to A the same way B‘ relates to B. In this case, A‘ is a blurred version of A. When these 2 are provided, along with the image B, image B‘ can be computed by using these defined analogies without the need to use the same filter applied to A‘, or even needing to know what filter it is.*

2.1 Patch-based Approaches

A cornerstone of guided texture synthesis was the work of Hertzmann et al. [2001], which presented a framework that is further expanded in many solutions to the style transfer problem. In this work, Hertzmann et al. defined the problem of *Image Analogies*, where given a set if input images A, A‘ and B, we’re trying to find an output image B‘ that relates to B the same way A‘ relates to A. This is further described in Figure 2.1. The authors then proposed a greedy guided patch-based algorithm, which could reproduce many filters as well as do some level of style transfer with paired input data. Wexler et al. [2007] and Kwatra et al. [2005] formulated optimization-based approach to texture synthesis which replaced greedy approach of Hertzmann et al. [2001] and was further embraced by research that followed. Benard et al. [2013] have proposed a keyframe-based stylization of video sequences, using a set of auxiliary guiding channels generated by a 3D renderer using a patch-based guided synthesis similar to the work of Hertzmann et al. [2001]. In 2015, works of Kaspar et al. [2015] and Jamriska et al. [2015] presented solutions to the problem formulated by Kwatra et al. [2005], trying to mitigate the effects of the wash-out effect, that was common in style transfer algorithms, by enforcing uniform patch usage. While this was a definite improvement, it only worked best on inputs where the same semantical content covered roughly the same area in both the source and the target. Disparity between these two would cause patches from one semantical area being forced to an area with entirely different semantical information, creating various artifacts.

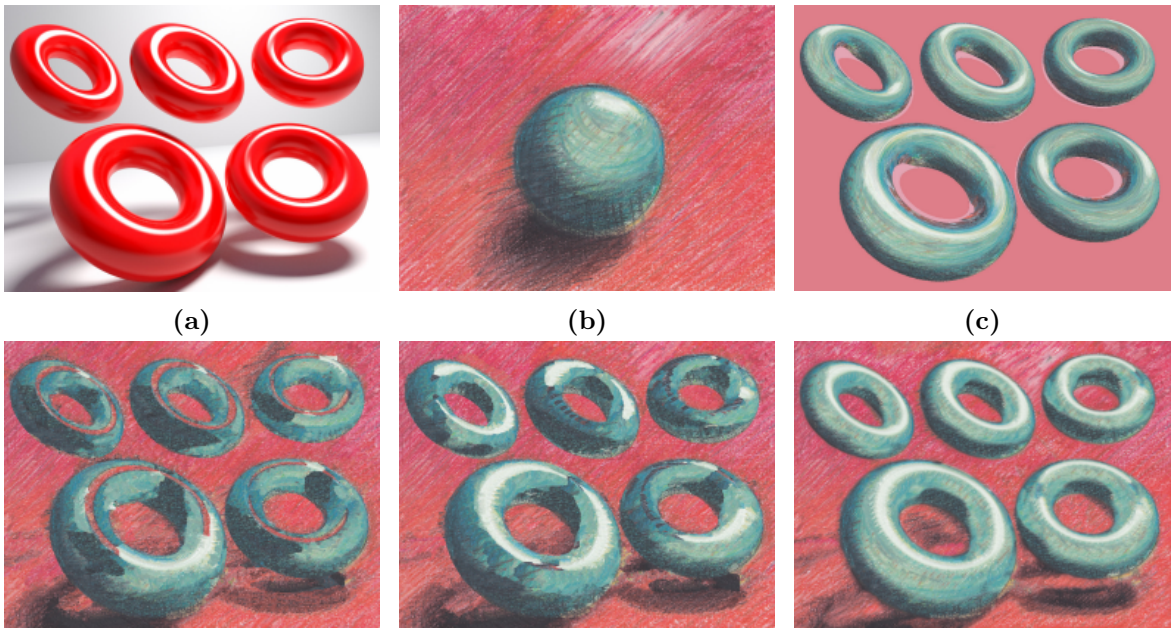


Figure 2.2: *Comparison of used guiding channels for the style transfer. (a) shows the source content and (b) the source style. (c) is the algorithm of Sloan et al. [2001], which uses the normals as guidance. (d) is the algorithm of Hertzmann et al. [2001] which uses RGB as the guiding channel. (e) is the same algorithm, but this time using LPEs as guiding channels. (f) is the optimized algorithm by Fišer et al., which also uses LPEs. All these images were taken from the StyLit paper Fišer et al. [2016]*

This issue was addressed by the work of Fišer et al. [2016], which introduced a mechanism where the uniform patch enforcement would reset if the assignment would produce too high of an error, allowing for some patches to be used multiple times, solving the issue of previous research. This work also proposed a new set a guiding channels for guided style transfer, based on light propagation throughout the scene rather than plain color as done in Hertzmann et al. [2001] or geometrical normals, as presented by Sloan et al. [2001]. The newly proposed guiding channels allowed to correctly guide the stylization in places where others struggled before, such as geometrical normals being unable to correctly place shadows and plain color guidance mistaking background with highlights (shown more closely in Figure 2.2). Since this set of guiding channels only worked on 3D renders, it was further extended in the follow-up research by Fišer et al. [2017] to work with human faces by introducing a new set of guiding channels. This time, the guidance would not need to be supplied by the user, but instead it was generated automatically using information from facial landmarks, which required both the target content and input style to contain a human face which a landmark detector could recognize. This research also extended its scope to videos and tackled the issue of temporal consistency, which is an important factor when dealing with a video, especially then, when reproducing an artistic style. As shown by Fišer et al. [2014], the temporal instability can be desired effect as it natural for traditional hand-painted animations. However, in certain cases or when stylizing long sequences, it can cause dizziness and be disturbing. To explicitly enforce temporal consistency, various patch-based methods [Bénard et al. 2013; Fišer et al. 2017; Dvorožňák et al. 2018; Jamriška et al. 2019; Frigo et al. 2019] were developed; they consider relations between individual frames od the video sequence.

The work of Jamriska et al. [2019] further expanded the research in style transfer video domain by using the video frames themselves as guidance by having the user provide a stylized exemplar of one or more of the video frames, making a paired input. This has improved the output quality significantly compared to other video stylization approaches, such as the work of Chen et al. [2013] using optical flow between consecutive frames or Li et al. [2019] and Wang et al. [2019c] using dense correspondences. Sýkora et al. [2019] has shown that the outputs of costly texture optimization using local guidance can be approximated inexpensively, so that it can be run in real-time even on single-core CPUs. However, in a face stylization scenario, this method requires a specific type of guidance that is costly to compute and thus the entire stylization cannot run in real-time.

2.2 Neural-based Approaches

The neural approaches to this problem gained significant popularity after the work of Gatys et al. [2016], that allowed unguided artistic style transfer from unpaired input content and style by reproducing responses from a VGG network of Simonyan et al. [2014]. They optimize the target image until its VGG responses match the style image as well as the target content. Such optimization is, however, computationally demanding and thus others [Johnson et al. 2016; Ulyanov et al. 2016c;a; Wang et al. 2017; Ulyanov et al. 2017; Wilmot et al. 2017] later propose to pre-calculate a larger dataset in a particular style, and then train a feed-forward network that is able to reproduce the stylized output notably faster. Although those approaches can perform stylization in real-time they still require lengthy pre-processing. Moreover, neural techniques also tend to omit

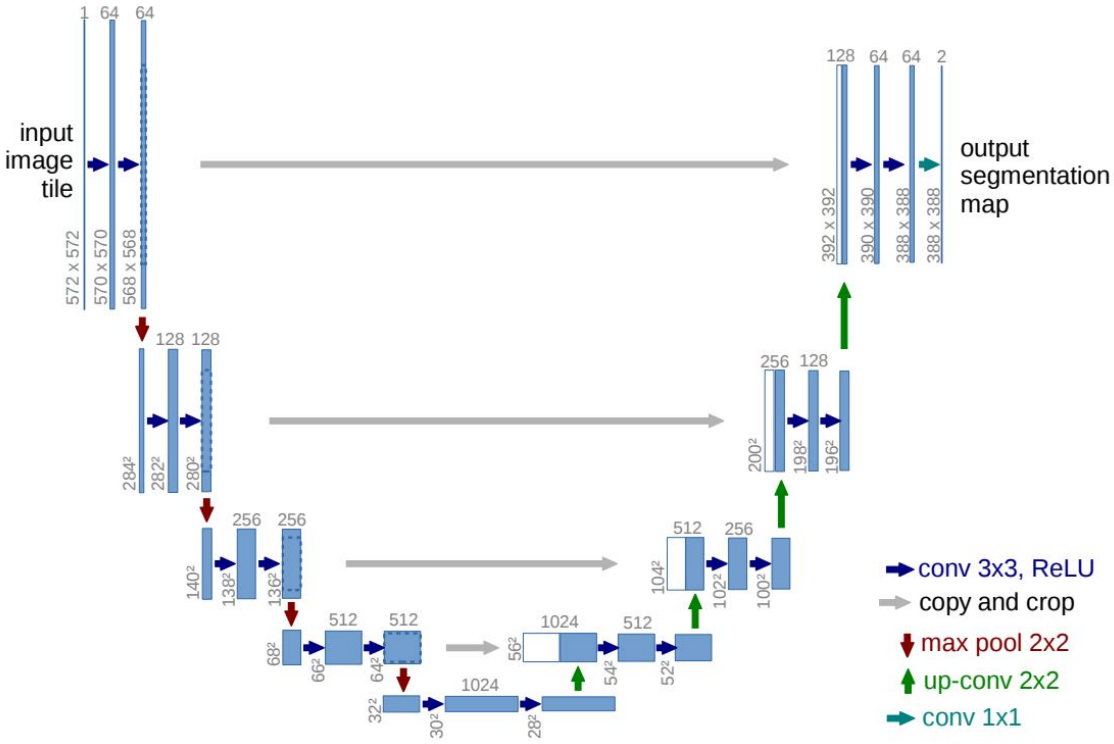


Figure 2.3: *The U-Net architecture proposed by Ronneberger et al. [2015]. The input image is first converted into feature space through a series of progressively smaller convolutional layers, and the converted back into an image following a similar process. Each blue box corresponds to a multi-channel feature map and arrows denote the different operations.*

important textural details presented in the original style exemplar and the transfer is not semantically meaningful, and because of that their ability to generalize and increase the robustness was not as apparent. The tendency to generalize and improve upon the original training dataset has been recently reported also in the case where corrupted datasets are used for training [Bora et al. 2018; Lehtinen et al. 2018]. In these works authors observed the ability of a generative network to recover from failures and produce comparable or sometimes even better visual quality as compared to a scenario when a clean dataset is used for training. The approach of Gatys et al. has been later extended into video domain by the papers of Kotovenko et al. [2019b] and Ruder et al. [2018].

There are also various successful attempts to combine patch-based synthesis and neural style transfer. Li et al. [2016b] search for neural patches in a style image while following the structure of a content image. Liao et al. [2017] and Gu et al. [2018] later extended this approach to perform patch-based synthesis directly in the domain of latent neural feature spaces, and then reconstruct the final image using deconvolution. Recently, Cao et al. [2018] proposed to perform geometric exaggeration on top of appearance transfer. Despite the impressive results, these techniques still suffer from common pixel-level artifacts which lead to lower quality of the synthesized imagery as compared to patch-based methods which can work directly image domain and preserve important pixel-level details. Texler et al. [2020a] propose to use patch-based synthesis method on top of the neural-based style transfer approach. In this setting, they are able to generate high-resolution stylized imagery which would be difficult for the original neural network.

Their method is able to convincingly preserve important texture details of the style exemplar. But semantically meaningful results are still not guaranteed as the method relies on the output from the underlying neural network.

Ronneberger et al. [2015] propose an image-to-image translation network architecture called *U-Net* (Figure 2.3), and while originally targeted for segmentation of biomedical data, it is well-suited for applications in the stylization domain. With arise of generative adversarial networks, such as the work of Goodfellow et al. [2014], research proposing the usage of image-to-image translation network has been introduced by Isola et al. [2017] and Zhu et al. [2017a][2017b], which offers real-time semantically meaningful style transfer, but unfortunately also requires a large learning dataset to be able to reproduce a style compellingly, which can be hardly accessible in a generic video stylization scenario, where only a few hand-drawn exemplars exist, let alone in the context of video-to-video translation [Chan et al. 2019; Tulyakov et al. 2018; Wang et al. 2018c] which is completely intractable. Research by Liu et al. [2019] and Wang et al. [2019b; 2019a] have tried to alleviate this issue by introducing few-shot learning techniques, which still unfortunately require a large domain-specific dataset for pre-training. Approaches based on deformation transfer [Siarohin et al. 2019a;b] are proposed to animate target photo in real-time using only a single exemplar. A key limitation of these approaches is that they transfer only coarse deformation characteristics while the identity of the subject in the driving video is often omitted. A similar drawback also holds for approaches based on generative adversarial networks such as Style- GAN v2 of Karras et al. [2020]. In this approach, a massive collection of artworks is used to train a network that can generate an artistic image for a given input latent vector. Those vectors can then be predicted and fine-tuned to align the generated image with the target image features. However, this process is inaccurate, leading to imprecise alignment that hinders ability of the network to preserve the structure or identity of the target subject.

A path to relax the requirements for domain-specific pre-training or a large dataset can be found in the works of Li et al. [2016a] or Ulyanov et al. [2016a], which train a network with limited receptive field on a single exemplar image and then use it to infer larger textures that retain essential low-level characteristics of the exemplary image. This potential venue is explored by our proposed method described in Section 4.1. Recently, the idea of patch-based training was further explored to accelerate training by Shocher et al. [2018] or to maintain high-level context [Shaham et al. 2019; Shocher et al. 2019; Zhou et al. 2018]; however, all those techniques deal only with a single image scenario, and video domain extension remains an open problem.

Similarly to the patch-based techniques, temporal consistency poses an important problem which needs to be addressed, which has been done in the neural domain papers of Chen et al. [2017], Gupta et al. [2017], Sanakoyeu et al. [2018] or Ruder et al. [2018]. To deal with temporal consistency in the post-process step, a blind temporal coherency method of Lai et al. [2018] stabilizes arbitrary input video sequence. Mullapudi et al. [2019] propose an approach where labelling is provided for a subset of frames by a more accurate predictor and then propagated the the rest of the sequence using a quickly trained lightweight network. To deliver sufficient quality, a relatively large number of keyframes is necessary. Also, full-frame training is employed which could suffer from strong overfitting artifacts and thus is not applicable in scenarios where a detailed texture needs to be propagated. Another successful approximation to patch-based synthesis is recently introduced by Hauptfleisch et al. [2020]. They pre-calculate

the latent representation of the stylized image in a sparse set of samples and then merge nearby pre-calculated representations to reconstruct the final stylized image during the interactive session. Although such an approach can deliver similar quality as full-fledged optimization, it requires costly pre-processing and works only on 3D models.

One of the most common scope limitation in style-transfer in general is a limitation to just human faces, which is a long-standing challenge for non-photorealistic rendering research community. In this domain, traditional filtering-based stylization techniques [Gooch et al. 2004; Tresset and Leymarie 2005; DiPaola 2007; Yang et al. 2010] have been extensively used to deliver compelling results for simple styles. However, they do not allow for greater appearance variations. Example-based techniques can be used to alleviate this limitation. One possible solution is to compose the final image using a set of stylized facial components prepared by an artist [Chen et al. 2003; 2002; 2004; Meng et al. 2010; Zhang et al. 2014]. Although this approach provides greater freedom for local regions, it is still challenging to preserve the identity of the target person due to the inability to adapt the templates to the unique geometry of target facial features. To overcome this drawback, researchers further propose to prepare a larger dataset of photo-style exemplary pairs (e.g., CUHK Face Sketch Database of Wang et al. [2009]), and then use multi-scale Markov Random Fields [Wang and Tang 2009; Li et al. 2011; Zhou et al. 2012; Wang et al. 2013; 2014] to estimate the stylization for a given target face. Although these techniques can deliver better identity adaption, they are highly impractical since many photo-style exemplars need to be prepared manually for each new artistic style. The success of the method of Gatys et al. [2016] motivated others [Selim et al. 2016; Lu et al. 2017] to develop custom neural-based stylization techniques for human portraits. Although those example-based methods can achieve generally compelling results, they usually fail on more complex structured exemplars where preserving high-frequency details is critical.

To achieve an arbitrary style transfer using a network trained on unpaired examples, encoder-decoder schemes are proposed [Huang and Belongie 2017; Li et al. 2017; Lu et al. 2017]. In this setup, an encoder, usually a subset of convolutional layers of the VGG network, is used to extract feature representation from both style and content image. These features are then combined and fed through the decoder, which is pre-trained to convert features into the image space. In a similar spirit, more complex encoder-decoder schemes are proposed by Kotovenko et al. [2019b; 2019a]. They are able to convincingly transfer even finer textural details. Nevertheless, as they measure only statistical correlations between the stylized image and the original image, semantically meaningful transfer is not guaranteed.

2.3 Stereoscopic Style Transfer

The research described in previous sections, however, little considers stereo images, which is a specialized topic with its own literature. Stavrakis et al. [2004] were the first to consider computer-generated stylized stereo images and identified many of the challenges in stylized stereo, including the need for planarity of style elements in the output. They used a stroke-based rendering system, ensuring consistency by enforcing similar stroke placement across right and left views. Northam et al. [2012] propose a more general framework for stylized stereo images which uses multiple discrete disparity layers and a

Figure 2.4: *Comparison of input style (left) to the right view of the output of Chen et al. [2018] (right). While the color scheme and some low-level characteristics of the style have been transferred, much of the fine detail is lost or muddled in the transfer, creating an output that somewhat resembles the output style, but does not faithfully reproduce it.*

separate stylization for each layer. While this approach was effective for still images, the discretization of layers is problematic for application to video. Considerable effort has also been directed towards synthesizing stereo line drawings. Kim et al. [2013] laid the groundwork for this area, working with 3D geometry as an input. They note that the simple approach of detecting and rendering silhouettes separately from each eye creates incoherent collections of lines. By rendering only matched pairs of lines and excluding lines that cannot be fused in stereo, they are able to create a high-quality experience of 3D stereo line drawings from geometry. Later work [Bukenberger et al. 2018; Istead and Kaplan 2018; Istead et al. 2021] produced stylized line drawings from stereo depth images. Other researchers have considered also specialized systems for particular effects and scenarios, such as film grain in stereo by Templin et al. [2014] or stylization of lightfields by Egan et al. [2021]. Application of neural style transfer to stereo images or to generation of novel views has enjoyed some success recently [Chen et al. 2018; Gong et al. 2018; Huang et al. 2021]. Such systems incorporate estimates of stereo or multi-view consistency into the loss function. However, the resulting stylization does not guarantee semantically meaningful transfer and also distorts visually important features seen in the original exemplar such as individual brush strokes or a canvas structure (Figure 2.4). Luo et al. [2015] use a patch-based approach for coherence-preserving modification of stereo images. However, they do not consider stereoconsistent example-based stylization of videos, which remains an open problem.

Chapter 3

Real-Time Stylization of Portraits

The stylization of human portraits becomes highly attractive thanks to the massive popularity of selfie photography and invention of mobile applications such as MSQRD or Snapchat which use facial landmarks together with CG rendering pipeline to deliver stylized look. This approach, however, requires professional artists to carefully design textured 3D models along with custom shaders to achieve the desired look.

This limitation can be alleviated using example-based approaches pioneered by Hertzmann et al. [2001]. This technique allows transferring style from a given artistic exemplary image to a target photo. State-of-the-art in this domain uses neural-based techniques [Selim et al. 2016], patch-based synthesis [Fišer et al. 2017], and their combinations [Liao et al. 2017] to deliver impressive stylization results. However, a key limitation of those techniques is that they consist of several algoritmic steps each of which may be a source of potential failure (see Figures 3.5, 3.6, and 3.7, two right columns) and introduces algoritmic complexity which leads to huge computational overhead, as well as difficult real-time applications.

In this Chapter, we propose two methods able to achieve real-time style transfer on facial video sequences. In Section 3.1 we propose a neural approach, able to learn an artistic style from a premade paired dataset and then perform the stylization inexpensively on GPU. In Section 3.2 we instead focus on a patch-based approach based on the paper done by Sýkora et al. [2019], where we precompute a lookup table based on facial landmarks, which we can then use to perform the facial stylization quickly even on low-end hardware.

Figure 3.1: *Given an input exemplar and a target portrait photo, we can generate stylized output with comparable or superior visual quality as compared to several state-of-the-art face stylization methods (Fišer et al. [2017], Liao et al. [2017], Selim et al. [2016], and Gatys et al. [2016]) while being able to run at interactive frame rates on a consumer GPU. Style exemplar:* © Scary Zara Mary.

3.1 Neural-based Approach

Generative adversarial networks [Goodfellow et al. 2014] have become a favorite technique for image-to-image translation tasks [Isola et al. 2017; Wang et al. 2018b;c] recently. Their principal drawback over classical style transfer techniques which require only a single style exemplar image [Gatys et al. 2016] is the necessity of training the network on a large dataset of paired appearance exemplars. This requirement is prohibitive in the case of artistic style transfer as tedious manual work is necessary to prepare the training dataset. Although unpaired alternatives exist [Zhu et al. 2017a;b] they still require many drawings of a particular style as an input. Another issue is related to the fact that current image-to-image network architectures have difficulties in reproducing delicate high-frequency details that are important to retain fidelity of used artistic media.

To address this, we present *FaceStyleGAN* [Futschik et al. 2019] which combines benefits of state-of-the-art high-quality patch-based synthesis with the power of image translation networks. Thanks to the ability of patch-based method of Fišer et al. [2017] to produce high-quality results we can generate a dataset which preserves the original artistic style precisely. We then use this dataset to train a variant of image-to-image translation network with improved structure that better preserves important high-frequency details. Although the method of Fišer et al. is prone to failure in more complex cases, we leverage the fact that the network can generalize even when the training dataset contains many failure exemplars. This behavior was recently demonstrated in a different context of generative models trained from partially observed samples [Bora et al. 2018] or without ground truth counterparts [Lehtinen et al. 2018]. Thanks to this ability to generalize while still being able to preserve high-frequency details, we can produce results which are comparable or sometimes more visually pleasing than the output of the original patch-based method. Moreover, since the trained network can be evaluated quickly on the GPU our approach enables real-time style transfer which was unattainable for previous high-quality techniques.

3.1.1 Our Approach

Our goal is to learn a mapping function F between color images of human faces $\mathbb{X}$, and their stylized counterparts $\mathbb{Y}$. Since in our case paired data can be produced easily using the algorithm of Fišer et al. [2017], we can model the mapping as a direct transformation $F : \mathbb{X} \rightarrow \mathbb{Y}$.

Given pairs of training samples: $(x_i, y_i)_{i=1}^{N}$ where $x_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$, our objective to learn F contains three different terms: *adversarial loss* $\mathcal{L}_{GAN}$ for matching the distribution of generated images to the distribution of the stylized images [Goodfellow et al. 2014], a *color loss* calculated directly on the stylized output $\mathcal{L}_1$, and finally a *perceptual loss* $\mathcal{L}_{VGG}$ calculated on features extracted by a VGG network pre-trained on ImageNet [Simonyan and Zisserman 2014]. In the following section we focus on each loss in more detail and state the final objective function. Then we describe our network architecture and discuss implementations details.

Adversarial Loss We apply adversarial loss to the output of the mapping function F and its discriminator D_Y using the following objective function:

$$\begin{aligned}\mathcal{L}_{GAN}(F, D_Y, \mathbb{X}, \mathbb{Y}) = \mathbb{E}_{y \sim p_{data}(y)}\big[\big(D_Y(y) - 1\big)^2\big] \\ + \mathbb{E}_{x \sim p_{data}(x)}\big[(D_Y(F(x))^2)\big]\end{aligned} \tag{3.1}$$

where instead of traditional binary cross entropy $\mathcal{L}_2$ norm is used as the adversarial criterion. This leads to a more stable training [Mao et al. 2017].

Color Loss While adversarial loss alone could be enough to learn mapping F, we observed that when an additional $\mathcal{L}_1$ loss [Isola et al. 2017] is computed between the output of the network and the original stylized image we can encourage the generator to better preserve identity as well as stabilize and speed up the training:

$$\mathcal{L}_1(F) = \mathbb{E}_{X,Y \sim p(X,Y)}||Y - F(X)||_1 \tag{3.2}$$

Perceptual Loss Additional improvement can be achieved using perceptual loss that is calculated on feature maps of the VGG-19 model pre-trained on ImageNet at different depths:

$$\mathcal{L}_{VGG}(F) = \sum_{d \in D} ||VGG_d(Y) - VGG_d(F(X))||_2 \tag{3.3}$$

where D is the set of depths of VGG-19 which are considered, in our case $D = 0, 3, 5, 10$. Similar approach was used also in Wang et al. [2018b], however, Wang et al. used $\mathcal{L}_1$ norm which we found has notably lower impact on the final visual quality as compared to our $\mathcal{L}_2$ norm (see Figures 3.2a and 3.2c).

Objective Using all mentioned losses our final objective function is as follows:

$$\mathcal{L}(F, D_Y, X, Y) = \lambda_1 \mathcal{L}_{GAN} + \lambda_2 \mathcal{L}_1 + \lambda_3 \mathcal{L}_{VGG} \tag{3.4}$$

where $\lambda_1, \lambda_2, \lambda_3$ influence the relative importance of the different loss functions.

3.1.2 Network Architecture

For our generator model we use the initial architecture from the work of Johnson et al. [2016], three convolution blocks (two of them with stride = 2) which are followed by several residual blocks [He et al. 2016], two upsampling blocks and finally a tanh activation. Compared with Johnson et al.'s solution, we make the following modifications (see Fig. 3.3) which we observed had a significant impact on the final perceptual quality: we changed the size of convolutional filters in the very first layer from 9×9 to 7×7 and in the very last layer of the original architecture from 9×9 to 5×5. We increased the number of residual blocks used from five to nine. Next, we added skip connections using concatenation of feature maps [Ronneberger et al. 2015] to the upsampling layers, which has been shown to improve gradient propagation, and we replace convolutions with fractional strides with nearest neighbor upsampling followed by an additional 3×3 convolution. Lastly, we attached two more convolutional layers before the output, which we observed have positive effect when the skip connections are added. All these

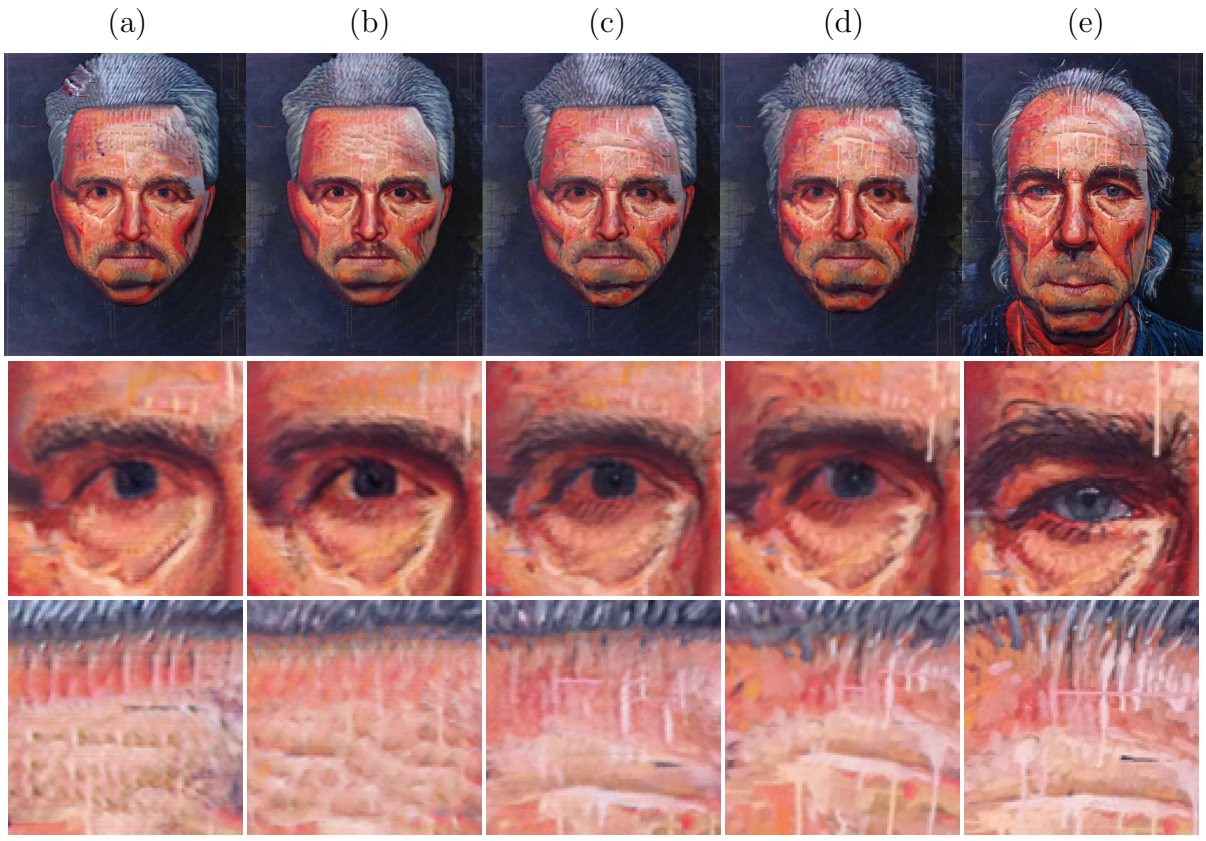


Figure 3.2: *Ablation study. A demonstration of visual quality improvement achieved using modified VGG loss and our improved network architecture: (a) result of our network trained without using VGG loss, (b) result generated using all losses, however, without our improved network architecture, i.e., using the original architecture of Johnson et al. [2016], (c) our result, (d) result generated using FaceStyle algorithm of Fišer et al. [2017], (e) style exemplar. Note how our full-fledged approach better reproduces the original style exemplar (see the avoidance of artificial repetitive patterns on forehead as well as sharper details around eyes) and also slightly improve upon the output of FaceStyle algorithm (c.f. better preservation of important facial features like ears or nose). Style exemplar:* © Matthew Cherry *via http://matthewivancherry.com/home.html and https://www.instagram.com/matthewivancherry.artist (HAT, oil on canvas, 48" x 48", 2011).*

modifications helped to preserve important high-frequency details in the generated image (see visual quality improvement over the initial generator's structure in Figures 3.2b and 3.2c).

For our discriminator model we use PatchGAN model [Isola et al. 2017] using progressively higher number of feature maps with instance normalization proposed by Ulyanov et al. [2016b] and leaky ReLUs as activation. This helped us to lower the number of parameters and achieve a more stable gradient propagation.

3.1.3 Implementation Details

We implemented our approach using C++ and the Python framework PyTorch.

For FaceStyle algorithm we used settings recommended in the original paper of Fišer et al. [2017]. For each artistic style we produced a training set of 5124 stylized facial

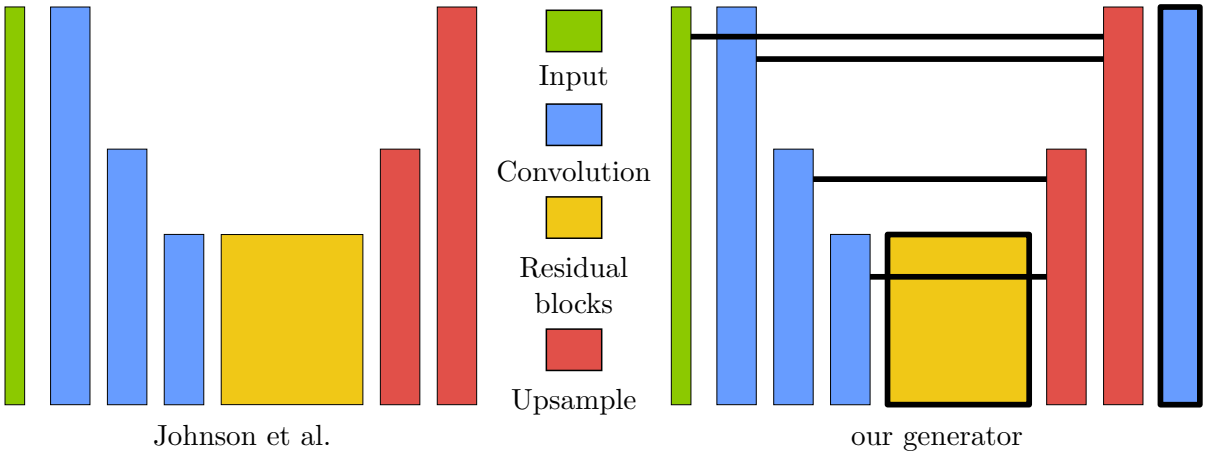


Figure 3.3: *The original generator network architecture of Johnson et al. [2016] (left) followed by our improved architecture (right). Modifications are denoted with black color: added skip connections, increased the number of residual blocks, two upsampling layers are followed by additional transposed convolution layer.*

images in a resolution of 512×512 which is supported by our network architecture. We used automatic portrait segmentation [Shen et al. 2016] to assure the training algorithm focus more on important facial parts of the input image. Since we did not pre-filter the dataset the resulting set of samples contains both successful as well as failure exemplars (c.f. two right columns in Figures 3.5, 3.6, and 3.7 to see examples of such failures).

For training of our models we used the Adam solver [Kingma and Ba 2014] with a batch size of 2. In total, our generator model has 14.7 million parameters, and our discriminator has total number of parameters of 694 thousand. We set $\lambda_1 = 0.3$, $\lambda_2 = 5$, and $\lambda_3 = 0.7$, which were chosen experimentally via grid search and manual tuning. Both generator and discriminator networks were trained from scratch with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $lr = 0.0002$. During the training phase we found that we could use as few as 2000 samples without significant loss of quality. Sufficiency of lower number of training samples can be explained by limited complexity of the appearance changes in the stylized output. We train our models in 50 epochs. Some styles proved to be more challenging to learn, and thus we allowed training in 100 epochs. In general, training for one epoch took around 83 minutes on a single NVIDIA Tesla P100 GPU, making the total training time for one style slightly shorter than 3 days.

3.1.4 Results

We trained our network on seven different style exemplars (see Figures 3.1, 3.2, 3.4 and 3.9) and applied it to 24 portraits not included in the training dataset. In Figures 3.1, 3.2, 3.5, 3.6, 3.7, and 3.9 results of our trained network are compared with the original FaceStyle algorithm [Fišer et al. 2017].

In the following sections, we discuss potential of our method to perform real-time high-quality style transfer, we also mention its ability to generalize and increase robustness over the original FaceStyle algorithm [Fišer et al. 2017] and describe a perceptual study we conducted to evaluate visual quality of our approach with respect to the output generated by FaceStyle algorithm. Finally, we compare our results with current state-of-the-art.

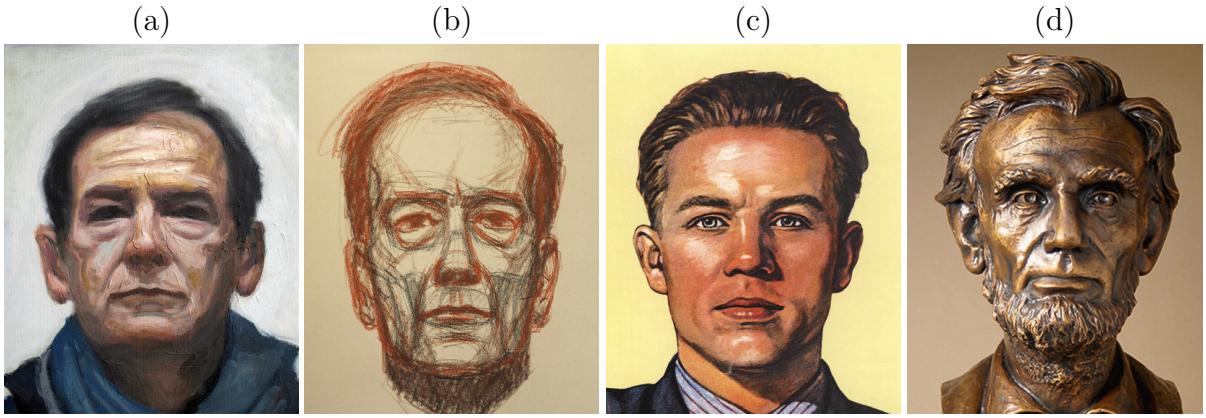


Figure 3.4: *Exemplars of styles used in Figures 3.6, 3.7, and 3.8. See Figures 3.1, 3.2, and 3.9 for the remaining style exemplars. Style exemplars: (a–b)* © Adrian Morgan*, (c)* Viktor Ivanovich Govorkov*, (d)* © Will Murray*.*

Figure 3.5: *Face stylization results. In each group of three images, from left to right, we show the input image, our stylization result, and the output from FaceStyle [Fišer et al. 2017]. The corresponding style exemplars are visible in Figures 3.1 and 3.2.*

3.1.5 Interactive Scenario

Thanks to the compactness of our network (47MB) we can perform feed-forward propagation in real-time (15 frames per second) on currently available consumer graphics cards (we use GeForce RTX 2080 Ti). This benefit enables us to implement the first high-quality style transfer on live video streams (please refer to our supplementary video). We can downsize our network architecture to 256 × 256 resolution (along with reduc-

Figure 3.6: *Face stylization results (continued). In each group of three images, from left to right, we show the input image, our stylization result, and the output from FaceStyle [Fišer et al. 2017]. The corresponding style exemplars are visible in Figure 3.4.*

ing the number of filters in each layer) and also achieve interactive response on mobile devices without significant loss of visual quality.

3.1.6 Generalization

During the training experiments we found that when we deliberately filter out failure exemplars from the training dataset the overall visual quality does not increase significantly, however, the robustness of the resulting trained network decreases. This behavior bears resemblance to findings reported by Lehtinen et al. [2018] although in our case the nature of corruption cannot be modelled by zero-mean noise, we can characterize this tendency as a convergence to an equilibrium which expresses a "mean" of stylized appearance. Thanks to this behavior the trained network can in practice repair failures of the original FaceStyle algorithm. In cases when the FaceStyle algorithm produces correct result our network can deliver stylization which is comparable or sometimes even more visually pleasing and better preserving the identity of a stylized person (see Figures 3.1, 3.5, 3.6, 3.7, 3.2, and 3.9).

Another important aspect of the equilibrium mentioned above is that it helps to preserve coherent stylization when the target image does not change considerably. This tendency is essential for achieving temporal coherency. In contrast to FaceStyle algorithm or other video stylization techniques [Chen et al. 2017; Ruder et al. 2018] that would require explicit treatment of consistency between adjacent frames our technique handles temporal coherency implicitly (see accompanying video demo).

Figure 3.7: *Face stylization results (continued). In each group of three images, from left to right, we show the input image, our stylization result, and the output from FaceStyle [Fišer et al. 2017]. The corresponding style exemplars are visible in Figures 3.4 and 3.9.*

3.1.7 Perceptual Study

To confirm the quality of results produced by our approach are comparable to those produced by the original FaceStyle algorithm [Fišer et al. 2017] we conducted a perceptual study. The study had the form of an online questionnaire, where we showed each user the input face, input style, and the output. We asked the user to rate the output in two categories: how well does the stylization preserve the identity of the stylized person, and how well does the stylization reproduce the input style. The ratings were from 1 to 10, 1 being the worst and 10 being the best. The questionnaire featured 6 sets of input images and their outputs for both of the tested methods, making a total of 12 image sets showed to users, which were all being rated in the 2 categories. We deliberately selected results which are comparable with no obvious failures. During the time the questionnaire was open, we have collected 194 responses.

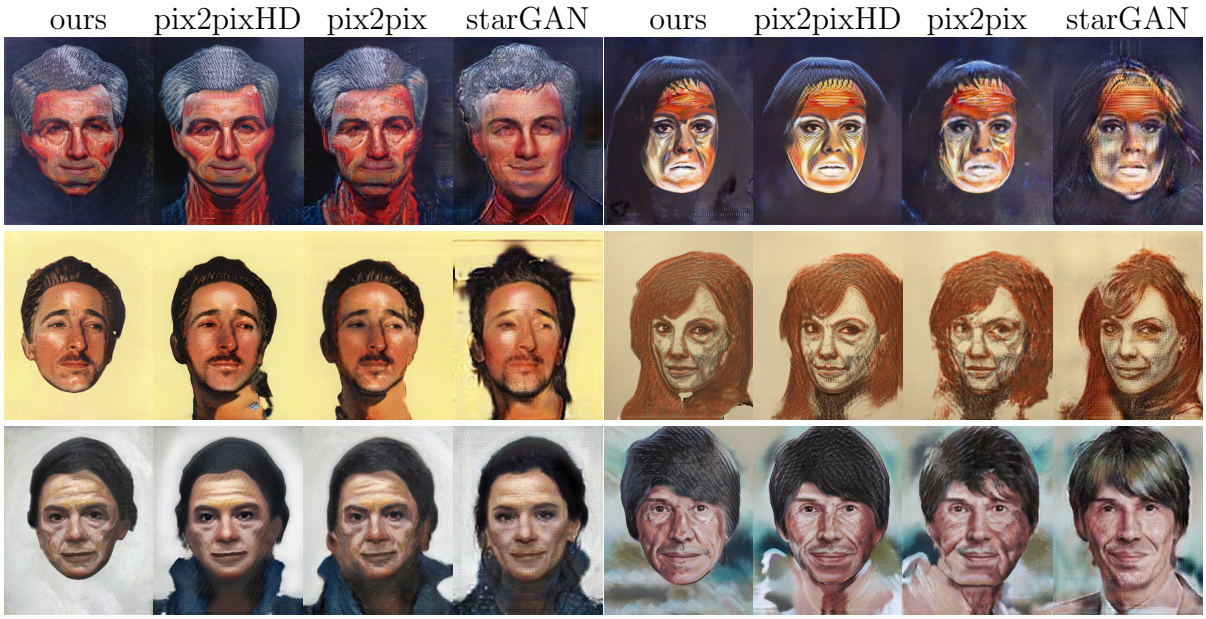


Figure 3.8: *Comparisons of our approach with current state-of-the-art in image-to-image translation:* pix2pixHD *[Wang et al. 2018b],* pix2pix *[Isola et al. 2017], and* starGAN *[Choi et al. 2018]. Note, how our combination of losses and a specific network architecture better preserve the original style exemplar. The corresponding style exemplars are visible in Figures 3.1, 3.2, 3.4, and 3.9.*

Figure 3.9: *Comparisons of our approach with current state-of-the-art face stylization methods. Note how our technique can deliver comparable visual quality to the original FaceStyle algorithm of Fišer et al. [2017] while significantly outperforms other concurrent neural-based techniques (Liao et al. [Liao et al. 2017], Selim et al. [Selim et al. 2016], and Gatys et al. [Gatys et al. 2016]). Style exemplar:* © Graciela Bombalova-Bogra.

We started with the null hypothesis that there is no statistically significant difference between the quality of the outputs of both tested methods, which we tried to reject based on the collected data using the Student's t-test. In the question of identity preservation, we can reject the null hypothesis with a probability of only 49%, which means there is no statistically significant difference between the scores in this category. Our approach scored an average of 6.76 points and FaceStyle scored an average of 6.87 points, which totals to approximately 1% difference on the 1 to 10 scale, supporting the conclusion of both methods being on par with each other. In regard to the style reproduction category, using the same procedure we can reject the null hypothesis with a probability of 63%, which once again does not represent a significant statistical difference. Our approach scored an average of 8.28 points and FaceStyle scored an average of 8.55 points, making

only 3% difference. From these results, we can conclude that the outputs of our approach are on par with the outputs of FaceStyle with only minor differences in the overall quality.

3.1.8 Comparisons

We compared the visual quality of our approach with current state-of-the-art in image-to-image translation (see Fig. 3.8). For training, we used the same dataset as for our method and tweak the parameters to get as close as possible to the appearance of the original style exemplar. Results produced by *pix2pix* method [Isola et al. 2017] bear a resemblance to our output concerning the ability to preserve the target person's identity. Nevertheless, the network produces several high-frequency artifacts which affect texture details of the original style exemplar. A part of the problem is caused by the fact that the pix2pix network supports only lower resolution (256×256), however, more importantly, the structure of pix2pix generator tends to introduce various uncanny high-frequency patterns. This issue becomes even more apparent in the case of *pix2pixHD* [Wang et al. 2018b] which can support 512×512 resolution, nevertheless, at high frequencies, it still contains disturbing repetitive patterns which are not present in the original style exemplar. The *StarGAN* method [Choi et al. 2018] roughly preserves basic facial structure, but it also introduces disturbing high-frequency patterns on top of various low-frequency anomalies which give rise to soft color transitions that are not visible in the original style exemplar.

We also compared our approach with concurrent neural-based techniques that do not require training (see Figures 3.1 and 3.9). From the comparison it is apparent that the generic neural-based technique of Gatys et al. [2016] has difficulty in preserving semantically meaningful transfer. Selim et al. [2016] provide an improvement over Gatys et al., nevertheless, they still suffer from a loss of critical visual details. Deep image analogies [Liao et al. 2017] produce compelling results concerning visual details, but they often fail to keep the consistency of high-level features which affect the identity of the target subject.

3.2 Patch-based Approach

While we have shown the strength of neural-based approaches to portrait stylization in the previous section, they tend to omit textural details in the style exemplar that are critical to the preservation of the visual characteristics in the artistic media. Those techniques also do not guarantee a semantically meaningful transfer, i.e., the use of specific local stylization decisions made by an artist in the exemplar image (e.g., use of a certain type of strokes around the mouth area).

On the other hand, although style transfer techniques powered by patch-based methods [Fišer et al. 2016; 2017] can preserve the textural richness and deliver high-quality semantically meaningful results, they are computationally expensive due to their optimization nature. This issue is partially addressed by a faster synthesis algorithm of Sýkora et al. [2019] that provides a real-time approximation to the fully-fledged optimization by leveraging the specific structure of the guiding channels used in the context of face stylization [Fišer et al. 2017]. Despite this great improvement, the time needed to compute the appearance guidance still hinders the real-time performance, which is

the reason that Sýkora et al. are not able to demonstrate real-time style transfer that preserves the identity of the target subject.

In this section, we present a method that allows for real-time stylization of an arbitrary facial video using a single stylized exemplar instantly without lengthy pre-calculation. To achieve this, we modify the existing example-based stylization method of Fišer et al. [2017] to compute guidance that is compatible with the fast synthesis method of Sýkora et al. [2019] yet still enables identity preservation of the target subject. To verify the practical utility of the proposed method we implemented the entire stylization pipeline which runs on a moderate mobile device in real-time, and achieves comparable stylization quality with previous techniques.

3.2.1 Our Approach

The input to our method is a style exemplar image S of a human portrait and a target face video sequence T. The assumption is that the face changes its expression, moves but is mostly looking towards the camera, and is not occluded by other objects. The output of our method is a stylized sequence O that retains important artistic features of S while preserving the identity of the target subject. Although such an output can already be produced using, e.g., a method of Fišer et al. [2017] a key drawback here is that their approach is suitable only for offline processing. To achieve real-time performance we need to change the way how guiding channels are computed and also replace the slow patch-based synthesis algorithm of Fišer et al. [2016] with its faster variant proposed by Sýkora et al. [2019].

In the paper of Fišer et al. [2017] four guiding channels are used to drive the synthesis. A segmentation guide G_{seg} that delineates important facial features by subdividing the face into a set of regions (hair, eyebrows, nose, lips, oral cavity, eyes, and skin) and a positional guide G_{pos} that encodes spatial correspondences between the source and target face. Those two channels ensure semantically meaningful transfer (i.e., strokes used to depict, e.g., eyes in S are used to stylize eyes in T as well). To preserve the identity of the target subject Fišer et al. employs an appearance guide G_{app} which reduces domain gap between the source and target image by equalizing their appearance using the photographic style transfer method of Shih et al. [2014]. Finally, a temporal guide G_{temp} represented by a motion-compensated version of the previously stylized frame is used to enforce temporal consistency.

Since the computation of guiding channels mentioned above takes tens of seconds on a desktop, their use is not tractable for our real-time scenario. Instead, we reduce those four channels into two essential G_{pos} & G_{app} (see Fig. 3.10), and change their underlying generation algorithms to reduce the preparation time to tens of milliseconds. Finally, we demonstrate how to plug those two new guiding channels into a fast synthesis algorithm of Sýkora et al. [2019].

3.2.2 Positional Guide

A key role of the positional guide G_{pos} is to ensure style consistency, i.e., encourage the synthesis to transfer patches from the source exemplar to a semantically meaningful location in the target image. The existence of the positional guide in the set of guid-

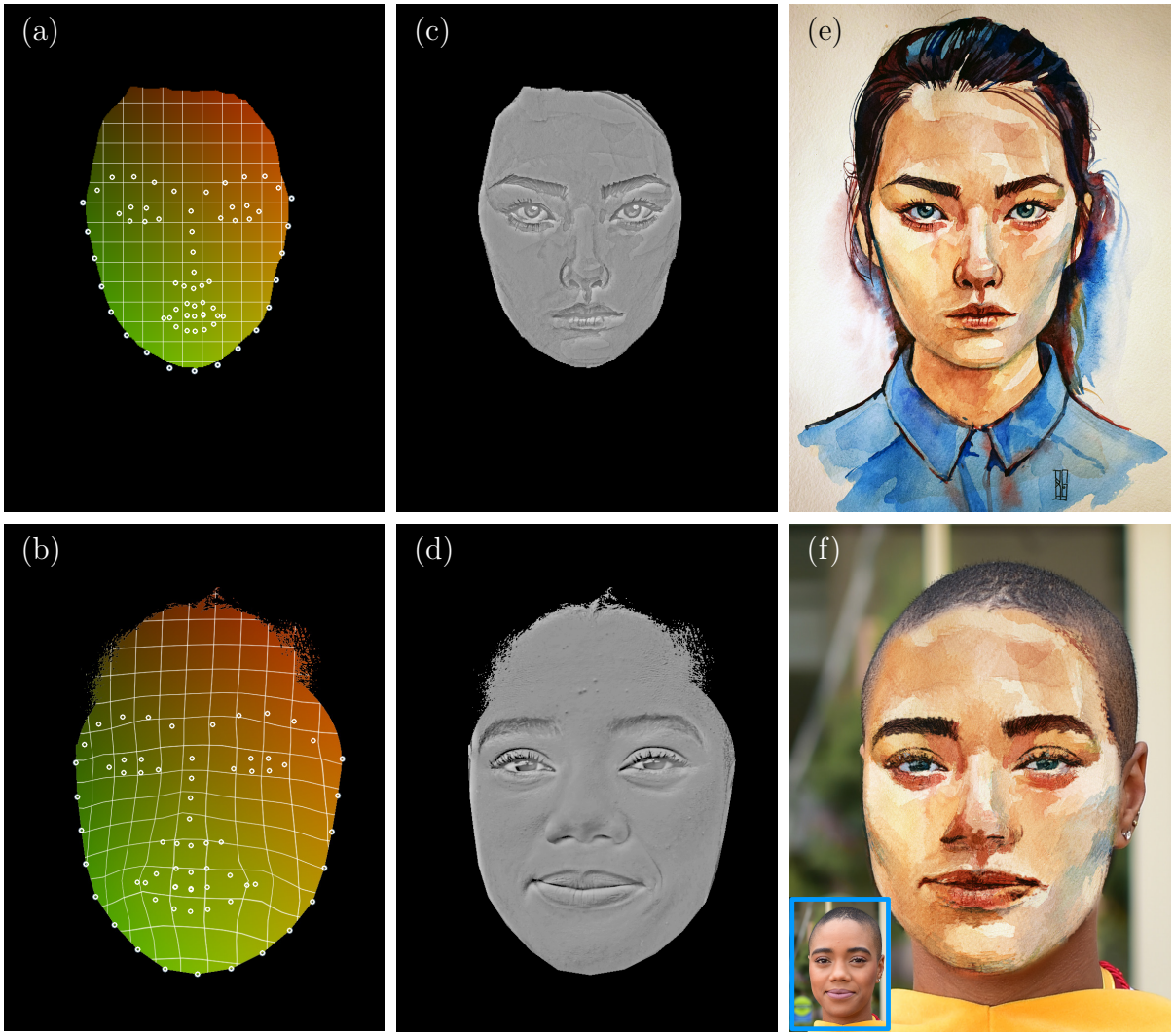


Figure 3.10: *Overview of the guiding channels used in our technique. The positional guide G_{pos} (a, b) secures the local consistency of the transfer from the style exemplar (e) to the target image (inset in blue). The target positional guide (b) is created by deforming the positional guide of the style image (a) according to the correspondence of facial landmarks, shown as white circles. Note that landmarks and the white grid is shown only for visualization purposes. The appearance guide G_{app} (c, d) encourages the synthesis to preserve subject's identity. See the text and Fig. 3.12 for detailed explanation of how G_{pos} & G_{app} is computed. Style exemplar (e) © Boris Groh, target photo (f) © Wilson Pumpernickel.*

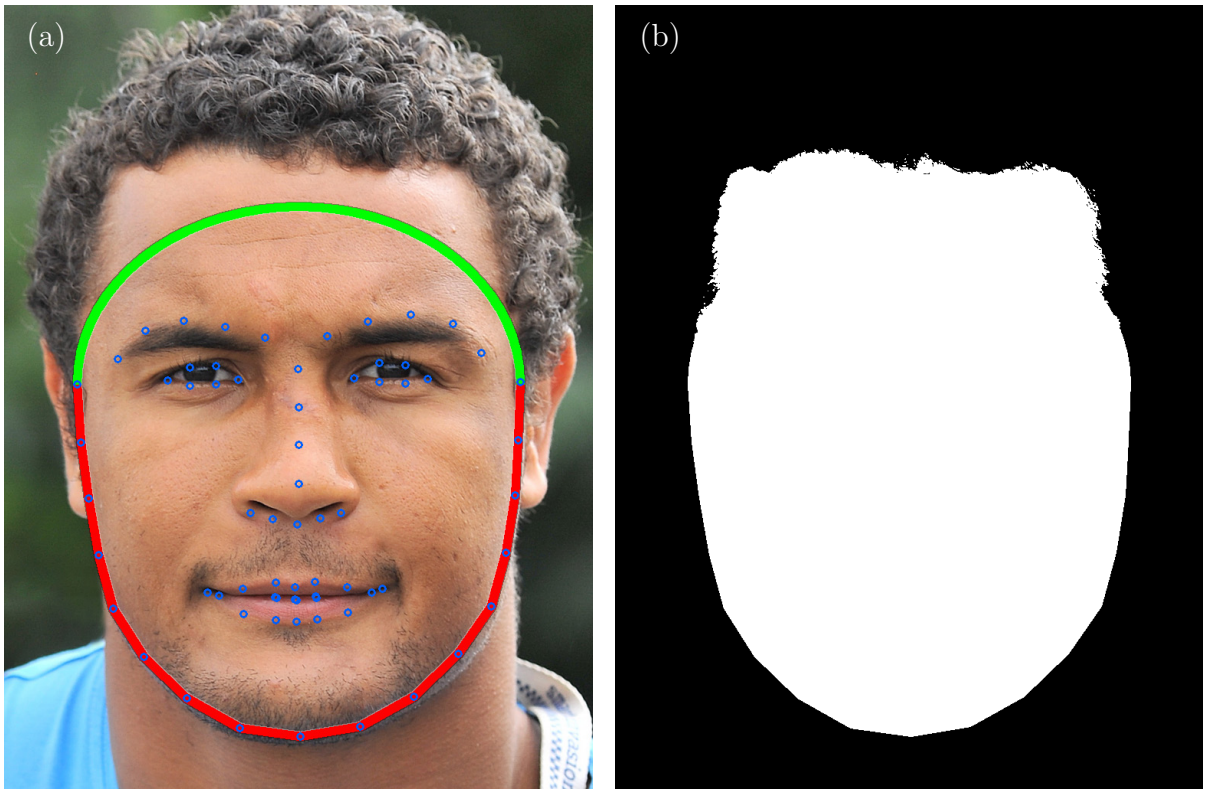


Figure 3.11: *Given a face (a), we compute a fast approximation of a segmentation mask (b) as follows. We take advantage of detected landmarks visualized as blue circles in (a). We first connect the chin landmarks, red line in (a). Then, we connect left and right uppermost chin landmark using an ellipse, green curve in (b). This gives us the segmentation of a lower and inner face. To include segmentation of forehead, we sample color components along the green curve and use a fast color thresholding operation and connected component analysis to determine the boundary between skin and hair, see the text for details. Target photo (a) © Patrick Subotkiewiez.*

ing channels is also an essential component for the fast synthesis method of Sýkora et al. [2019] which requires one of the guides to provide good localization.

Obtaining positional guide G^S_{pos} (Fig. 3.10a) for the style exemplar S is straightforward. All pixels are simply set to a color determined by their coordinates: x-coordinate corresponds to the red channel, y to the green channel. For $G^{T_i}_{\text{pos}}$ we need to generate an image that encodes a warping field between S and T_i where each target pixel is storing color-coded coordinates of its corresponding pixel in the source image (Fig. 3.10b). To create $G^{T_i}_{\text{pos}}$ we detect facial landmarks in the style exemplar S as well as in the target frame T_i using the method of Kazemi et al. [2014]. They provide a set of point correspondence from which a warping field between the source and target face can be computed using the moving least-squares method of Schaefer et al. [2006].

Since the style image S is static, facial landmarks can be detected in advance to save computational time. Sometimes landmark detector of Kazemi et al. may fail on artistic images due to the fact that it is trained on real photographs. In such a case, the method of Yaniv et al. [2019] tailored to artistic images could be used instead. In the target frames T_i the detection of landmarks needs to be performed on the fly. Therefore, to increase the detection speed, we subsample the target portrait to half resolution before passing it to the detector. It affects the accuracy negligibly while makes the detection significantly faster.

In contrast to the method of Fišer et al. [2017] we do not explicitly compute G_{seg} to reduce the computational overhead. Instead, we encode a simplified version of the facial mask directly into G_{pos}. In addition to color-coded pixel coordinates, we use the remaining blue channel to store a mask of the facial segment which for the style image S is computed offline using the method of Lee et al. [2020]. For T_i we need a faster algorithm as the target mask is computed on the fly. We use a subset of chin landmarks to define the lower part of the mask boundary. The upper part is constructed by sampling color components of pixels along the upper part of an ellipse going through the left and right uppermost chin landmark. Those samples are then used to perform fast color threshold operation followed by a connected component analysis that extracts the largest region of which upper contour defines the remaining upper boundary of the facial mask (see Fig. 3.11).

3.2.3 Appearance Guide

A primary role of appearance guide G_{app} is to preserve the identity of the target subject. In Fišer et al. [2017] a method of Shih et al. [2014] is employed to equalize the target image to have a similar appearance as the source style. However, this approach requires several seconds to compute. The entire Laplacian pyramid for both source and target image needs to be constructed. Then a robust gain mask is computed, applied at each pyramid level. And finally, a pyramid collapsing operation is performed. In our experiments, we found that such a costly operation can be approximated by a computation of only a single pyramid level on which histogram equalization is applied (see Fig. 3.12).

3.2.4 Style Transfer

Once the guiding channels for the source image S and the target video frame T_i are computed, the style transfer can be performed using the method of Sýkora et al. [2019].

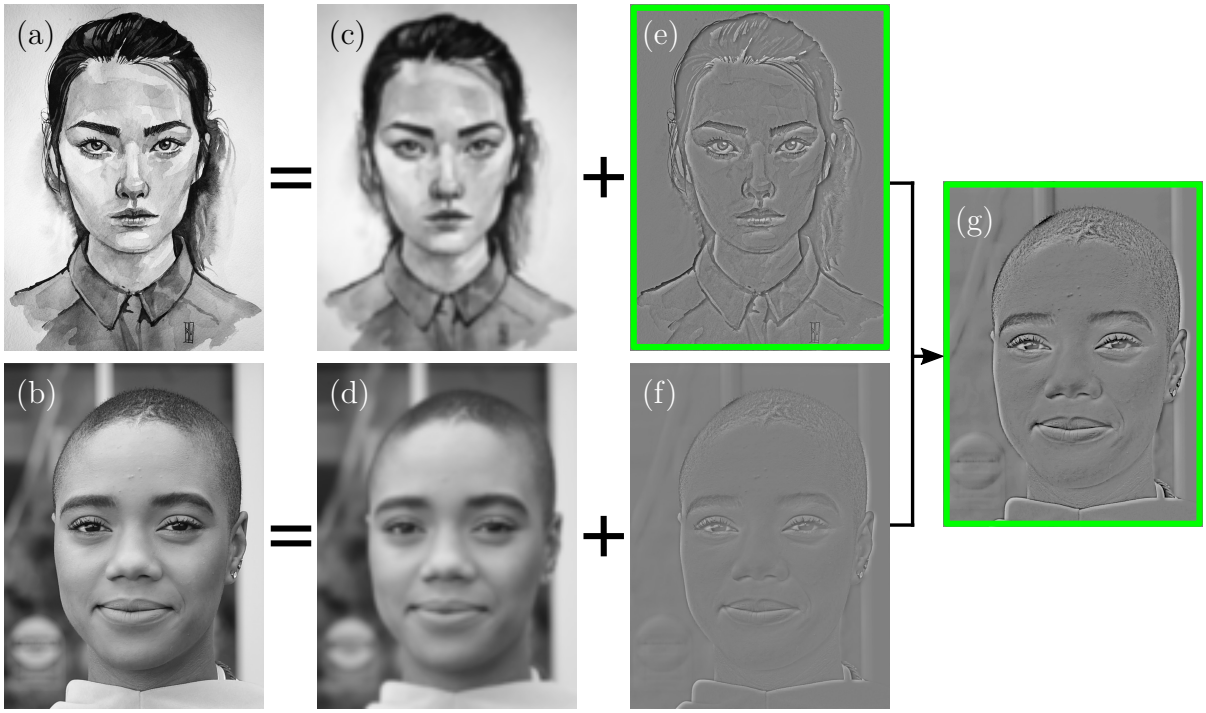


Figure 3.12: *The process of generating appearance guides G_{app} for the style exemplar S and the target frame T_i. The original images are converted into a grayscale domain (a, b), and filtered using Gaussian blur (c, d). To simulate the result of Laplacian of Gaussian filter (e, f) we subtract the blurred images (c, d) from their originals (a, b). Image (e) is the source part of appearance guide G^S_{app} and to produce its target counterpart $G^{T_i}_{app}$ (g) we modify (f) to match its histogram to that of (e). Style exemplar (a) © Boris Groh, target photo (b) © Wilson Pumpernickel.*

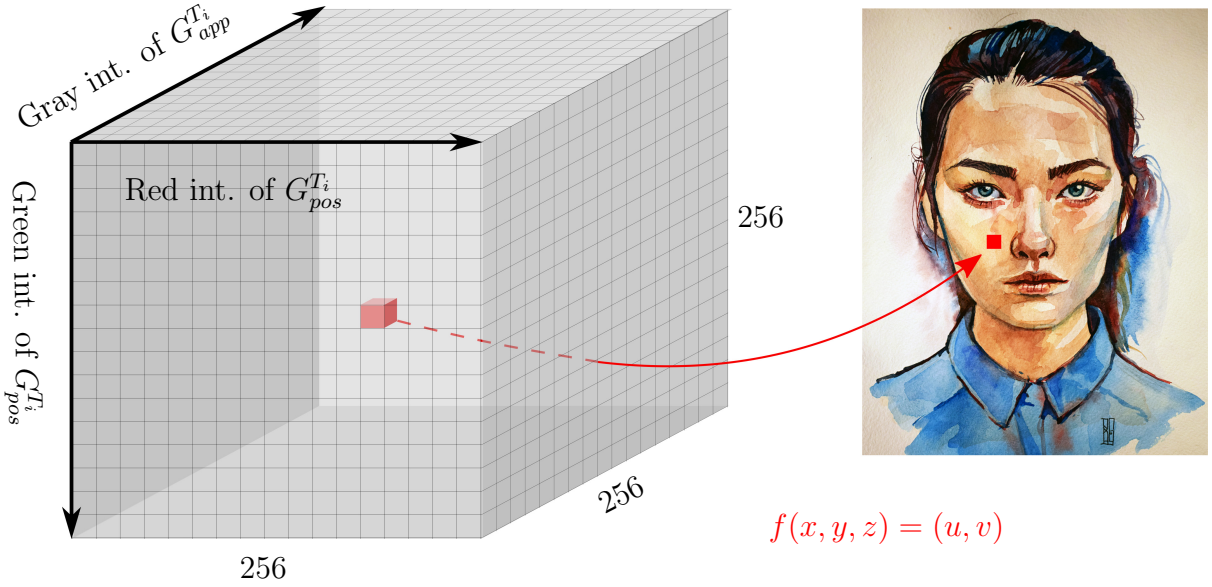


Figure 3.13: *The utilization of a 3D lookup table to obtain the corresponding source pixel for each target pixel. The cube stores coordinates of the best matching style exemplar pixel for a given red and green channel value in G_{pos} and the gray intensity in G_{app}. It allows to find the corresponding source pixel with complexity $\mathcal{O}(1)$ during the synthesis using method of Sýkora et al. [2019]. Style exemplar (right) © Boris Groh.*

Since in our case we have more than two values indirectly specifying the corresponding pixel location in the source exemplar (three dimensions for G_{pos} and one dimension for G_{app}), we need to leverage a kind of data structure which for each target pixel q quickly retrieves the closest source pixel p given by the following error metric:

$$E(p,q) = \|G^S_{\text{pos}}(p) - G^{T_i}_{\text{pos}}(q)\|^2 + \lambda |G^S_{\text{app}}(p) - G^{T_i}_{\text{app}}(q)|^2 \tag{3.5}$$

where λ is weighting the contribution of G_{pos} & G_{app} terms. We first reduce the original 4D mapping into 3D by encoding the blue channel of G_{pos} using zeros in red and green channels (see Fig. 3.10) and then we pre-calculate a 3D lookup table (see Fig. 3.13) that will require some additional memory but enables constant retrieval time. Such a 3D lookup table is then plugged into the parallel StyleBlit algorith described in the paper by Sýkora et al. [2019] (Algorithm 1).

3.2.5 Results

We implemented our approach using Java and C++. For all results presented in the method we use the following setting of parameters in the StyleBlit algorithm of Sýkora et al. [2019]: $\lambda = 0.2$ and $t = 50$. For each new style exemplar it takes several seconds to pre-calculate necessary data (3D lookup table, landmarks, and guiding channels) before the real-time stylization starts. The most critical is the computation of the 3D lookup table, the structure that stores coordinates pointing to the closest pixels in the source image (see Fig. 3.13). To obtain the coordinates, entire source image has to be searched. This process is computationally expensive as the search needs to be done for every position of the 3D lookup cube, i.e., 256^3 times. However, we reduce the processing time significantly by restricting the radius for searching the best matching pixel candidate to

20 pixels from the location estimated only by G_{pos}. We empirically verified that for all styles used in our experiments larger radius do not significantly increase the stylization quality. When a multicore CPU or a GPU is available lookup table pre-calculation can easily be accelerated by subdividing the entire 3D space into a set of smaller cubes that can be evaluated in parallel.

On a half megapixel image our implementation runs at 15 frames per second on *Samsung Galaxy Note8* with CPU *Samsung Exynos 8895, 2.3 GHz*, GPU *Mali G71 MP20* and *6 GB* of RAM. The framerate scales roughly linearly with the increasing number of pixels. On the fly detection of landmarks in the target video frame takes 10 ms, generation of guidance channels 12 ms, style transfer 20 ms, and other miscellaneous steps, (camera handling, frame flipping and rotating, conversions between color spaces, copying data between Java and C++) take 28 ms.

We tested our method with various style exemplars applied on several target faces from FFHQ dataset [Karras et al. 2019] (see Fig. 3.14) and videos captured on a mobile device (see our supplementary video). Those experiments verified that our method can carry the exemplar's textural details while still being able to respect the target subject's identity. The quality of stylization results is comparable to those produced by the previous offline method of Fišer et al. [2017] as well as real-time method of Futschik et al. [2019] that requires lengthy pre-processing phase (see Fig. 3.15 and Fig. 3.20). A detailed banchmark measuring pre-processing and synthesis time for a half megapixel image on 3 GHz Quad-core CPU with Nvidia RTX 2080 GPU is available in Table 3.1. Note that as compared to Fišer et al. and Futschik et al. our method uses only CPU.

Method	**Pre-calculation**	**Synthesis**
Our approach (CPU only)	10 s	0.05 s
[Fišer et al. 2017] (CPU + GPU)	5 s	10 s
[Futschik et al. 2019] (CPU + GPU)	2 days	0.06 s

Table 3.1: *Comparison of processing times w.r.t. state-of-the-art.*

In addition to method comparison, we also performed various ablation experiments.

In Fig. 3.16, we demonstrate the importance of using both the G_{pos} & G_{app} guidance channels. The absence of G_{pos} may cause that coherent chunks from style exemplar are transferred to wrong locations in the target portrait, (see Fig. 3.16c, d). Without G_{app}, the subject's identity is not preserved well (see, e.g., wrong eyebrows or the absence of wrinkles in Fig. 3.16e, f). When using both guides, stylized results faithfully represent artistic medium, the transfer is semantically meaningful, and the identity of the target subject is well-preserved (see Fig. 3.16g, h).

In Fig. 3.17 we show the necessity of the histogram matching operation during the generating of the target appearance guide G^T_{app}. Without matching the appearance guides' histograms, the error E overcomes the threshold t too soon which leads to notably smaller chunks and the result may seem blurry (see Fig. 3.17).

We also tried to execute our algorithm with the same G^T_{app} as described in the original approach of Fišer et al. [2017] (see Fig. 3.18). It is visible that their more sophisticated G^T_{app} preserves the subject's identity a bit better, nevertheless, it is notably slower to compute.

Figure 3.14: *FaceBlit applied on several target subjects (leftmost column), using various style exemplars (topmost row). Style exemplars: (a) © Boris Groh, (b) Viktor Ivanovich Govorkov, (c) © Matthew Ivan Cherry (HAT, oil on canvas, 48" x 48", 2011), (d, e) © Adrian Morgan, (f) Peter Zelizňák (sculpture by Stanislav Mikuš), target photos: (g) PFA SEAL, (h) © Ajuntament de Sabadell, (i) © Raziel Janeway, (j) lam_anh2005.*

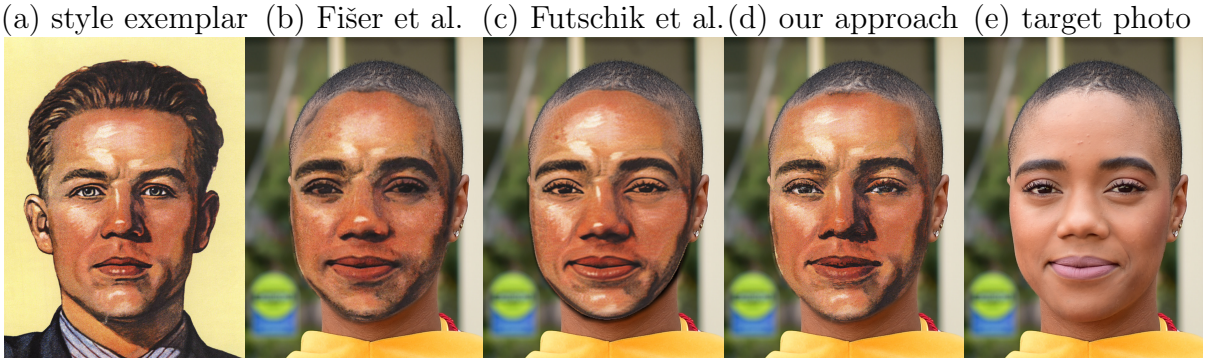


Figure 3.15: *FaceBlit vs. state-of-the-art—the task at hand is to transfer a style from an exemplar image (a) to a face in the target image (e) while preserving important visual characteristics of the used artistic media in (a) and the identity of the subject in (e). In contrast to current state-of-the-art [Fišer et al. 2017] (b) and [Futschik et al. 2019] (c), our approach (d) is able to deliver comparable stylization quality and identity preservation without the need to perform costly computation during the synthesis (tens of seconds for Fišer et al.) or lengthy data set generation and training (days for Futschik et al.). Thanks to this advantage our approach can perform instant style transfer to facial videos in real-time even on mobile device. Source style (a) Viktor Ivanovich Govorkov, target photo (f) © Wilson Pumpernickel.*

Figure 3.16: *Importance of individual guidance channels. The positional guide G_{pos} is essential. Its absence (c, d) causes that the chunks from the style are not transferred in a semantically meaningful way. Without the appearance guide G_{app}, the identity of target subjects (a, b) is not preserved well (e, f). The full guidance (g, h) secures the local consistency of style transfer while retaining the target subject's identity. Style exemplars: (i) © Boris Groh, (j) © Adrian Morgan, target photos: (a) © LEMON Studio, (b) © Mark Peers.*

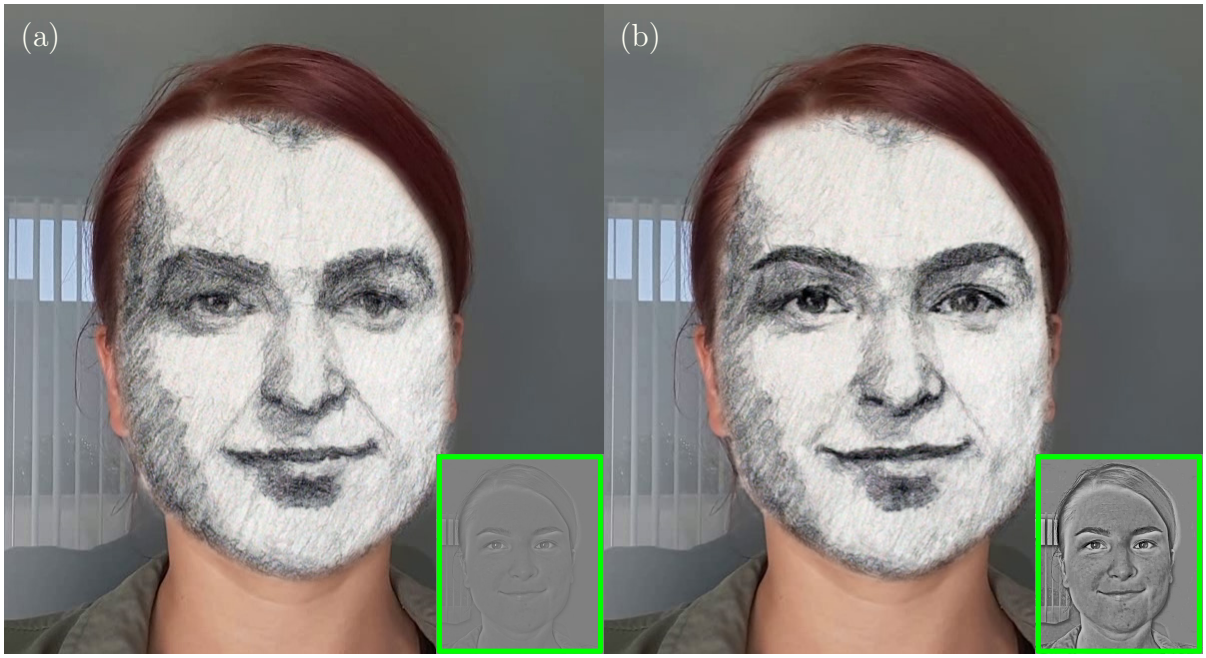


Figure 3.17: *Importance of the histogram matching phase during the generation of the target appearance guide G^T_{app}. Without the histogram matching, the subject's identity is not preserved well, and the result may seem blurry. See (a) and its respective appearance guide in green inset. After equalizing histograms, the gain in quality is significant. See (b) and the green inset.*

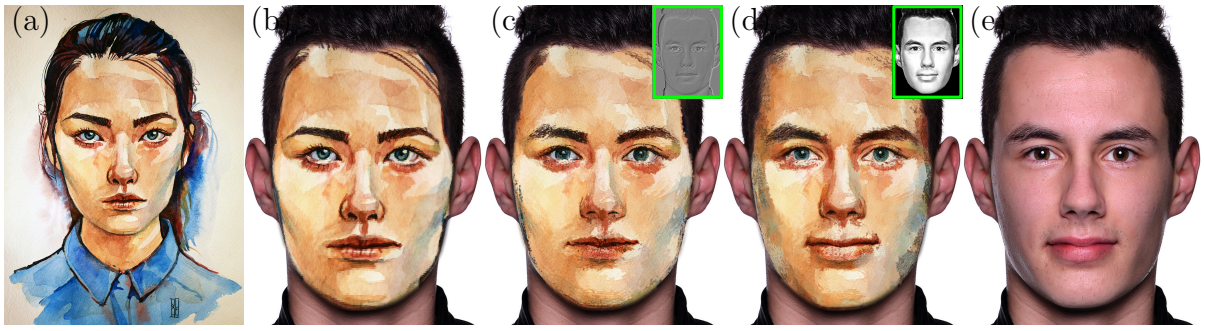


Figure 3.18: *Comparison of using our appearance guide with the one proposed in the method of Fišer et al. [2017]—style from the exemplar (a) is transferred to the target image (e). A stylization result without appearance guide (b), with appearance guide generated by our method (c), and with appearance guide generated by Fišer et al. (d). Note, how the identity of the target subject is bit less pronounced as compared to the solution of Fišer et al., which however is orders of magnitude slower than ours. Style exemplar (a) © Boris Groh, target photo (e) SKV Florbal.*

3.2.6 Extensions

A visible limitation of our approach when compared to state-of-the-art is the absence of hair stylization (c.f. Fig. 3.20). Although the face parsing network of Lee et al. [2020] can be used to estimate hair mask its computational overhead is too demanding to preserve real-time response. Also the computation of positional guide could be complicated when the shapes of source and target hair segments differ significantly. The resulting warping field may violate good localization property of the positional guide which is crucial for the method of Sýkora et al. [2019] to produce reasonable stylization results.

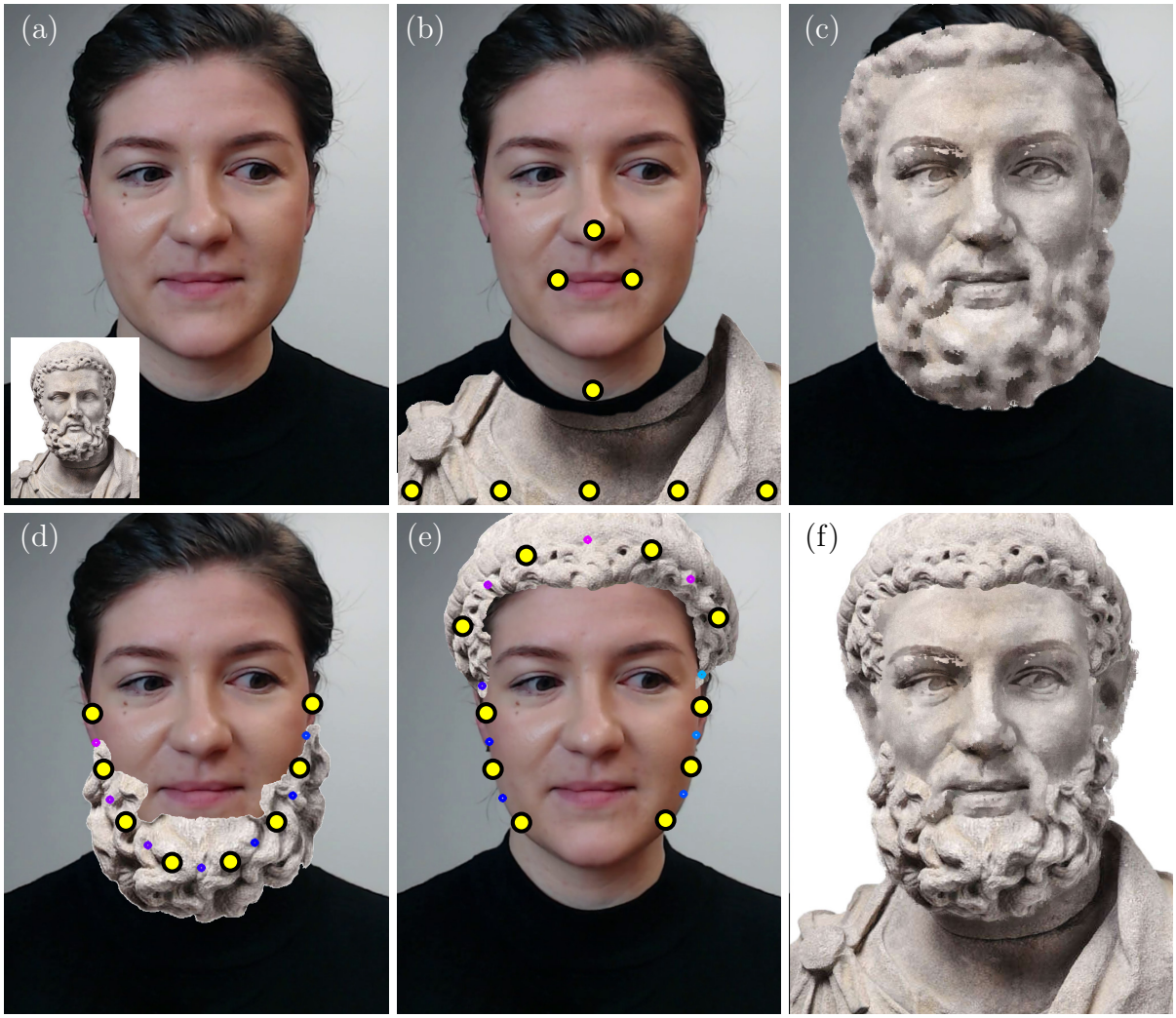


Figure 3.19: *An example of a hybrid approach where the aim is to stylize a person in the video (a) to look like the statue in the inset reassembling her identity. To do that we subdivide the statue into a set of separate layers: torso (b), face (c), beard (d), and hair (e). The facial layer (c) is animated using our approach while for the torso (b), beard (d), and hair (e) layer we use moving least-squares deformation [Schaefer et al. 2006] driven by a set of control points (yellow dots) of which position is derived from detected landmarks. Such a set of deformed and stylized layers is then blended in a predefined depth order to produce the final composition (f). See our supplementary video for this example in montion. Style exemplar (a) © Country French Interiors, target photo (a) Šárka Sochorová.*

To alleviate this drawback, we implemented a hybrid method that uses our new face stylization approach to bring an existing portrait painting to life while adapting the identity of the portrayed person to the one seen in the target video (see Fig. 3.19 and our supplementary video). In this extension we separate the style image into a set of segments (face, hair, beard, torso, and background). These segments are processed independently and then stiched together to form the final output frame. The facial segment is stylized using the algorithm described in this Chapter. For hair, beard (if applicable), and torso segments, we use moving least-squares deformation [Schaefer et al. 2006] driven by a set of facial landmarks (c.f. Fig. 3.19).

3.3 Conclusion and Future Work

In this Chapter, we have presented two methods to address common issues and limitations of algorithms performing facial stylization. In Section 3.1, we proposed an image-to-image translation network, being able to reproduce the outputs of the method of Fišer et al. [2017] in real-time, which was previously not possible due to the expensive optimization scheme the approach of Fišer et al. utilized. In Section 3.2, we explored a different option to achieve real-time face stylization performance without the use of neural networks, by precomputing required positional and appearance data and using them with the algorithm of Sýkora et al. [2019], which allowed real-time performance even on low-end devices, such as smartphones.

One of the critical challenges in the method described in Section 3.1 is the accuracy and smoothness of head and hair segmentation masks. Although our method often outperforms FaceStyle algorithm [Fišer et al. 2017] concerning the quality of separation of head and hair segments, in general (especially) the outer hair boundary has some issues with smoothness and shape details (see Figures 3.5, 3.6, and 3.7). One can mitigate this inaccuracy by preparing a broader set of training exemplars containing a greater variety of input photos under different illumination conditions with more accurately specified head and face masks. For some styles our method tends to produce repetition artifacts visible principally on hair segments depending on the overall spatial extent (see Figures 3.5, 3.6, and 3.7). Although a similar effect is apparent also on the original output from the FaceStyle algorithm, our solution tends to exaggerate it. Techniques to reduce visible repetition on the level of patch-based synthesis as well as during the training phase (e.g., using a specific penalizing loss) would be a promising avenue for future work. When inspecting results closely on a pixel level our approach has still a difficulty in preserving the original sharpness of the texture visible in the original from the FaceStyle algorithm. Such a visual smoothing effect is caused by the fact that the network has parametric nature while the output from FaceStyle represents a non-parametric mosaic of patches that represent exact copies of the original style exemplar. As a future work, we plan to investigate more the possibility to train pixel mapping instead of color information which can enable the formation of the final image using an explicit pixel copy-and-paste operation as in patch-based techniques. Although our approach delivers stable results when the target does not change considerably and enables rough temporal coherency for video sequences it still suffers from subtle temporal flicker which can be disturbing in some applications. To gain control over the temporal dynamics an addition of specific temporal

smoothness terms similar to those used in video-to-video transfer approaches [Wang et al. 2018d] need to be considered.

While we show that our method in Section 3.2 can deliver comparable visual quality with significantly lower computational overhead than the state-of-the-art, some limitations stem from this performance gain. A compromise we accepted in our real-time solution is the omission of explicit guidance that allows to control the level of temporal coherence. Although the flickering our approach is producing resembles temporal dynamics of hand-colored animations and can be perceived as an important feature (c.f. Fišer et al. [2014]), some sort of control over its behavior would be valuable since it may become disturbing after a while. To control the strength of temporal flickering [Sýkora et al. 2019] proposes to lower the threshold t of their fast stylization algorithm which in fact leads to smaller copied exemplar chunks and thus become close to texture mapping scenario which breaks the planarity of brush strokes present in the original style exemplar. This problem opens an interesting direction for future work. Also, the addition of appearance guide causes the overall guidance to become a bit more discontinuous when compared to the case of clean positional guide which better suits fast stylization method of Sýkora et al. [2019]. Due to this reason the size of transferred chunks can be notably smaller and thus cause suppression of artistic features that have larger scale in the original style exemplar (see, e.g., Fig. 3.16f vs. 3.16h). Lastly, this approach shares similar limitations as other techniques that use guided patch-based synthesis, such as the methods of Fišer et al. [2017] and Sýkora et al. [2019]. The style exemplar needs to have a compatible scale with the target image otherwise artifacts may appear (see, e.g., Fig. 13 in [Sýkora et al. 2019]). Patch-based synthesis also encounters difficulties when adapting to different lighting conditions or an absence of important features (e.g., wrinkles or moustach) that are present in the target image, however, are missing in the style exemplar or vice versa. A viable avenue for future work could be to alter between a set of exemplars drawn in a similar style that would better suit the target image (e.g., various lighting directions, man/woman, old/young, etc.).

(g) Fišer et al. (h) Futschik et al. (i) our approach

Figure 3.20: *Comparison of our method with state-of-the-art: style from an exemplar (a, c, e) is transferred to the target photo (b, d, f) using the method of Fišer et al. [2017] (g), Futschik et al. [2019] (h), and our approach (i). Note, how our approach produces comparable stylization quality while is notably faster than the method of Fišer et al. and does not require lengthy pre-calculation contrary to Futschik et al. A limitation of our method is that it does not support hair stylization. Style exemplars: (a) © Matthew Ivan Cherry (HAT, oil on canvas, 48" x 48", 2011), (c, e) © Adrian Morgan, target photos: (b) © MPCA Photos, (d) © LEMON Studio, (f) © Patrick Subotkiewiez.*

Chapter 4

Stylization of Video Sequences

Example-based stylization of videos became recently popular thanks to significant advances made in neural techniques [Ruder et al. 2018; Sanakoyeu et al. 2018; Kotovenko et al. 2019a]. Those extend the seminal approach of Gatys et al. [2016] into the video domain and improve the quality by adding specific style-aware content losses. Although these techniques can deliver impressive stylization results on various exemplars, they still suffer from the key limitation of being difficult to control. This is due to the fact that they only measure statistical correlations and thus do not guarantee that specific parts of the video will be stylized according to the artist's intention, which is an essential requirement for use in a real production pipeline.

This important aspect is addressed by a concurrent approach—the keyframe-based video stylization [Bénard et al. 2013; Jamriška et al. 2019]. Those techniques employ guided patch-based synthesis [Hertzmann et al. 2001; Fišer et al. 2016] to perform a semantically meaningful transfer from a set of stylized keyframes to the rest of the target video sequence. The great advantage of a guided scenario is that the user has a full control over the final appearance, as she can always refine the result by providing additional keyframes. Despite the clear benefits of this approach, there are still some challenges that need to be resolved to make the method suitable for a production environment.

One of the key limitations of keyframe-based stylization techniques is that they operate in a sequential fashion, i.e., their outputs are not *seekable*. When the user seeks to any given frame, all the preceding frames have to be processed first, before the desired result can be displayed. This sequential processing does not fit the mechanism of how frames are handled in professional video production tools, where random access and parallel processing are inevitable.

Another important aspect that needs to be addressed is merging, or blending, the stylized content from two or more (possibly inconsistent) keyframes to form the final sequence. Although various solutions exist to this problem (e.g., [Shechtman et al. 2010; Jamriška et al. 2019]), the resulting sequences usually suffer from visible clutter or ghosting artifacts. To prevent the issues with merging, the user has to resort to a tedious incremental workflow, where she starts by processing the whole sequence using only a single keyframe first. Next, she prepares a corrective keyframe by painting over the result of the previous synthesis run. This requires re-running the synthesis after each new correction, which leads to additional computational load and slows the overall process down.

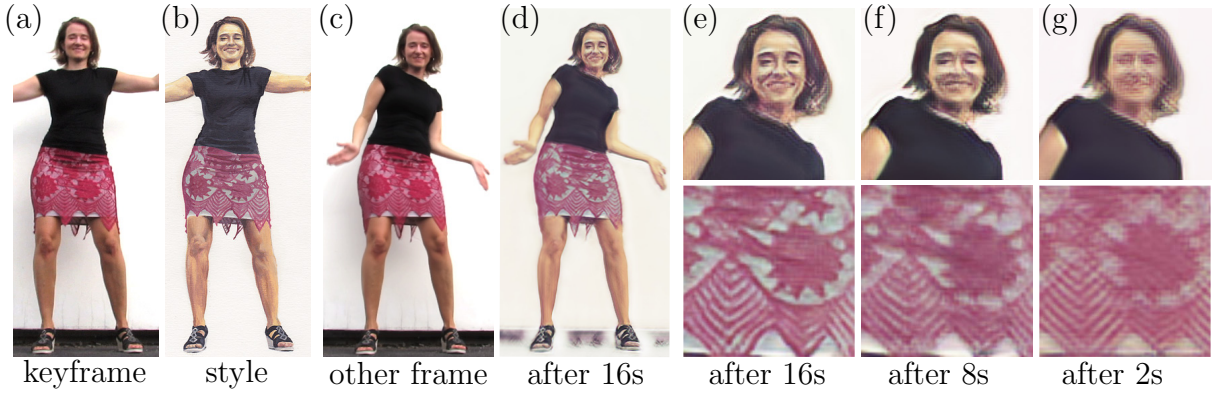


Figure 4.1: *An example of a sequence stylized using our approach. One frame from the original sequence is selected as a keyframe (a) and an artist stylizes it with acrylic paint (b). We use this single style exemplar as the only data to train a network. After 16 seconds of training, the network can stylize the entire sequence in real-time (c-d) while maintaining the state-of-the-art visual quality and temporal coherence. See the zoom-in views (e-g); even after 2 seconds of training, important structures already start to show up. Video frames (a, c) and style exemplar (b) courtesy of © Zuzana Studená.*

To summarize, it would be highly beneficial to develop a guided style transfer algorithm that would act as a fast image filter. Such a filter would perform a semantically meaningful transfer on individual frames without the need to access past results, while still maintaining temporal coherence. In addition, it should also react adaptively to incoming user edits and seamlessly integrate them on the fly without having to perform an explicit merging.

Such a setting resembles the functionality of appearance translation networks [Isola et al. 2017; Wang et al. 2018a], which can give the desired look to a variety of images and videos. In these approaches, generalization is achieved by a large training dataset of aligned appearance exemplars. In our scenario, however, we only have one or a few stylized examples aligned with the input video frames, and we propagate the style to other frames with similar content. Although this may seem like a simpler task, we demonstrate that when existing appearance translation frameworks are applied to it naively, they lead to disturbing visual artifacts. Those are caused by their tendency to overfit the model when only a small set of appearance exemplars is available.

Our scenario is also similar to few-shot learning techniques [Liu et al. 2019; Wang et al. 2019b] where an initial model is trained first on a large generic dataset, and then in the inference time, additional appearance exemplars are provided to modify the target look. Although those methods deliver convincing results for a great variety of styles, they are limited only to specific target domains for which large generic training datasets exist (e.g., human bodies, faces, or street-view videos). Few-shot appearance translation to generic videos remains an open problem.

In the following sections, we focus on the key aspects of video sequence stylization. First of those is interactivity, being able to provide fast stylization of videos and provide random access to any stylized frame in a given sequence, providing real-time feedback to the artist, allowing them to make changes as necessary. Second is robustness, avoiding the loss of fine textural details over time and preserving the integrity of the style througout

the entire video sequence. In the following sections, we address each of these issues individually and propose methods and solutions to tackle them.

4.1 Real-Time Interactive Video Stylization

To address the demanding time requirements of state-of-the-art video sequence stylization methods, in this section we present a new appearance translation framework for arbitrary video sequences that can deliver semantically meaningful style transfer with temporal coherence without the need to perform any lengthy domain-specific pre-training. We introduce a patch-based training mechanism that significantly improves the ability of the image-to-image translation network to generalize in a setting where larger dataset of exemplars is not available. Using our approach, even after a couple of seconds of training, the network can stylize the entire sequence in parallel or a live video stream in real-time.

Our method unlocks a productive workflow, where the artist provides a stylized keyframe, and after a couple of seconds of training, she can watch the entire video stylized. Such rapid feedback allows the user to quickly provide localized changes and instantly see the impact on the stylized video. The artist can even participate in an interactive session and watch how the progress of her painting affects the target video in real-time. By replacing the target video with a live camera feed, our method enables an unprecedented scenario where the artist can stylize an actual live scene. When we point the camera at the artist's face, for instance, she can simultaneously paint the keyframe and watch a stylized video-portrait of herself. Those scenarios would be impossible to achieve with previous keyframe-based video stylization methods, and our framework thus opens the potential for new unconventional applications.

4.1.1 Our Approach

The input to our method is a video sequence I, which consists of N frames. Optionally, every frame I_i can be accompanied by a mask M_i to delineate the region of interest; otherwise, the entire video frame is stylized. Additionally, the user also specifies a set of keyframes $I^k \subset I$, and for each of them, the user provides stylized keyframes S^k, in which the original video content is stylized. The user can stylize the entire keyframe or only a selected subset of pixels. In the latter case, additional keyframe masks M^k are provided to determine the location of stylized regions (see Fig. 4.2 for details).

Our task is to stylize I in a way that the style from S^k is transferred to the whole of I in a semantically meaningful way, i.e., the stylization of particular objects in the scene remains consistent. We denote the output sequence by O. The aim is to achieve visual quality and temporal consistency comparable to the state-of-the-art in the keyframe-based video stylization [Jamriška et al. 2019]. However, in contrast to this previous work, we would like to stylize the video frames in random order, possibly in-parallel, or on-demand in real-time, without the need to wait for previous frames to be stylized or to perform explicit merging of stylized content from different keyframes. In other words, we aim to design a translation filter that can quickly learn the style from a few heterogeneously hand-drawn exemplars S^k and then stylize the entire sequence I in parallel, or any single frame on demand. It would also be beneficial if the learning phase

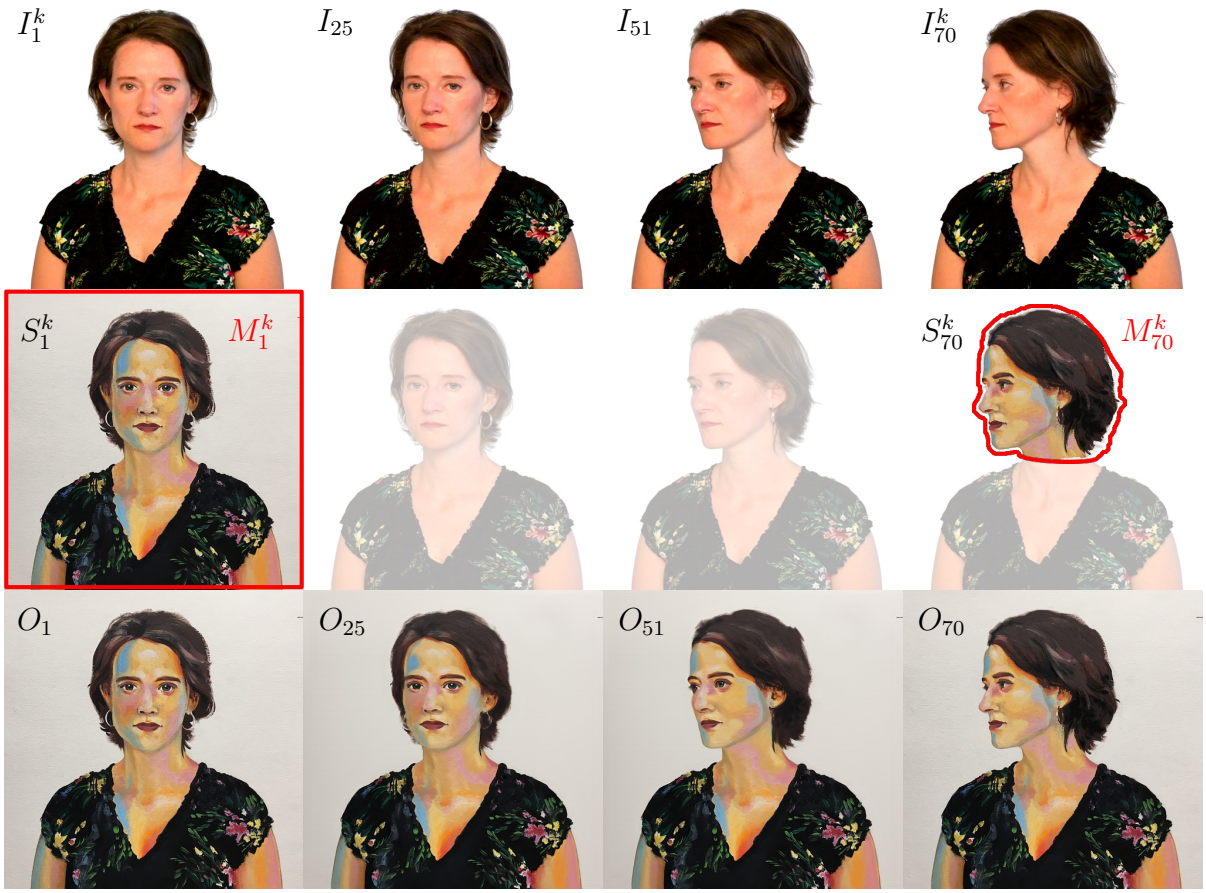


Figure 4.2: *The setting of video stylization with keyframes. The first row shows an input video sequence I. There are two keyframes painted by the user, one keyframe is painted fully (S_1^k) and the other is painted only partially (S_{70}^k). Mask M_1^k denotes that the entire keyframe is used; mask M_{70}^k specifies only the head region. Our task is to stylize all frames of the input sequence I while preserving the artistic style of the keyframes. The sequence O in the bottom row shows the result of our method. Video frames (I) and style exemplars (S) courtesy of © Zuzana Studená.*

was fast and incremental so that the stylization of individual video frames could start immediately, and the stylization quality would progressively improve over time.

To design such a filter, we adopt the U-net-based image-to-image translation framework of Futschik et al. [2019], which was originally designed for the stylization of faces. It uses a custom network architecture that can retain important high-frequency details of the original style exemplar. Although their network can be applied in our scenario directly, the quality of results it produces is notably inferior as compared to state-of-the-art (see Fig. 4.3c and our supplementary video at 2:20). One of the reasons why this happens is that the original Futschik et al.'s network is trained on a large dataset of style exemplars produced by FaceStyle algorithm [Fišer et al. 2017]. Such many exemplars are not available in our scenario, and thus the network suffers from strong overfitting. Due to this reason, keyframes can be perfectly reconstructed; however, the rest of the frames are stylized poorly, even after applying well-known data augmentation methods. See the detailed comparison in Figures 4.3 and 4.9. Furthermore, the resulting sequence also

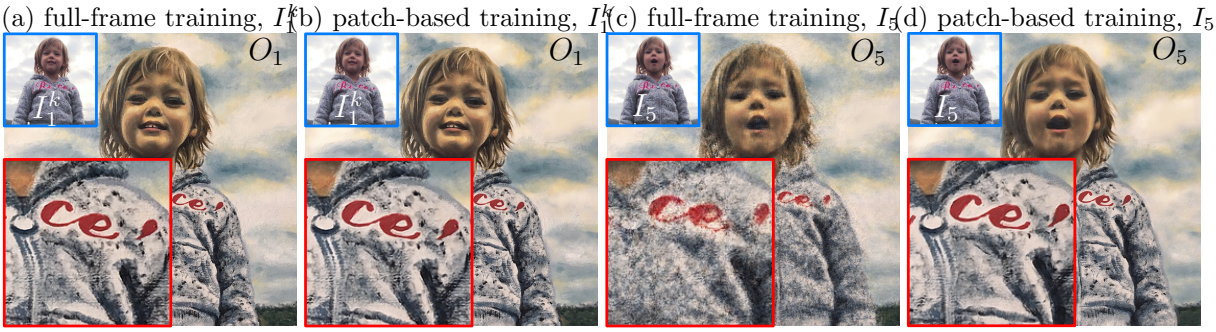


Figure 4.3: *Comparison of full-frame training vs. our patch-based approach: the original frames from the input sequence I are marked in blue and details of their stylized counterparts O are marked in red. The full-frame training scheme of Futschik et al. [2019] (a) as well as our patch-based approach (b) closely reproduce the frame on which the training was performed (see the frame S_1^k in Fig. 4.6). Both stylized frames (a, b) look nearly identical, although the training loss is lower for the full-frame scheme. Nevertheless, the situation changes dramatically when the two networks are used to stylize another frame from the same sequence (here frame I_5). The network which was trained using the full-frame scheme produces images that are very noisy and have fuzzy structure (c). This is due to the fact that the full-frame training causes the network to overfit the keyframe. The network is then unable to generalize to other frames in the sequence even though they structurally resemble the original keyframe. The network which was trained using our patch-based scheme retains the fidelity and preserves the important artistic details of the original style exemplar (d). This is thanks to the fact that our patch-based scheme better encourages the network to generalize to unseen video frames. Video frames (I) courtesy of © Zuzana Studená.*

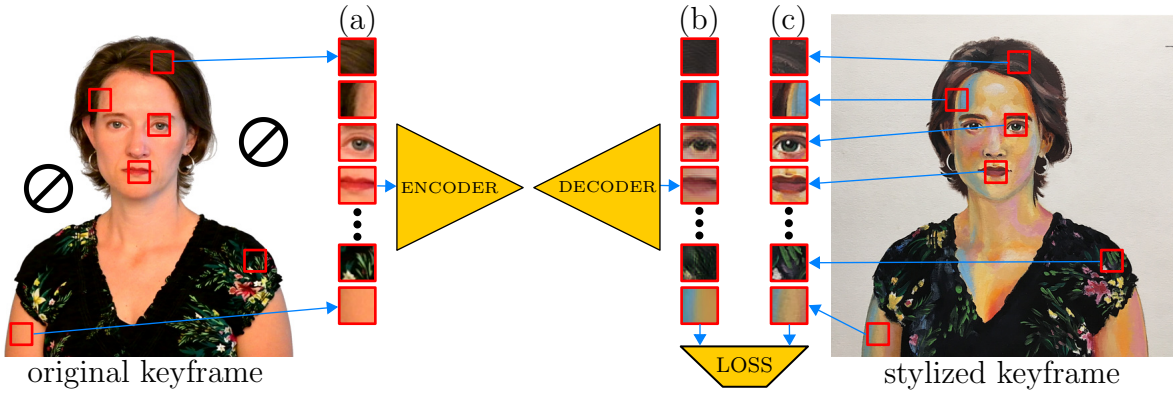


Figure 4.4: *Training strategy: we randomly sample a set of small patches from the masked area of the original keyframe (a). These patches are then propagated through the network in a single batch to produce their stylized counterparts (b). We then compute the loss of these stylized counterparts (b) with respect to the co-located patches sampled from the stylized keyframe (c) and back-propagate the error. Such a training scheme is not limited to any particular loss function; in our method, we use a combination of L1 loss, adversarial loss, and VGG loss. Video frame (left) and style exemplar (right) courtesy of © Zuzana Studená.*

contains a disturbing amount of temporal flickering because the original method does not take into account temporal coherence explicitly.

To address the drawbacks mentioned above, we alter how the network is trained and formulate an optimization problem that allows fine-tuning the network's architecture and its hyperparameters to get the stylization quality comparable to the state-of-the-art, even

with only a few training exemplars available and within short training time. Also, we propose a solution to suppress temporal flicker without the need to measure consistency between individual video frames explicitly. In the following sections, those improvements are discussed in further detail.

4.1.2 Patch-based Training Strategy

To avoid network overfitting to the few available keyframes, we adopt a patch-based training strategy. Instead of feeding the entire exemplar to the network as done in the paper of Futschik et al. [2019], we randomly sample smaller rectangular patches from all stylized keyframes S^k (see Fig. 4.4) and train the network to predict a stylized rectangular area of same size as input. The sampling is performed only within the area of masked pixels M^k. Note that thanks to the fully convolutional nature of the network, once trained, it can be directly used to stylize the entire video frame even though the training was performed on smaller patches (see Fig. 4.5). The key benefit of this explicit cropping and randomization step is that it simulates the scenario when a large and diverse dataset is used for training. It prevents the network from overfitting and generalizes to stylize the other video frames better. This training strategy is similar to one previously used for texture synthesis [Zhou et al. 2018].

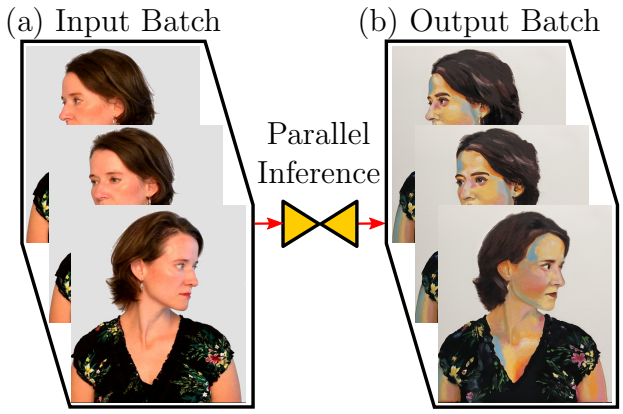


Figure 4.5: *Inference: thanks to the fully convolutional nature of the network, we can perform the inference on entire video frames, even though the training is done on small patches only. Since the inference does not depend on other stylized frames, all video frames can be stylized in parallel or in random order. This allows us to pass many or even all of the input frames (a) through the network in a single batch and get all output frames (b) at once. Video frames (left) courtesy of © Zuzana Studená.*

Although the reconstruction loss measured on keyframes S^k is higher when compared to full-frame training after comparable amount of time, on the remaining frames of I the reconstruction loss is considerably lower when comparing to the frames stylized using state-of-the-art keyframe-based video stylization method of Jamriška et al. which we purposefully consider as a ground truth (cf. supplementary video at 0:08 and 1:08). This lower loss w.r.t. Jamriška et al. translates to much better visual quality.

4.1.3 Hyper-parameter Optimization

Although the patch-based training strategy considerably helps to resolve the overfitting problem, we find that it is still essential to have a proper setting of critical network hyperparameters, as their naive values could lead to poor inference quality, especially when the training performance is of great importance in our applications (see Fig. 4.8). Besides that, we also need to balance the model size to capture the essential characteristics of the style yet being able to perform the inference in real-time using off-the-shelf graphics card.

We formulate an optimization problem in which we search for an optimal setting of the following hyperparameters: W_p—size of a training patch, N_b—number of patches used in one training batch, α—learning rate, and N_r—number of ResNet blocks used in our network architecture. The aim is to minimize the loss function used in the method of Futschik et al. [2019] computed over the frames inferred by our network and their counterparts stylized using the method of Jamriška et al. [2019]. The minimization is performed subject to the following hard constraints: T_t—the time for which we allow the network to be trained for and T_i—the inference time for a single video frame. Since T_t as well as T_i are relatively short (in our setting $T_t = 30$ and $T_i = 0.06$ seconds) full optimization of hyperparameters becomes tractable. We used the grid search method on a GPU cluster, to find the optimal values (see detailed scheme Fig. 4.6). In-depth elaboration can be found in Section 4.1.5.

In our experiments, we found that hyperparameter optimization is relatively consistent when different validation sequences are used. We thus believe the setting we found is useful for a greater variety of styles and sequences. Note also that the result of Jamriška et al. is used only for fine-tuning of hyperparameters. Once this step is finished, our framework does not require any guided patch-based synthesis algorithm and can act fully independently.

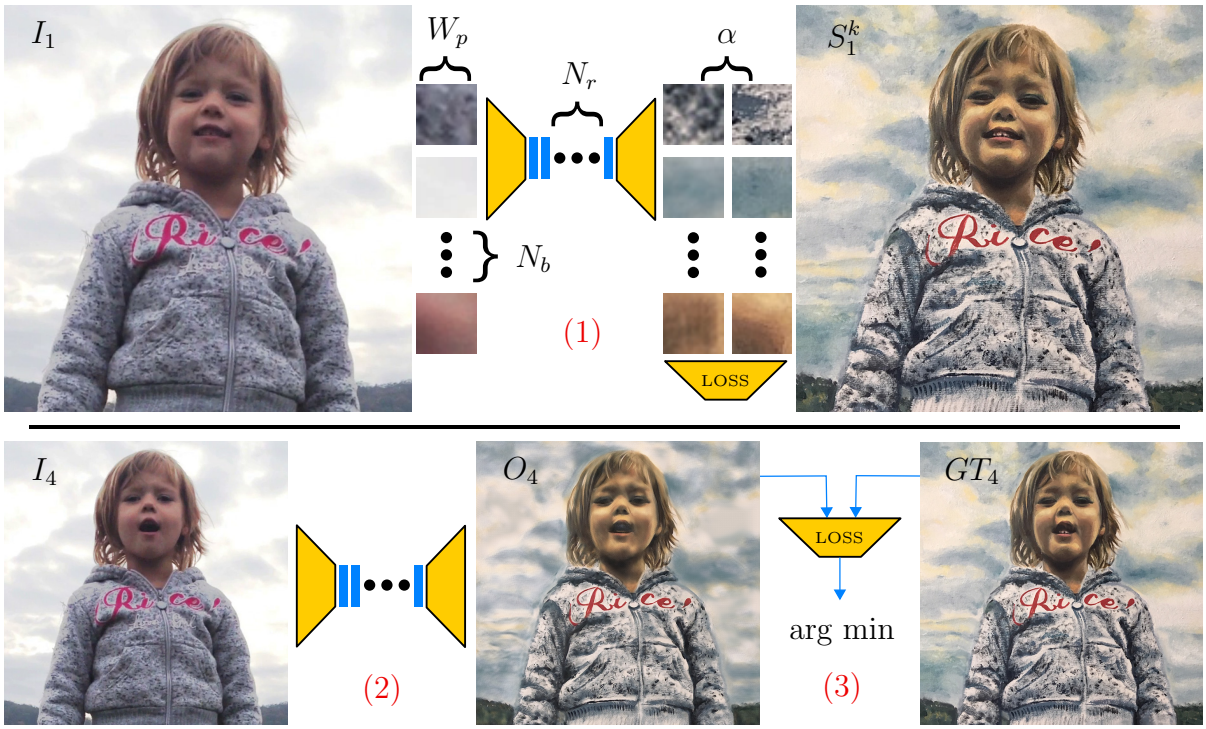


Figure 4.6: *To fine-tune critical hyperparameters of our network, we propose the following optimization scheme. We tune batch size N_b, patch size W_p, number of ResNet blocks N_r, and learning rate α. Using the grid search method we sample 4-dimensional space given by these hyperparameters and for every hyperparameter setting we (1) perform a training for a given amount of time, (2) do inference on unseen frames, and (3) compute the loss between inferred frames (O_4) and result of Jamriška et al. [2019] (GT_4) - which we consider to be ground truth. The objective is to minimize this loss. Note that the loss in step (1) and the loss in step (3) are both the same. Video frames (I) and style exemplar (S) courtesy of © Zuzana Studená.*

4.1.4 Temporal Coherency

Once the translation network with optimized hyperparameters is trained using the proposed patch-based scheme, style transfer to I can be performed in real-time or in parallel on the off-the-shelf graphics card. Even though such a frame-independent process yields relatively good temporal coherence on its own (as noted by Futschik et al.), in many cases, temporal flicker is still apparent. We aim to suppress it while keeping the ability of the network to perform frame-independent inference. We analyzed the source of the temporal instability and found two main reasons: (1) temporal noise in the original video and (2) visual ambiguity of the stylized content. We discuss our solution to those issues in the following paragraphs.

We observed that the appearance translation network tends to amplify temporal noise in the input video, i.e., even a small amount of temporal instability in the input video causes visible flicker in the output sequence. To suppress it, we use the motion-compensated variant of bilateral filter operating in the temporal domain [Bennett and McMillan 2005]. See our supplementary video (at 2:40) for the flicker reduction that can be achieved using this pre-filtering. Although bilateral filter requires nearby frames to be fetched into the memory, it does not violate our requirement for frame-independent processing.

Another observation we made is that filtering the input video reduces temporal flicker only on objects that have distinct and variable texture. Those that lack sufficient discriminatory information (e.g., homogeneous regions) flicker due to the fact that the visual ambiguity correlates with the network's ability to recall the desired appearance. To suppress this phenomenon, one possibility is to prepare the scene to contain only well distinctive regions. However, such an adjustment may not always be feasible in practice.

Instead, we provide an additional input layer to the network that will improve its discriminative power explicitly. This layer consists of a sparse set of randomly distributed 2D Gaussians, each of which has a distinct randomly generated color. Their mixture represents a unique color variation that helps the network to identify local context and suppress the ambiguity (see Fig. 4.7). To compensate for the motion in the input video, Gaussians are treated as points attached to a grid, which is deformed using as-rigid-as-possible (ARAP) image registration technique [Sýkora et al. 2009]. In this approach, two steps are iterated: (1) block-matching estimates optimal translation of each point on the grid, and (2) rigidity is locally enforced using the ARAP deformation model to regularize the grid structure. As this registration scheme can be applied independently for each video frame, the condition on frame independence is still satisfied.

The reason why the mixture of Gaussians is used instead of directly encoding pixel coordinates as done, e.g., in Liu et al. [2018] or Jamriška et al. [2019], is the fact that random colorization provides better localization and their sparsity, together with rotational symmetry, reduces the effect of local distortion, which may confuse the network. In our supplementary video (at 3:20) we, demonstrate the benefit of using the mixture of Gaussians over the layer with color-coded pixel coordinates. In case of extreme non-planar deformation (e.g., head rotation) or strong occlusion (multiple scene planes), additional keyframes need to be provided or the scene separated into multiple layers. Each keyframe or a scene layer has then its own dedicated deformation grid. We demonstrate this scenario in our supplementary video (at 2:56).

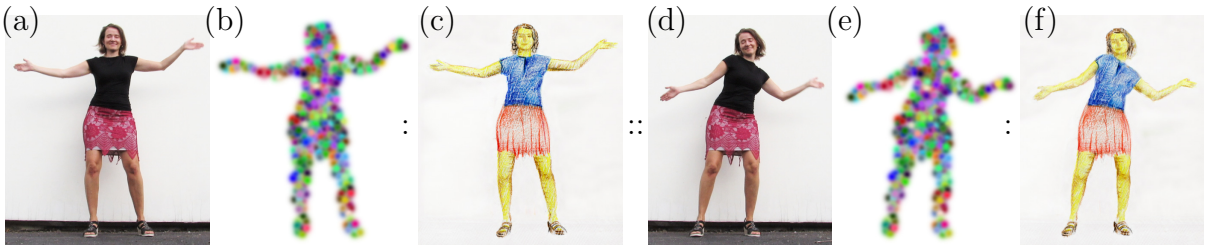
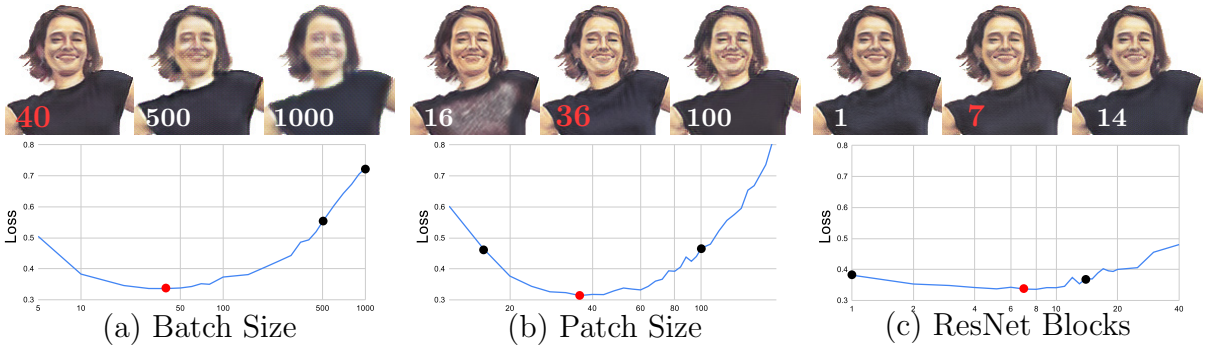


Figure 4.7: *To suppress visual ambiguity of the dark mostly homogeneous T-shirt in (a) an auxiliary input layer is provided that contains a mixture of randomly distributed and colored Gaussians (b). The translation network is trained on patches of which input pixels contain those additional color components. The aim is to reproduce the stylized counterpart (c). Once the network is trained a different frame from the sequence can be stylized (d) using adopted version of the auxiliary input layer (e). The resulting sequence of stylized frames (f) has notably better temporal stability (cf. our supplementary video at 2:40). Video frames (a, d) courtesy of © Zuzana Studená and style exemplar (b) courtesy of © Pavla Sýkorová.*

4.1.5 Results

We implemented our approach in C++ and Python with PyTorch, adopting the structure of the appearance translation network of Futschik et al. [2019] and used their recommended settings including training loss. Ground truth stylized sequences for hyperparameter tuning and comparison were produced using the video stylization method of Jamriška et al. [2019].



(a) Batch Size (b) Patch Size (c) ResNet Blocks

Figure 4.8: *Influence of important hyperparameters on visual quality of results. The loss, y-axes, is computed w.r.t. the output of Jamriška et al. [2019]. The best setting for each hyperparameter is highlighted in red: (a) The loss curve for the batch size N_b—the number of patches in one training batch (other hyperparameters are fixed). As can be seen, increasing N_b deteriorates visual quality significantly; it indicates that there exists an ideal amount of data to pass through the network during the back-propagation step. (b) The loss curve for the patch size W_p. The optimal size of a patch is around 36x36 pixels. This fact indicates that smaller patches may not provide sufficient context while larger ones could make the network less robust to deformation changes. (c) The loss curve for the number of ResNet blocks N_r that corresponds to the capacity of the network. As can be seen, settings with 7 ResNet blocks is slightly better than other results; however, this hyperparameter does have major impact on the quality of results. For additional experiments with hyperparameter setting, refer to our supplementary text.*

We performed fine-tuning of hyperparameters on a selection of frames from our evaluation sequences. We computed their stylized counterparts using the method of Jamriška

et al. [2019] and performed optimization using grid search on a cluster with 48 Nvidia Tesla V100 GPUs in 3 days. We searched over the following intervals: $W_p \in (12, 188)$, $N_b \in (5, 1000)$, $N_r \in (1, 40)$, $\alpha \in (0.0002, 0.0032)$. In total we sampled around 200,000 different settings of those hyperparameters. We found the optimal patch size to be $W_p = 36$ pixels, the number of patches in one batch $N_b = 40$, learning rate $\alpha = 0.0004$, and the number of ResNet blocks $N_r = 7$.

See Fig. 4.8 to compare visual quality for different hyperparameter settings. Note the substantial improvement in visual quality over different settings, which confirms the necessity of this optimization. An interesting outcome of the proposed hyperparameter optimization is a relatively small number of patches in one batch $N_b = 40$ (Fig. 4.8a). This value interplays with our choice of patch-based training scheme. Although a common strategy would be to enlarge N_b as much as possible to utilize GPU capability, in our case, increasing N_b is actually counterproductive as it turns training scheme into a full-frame scenario that tends to overfit the network on the keyframe and produce poor results on unseen video frames. A smaller number of randomly selected patches in every batch increases the variety of back-propagation gradients and thus encourages the network to generalize better. From the optimal patch size $W_p = 36$ (Fig. 4.8b) it is apparent that smaller patches may not provide sufficient context, while larger patches may make the network less resistant to appearance changes caused by deformation of the target object and less sensitive to details. Surprisingly, the number of ResNet blocks N_r (see Fig. 4.8c) does not have a significant impact on the quality, although there is a subtle saddle point visible. Similar behavior also holds true for the learning rate parameter α. In addition, we also examined the influence of the number of network filters on the final visual quality (see our supplementary material). The measurements confirmed that the number of filters needs to be balanced as well to capture the stylized content while still avoid overfitting.

With all optimized hyperparameters, a video sequence of resolution 640×640 with 10% of active pixels (inside the mask M^k) can be stylized in good quality at 17 frames per second after 16 seconds of training (see Fig. 4.1).

We evaluated our approach on a set of video sequences with different resolutions ranging from 350×350 to 960×540, containing different visual content (faces, human bodies, animals), and various artistic styles (oil paint, acrylic paint, chalk, color pencil, markers, and digital image). Simpler sequences were stylized using only one keyframe (see Figures 4.1, 4.3, 4.7, 4.11, and 4.12) while the more complex ones have multiple (ranging from two to seven, see Figures 4.14, 4.13, 4.15, and 4.16). Before training, the target sequence was pre-filtered using the bilateral temporal filter. In case that the sequence contains regions having ambiguous appearances, we compute an auxiliary input layer with the mixture of randomly colored Gaussians that follows the motion in the target sequence. During the training phase, we randomly sample patches inside the mask M^k from all keyframes k and feed them in batches to the network to compute the loss and backpropagate the error. Training, as well as inference, were performed on Nvidia RTX 2080 GPU. The training time was set to be proportional to the number of input patches (number of pixels inside the mask M^k), e.g., 5 minutes for a 512×512 keyframe with all pixels inside the mask. After training, the entire sequence can be stylized at the speed of roughly 17 frames per second. See our supplementary video (at 0:08 and 1:08) for the resulting stylized sequences.

4.1.6 Comparison

To confirm the importance of our patch-based training strategy, we conducted comparisons with other commonly used methods for data-augmentation that can help avoid overfitting such as adding Gaussian noise to the input, randomly erasing selected pixels, occluding larger parts of the input image, or performing dropout before each convolution layer. We found that none of these techniques can achieve comparable visual quality to our patch-based training strategy (see Fig. 4.9).

Figure 4.9: *To deal with the overfitting caused by a minimal amount of training data, we tried several commonly used techniques to enforce regularization. In all cases shown in this figure, we trained the network on the first frame; the shown results are zoomed details of the fifth frame. (a) is a result of the original full-frame training. (b-h) are results of full-frame training with some data augmentation. (i) is a result of our patch-based training strategy—see how our technique can deliver much sharper and significantly better visual quality results, please, zoom into the figure to better appreciate the difference. In case of (b-c), Gaussian noise was used to augment the data; (d) some pixels were randomly set to black; (e-f) some parts of the image were occluded; (g) dropout of entire 2D feature maps; (h) dropout of individual pixels before each convolution layer.*

We compared our approach with the state-of-the-art in keyframe-based video stylization [Jamriška et al. 2019]. For the results see Figures 4.10, 4.12, 4.14, 4.15, and our

supplementary video (at 0:08 and 1:08). Note how the overall visual quality, as well as the temporal coherence, is comparable. In most cases, our approach is better at preserving important structural details in the target video, whereas the method of Jamriška et al. often more faithfully preserves the texture of the original style exemplar. This is caused by the fact that the method of Jamriška et al. is non-parametric, i.e., it can copy larger chunks of the style bitmap to the target frame. Our method is parametric, and thus it can adapt to fine structural details in the target frame, which would otherwise be difficult to reproduce using bitmap chunks from the original style exemplar.

Regarding the temporal consistency, when our full-fledged flicker compensation based on the mixture of Gaussians is used our approach achieves comparable coherency in time to the method of Jamriška et al. It is also apparent that when multiple keyframes are used for stylization, ghosting artifacts mostly vanish in our method, unlike in Jamriška et al. When the original noisy sequence is used, or only the bilateral filtering is applied, the resulting sequence may flicker a little more when compared to the output of Jamriška et al. However, we argue that the benefits gained from random access and parallel processing greatly outweigh the slight increase of temporal flicker. Moreover, the order-independent processing brings also a qualitative improvement over the method of Jamriška et al. that tends to accumulate small errors during the course of the sequence, and visibly deteriorates after a certain number of frames.

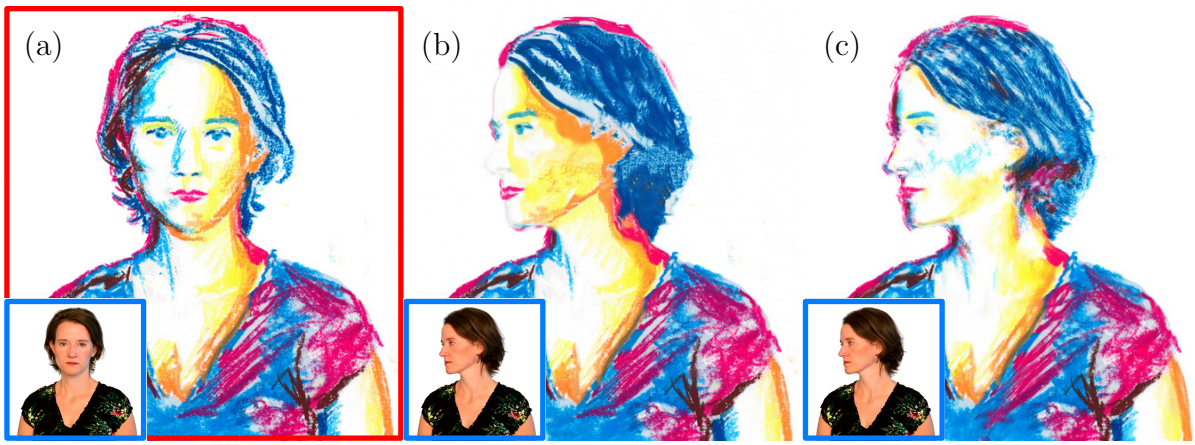


Figure 4.10: *When the target subject undergoes a substantial appearance change, the results of both [Jamriška et al. 2019] (b) and our method (c) exhibit noticeable artifacts. The parts that were not present in the keyframe are reconstructed poorly—see the face and hair regions where [Jamriška et al. 2019] produces large flat areas, while our approach does not reproduce the color of the face well. Video frames (insets of a–c) and style exemplars (a) courtesy of © Zuzana Studená.*

Performance-wise a key benefit of our approach is that once the network is trained, one can perform stylization of a live video stream in real-time. Even in the offline setting, when the training phase is taken into account, the overall end-to-end computation overhead is still competitive. On a 3 GHz quad-core CPU with Nvidia RTX 2080 GPU, a 512×512 sequence with 100 frames takes around 5 minutes to train until convergence and stylize using our approach, whereas the method of Jamriška et al. requires around 15 minutes.

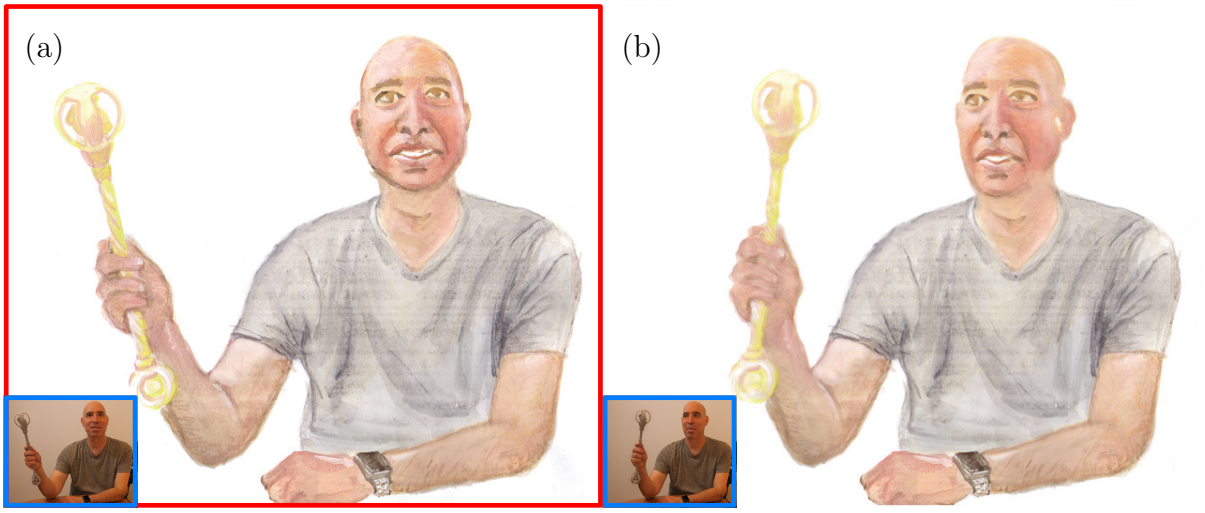


Figure 4.11: *Given one keyframe (a) and a video sequence (in blue), our method produces the stylized result (b). Video frames (insets of a, b) courtesy of © Adam Finkelstein and style exemplars (a) courtesy of © Pavla Sýkorová.*

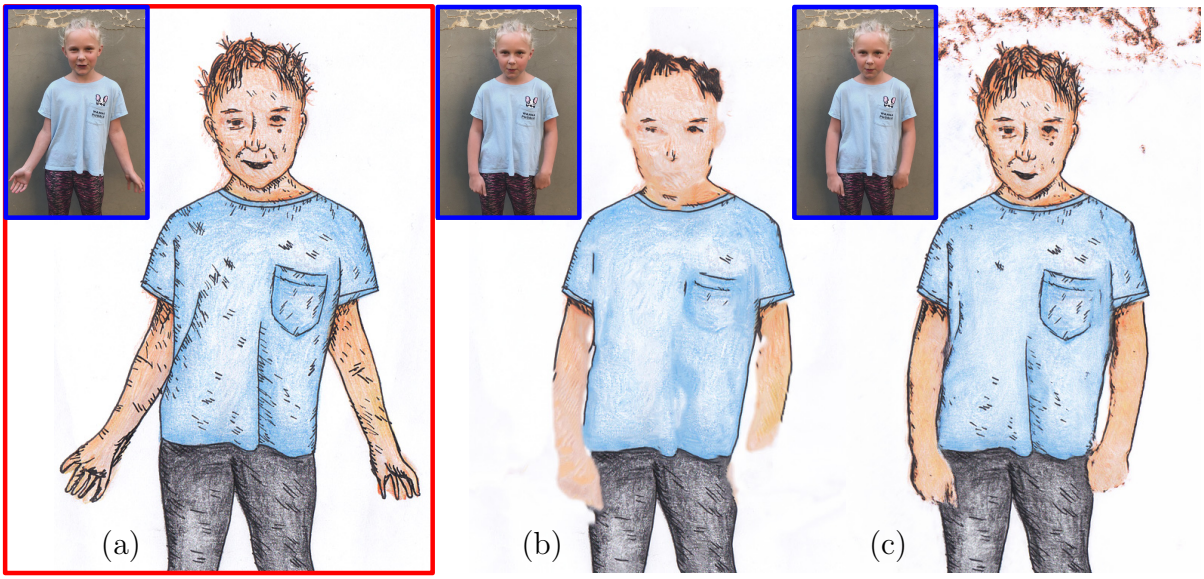


Figure 4.12: *For the state-of-the-art algorithm of Jamriška et al. [2019], contour based styles (a) present a particular challenge (b). Using our approach (c), the contours are transferred with finer detail and remain sharp even as the sequence undergoes transformations. Video frames (insets of a–c) and style exemplar (a) courtesy of © Štěpánka Sýkorová.*

4.1.7 Interactive Applications

To evaluate the ideas we presented in practice, we invited artists to work with our framework. We implement and experiment with three different setups in which the artists created physical as well as digital drawings. The goal of these sessions was to stylize one or more video keyframes artistically. Using a workstation PC, we provided the artists with a version of our framework that implements real-time interactive stylization of pre-prepared video sequences and stylization of live camera feeds.

Figure 4.13: *The Lynx sequence stylized using two keyframes (a, d). Notice how our method produces seamless transition between the keyframes while preserving fine texture of the style (b, c). Watch our supplementary video (at 1:22) to see the sequence in motion. Style exemplars (a, d) courtesy of © Jakub Javora.*

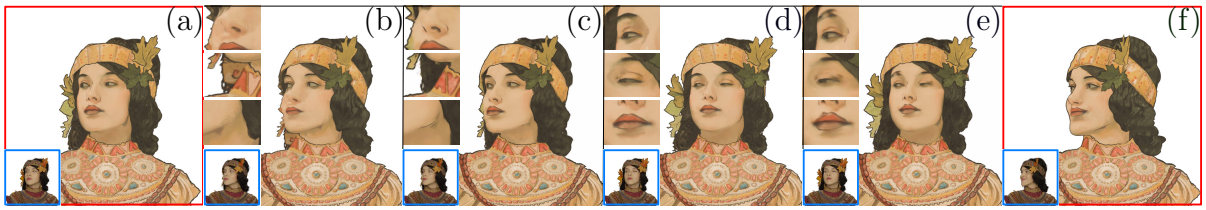


Figure 4.14: *Keyframes (a, f) were used to stylize the sequence of 154 frames. See the qualitative difference between [Jamriška et al. 2019] (b) and our result (c). Focusing mainly on zoom-in views, our approach better preserves contour lines around the nose and chin; moreover, the method of Jamriška et al. suffers from blending artifacts—the face is blended into the hair region. On the other hand, comparison on a different frame from the same sequence shows that the result of Jamriška et al. (d) is qualitatively superior to our result (e) on this particular frame. See the corresponding zoom-in views where the approach of Jamriška et al. produces cleaner results. Video frames (insets of a–f) and style exemplars (a, f) courtesy of © Muchalogy.*

These applications, all of which rely on and strongly benefit from the near real-time nature of patch-based training as well as the real-time performance of full-frame inference, naturally lend themselves to fast iteration. The artist is provided with real-time feedback that approximates what the final result of video stylization might look like, thus reducing the possibility of running into issues with artifacts that would be difficult to alleviate later on.

During the sessions, artists especially appreciated seeing video results very quickly, as it helps steer creative flow and offers the possibility of perceiving the effect of individual changes in the style exemplar at a glance. The overall experience was described as incredibly fun and paradigm-changing, with little to no negative feedback. Using this system is intuitive and even suitable for children. These different scenarios are described in detail in the supplementary material.

4.2 Robust Neural Video Stylization

An important yet still missing contribution to the subfield of video sequence stylization is the ability to allow artists to stylize a set of images with arbitrary yet similar content in a semantically meaningful way, while preserving the target subjects' critical structural features. In this section, we propose a solution to this task. In contrast to previous neural techniques, in our proposed framework, the user explicitly encodes the semantic intent by specifying a stylized counterpart for a selected image from the set that needs to

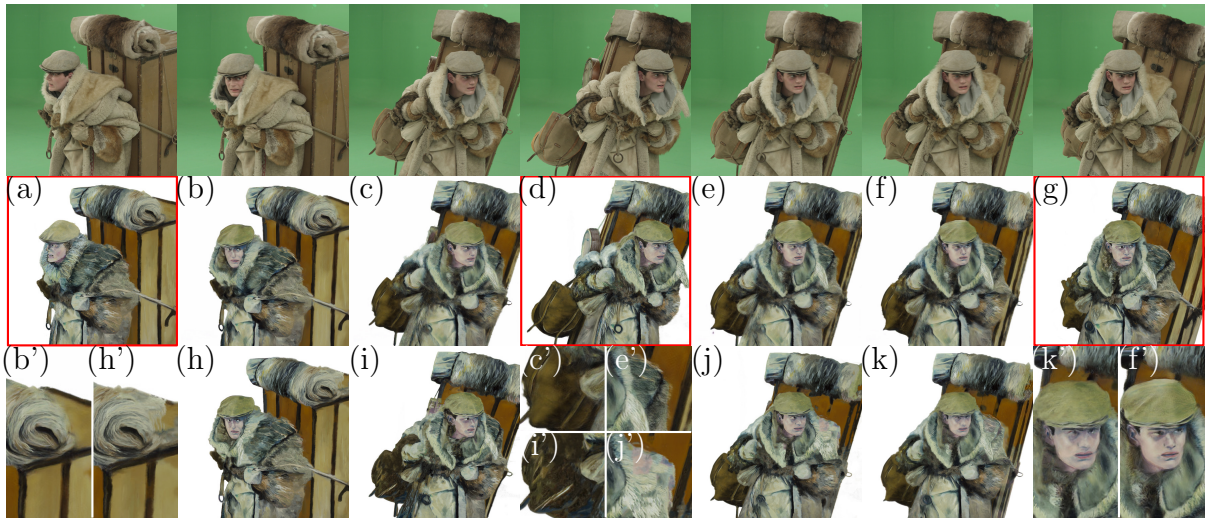


Figure 4.15: *A complex input sequence (the first row) with seven keyframes, three of them are shown in (a, d, g). Here we compare our approach to the approach of Jamriška et al. [2019]. See our result (b) and theirs (h) along with the close-ups (b', h'); due to their explicit handling of temporal coherence, the texture of the fur leaks into the box (h'). Next, compare our result (c) to theirs (i); our approach better reconstructs the bag (c', i'). Their issue with texture leakage manifests itself again on the shoulder in (j, j'), notice how our approach (e, e') produces a clean result. Lastly, see how our result (f, f') is sharper and the face is better pronounced compared to the result of Jamriška et al. [2019] (k, k'), which suffers from artifacts caused by their explicit merging of keyframes. Video frames (top row) and style exemplars (a, d, g) courtesy of © MAUR film.*

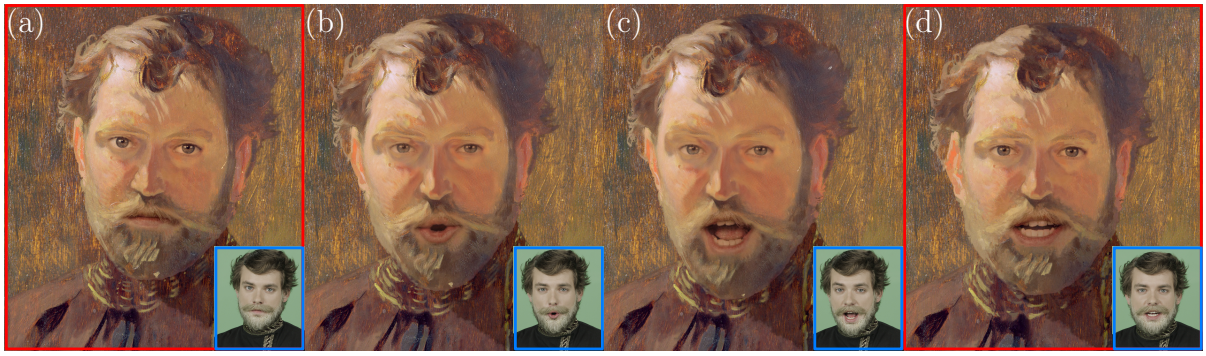


Figure 4.16: *An example sequence of 228 video frames (in blue) as stylized from two keyframes (a, d). Results of our method (b, c) stay true to style exemplars over the course of the sequence. Video frames (insets of a–d) and style exemplars (a, d) courtesy of © Muchalogy.*

be stylized. Using this single style exemplar, we then train an image-to-image translation network that stylizes the remaining images. While this approach bears a resemblance to the method presented in previous section, where a similar workflow is used, a key difference in this technique is that we consider other frames from the input sequence during the training phase. This enables us to ensure temporal stability without explicit guidance and better preserve style when the remaining video frames deviate from the original keyframe. Moreover, thanks to this increased robustness, our framework goes beyond video stylization. One can use it also in more challenging scenarios, including

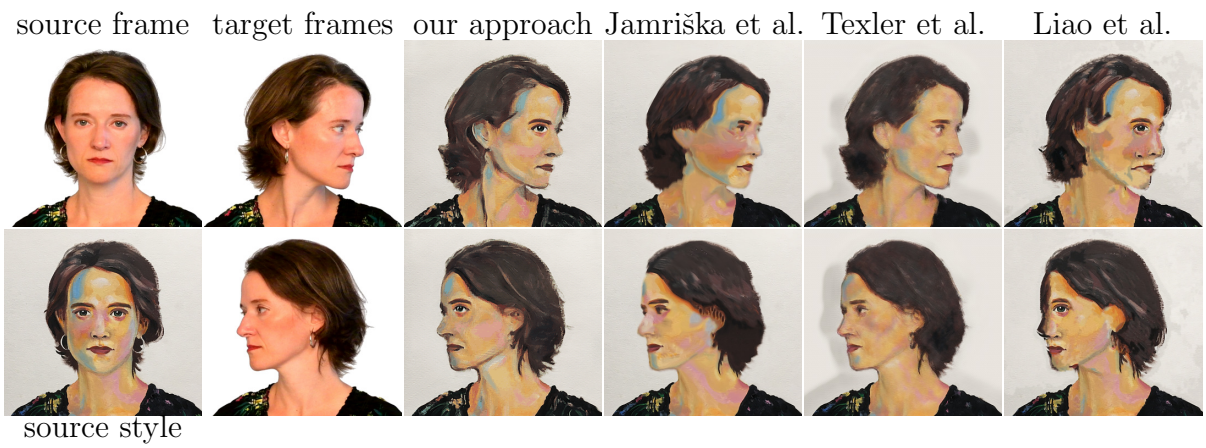


Figure 4.17: *An example of style transfer with limited auxiliary pairing—an artist prepares a stylized version (source style) of a selected video frame (source frame). Then an image-to-image translation network is trained to transfer artist's style to other video frames (target frames). During the training phase a subset of target frames as well as the source frame and its stylized counterpart are taken into account. Once the network is trained, the entire sequence can be stylized in real-time (our approach). In contrast to state-of-the-art in example-based video stylization [Jamriška et al. 2019] our approach better preserves important visual characteristics of the style exemplar even though the scene structure changed considerably (head rotation). The advantage of having an auxiliary stylized pair is also visible in comparison with the output of Deep Image Analogies of Liao et al. [2017]. Although the style's texture is preserved reasonably well, the transfer is not semantically meaningful.*

auto-completion of a panorama painting, stylization of 3D renders, or different portraits captured under similar illumination conditions.

4.2.1 Our Approach

As input to our method, we take pairs of images $K = (X, Y)$ called *keyframes*. They represent a visual translation from a source visual domain of X into a target domain of Y. For instance X can be a photo and Y its stylized counterpart prepared by an artist (see Fig. 4.18). Note that our key assumption about K is that it should be as small as possible, in practice even a *single* keyframe is usually sufficient. This is in line with our central motivation to reduce the amount of manual work since the creation of keyframes is time-consuming and thus prohibitive. In addition to K, the user also provides a set of unpaired images Z, which they would like to stylize. The images in Z can be arbitrary, but our method works best if their domain is similar or same as X. For instance Z and X can be frames from the same video sequence or photos from the same location, etc. If there is a larger number of images in Z, it is beneficial to prune it as smaller number of images in Z usually has a positive effect on the resulting quality (see Fig. 4.24). Both keyframes K as well as unpaired images Z are used during an optimization process that produces a neural translation model $\mathcal{F}$. Using $\mathcal{F}$ one can stylize Z in a semantically meaningful way, i.e., produce a set of output images O in which important visual features of artistic style Y are reproduced at appropriate locations.

As $\mathcal{F}$, we use the network architecture design of Futschik et al. [2019] (see Fig. 4.19), a U-Net-type network, which is particularly suitable for style transfer tasks as it allows to reproduce important high-frequency details that are crucial for generating believable

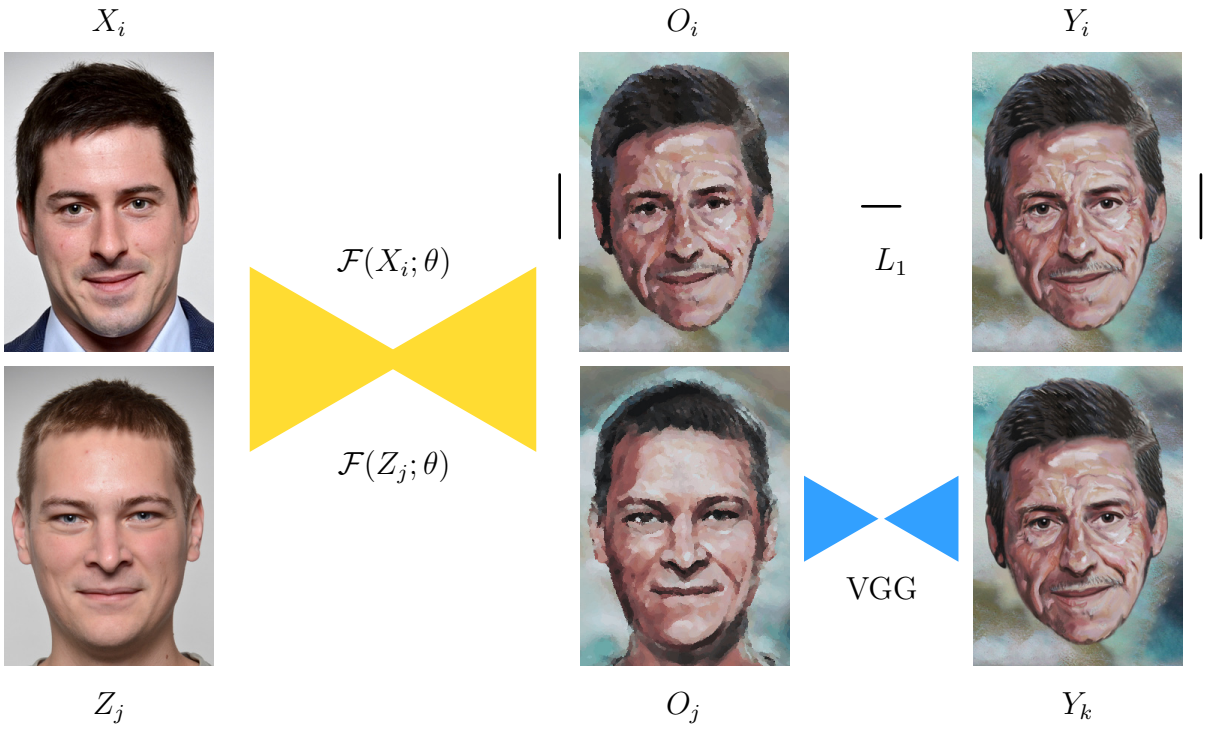


Figure 4.18: *An overview of our approach—we optimize weights θ of a translation network $\mathcal{F}$ which accepts images from a source domain X or Z and produces output images O with a similar appearance as those in the target domain Y. The high-frequency details are preserved well, thanks to the L_1 loss computed on the artist-created style images Y which have the same structure as the input images X, while the style consistency on other images Z is enforced due to the VGG loss. Source style © Graciela Bombalova-Bogra, used with permission.*

artistic styles. In the original method of Futschik et al. $\mathcal{F}$ was trained on a large dataset of K which is intractable in our scenario. Our previous patch-based approach uses the network architecture of $\mathcal{F}$ as well in a similar setting as ours, i.e., small number of keyframes K, however, that method struggles with larger structural changes in the target images Z.

To address this issue, we leverage the fact that the set of target images Z is known beforehand and thus we can incorporate this additional knowledge into the optimization process. To do that, we introduce a different training strategy. The process is a combination of two complementary objectives, illustrated in Fig. 4.18, which we minimize as we train $\mathcal{F}$:

- L_1 *loss* on the original translation pairs K, ensuring that keyframes are represented as closely as possible.
- *VGG loss* between the images from set Z and set Y, which acts as a regularizer for the stylized images O.

Combining these two, we obtain the objective function we would like to minimize:

$$\sum_i |\mathcal{F}(X_i;\theta) - Y_i| + \lambda \sum_{j,k} \sum_l \|\mathcal{G}^l(\mathcal{F}(Z_j;\theta)) - \mathcal{G}^l(Y_k)\|^2 \tag{4.1}$$

where θ is a set of weights of $\mathcal{F}$ which we would like to optimize, $\mathcal{G}^l$ stands for Gram correlation matrix calculated at layer $l \in L$ after extracting VGG network responses [Si-

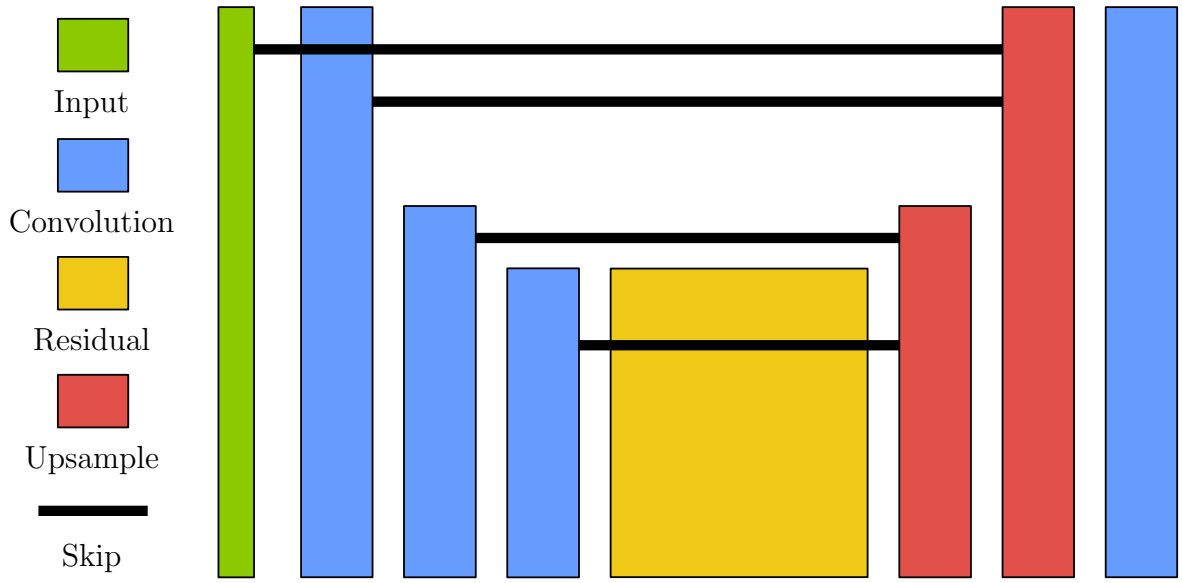


Figure 4.19: *A network architecture used for our model $\mathcal{F}$: input layer (green), one 7×7 and two 3×3 convolution blocks (blue), nine 3×3 residual blocks (yellow), two 3×3 upsampling blocks (red), and one additional block with 7×7 convolutions (blue). Skip connections (black) are used to connect downsampling and upsampling layers.*

monyan and Zisserman 2014] of the given image, and λ is a weighting coefficient which we set to $100/(|Z||L|)$ for all conducted experiments.

Contrary to previous techniques [Gatys et al. 2016; Johnson et al. 2016] which compute Gram matrix from a subset of layers we found that evaluating the loss at every layer $l \in L$ of VGG is beneficial in terms of measuring the overall style quality. However, this is computationally more expensive and thus our method generally requires an order of magnitude more time to produce the final results. These previous methods use the term purely as a proxy for style transfer. In our case we use it as regularizer to prevent the model from overfitting to the keyframes. This effect is visible in Fig. 4.20, where if we take away the VGG loss, the resulting $\mathcal{F}$ is unable to generalize beyond K whereas using VGG loss only will negatively affect the content.

By minimizing the objective (4.1) we produce a trained model $\mathcal{F}$, which in turn is able to stylize the images from Z via a feed-forward pass. An important aspect to notice is that unlike most previous style transfer techniques, our approach does not enforce any content loss explicitly. We find that content losses found in literature [Gatys et al. 2016; Kolkin et al. 2019] tend to be detrimental to the quality of style transfer, especially when higher frequencies are concerned. It causes a particular washed-out look where important style details are missing (see Fig. 4.21). An objection to our argument could be that without explicit penalty on the content preservation, the model can resort to memorizing the keyframes and return Y regardless the content in target images Z. This would eventually minimize both the L_1 error as well as the VGG loss. The reason why the optimization process does not end up using this trivial solution is twofold. We argue that due to the limited receptive field of $\mathcal{F}$, it has to learn an effective encoding of the input; in addition, since the VGG loss is relatively weak and serves only as a non-linear regularizer, it makes the trivial solution difficult to find during the optimization process. Moreover, by optimizing a one-to-one mapping between images of perceptually similar

Figure 4.20: *An ablation study demonstrating the importance of individual terms in our objective function (4.1)—a stylized pair (X_1, Y_1) (source photo, source style) is used together with Z_1 (target photo) to optimize weights of model $\mathcal{F}$. When only VGG loss is used, the identity of a person in the target photo deteriorates. On the other hand when only L_1 loss is used during optimization source, style is not preserved well. By combining L_1 loss and VGG loss in (4.1) we get the result which produces a good balance between identity and style preservation. Source style © Graciela Bombalova-Bogra, used with permission.*

semantic structure (X to Y), we posit that this acts as an implicit content preservation technique.

4.2.2 Results

We implemented our approach using PyTorch [Paszke et al. 2019]. For all experiments, we use Adam optimizer with learning rate 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. We found that higher rate does not work well when performing many Gram matrix operations that are prone to producing exploding gradients. For the network model $\mathcal{F}$, we use 9 residual blocks, which is in line with previous approaches [Futschik et al. 2019; Texler et al. 2020b]. However, since in our optimization batch size is equal to 1 we use instance normalization [Ulyanov et al. 2016c] instead of batch normalization. All layers used for Gram matrix computation are post-activated with ReLU to better incorporate non-linearity. In each experiment, we let the optimization process run for approximately 100k iterations, which translates into roughly 3–6 hours of wall time on a single NVIDIA V100 GPU, depending on the target resolution. The resolutions we produce range from 512px to 768px as longer side of the image, with the shorter side scaled appropriately to preserve correct aspect ratio given by the input images.

We evaluated our approach in five different use cases to demonstrate its wider range of applicability: (1) keyframe-based video stylization, (2) style transfer to 3D models, (3) autopainting panorama images, (4) example-based stylization of portraits, and (5) real-time stylization of video calls.

Video stylization results together with a side-by-side comparison of the output from previous techniques [Jamriška et al. 2019; Texler et al. 2020b] is presented in Figures 4.17 and 4.22 as well as in our supplementary video. In each experiment, we selected a keyframe X from the input video sequence V which was stylized by an artist to produce Y. Then a 10% of video frames from V were sampled uniformly to get the set Z. Using this input, the weights θ of the network $\mathcal{F}$ were optimized and used to stylize the entire sequence V. In Fig. 4.23 we compare the scenario where multiple keyframes K are used to stylize V. We also considered an option that all frames from V are used as Z,

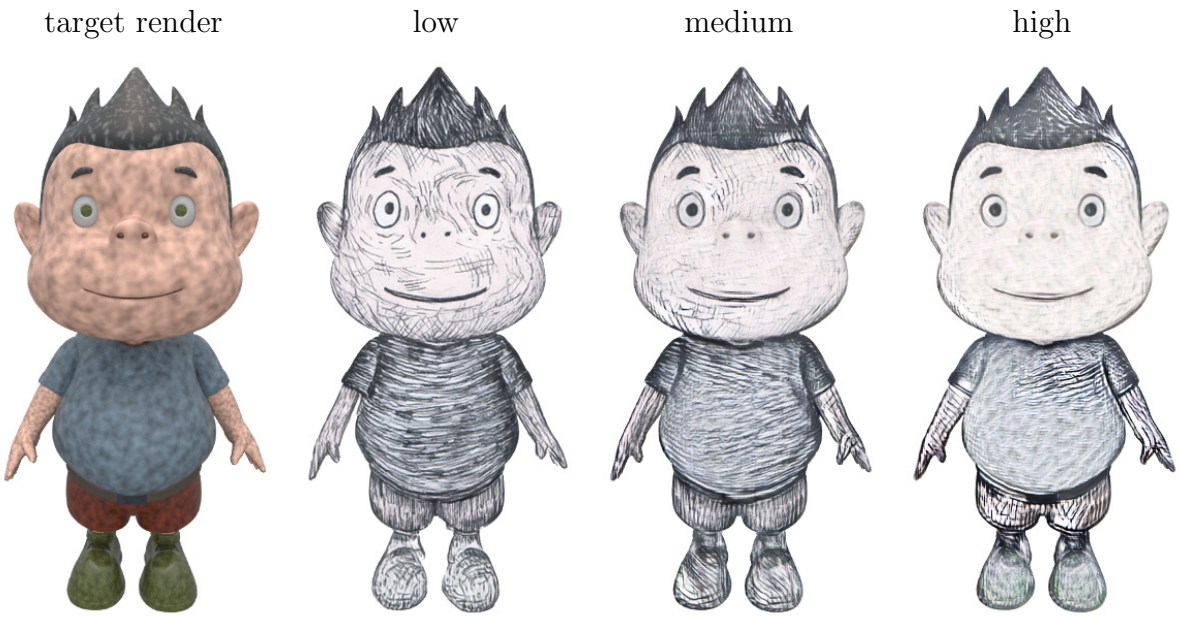


Figure 4.21: *An illustration of a wash-out effect caused by adding an explicit content loss term [Kolkin et al. 2019] into our objective function (4.1). Target render stylized using model $\mathcal{F}$ optimized on a stylized pair from Fig. 4.25 with low, medium, and high content loss weight. Note how style details deteriorate gradually with the increasing content loss. Source style © Štěpánka Sýkorová, used with permission.*

or instead of using uniform sampling we selected 10% of frames that represent the most significant changes in the scene. We found that sparse uniform sampling has usually the best performance (see Fig. 4.24).

As visible from the results and comparisons, our approach can better preserve style details during a longer time frame even if the scene structure changes considerably with respect to X. Also, note how the resulting stylized sequence has better temporal stability implicitly without performing any additional treatment, which contrasts with previous techniques [Jamriška et al. 2019; Texler et al. 2020b] that need to handle temporal consistency explicitly.

Style transfer to 3D models resembles video stylization use case, however, there are specific features worth separate discussion. In this scenario we let the user select a camera viewpoint from which a 3D model is rendered to get image X. As the network $\mathcal{F}$ is sensitive to local variations in X, it is important to avoid larger flat regions which can make the translation ambiguous. Due to this reason we add a noisy texture to the 3D model to alleviate the ambiguity (see source render in Fig. 4.25). An artist then prepares the stylized counterpart Y and the model is rendered again from a few different viewpoints to produce Z. Using those inputs, weights θ of the network $\mathcal{F}$ are optimized and the translation network can then be used in an interactive scenario where the user changes the camera viewpoint, the 3D model is rendered on the fly, and immediately stylized using $\mathcal{F}$. See Figures 4.25 and 4.26 and our supplementary video for results in this scenario. As in the video stylization case when compared to other techniques [Gatys et al. 2016; Kolkin et al. 2019; Jamriška et al. 2019; Texler et al. 2020b] our approach better preserves the style exemplar (c.f. Fig. 4.25) and implicitly maintains temporal consistency.

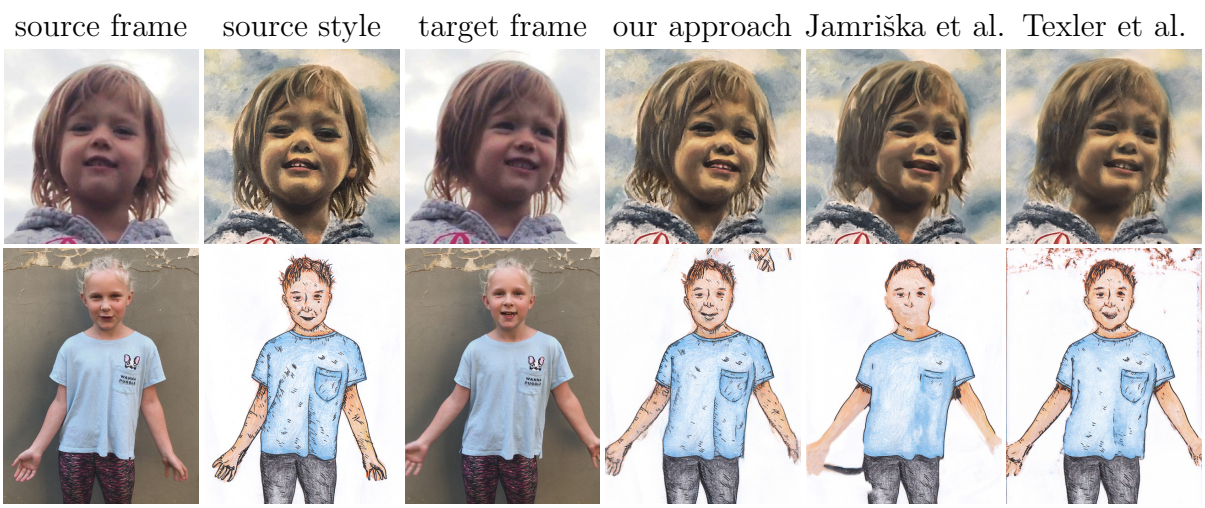


Figure 4.22: *Video stylization results—in each video sequence (rows) a selected frame (source frame) is stylized using different artistic media (source style). The network is then trained using this stylized pair and a subset of frames from the entire video sequence (target frame). The results of our method (our approach) are compared with the output of concurrent techniques: [Jamriška et al. 2019] and [Texler et al. 2020b]. Note how our method better preserves important style details and visual features of the target frames. Previous style transfer techniques tend to produce wash out artifacts due to significant structural changes with respect to the source frame. Video frames and style (top row) © Zuzana Studená, and (bottom row) © Štěpánka Sýkorová, used with permission.*

In the panorama auto-painting scenario we consider a set of photos P taken from the same location by rotating the camera around its center of projection. We compute a set of homographies H between photos in P using the method of Brown et al. [2007]. Then we let the artist pick one photo from P as X and produce its stylized counterpart Y. Remaining photos in P are used as Z. After the optimization one can use $\mathcal{F}$ to stylize all photos in P, stitch them together using H, and either produce a cylindrical unwrap or alternatively use an interactive scenario where the user changes the relative camera rotation from which a pinhole projection can be computed and stylized in real-time using $\mathcal{F}$. As visible in Fig. 4.27 and 4.28 from the comparisons with method of Liao et al. [2017] and Kolkin et al. [2019] our approach better preserves the original style details as well as semantic context.

In the example-based portrait stylization use case a set of portraits U is assumed to be taken under similar lighting conditions. One portrait from U is used as X and stylized to get Y. The rest of portraits in U is used in Z. Resulting model $\mathcal{F}$ can then be used to stylize all portraits in U. In Fig. 4.29 stylization results for two different style exemplars are presented. It is apparent that our approach produces a reasonable compromise between identity and style preservation whereas previous neural methods such as the methods of Gatys et al. [2016] or Kolkin et al. [2019] tend to preserve identity better, but lose style details. On the other hand, the patch-based technique of Fišer et al. [2017] reproduces style better, nevertheless, has difficulties retaining identity.

In real-time stylization of video calls we let the user record a short video sequence V which captures her face during a regular video meet. A most representative frame is selected from V and used as X. An artist then produces its stylized counterpart Y and 10% of other frames in V are used as Z. A model $\mathcal{F}$ is optimized using those inputs.

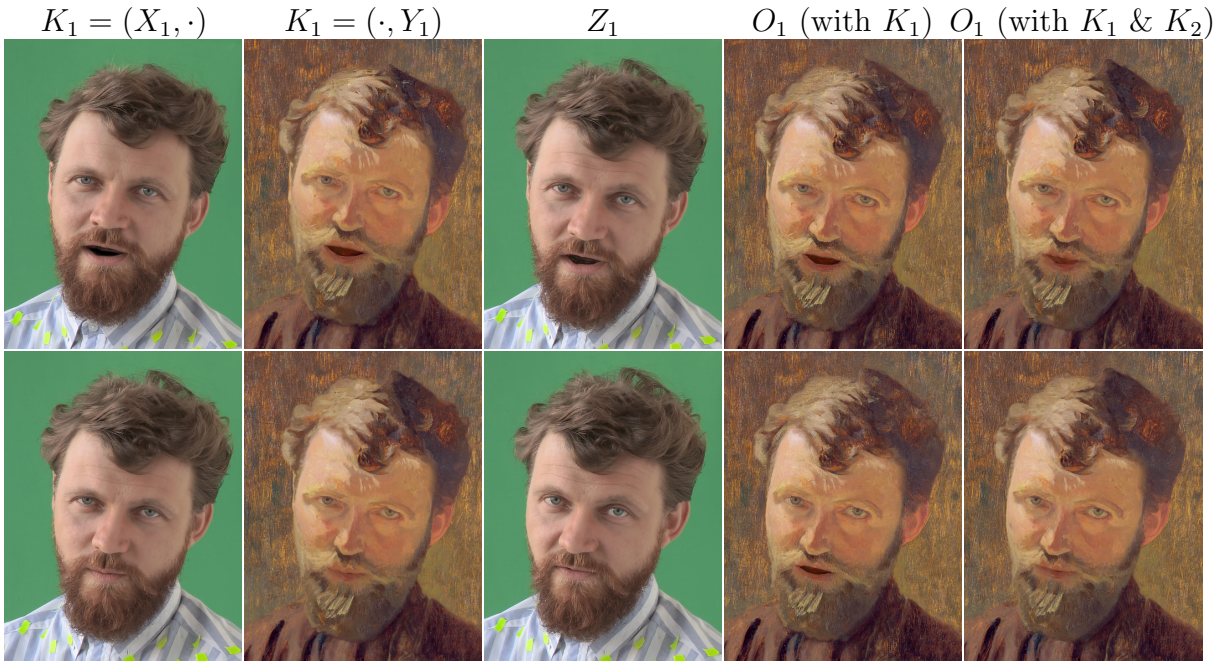


Figure 4.23: *Example of video stylization with multiple keyframes—two keyframes* $K_1 = (X_1, Y_1)$ *and* $K_2 = (X_2, Y_2)$ *were created by painting over the input video frames* X_1 *&* X_2 *to get their stylized counterparts* Y_1 *&* Y_2*. First, our network* $\mathcal{F}$ *was trained using only single keyframe* K_1 *and applied to stylize input video frames* Z_1 *&* Z_2 *to produce* O_1 *&* O_2 *(with* K_1*). Note, how closed mouth in* Z_2 *was not stylized properly in* O_2 *(with* K_1*). By adding* K_2 *to the list of keyframes used during training phase, open and closed mouth is stylized better, see* O_1 *&* O_2 *(with* K_1 *&* K_2*). Frames* X_1*,* X_2*,* Y_1*,* Y_2*,* Z_1 *&* Z_2 *© Muchalogy, used with permission.*

Then, during the next video call $\mathcal{F}$ is used to stylize captured video frames in real-time. See Fig. 4.30 and our supplementary video for an example of such interactive stylized video call. From the comparison with the method of Texler et al. [2020b] it is visible that our approach not only better preserves the overall style quality but also retains temporal stability which is difficult to accomplish by the method of Texler et al. in this kind of interactive scenario.

4.2.3 Perceptual study

In order to qualitatively evaluate our approach, we performed a perception study comparing the outputs of our method with the outputs of three state-of-the-art techniques (Jamriška et al. [2019], Kolkin et al. [2019], and Texler et al. [2020a] (green points)). In our experiment we wanted to evaluate how well our method reproduces the given artistic style and how well it preserves the content of the target image. To perform the evaluation, we collected data via an online survey, where we presented 170 participants with a randomized set of comparisons (2AFC) asking to choose which anonymized stylization reproduces style or preserves content better. In total each participant responded to 28 questions. In each question, an output from a different method was paired with the output from our technique using the same input data.

We set out a null hypothesis that "there is no statistically significant difference in the content preservation or style reproduction between the results of our method and the

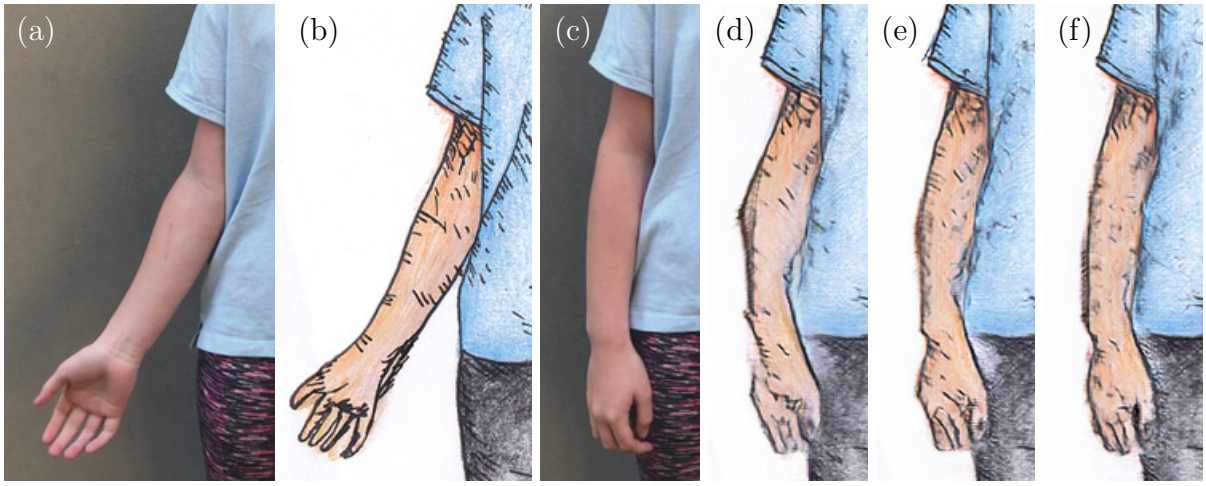


Figure 4.24: *A different sampling strategy for a selection of frames in Z—a source frame from a sequence V (a) and its stylized counterpart (b) are used as K. Then weights of $\mathcal{F}$ are optimized with K and Z, where Z contains all frames from V (d),* 10% *of uniformly sampled frames from V (e), and* 10% *of adaptivelly sampled frames from V (f). Note how dense sampling tends to produce distortion artifacts on a rare hand pose (c) due to overfitting on a different pose that is more frequent in the sequence V (a) whereas sparse sampling generalizes better. Source video frames (a, c) and style (b) © Štěpánka Sýkorová, used with permission.*

other methods." Then we discussed the probability of rejection of the null hypothesis using the data we collected via Student's t-test. In the style reproduction category, we were able to reject the null hypothesis with more than 99% probability in comparison to all tested methods in favor of our method. In the content preservation category, we were able to reject the null hypothesis with more than 99% probability, but only the comparison with the method of Jamriška et al. was in favor of our method while the other two were not.

4.3 Conclusion and Future Work

In this Chapter we proposed two new approaches for example-based stylization of video sequences. In Section 4.1, we have employed the core architecture of our approach described in Section 3.1 together with a patch-based learning approach to present an interactive framework, allowing real-time video sequence stylization as well as real-time feedback to changes being made in the given style example. On top of that, the patch-based learning approach allows learning target artistic styles without a large dataset, which would previously be infeasible. In Section 4.2, we have presented an image translation network for video sequences, that addresses the common quality deterioration, once the frames being stylized becomes too different from a keyframe, by employing an alternative loss function which allows better detail retention and style reproduction, consistent over the entire sequence.

Although our framework described on Section 4.1 brings substantial improvements over the state-of-the-art and makes keyframe video stylization more flexible and interactive, there are still some limitations that could represent a potential for further research. Despite the fact our technique uses different computational machinery than

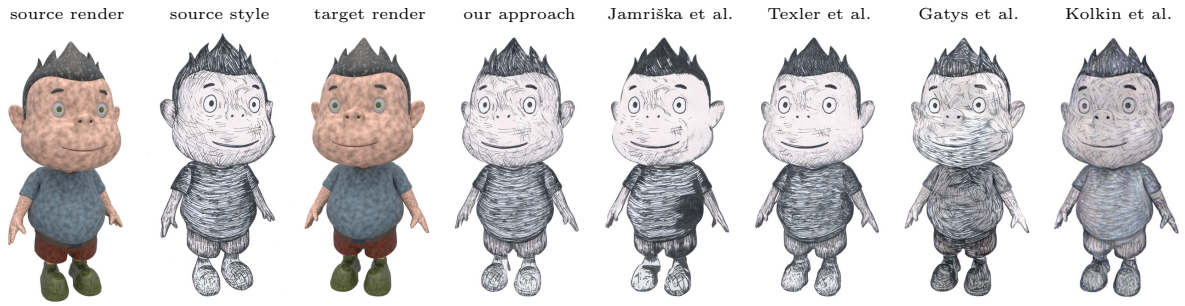


Figure 4.25: *Stylization of 3D renders—a colored 3D model enhanced with an artificial noisy texture to avoid large flat regions (source render) is stylized at a selected viewpoint by an artist (source style). The network is then trained using the stylized pair and a set of additional renders of the same model viewed from a different direction (target render). The trained network can then be used to stylize the rendered 3D model from a different user-specified position in real-time (our approach). When compared to other concurrent style transfer techniques ([Jamriška et al. 2019; Texler et al. 2020b; Gatys et al. 2016; Kolkin et al. 2019]) our approach better preserves important high-frequency details of the original style exemplar while being able to adapt to a new pose in a semantically meaningful way. Source style © Štěpánka Sýkorová, used with permission.*

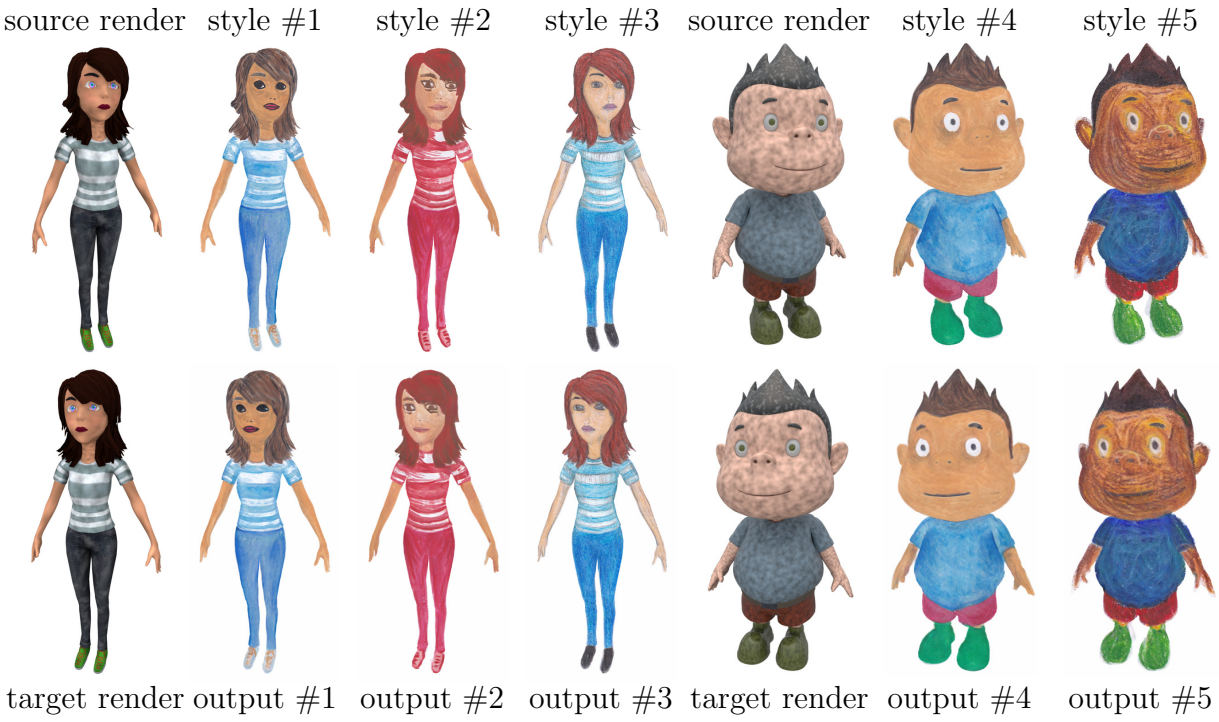


Figure 4.26: *Stylization of 3D renders (cont.)—a colored 3D model enhanced by a noisy texture (source render) is stylized by hand using various artistic media (style #1–#5). The resulting image translation network $\mathcal{F}$ is then used to stylize the same 3D model (output #1–#5) rendered from a different viewpoint (target render) in real-time. Source styles (#1–#5) © Štěpánka Sýkorová, used with permission.*

current state-of-the-art [Jamriška et al. 2019] (deep convolutional network vs. guided patch-based synthesis), both approaches share similar difficulties when stylized objects change their appearance substantially over time, e.g., when the object rotates and thus reveals some unseen content. Although our approach often resists slightly longer than

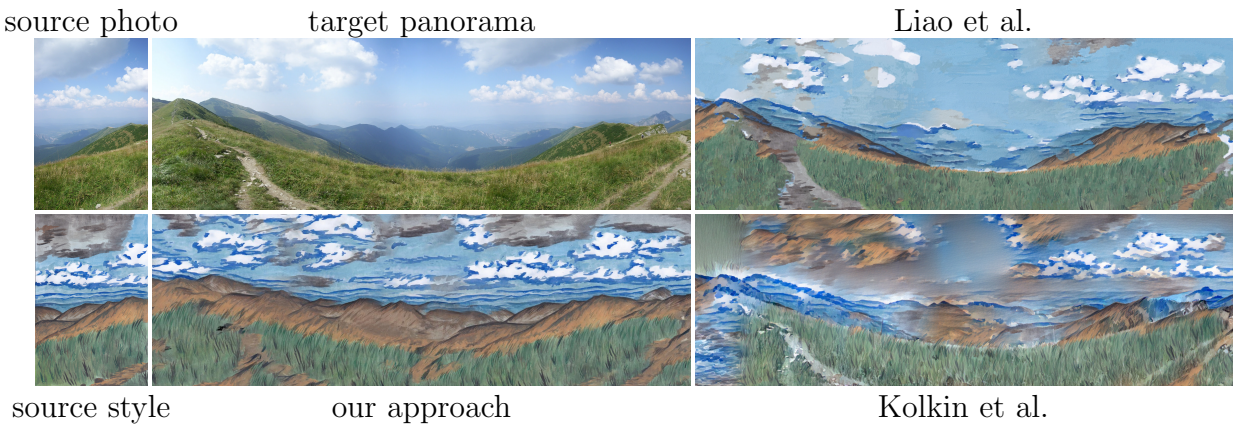


Figure 4.27: *Panorama stylization results—a photo (source photo) is selected from a set of shots taken around the same location by rotating a camera (target panorama) and stylized using different artistic media (source style). The network is then trained using the stylized pair and a subset of photos of the panoramic image (target panorama). Finally, the network is used to stylize each shot, and the entire panorama is stitched together (our approach). In contrast to previous techniques [Liao et al. 2017; Kolkin et al. 2019] our approach better preserves essential artistic features and transfers them into appropriate semantically meaningful locations. See also results with additional styles in Fig. 4.28. Source style © Štěpánka Sýkorová, used with permission.*

Figure 4.28: *Panorama stylization results (cont.)—two additional artistic styles (source style) used to stylize the panorama shown in Fig. 4.27. Note how our approach (stylized panorama) handles also a higher level of abstraction (first row). Source style (top row) © Jolana Sýkorová, used with permission.*

patch-based synthesis due to the ability to generalize better, it usually cannot invent consistent stylization for new features that were not stylized in the original keyframe, see Fig. 4.10. In this case, the user needs to provide additional keyframes to make the stylization consistent. As compared to the method of Jamriška et al. our approach may

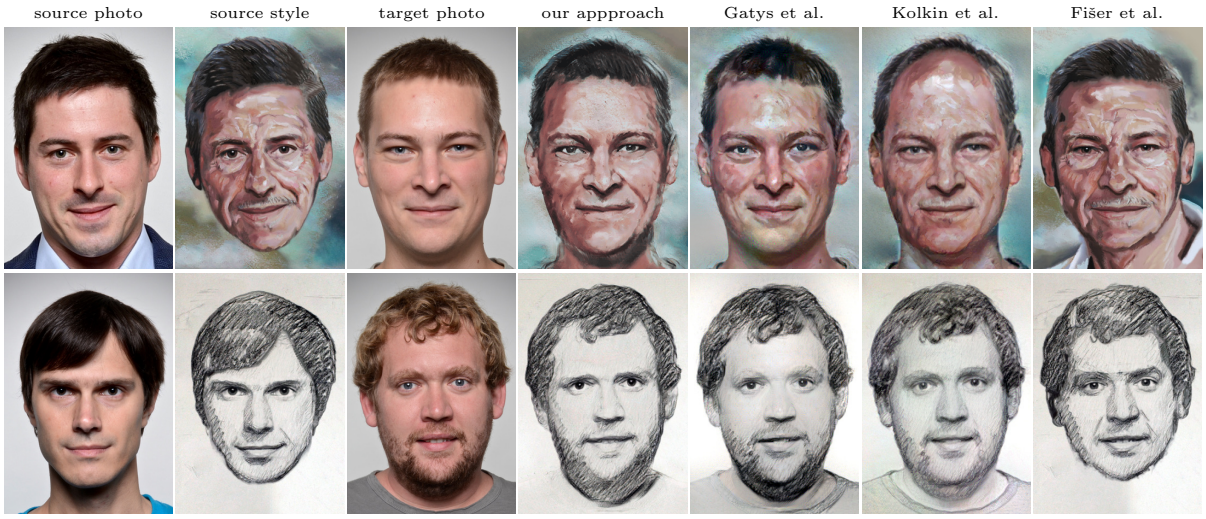


Figure 4.29: *Stylization of portraits—a portrait photo (source photo) taken from a set of portraits captured under similar lighting conditions is stylized by an artist (source style). The network is then trained on the stylized pair and other portraits from the original set (target photo). Once trained the network can be used to stylize the other portraits (our approach). Even in this more challenging scenario our method produces a reasonable compromise between style and identity preservation whereas concurrent techniques suffer either from loosing important high-frequency details ([Gatys et al. 2016; Kolkin et al. 2019]) or have difficulties to retain identity ([Fišer et al. 2017]). Source style (top row) © Graciela Bombalova-Bogra and style (bottom row) © Adrian Morgan, used with permission.*

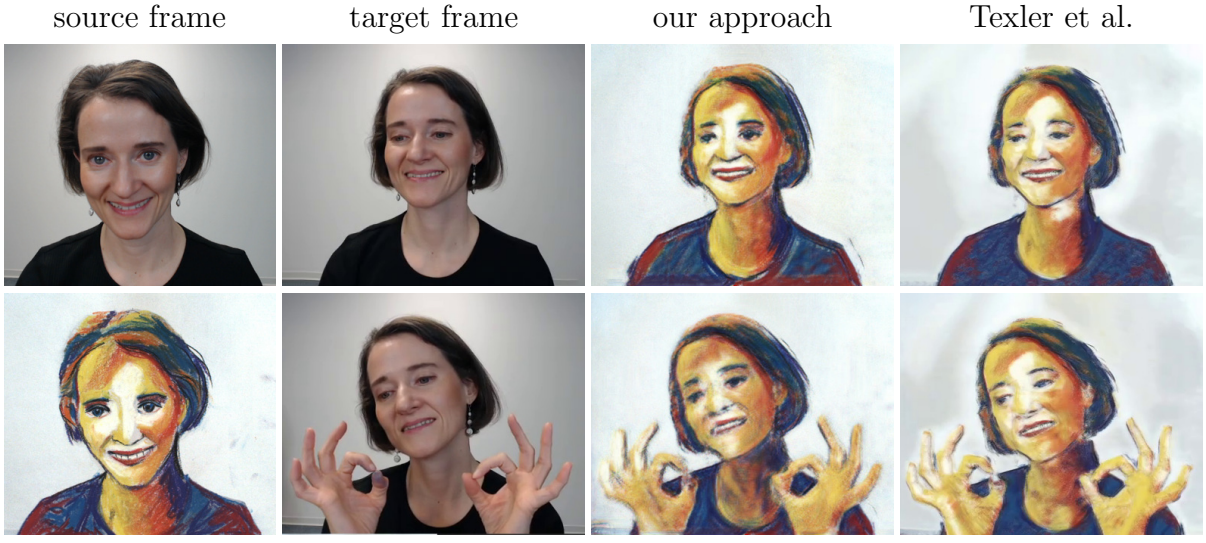
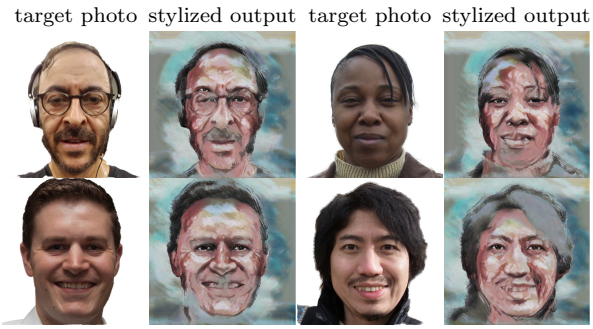
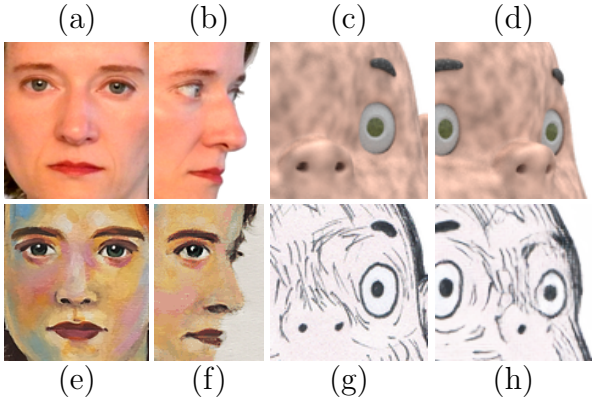


Figure 4.30: *Real-time stylization of video calls—a frame from a training sequence (source frame) is stylized by an artist (source style). The network weights are then optimized using this stylized pair and remaining frames from the training sequence. The final image translation model can be used for real-time stylization of a new video conference call that contains the same person and have similar lightihg conditions (target frames). Note that in contrast to the method of Texler et al. [2020b] our approach better preserves style details and keeps the stylization more consistent in time (see also our supplementary video). Video frames and source style © Zuzana Studená, used with permission.*

encounter difficulties when processing keyframes at a higher resolution (e.g., 4K) to stylize high-definition videos. Although the size of patches, as well as the network capacity, can be increased accordingly, the training may take notably longer time, as a different multi-scale approach [Wang et al. 2018c] could be necessary. However, the problem of training of larger models is an active research topic in machine learning, so we believe that soon, more efficient methods will be developed so that our technique would be applicable also at higher resolutions. Although our approach does not require the presence of previous stylized frames to preserve temporal coherency, the motion-compensated bilateral filter, as well as the creation of layer with a random mixture of colored Gaussians, requires fetching multiple video frames. Even though those auxiliary calculations can still be performed in parallel, they need additional computation resources. Those may cause difficulties when considering real-time inference from live video streams. In our prototype, during the live capture sessions, treatment for improving temporal coherence was not taken into account. A fruitful avenue for future work would be to implement real-time variants of the motion-compensated bilateral filter as well as a mixture of colored Gaussians. Also, different methods could be developed that would enable the network to keep stylized video temporally coherent without the need to look into other video frames.



(i) *Limitation on greater appearance change in the target photo—a key assumption of our method is that the domain of source and target photos is similar, e.g., photos have same content and are taken under comparable illumination conditions. When this requirement is not satisfied, the resulting stylization may start to show artifacts as is visible in those examples of photos taken from the FFHQ dataset [Karras et al. 2019] where the illumination conditions are different to those used for the capture of source photo in Fig. 4.29.*



(ii) *Limitation on generalization—although our approach usually generalizes better than concurrent stylization techniques [Jamriška et al. 2019; Texler et al. 2020b], some specific features like eyes (a, c) that tend to generate strong activation in selected layers of VGG network may bias the VGG loss and make the network $\mathcal{F}$ reproduce their mostly unchanged copies (f, h) instead of adapting to their actual geometric distortion (b, d).*

Figure 4.31: *Illustration of common limitations of our method.*

The most important limitation of the method presented in Section 4.2 as compared to related approaches is notably longer time frame required to finish the optimization, which might be prohibitive for artist's exploration. To alleviate this drawback we envision a combination of fast patch-based training strategy presented in Section 4.1 with the computation of VGG loss which needs to be performed in a full-frame setting. In our proposed workflow an artist is responsible for keyframe selection. While some rules of thumb can be applied, such as selecting a frame that contains all features that are descriptive for most other frames, a mechanism which would select the keyframe automatically would improve ease of use. Most significantly, the method does not seem to generalize

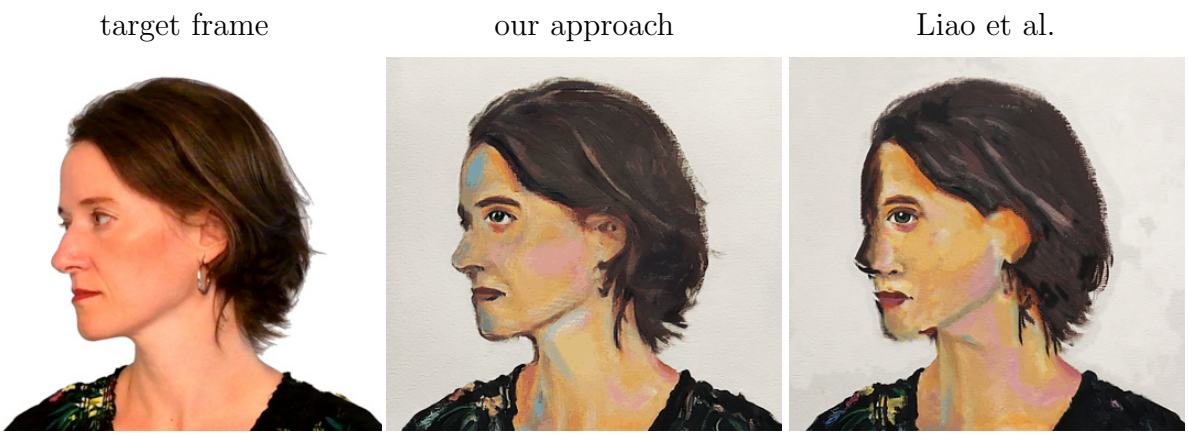


Figure 4.32: *The advantage of using style transfer with auxiliary pairing in visual attribute transfer scenario of Deep Image Analogy [Liao et al. 2017]. Although the style's texture and semantics (see source style in Fig. 4.17) are preserved well in both techniques, Deep Image Analogy (Liao et al.) has difficulties in adapting to certain structural changes. Target video frame © Zuzana Studená, used with permission.*

very well for completely generic use cases, for example in Fig. 4.31i, where input images are sampled from different underlying distributions. Thus, the set of potential applications is limited to groups of images of visually similar settings created under comparable conditions. A key advantage of this approach over existing methods in example-based video stylization such as the one described in Section 4.1 or state-of-the-art method of Jamriška et al. [2019] is greater robustness to structural discrepancies in the target frames. Even a relatively significant change such as head rotation is handled relatively well (see Fig. 4.17). In this case the network can successfully reproduce newly appearing content while still being able to preserve the notion of important planar structures of the original artistic media. On the other hand, some specific localized features such as eyes, may remain unchanged (see Fig. 4.31ii). A similar issue is known from visual attribute transfer approaches such as Deep Image Analogy [Liao et al. 2017]. As compared to them our method is able to adapt to structural changes better (see Fig. 4.32).

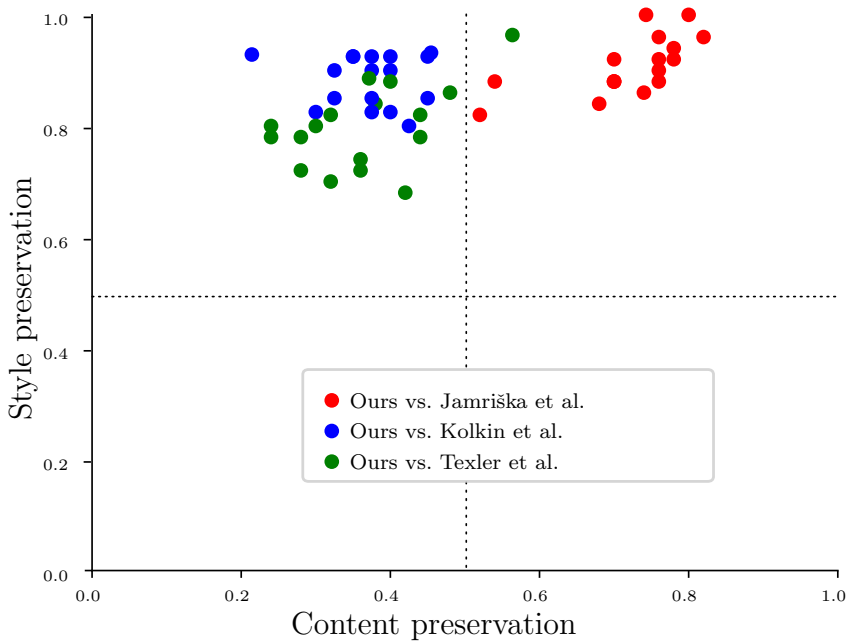


Figure 4.33: *Results of perceptual study—each point represents aggregated votes over a group of 10 participants. On the x axis we depict the percentage of answers in favor of content preservation of our method while on the y axis we show the style reproduction percentage. Comparisons were performed with the method of Jamriška et al. [2019] (red points), Kolkin et al. [2019] (blue points), and Texler et al. [2020a] (green points). From the graph it is visible that our method is observed to reproduce style notably better than previous works. It also outperforms the method of Jamriška et al. w.r.t. the content preservation, however, Kolkin et al. as well as Texler et al. are better in content preservation.*

Chapter 5

Stereoscopic Style Transfer

Example-based style transfer gained significant interest recently thanks to advances made in neural approaches [Gatys et al. 2016; Liao et al. 2017; Kolkin et al. 2019] as well as techniques based on guided texture synthesis [Jamriška et al. 2015; Fišer et al. 2016; Sýkora et al. 2019]. Great effort has also been devoted to example-based stylization of videos [Fišer et al. 2017; Ruder et al. 2018; Jamriška et al. 2019; Texler et al. 2020b; Futschik et al. 2021], where temporal consistency needs to be taken into account. Surprisingly, despite current trends in development of stereoscopic displays for virtual reality, cinemas, or metaverse, only a few researchers have tried to address the problem of example-based stylization in a binocular setting [Gong et al. 2018; Chen et al. 2018]. This lack of exploration can partly be explained by the fact that paintings are a priori assumed to be 2D projections of a 3D world where instead of binocular parallax, different depth cues are used. From this limited perspective, it may seem unnatural to transfer an inherently planar style to an image that will be depicted using a stereoscopic display. However, as recently demonstrated by Gong et al. [2018] and Chen et al. [2018], there is some interesting potential to better explore ways in which the human visual system can interpret artistic images under binocular vision. Both Gong et al. and Chen et al. approach this problem by improving neural style transfer [Gatys et al. 2016] to produce images that are consistent under binocular parallax. Their setting is, however, only an approximation to the more strict scenario we would like to consider. Since neural style transfer does not preserve the planarity of the style exemplar, structures such as strokes or canvas patterns can be distorted arbitrarily. This fact may lead to noticeable geometric distortion [Sýkora et al. 2019] where the stylized image looks as if the style exemplar is mapped onto the target 3D object which is then projected to 2D in each viewpoint.

To explore this research gap and address the new technical challenges, we propose a method we have published under the name *StyleBin: Stylizing Video by Example in Stereo*. The aim of our solution is to preserve the planarity of the original style exemplar while still being able to synthesize images that are consistent under binocular parallax. Given an input video and one or more stylized keyframes accompanied by information about depth in the scene, we synthesize a stylized output sequence for each eye. Our approach is a patch-based synthesis process where patch selection is informed by a family of guidance channels seeking to match aspects of the images, including color, position, and edges; the method is similar to that of Jamriška et al. [2019], though we must contend with the added difficulty of adapting the guidance channels to right and left eye views and then synthesizing both views consistently in time and space. Our

use of patches guarantees accurate reproduction of important planar structures in the style exemplar and the disparity-adapted guidance channels ensure their semantically meaningful transfer.

Our main contribution is a versatile framework for producing stylized stereoscopic sequences from an input monocular video with a semantically-meaningful style/depth transfer using a set of sparse style/depth keyframes. It extends the works of Jamriška et al. [2019] and Luo et al. [2015] to the stereo stylization setting. Its key technical contribution is the joint synthesis of stereo and temporal consistency. We demonstrate the effectiveness of the approach with several examples and a qualitative user study.

5.1 Our Approach

The input to our method is a target sequence T and a selection of one or more keyframes $K \subset T$ for which the user will provide (i) a stylized counterpart S_k and (ii) a disparity map D_k (see Fig. 5.1) that can be obtained manually or automatically. In our experiments we employ boosted monocular depth estimation [Miangoleh et al. 2021], and when applicable, also use Attention Mesh [Grishchenko et al. 2020] with Poisson image editing [Pérez et al. 2003] to improve disparity in facial regions. The precise choice of method is unimportant; any other depth estimation technique or additional depth sensor can be used to obtain D_k. The user may also decide to refine D_k manually to achieve the desired disparity.

The goal of our method is to produce two temporally coherent output sequences, a left sequence O^L and a right sequence O^R (see Fig. 5.1), in which the target sequence T will be stylized according to the style exemplar S_k such that when the frames from O^L and O^R are displayed to the corresponding eyes, the viewer will see a stereo effect driven by the disparity map D_k. This also means that O^L and O^R need to be consistent both in space and time to avoid ghosting and flickering artifacts.

We first describe the general approach to produce O^L and O^R from T using S_k and D_k. Further in this Section we demonstrate that the individual building blocks of our method can be applied in different scenarios: for example, we may have a target sequence T that is already fully stylized, or we may know D for each frame beforehand, perhaps because T was generated by 3D rendering or captured using a depth sensor.

To obtain O^L and O^R we use a guided patch-based synthesis framework similar to that described by Jamriška et al. [2019]. Like Jamriška et al., we want to transfer the style to the video in a semantically meaningful way. Unlike Jamriška et al., who create a single view, we need to jointly synthesize two views such that both stylized views are consistent in time and space according to the motion in the scene and the disparity given by D_k.

5.1.1 Disparity Propagation

As an initial step, we need to propagate the disparity stored in D_k from each keyframe $T_k \in K$ to the rest of the target sequence T (see Fig. 5.2). To do that, we employ the guided patch-based synthesis of Jamriška et al. [2019], providing the disparity map D_k as the style exemplar. In the case of multiple keyframes K, we propagate disparity to T from each keyframe separately and then blend the resulting frames using a weight proportional to the distance in time between the currently blended frame and the keyframe.

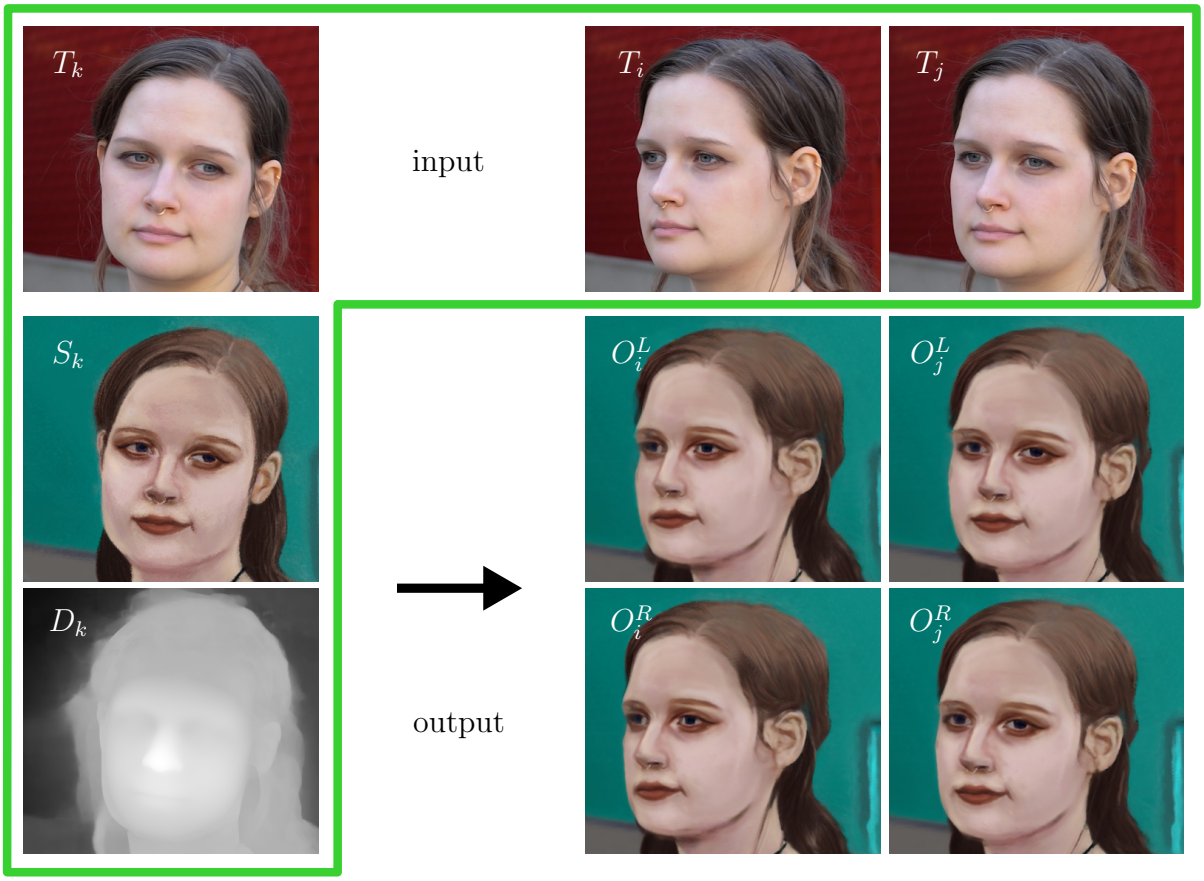


Figure 5.1: *An overview of the inputs and outputs of our method. The user provides a target sequence T in which one or more keyframes $T_k \in K$ are stylized S_k and contain information about disparity D_k. We propagate the disparity from D_k to the entire sequence T and transfer the style from S_k to T such that two stylized sequences O^L and O^R are produced, each of which can then be viewed by the corresponding eye to achieve a stereoscopic effect. Video frames T and style exemplar S_k © Jana Kyllerová.*

5.1.2 Disparity Shifting

As a byproduct of the previous disparity propagation step, a set of auxiliary channels $C = \{F, G_{\text{color}}, G_{\text{edge}}, G_{\text{pos}}\}$ is produced for each frame in T (see Fig. 5.3 and c.f. [Jamriška et al. 2019]). Here F is the optical flow computed between the consecutive frames in T using the method of Kroeger et al. [2016]. During the synthesis, F is used to help enforce temporal consistency. The channel G_{color} is a color guide that stores copies of individual frames of T. It helps to ensure that the style from S_k is transferred to locations where T_i has similar colors to those in T_k. G_{edge} denotes an edge guide that encourages salient features in T_i to be stylized consistently with those stored in T_k. G_{edge} is computed as follows: $G_{\text{edge}}(T_i) = T_i - \mathcal{N}_\sigma \circ T_i$, where $\mathcal{N}_\sigma$ is a Gaussian filter with standard deviation σ and $\circ$ denotes convolution. Finally, G_{pos} is a positional guide that encourages transfer of style pixels from keyframe T_k to the corresponding positions in the current frame T_i. G_{pos} is computed by accumulating a series of consecutive optical flows $F_{i-k} \in F$ between frames T_i and T_k.

The sequence of optical flows F plus the above-mentioned guiding channels G are sufficient to perform style transfer using the original method of Jamriška et al. In our

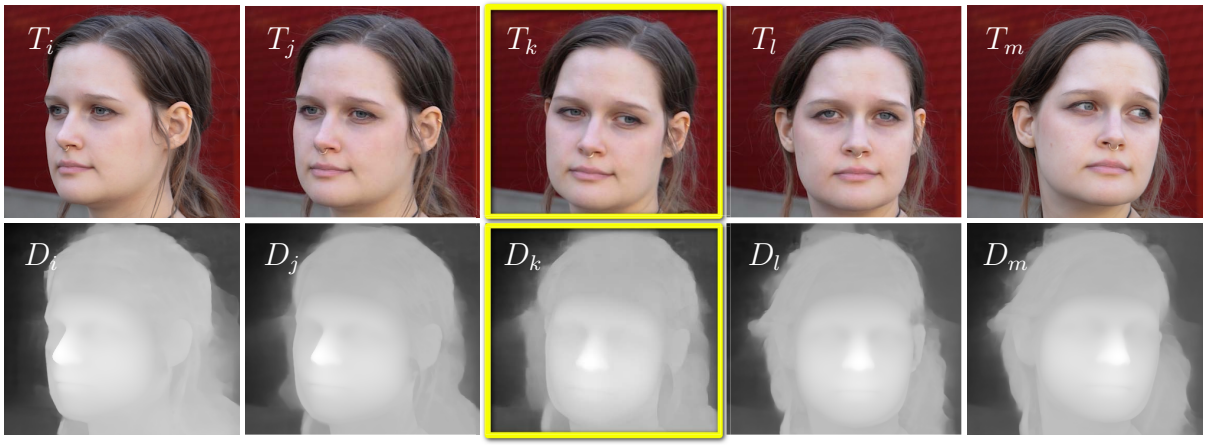


Figure 5.2: *An example of disparity map D_k propagation from a keyframe T_k to the rest of the sequence T. An output of this process is a sequence of disparity maps D aligned with every frame in T. Video frames T © Jana Kyllerová.*

setting, however, we need to produce a binocular sequence, for which we need a left C^L and right C^R view for each channel in C. Those new views can be obtained by shifting the content in C using the disparities stored in D; see Fig. 5.3.

Note that motion vectors stored in F are relative to the position of the underlying pixels, and therefore there is no need to modify their values during the shifting phase (we only need to shift their origins). Conversely, color-coded correspondences stored in G_{pos} are absolute; however, since they point to the original pixels in the monocular version of the keyframe $T_k \in K$, there is no need to modify them, as shifting their locations is sufficient.

5.1.3 Handling Disocclusion

After the shifting phase, a subset of the pixels in channels $\overleftarrow{C}$ and $\overrightarrow{C}$ may remain untouched due to disocclusion (see blue areas in Fig. 5.3). To fill those gaps, we first apply the disparity completion approach of Wang et al. [2008] to obtain consistent left D^L and right D^R disparity maps. Once D^L and D^R are available, we can employ disparity-guided patch-based synthesis, similar to that used by Luo et al. [2015]. Here the goal is to minimize the following:

$$\mathrm{E}_D(C^S, C^V) = \sum_{\hat{t} \in C^V} \min_{\hat{s} \in C^S} \mathrm{Q}(\hat{s}, \hat{t}), \tag{5.1}$$

where C^S is the source monocular channel and C^V is one of the shifted auxiliary channels (substituting for either C^L or C^R). For each disoccluded patch $\hat{t}$ in C^V, we search for a source patch $\hat{s}$ in C^S such that the following dissimilarity metric is minimized:

$$\begin{aligned} \mathrm{Q}(\hat{s}, \hat{t}) = \sum_{s \in \hat{s}, t \in \hat{t}} & w_{\text{dis}} |D^S(s) - D^V(t)|^2 + \\ & w_{\text{val}}(s,t) |C^S(s) - C^V(t)|^2 + w_{\text{uni}} \Omega(s). \end{aligned} \tag{5.2}$$

Here s and t are individual pixels from patches $\hat{s}$ and $\hat{t}$, and w_{dis} is the weight of a disparity term that compares the disparity of the source pixel s stored in D^S with the disparity of

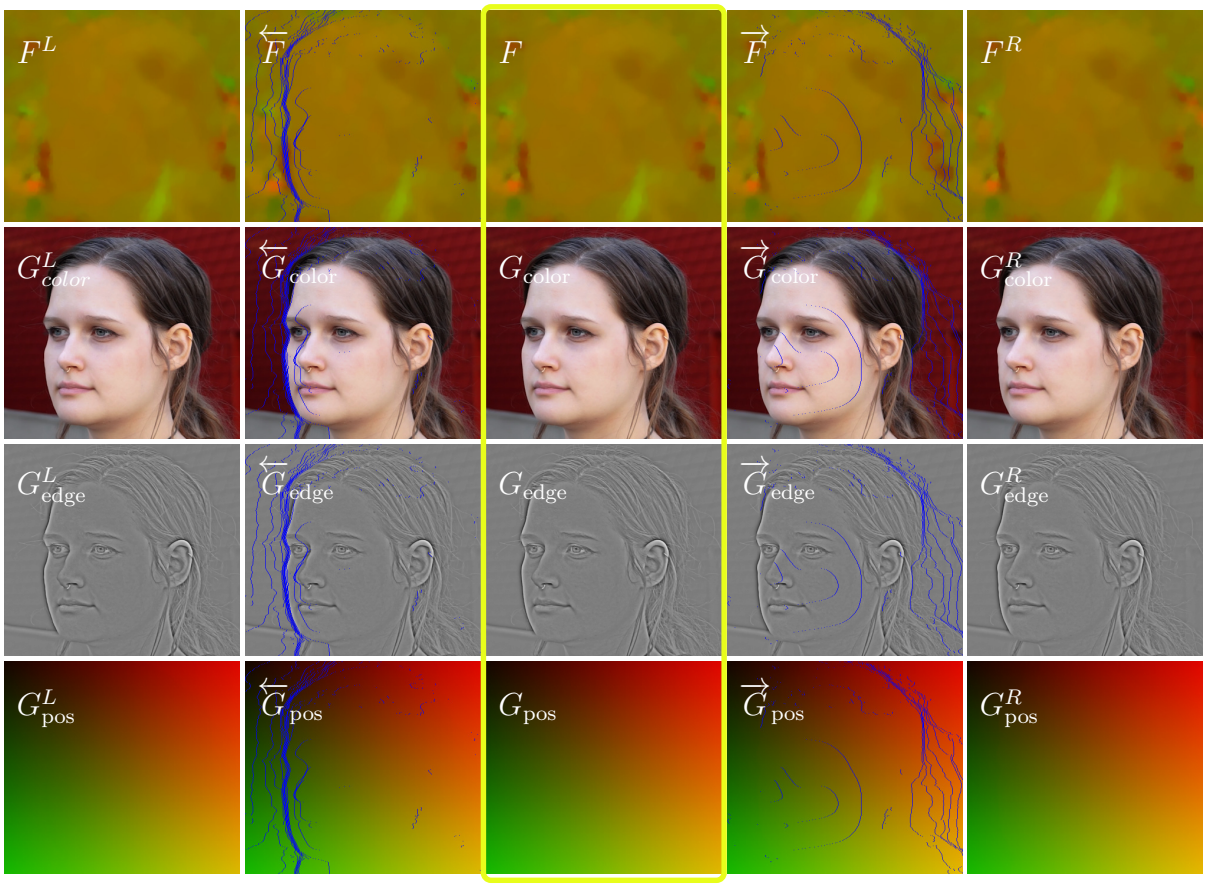


Figure 5.3: *An example of shifting and completion of auxiliary channels* $C = \{F, G_{color}, G_{edge}, G_{pos}\}$*: optical flow* F *as well as guiding channels* G *are first shifted to the left* $\overleftarrow{C}$ *and to the right* $\overrightarrow{C}$ *using disparities stored in* D*, and then disoccluded areas are filled using disparity-guided patch-based synthesis to obtain complete properly aligned auxiliary channels* C^L *and* C^R *for the left and right views. Video frame* G_{color} *© Jana Kyllerová.*

the target pixel t stored in D^V (substituting here either for D^R or D^L). Note that D^S was obtained in the disparity propagation phase (Section 5.1.1) while D^V originates from the preceding disparity completion step. The following disparity-dependent dissimilarity term helps to control the smoothness of the synthesized channel C^V (here C is one of F, G_{color}, G_{edge}, or G_{pos}, and V stands for L or R). By setting

$$w_{\text{val}}(s,t) = \exp(-|D^S(s) - D^V(t)|^2/\sigma^2) \tag{5.3}$$

as per the paper of Luo et al. [2015], we can encourage smooth transitions of synthesized channel values at the areas where the original disparity is continuous, while at discontinuities it enables abrupt changes. Finally, w_{uni} is the weight for the occurrence term Ω that prevents excessive repetition of source patches by counting frequency of their usage. More frequently used source patches have higher values of Ω and thus are less preferred during the search phase; see [Kaspar et al. 2015] for further details.

5.1.4 Final Synthesis

Once auxiliary channels for both views C^L and C^R are available in each target frame, we can begin to synthesize the stylized output sequences O^L and O^R. We start from a selected keyframe $T_k \in K$ and continue frame by frame forward/backward in time (or in both directions when k is neither the starting or final frame of T). For each input frame $T_i \in T$, we compute output frames that minimize the following energy:

$$\mathrm{E}_S(S_k, O_i^L, O_i^R) = \sum_{\hat{t}^L \in O_i^L} \min_{\hat{s}^L \in S_k} \mathrm{M}^L(\hat{s}^L, \hat{t}^L) + \sum_{\hat{t}^R \in O_i^R} \min_{\hat{s}^R \in S_k} \mathrm{M}^R(\hat{s}^R, \hat{t}^R), \tag{5.4}$$

which is a sum of two partial energies computed over the left O_i^L and right O_i^R stylized views. The aim is to find a source patch $\hat{s}^L \in S_k$ for each target patch $\hat{t}^L \in O_i^L$ in the left view and a source patch $\hat{s}^R \in S_k$ for each target patch $\hat{t}^R \in O_i^R$ in the right view that minimizes the following patch dissimilarity metric (see Fig. 5.4):

$$\begin{aligned} \mathrm{M}^V(\hat{s}, \hat{t}) = \sum_{s \in \hat{s}, t \in \hat{t}} & w_{\text{tex}} \mathrm{M}^V_{\text{tex}}(s,t) + w_{\text{color}} \mathrm{M}^V_{\text{color}}(s,t) + \\ & w_{\text{pos}} \mathrm{M}^V_{\text{pos}}(s,t) + w_{\text{edge}} \mathrm{M}^V_{\text{edge}}(s,t) + \\ & w_{\text{temp}} \mathrm{M}^V_{\text{temp}}(s,t) + w_{\text{uni}} \Omega(s). \end{aligned} \tag{5.5}$$

Here s denotes a pixel within the source patch $\hat{s}$ and t is a pixel within the target patch $\hat{t}$. The overall energy is a sum of dissimilarity terms, each with its own weight. The first texture dissimilarity term $\mathrm{M}^V_{\text{tex}}$ (V stands for left L or right R) with its weight w_{tex} measures the similarity between pixels in the style exemplar S_k and the corresponding pixels in the synthesized views (O_i^L and O_i^R). At the same time, it also evaluates the stereo consistency in the other view using the disparity maps D_i^L and D_i^R of the current frame i:

$$\mathrm{M}^V_{\text{tex}}(s,t) = |S_k(s) - O_i^V(t)|^2 + w_{\text{stereo}} |S_k(s) - O_i^{\neg V}(t \pm D_i^V(t))|^2, \tag{5.6}$$

Again V denotes L or R, $\neg V$ denotes the complement (R or L respectively), and $\pm$ refers to adding the disparity going left, and subtracting it going right. The stereo weight w_{stereo} balances the influence of texture and stereo consistency. The following terms in the energy formulation represent additional weighted guidance (w_{color}, w_{edge}, and w_{pos}) using channels G_{color}, G_{edge}, and G_{pos}:

$$\mathrm{M}^V_{\text{guide}}(s,t) = |G_k^S(s) - G_i^V(t)|^2 + w_{\text{stereo}} |G_k^S(s) - G_i^{\neg V}(t \pm D_i^V(t))|^2. \tag{5.7}$$

Here $\mathrm{M}^V_{\text{guide}}$ substitutes for $\mathrm{M}^V_{\text{color}}$, $\mathrm{M}^V_{\text{edge}}$ or $\mathrm{M}^V_{\text{pos}}$, while G^V stands for G^V_{color}, G^V_{edge} or G^V_{pos}. G_k^S are monocular guiding channels that correspond to a keyframe T_k. Each dissimilarity measure is accompanied by a corresponding dissimilarity for the disparity-adjusted pixel, promoting stereo consistency across the two views. In addition, temporal coherence is taken into account with a weight w_{temp} in both views:

$$\mathrm{M}^V_{\text{temp}}(s,t) = |S_k(s) - F_i^V[O_{i-1}^V](t)|^2. \tag{5.8}$$

Here $F_i^V[\ldots]$ denotes a warp driven by the shifted optical flow F_i^V of the previously synthesized output frame O_{i-1}^V. Again, V refers to either the left L or the right R view. Finally, Ω is the patch occurrence term with a weight w_{uni}, used to prevent overuse of particular exemplar patches as described by Kaspar et al. [2015].

5.1.5 Optimization

To minimize E_D and E_S, we use the EM-like optimization scheme proposed by Wexler et al. [2007] and later refined by Kaspat et al. [2015] to update the patch occurrence term. During the optimization of E_D, only patches whose central pixel lies within the disoccluded area are modified. All others remain unchanged and serve as boundary conditions to encourage the synthesis to produce seamless transitions between the original shifted pixels and those being synthesized to fill in dissocluded areas. In the case of E_S, the optimization runs over all target pixels since the style S needs to be consistently propagated to the entire frames. In the case of multiple keyframes, we transfer the style from each exemplar S separately and then perform linear blending to obtain the final merged sequence. Alternatively, a more advanced merging based on a screened Poisson equation can be used as described in the paper of Jamriška et al. [2019].

5.2 Results

We implemented our approach using C++. A table providing settings of all tunable parameters can be found in our supplementary material. To reduce computational overhead during the optimization of E^D and E^S, we employed PatchMatch [Barnes et al. 2009] to accelerate nearest-neighbour retrieval. On average, it takes 2.5 minutes on a ten-core CPU to synthesize one stereo pair for a single half-megapixel video frame.

To demonstrate the versatility of our framework, we prepared a selection of testing sequences with a variety of input data. These include one or more stylized keyframes in different styles with depth information obtained via boosted monocular depth estimation [Miangoleh et al. 2021] or rendered from a 3D model aligned with the target scene [Grishchenko et al. 2020]. We also demonstrate a use case when the target sequence is partly or entirely stylized and where keyframes are produced using a different style transfer method or contain only information about the depth in the scene. All results are presented in Figures 5.5 and 5.6 where the stereo effect can be seen using red-cyan anaglyph glasses. The full stylized sequences are also presented in the supplementary video, rendered both in red-cyan anaglyph and side-by-side mode. The latter is suitable for a cardboard or a VR headset, where the resulting stereo effect is most apparent.

The *Lili* sequence (see Fig. 5.5.1) contains subtle head motions. We prepared a single keyframe with an oil painting as a style exemplar and obtained depth information by combining boosted monocular depth estimation [Miangoleh et al. 2021] and a rendering of a 3D face model aligned with the head pose in the keyframe [Grishchenko et al. 2020]. We merged those two sources using Poisson image editing [Pérez et al. 2003]. The rest of the sequence was stylized using our approach, i.e., depth was propagated to the remaining frames, left and right auxiliary channels were produced, and finally the synthesis was executed to obtain the final stylized views.

In the *Jana* sequence (see Fig. 5.5.2) with its more dramatic head motion, a single keyframe was digitally painted by hand and then three other keyframes were generated using STALP [Futschik et al. 2021]—a neural style transfer method that handles more dramatic changes in the scene. For those additional keyframes, depth information was estimated using the same approach as for the *Lili* sequence, i.e., we combined estimated

and rendered depth maps. We used our approach to propagate depth and stylize the sequence from each keyframe and then we blended them to produce the final output.

The *Selfie* sequence (see Fig. 5.6.1) shows a human head with moving body and the *Lynx* sequence (see Fig. 5.6.2) depicts an animal in motion. For each of these sequences, two keyframes were digitally painted and depth was estimated using [Miangoleh et al. 2021]. Our approach was used to propagate depth and stylize the sequence using both keyframes. The final output was produced by blending.

Finally, sequences *Knights* and *Alchemist* (see Figures 5.5.3 and 5.6.3) were created in monocular view by an artist using a combination of hand-painted layers that undergo parallax motion and the video style transfer method of Jamriška et al. [2019]. Depth for those two sequences was obtained by generating eight keyframes using [Miangoleh et al. 2021]. Our method was then used to propagate the depth from the keyframes, construct auxiliary channels, and perform the synthesis to produce the resulting stereo pairs.

To evaluate our method, we conducted an informal user study. We presented each participant with the sequences produced using our approach, and interviewed them to gain some qualitative feedback about the outputs. The interviews took place in a VR environment, with both the interviewer and the interviewee being in the same virtual room with a screen. The interviewer controlled the sequences being shown and asked questions about them. There were in total eight participants, selected specifically to include a range of experience with VR, 3D movies, and hand-drawn art, from complete novices to professional artists. Participants were asked about their overall feeling from the sequence and whether they saw any artifacts; they were also given the opportunity to comment generally on the sequences.

Participants in general enjoyed watching our sequences. Without prompting, they immediately noticed clear stereo effect, which was more vivid in sequences with dynamic camera (*Knights*, *Alchemist*, and *Lynx*). They expressed no objections about understanding the depth layout in the scene, nor did they report any discomfort with respect to the stereo consistency. Participants were more interested in aspects that were not directly related to our method, such as expressing a preference for some particular artistic style or the selection of colors in the background plane. After several repeated viewings, two participants spotted subtle artifacts produced by our method, relating to the temporal coherency of newly uncovered regions in each view, comparing them to a shimmer caused by heat. Some participants commented on aspects of the sequences that were already present in the input, such as the lack of movement in the candle flames in the *Alchemist* sequence. Overall, the participants were enthusiastic about the potential for stereo stylization.

To further highlight the benefits of our approach, we performed quantitative and qualitative evaluation with two baseline stereo stylization techniques: (1) *stylize-and-warp*—a method where we use known disparity to warp the input stylized monocular video to left and right view; and (2) *warp-and-stylize*—an approach in which an input monocular video is warped to left and right views and then each view is stylized separately. Results of these two evaluations are presented in the supplementary material. They clearly demonstrate that our approach reproduces the style more faithfully and achieves better stereo consistency.

5.3 Conclusion and Future Work

In this Chapter, we have presented a method allowing the creation of stereo-consistent example-based stylizations from monocular video sequences, utilizing the state-of-the-art methods in depth estimation to fill in all the required information and relax the requirements on input data format, which would otherwise be too restrictive to reach its full potential.

The method presented in this Chapter shares some limitations with the selected baseline approach [2019]. Both techniques are sensitive to significant changes in the input video (e.g., viewpoint, pose or illumination) and can find it difficult to propagate high-frequency details from the style exemplar through the full video sequence. This drawback can be mitigated by providing additional corrective keyframes, either manually or using a more advanced style transfer technique, such as the algorithms described in Chapter 4, as we demonstrated in the *Jana* example. There is also some dependence on the quality of input depth. While monocular depth estimation is outside the scope of our contribution, inaccurate depth maps may impose problems on us, sometimes manifesting as inconsistent halo effects or a lack of depth perception. We demonstrated how to partially mitigate this by fitting a 3D mesh [Grishchenko et al. 2020] into the input sequence to obtain higher-quality depth values in facial regions, but a more general solution remains an open problem. This method may encounter difficulties in scenarios where more accurate reconstruction of disocclusions is necessary. Our expectation is that holes are relatively small and thus there is no need to handle continuation of semantically meaningful structures in the scene. For larger holes or a complex configuration of occluders (e.g., a dense forest with leaves and branches blowing in the wind), more elaborate methods would be required.

Overall, while we have certainly made definite progress towards out ultimate goal of being able to create real-time, interactive and stereoscopic experiences using style-transfer methods, there is still much that could be done to further this goal. The work we have done so far serves more as a platform for out future research, we have one-by-one addressed the individual requirements of our goal in a divide and conquer strategy, with the plan to continue or work by combining the individual solutions into a complete, robust one, solving larger and larger portions of the task until the task is completed entirely. The work attempting to solve multiple subtasks at once, however, remains open for now, as we have thought about some of these needed subtasks and believed the solution would be a relatively simple combination of our previous work without much room for additional orthogonal contributions, making it difficult to publish on conferences or publications on par with our previous research. Our work in this Chapter has made it possible to generate stereo-consistent content, but optimization-based techniques are known to be computationally expensive and not feasible for real-time applications. For applications such as 3D cinemas, where the point of view and perspective are fixed, our algorithm can be used to generate the stylized sequences offline, where performance does not matter as much. For use with VR headsets, however, where the display content must react in real-time to movement of the user's head, our algorithm is not feasible, and a real-time alternative must be found instead. Taking inspiration from our neural real-time work in Chapters 3 and 4, we believe that training a network for stereoscopic content following our previously done research is possible, and would most likely lead to a feasible solution that would be both interactive and would perform well in a real-time setting. In this

thesis, however, we have instead opted to commit our time to the optimization-based technique instead for two reasons: first, simply training a network following our previous frameworks would probably present little opportunity for contribution, and would most likely be reduced to a simple extension, and second, we believe that commiting time to researching a traditional technique is preferable to commiting time to a neural technique approximating the desired results, as traditional techniques lead to better understanding of the underlying problem, allowing us to identify and address issues and phenomena found during the research process, whereas training a network leads to a solution that works without proper understanding why.

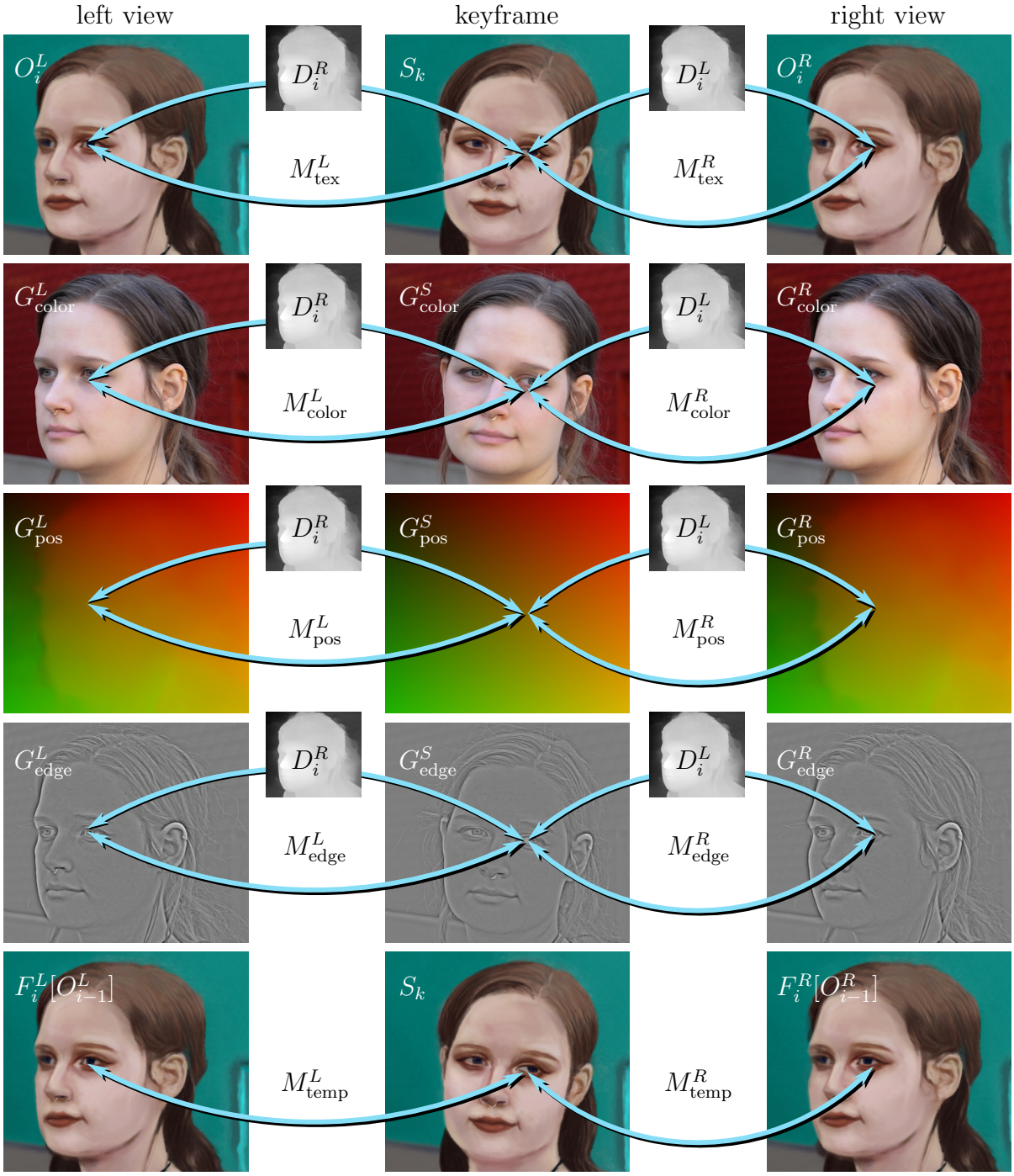


Figure 5.4: *An overview of terms consisting of patch dissimilarity metrics M^L and M^R and their dependence on auxiliary channels. See the text for the detailed explanation. Video frame G^S_{color} and style exemplar S_k © Jana Kyllerová.*

Figure 5.5: *A collection of three different sequences stylized using our approach—*Lili *Fig. 5.5.1,* Jana *Fig. 5.5.2, and* Knights *Fig. 5.5.3. From* Lili*'s and* Jana*'s input sequences (1d & 2d) a single keyframe was selected (1a & 2a) for which a stylized counterpart was prepared by an artist (1b & 2b) and also a depth map specified (1c & 2c). Our method then produced the final binocular sequences (1e & 2e) of which anaglyph examples are shown in (1f & 2f). In the case of* Knights*. the input sequence (3d) was already stylized by an artist, and the aim here is to add a stereoscopic effect (3e). To do that, our method propagates depth information (3b) from a set of keyframes (3a) to the entire sequence and synthesizes the stylized stereo view (3f). See also our supplementary video for a side-by-side version of this result. Video frames (1a) & (1d) © Michal Dvořák, video frames (2a) & (2d) and style exemplar (2b) © Jana Kyllerová, stylized video frames (3a) & (3d) © Jakub Javora.*

Figure 5.6: *Our approach applied to three different sequences*—Selfie *Fig. 5.6.1,* Lynx *Fig. 5.6.2, and* Alchemist *Fig. 5.6.3. From* Selfie*'s and* Lynx*'s input sequences (1g & 2g) the user will pick two keyframes (1a, 1d, 2a, 2d), prepare their stylized variants (1b, 1e, 2b, 2e), and provide an estimate of depth in the scene (1c, 1f, 2c, 2f). Our method then transfers the style from those keyframes onto the rest of the video (1g & 2g) producing a consistent stereo sequence (1h & 2h) of which one frame is displayed here as a red-cyan anaglyph (1i & 2i). In the case of* Alchemist*, the input video (3c) was already stylized by an artist. A set of depth maps (3b) is provided for a selection of keyframes (3a). Our algorithm then propagates the information about depth to the entire stylized video and synthesizes a stereo sequence (3d). An anaglyph close-up of one frame from our stereoscopic output is shown in (3e). See also our supplementary video for a side-by-side version of this result. Video frames (1a), (1d) & (1g) and style exemplars (1b) & (1e) © Jana Kyllerová, style exemplars (2b) & (2e) and stylized video frames (3a), (3c) & (3d) © Jakub Javora.*

Chapter 6

Conclusion

In this thesis, we have presented five contributions to the field of example-based style-transfer, specifically in the domain of video stylization. Each of these helps push the state-of-the-art further towards feasible real-time applications in the field od virtual and augmented reality, and further beyond.

We have shown that neural approaches are able to provide real-time and high quality interactive experiences, making them usable for artists in their creative process. Our proposed methods for overfitting prevention, either by limiting the receptive field or by adding style preservation term to the training, allows to train such networks without requiring large sets of paired training inputs or large domain-specific datasets. We also show that a learning-based method can be used to solve the sequential nature of many video stylization approaches, allowing for easy and quick random access to any stylized frame within the sequence, as well as parallel processing. In our non-neural stylization methods, we have presented a framework for inexpensive approximation of existing optimization based techniques that achieves similar results on facial videos, enabling their use on mobile devices in real-time or in places where minimal impact on overall performance is required. We have also proposed an optimization based technique for stereo-consistent video stylizations, allowing the generation of stylized video content targeting stereoscopic devices without the artifacts and inconsistencies that previously prevented such content form being created. In our studies we have also compared the outputs of our newly proposed algorithms to other state-of-the-art style transfer methods and reported definite improvements in image quality. In this chapter we summarize the contributions and conclusions made throughout our research.

6.1 Summary

In Chapter 3 we presented a neural approach for real-time stylization of facial videos. By utilizing the state-of-the-art work of Fišer et al. [2017], we were able to create a suffient dataset of image pairs of unstylized original subjects and their stylized counterparts, to be then used with a state-of-the-art neural framework to learn a particular artistic style. Since the network is inexpensive to evaluate, we presented an interactive application to stylize subjects captured on a camera feed in real-time while achieving comparable output quality, running on consumer-grade GPUs. We have presented this method as a technical paper at the Expressive 2019 conference [Futschik et al. 2019].

In Chapter 3, we also presented our work allowing real-time example-based stylization of facial videos even on low-end devices. By taking inspiration from [Sýkora et al. 2019] and limiting our scope only to the domain of facial videos, we were able to utilize detected facial landmarks to precompute majority of information required during the stylization process, reducing the expensive optimization step to a simple lookup, which is also easily parallelizable in implementations, e.g. in shaders. We developed a real-time application to present the benefits of our method, which ran without issues even on phones. We compared the outputs of our method to those of already existing state-of-the-art approaches, and also presented an extension to our approach, which allows higher quality stylization of hair and facial hair, not normally possible using style-transfer methods, by deforming masked segments from the style exemplar using the extracted facial landmarks. We have presented this paper at the I3D 2021 conference and published in Proceedings of the ACM in Computer Graphics and Interactive Techniques journal [Texler et al. 2021].

In Chapter 4 we introduced a novel learning-based method for keyframe-based video sequence stylization. We addressed the common issue of neural networks often requiring massive amounts of training data, especially for style transfer, by purposely limiting the training receptive field to randomly selected batches of pairs of corresponding patches. This approach allows an image-to-image network to be trained even on a single keyframe in very short amount of time even on a consumer-grade GPU. Combined with the real-time inference, already demonstrated in our previous work in Chapter 4, we developed and presented a framework allowing real-time, interactive stylization of video sequences, not previously achievable, which we have successfully applied in several scenarios; having an artist stylize a video sequence by painting one of the frames and seeing the intermediate results in real-time, or stylizing a real-time video of the artist coming from a camera feed by painting a previously captured photo. The nature of the network's inference also allows frames of a video sequence to be stylized in parallel by preserving the temporal coherence implicitly, offering a fast alternative to many state-of-the-art patch-based stylization algorithms, which are sequential in nature. We have presented this paper at SIGGRAPH 2020 conference and published in ACM Transactions on Graphics journal [Texler et al. 2020b]. We have also won the Best in Show award in the *Real-time Live!* showcase on SIGGRAPH 2020.

Also in Chapter 4 we presented another of our contributions to the domain of neural keyframe-based style-transfer methods targeting video sequences. In this method, we aim to alleviate the common artifacts and qualitative issues from which neural methods are still suffering, mainly the temporal coherence and loss of small precise details occuring during changes in the video to be stylized. For that, we present an alternative loss function to our previous method, which allows us to train the network properly while still only requiring a single keyframe, while giving more emphasis on style detail retention and the quality of style reproduction. To verify our claims of better quality, we performed a user study comparing our method with a set of other state-of-the-art methods in the same domain, showing that our method perform substantially better in terms of style preservation than the competing methods. We have presented this method at Eurographics 2021 conference and published it in Computer Graphics Forum journal [Futschik et al. 2021].

Finally, in Chapter 5, we have presented a patch-based style-transfer method able to produce consistent stereoscopic stylization from monocular video sequences. By utilizing a combination of state-of-the-art methods for depth estimation and hole-filling, we are

able to generate a set of stereoscopic guiding channels, which can then be used with our EM-like optimization scheme. The core of our contribution is an augmented error function used during the optimization process, which takes into account the inconsistencies between corresponding patches in both views, eliminating unpleasant artifacts which prevented previous patch-based methods to be used in stereoscopic applications. We demonstrate the potential of this method on a variety of settings; stylizing video sequences with a single keyframe, stylizing video sequences with multiple keyframes, or stylizing an already stylized sequence without having the original unstylized counterpart. The method was also tested in several experiments and users studies, confirming the added benefit of our contribution. We have presented this method as a technical paper at the SIGGRAPH Asia 2022 conference [Kučera et al. 2022].

References

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Proceedings of European Conference on Computer Vision*, pages 707–723, 2022.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.

Pierre Bénard, Ares Lagae, Peter Vangorp, Sylvain Lefebvre, George Drettakis, and Joëlle Thollot. A dynamic noise primitive for coherent stylization. *Computer Graphics Forum*, 29(4):1497–1506, 2010.

Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. Stylizing animation by example. *ACM Transactions on Graphics*, 32(4):119, 2013.

Eric P. Bennett and Leonard McMillan. Video enhancement using per-pixel virtual exposures. *ACM Transactions on Graphics*, 24(3):845–852, 2005.

Ashish Bora, Eric Price, and Alexandros G. Dimakis. AmbientGAN: Generative models from lossy measurements. In *Proceedings of International Conference on Learning Representations*, 2018.

Adrien Bousseau, Matthew Kaplan, Joëlle Thollot, and François X. Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 141–149, 2006.

Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics*, 26 (3):104, 2007.

Simon Breslav, Karol Szerszen, Lee Markosian, Pascal Barla, and Joëlle Thollot. Dynamic 2D patterns for shading 3D scenes. *ACM Transactions on Graphics*, 26(3):20, 2007.

Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

Dennis R. Bukenberger, Katharina Schwarz, and Hendrik P. A. Lensch. Stereo-consistent contours in object space. *Computer Graphics Forum*, 37(1):301–312, 2018.

Kaidi Cao, Jing Liao, and Lu Yuan. Carigans: Unpaired photo-to-caricature translation. *ACM Transactions on Graphics*, 37(6):244:1–244:14, 2018.

Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of IEEE International Conference on Computer Vision*, pages 23206–23217, 2023.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of IEEE International Conference on Computer Vision*, pages 5933–5942, 2019.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1114–1123, 2017.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stereoscopic neural style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 6654–6663, 2018.

Hong Chen, Nanning Zheng, Lin Liang, Yan Li, Ying-Qing Xu, and Heung-Yeung Shum. PicToon: A personalized image-based cartoon system. In *Proceedings of ACM International Conference on Multimedia*, pages 171–178, 2002.

Hong Chen, Lin Liang, Ying-Qing Xu, Heung-Yeung Shum, and Nan-Ning Zheng. Example-based automatic portraiture. *Chinese Journal of Computers (Chinese Edition)*, 26(2):147–152, 2003.

Hong Chen, Ziqiang Liu, Chuck Rose, Yingqing Xu, Heung-Yeung Shum, and David Salesin. Example-based composite sketching of human portraits. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 95–102, 2004.

Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu. Large displacement optical flow from nearest neighbor fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2450, 2013.

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018.

Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. In *ACM SIGGRAPH Conference Proceedings*, pages 421–430, 1997.

Niraj Ramesh Dayama, Simo Santala, Lukas Brückner, Kashyap Todi, Jingzhou Du, and Antti Oulasvirta. Interactive layout transfer. In *26th International Conference on Intelligent User Interfaces*, pages 70–80, 2021.

Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. *Computer Graphics Forum*, 39(4):91–105, 2020.

Steve DiPaola. Painterly rendered portraits from photographs using a knowledge-based approach. In *Proceedings of SPIE Human Vision and Electronic Imaging*, volume 6492, pages 33–43, 2007.

Marek Dvorožňák, Wilmot Li, Vladimir G. Kim, and Daniel Sýkora. ToonSynth: Example-based synthesis of hand-colored cartoon animations. *ACM Transactions on Graphics*, 37(4):167, 2018.

Dónal Egan, Martin Alain, and Aljosa Smolic. Light field style transfer with local angular consistency. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2300–2304, 2021.

Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente, and Daniel Sýkora. Color Me Noisy: Example-based rendering of hand-colored animations with temporal noise control. *Computer Graphics Forum*, 33(4):1–10, 2014.

Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. StyLit: Illumination-guided example-based stylization of 3D renderings. *ACM Transactions on Graphics*, 35(4):92, 2016.

Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics*, 36(4):155, 2017.

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

Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. Video style transfer by consistent adaptive patch sampling. *The Visual Computer*, 35(3):429–443, 2019.

David Futschik, Menglei Chai, Chen Cao, Chongyang Ma, Aleksei Stoliar, Sergey Korolev, Sergey Tulyakov, Michal Kučera, and Daniel Sýkora. Real-time patch-based stylization of portraits using generative adversarial network. In *Proceedings of the ACM/EG Expressive Symposium*, pages 33–42, 2019.

David Futschik, Michal Kučera, Michal Lukáč, Zhaowen Wang, Eli Shechtman, and Daniel Sýkora. STALP: Style transfer with auxiliary limited pairing. *Computer Graphics Forum*, 40(2):563–573, 2021.

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

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *Proceedings of International Conference on Learning Representations*, 2024.

Xinyu Gong, Haozhi Huang, Lin Ma, Fumin Shen, Wei Liu, and Tong Zhang. Neural stereoscopic image style transfer. In *Proceedings of European Conference on Computer Vision*, pages 56–71, 2018.

Bruce Gooch, Erik Reinhard, and Amy Gooch. Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics*, 23(1):27–44, 2004.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of Conference on Neural Information Processing Systems*, pages 2672–2680, 2014.

Ific Goudé, Rémi Cozot, Olivier Le Meur, and Kadi Bouatouch. Example-based colour transfer for 3D point clouds. *Computer Graphics Forum*, 40(6):428–446, 2021.

Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. Attention Mesh: High-fidelity face mesh prediction in real-time. In *Proceedings of the CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, 2020.

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

Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4087–4096, 2017.

William Van Haevre, Tom Van Laerhoven, Fabian Di Fiore, and Frank Van Reeth. From Dust Till Drawn: A real-time bidirectional pastel simulation. *The Visual Computer*, 23(9–11):925–934, 2007.

Fangzhou Han, Shuquan Ye, Mingming He, Menglei Chai, and Jing Liao. Exemplar-based 3D portrait stylization. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1371–1383, 2021.

Filip Hauptfleisch, Ondřej Texler, Aneta Texler, Jaroslav Křivánek, and Daniel Sýkora. StyleProp: Real-time example-based stylization of 3D models. *Computer Graphics Forum*, 39(7):575–586, 2020.

James Hays and Irfan A. Essa. Image and video based painterly animation. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 113–120, 2004.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *ACM SIGGRAPH Conference Proceedings*, pages 327–340, 2001.

Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. *Computer Graphics Forum*, 41(4):117–128, 2022.

Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *Proceedings of IEEE International Conference on Computer Vision*, pages 13869–13878, 2021.

Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *Proceedings of IEEE International Conference on Computer Vision*, pages 1510–1519, 2017.

Loc Huynh, Bipin Kishore, and Paul Debevec. A new dimension in testimony: Relighting video with reflectance field exemplars, 2021. arXiv:2104.02773.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2017.

Lesley Istead and Craig S. Kaplan. Stylized stereoscopic 3D line drawings from 3d images. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, page 20, 2018.

Lesley Istead, Andreea Pocol, Craig S. Kaplan, Isaac Watt, Nick Lemoing, and Alicia Yang. Generating rough stereoscopic 3D line drawings from 3D images. In *Proceedings of Graphics Interface*, pages 178–185, 2021.

Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. LazyFluids: Appearance transfer for fluid animations. *ACM Transactions on Graphics*, 34(4):92, 2015.

Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Stylizing video by example. *ACM Transactions on Graphics*, 38(4):107, 2019.

Ruixiang Jiang, Can Wang, Jingbo Zhang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. AvatarCraft: Transforming text into neural human avatars with parameterized shape and pose control. In *Proceedings of IEEE International Conference on Computer Vision*, pages 14371–14382, 2023.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision*, pages 694–711, 2016.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8107–8116, 2020.

Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. Self tuning texture optimization. *Computer Graphics Forum*, 34(2):349–360, 2015.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics*, 40(6):210, 2021.

Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.

Yongjin Kim, Yunjin Lee, Henry Kang, and Seungyong Lee. Stereoscopic 3D line drawing. *ACM Transactions on Graphics*, 32(4):57, 2013.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. arXiv:1412.6980.

Nicholas I. Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2019.

Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum*, 40(4): 29–43, 2021.

Dmytro Kotovenko, Artsiom Sanakoyeu, Sabine Lang, and Björn Ommer. Content and style disentanglement for artistic style transfer. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4421–4430, 2019a.

Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Björn Ommer. A content transformation block for image style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 10032–10041, 2019b.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil D. B. Bruce, Richard P. Wildes, and Konstantinos G. Derpanis. Quantifying and learning static vs. dynamic information in deep spatiotemporal networks, 2022a. arXiv:2211.01783.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil DB Bruce, Richard P Wildes, and Konstantinos G Derpanis. A deeper dive into what deep spatiotemporal networks encode: Quantifying static vs. dynamic information. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 13999–14009, 2022b.

Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *Proceedings of European Conference on Computer Vision*, pages 471–488, 2016.

Michal Kučera, David Mould, and Daniel Sýkora. StyleBin: Stylizing video by example in stereo. *SIGGRAPH Asia 2022 Conference Papers*, art. no. 15, 2022.

Vivek Kwatra, Irfan A. Essa, Aaron F. Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics*, 24(3):795–802, 2005.

Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2013.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of European Conference on Computer Vision*, pages 179–195, 2018.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *Proceedings of the International Conference on Machine Learning*, pages 2971–2980, 2018.

Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of European Conference on Computer Vision*, pages 702–716, 2016a.

Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016b.

Hongliang Li, Guanghui Liu, and King Ngi Ngan. Guided face cartoon synthesis. *IEEE Transactions on Multimedia*, 13(6):1230–1239, 2011.

Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In *Proceedings of Conference on Neural Information Processing Systems*, pages 317–327, 2019.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Proceedings of Conference on Neural Information Processing Systems*, pages 385–395, 2017.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics*, 36(4):120, 2017.

Peter Litwinowicz. Processing images and video for an impressionist effect. In *ACM SIGGRAPH Conference Proceedings*, pages 407–414, 1997.

Feng-Lin Liu, Shu-Yu Chen, Yukun Lai, Chunpeng Li, Yue-Ren Jiang, Hongbo Fu, and Lin Gao. Deepfacevideoediting: Sketch-based deep editing of face videos. *ACM Transactions on Graphics*, 41(4):167, 2022.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of IEEE International Conference on Computer Vision*, pages 10551–10560, 2019.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Proceedings of Conference on Neural Information Processing Systems*, pages 9628–9639, 2018.

Cewu Lu, Li Xu, and Jiaya Jia. Combining sketch and tone for pencil drawing production. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 65–73, 2012.

Ming Lu, Hao Zhao, Anbang Yao, Feng Xu, Yurong Chen, and Xiang Lin. Decoder network over lightweight reconstructed feature for fast semantic style transfer. *Proceedings of IEEE International Conference on Computer Vision*, pages 2488–2496, 2017.

Wanglong Lu, Xianta Jiang, Xiaogang Jin, Yong-Liang Yang, Minglun Gong, Tao Wang, Kaijie Shi, and Hanli Zhao. Grig: Few-shot generative residual image inpainting, 2023.

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

Lei Ma, Yuhui Zheng, Zhao Zhang, Yazhou Yao, Xijian Fan, and Qiaolin Ye. Motion stimulation for compositional action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(5):2061–2074, 2023.

Akinobu Maejima, Hiroyuki Kubo, Seitaro Shinagawa, Takuya Funatomi, Tatsuo Yotsukura, Satoshi Nakamura, and Yasuhiro Mukaigawa. Anime character colorization using few-shot learning. *SIGGRAPH Asia 2021 Technical Communications*, art. no. 8, 2021.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2813–2821, 2017.

Meng Meng, Mingtian Zhao, and Song Chun Zhu. Artistic paper-cut of human portraits. In *Proceedings of ACM Multimedia*, pages 931–934, 2010.

S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 9685–9694, 2021.

Santiago E Montesdeoca, Hock Soon Seah, Amir Semmo, Pierre Bénard, Romain Vergne, Joëlle Thollot, and Davide Benvenuti. MNPR: A framework for real-time expressive non-photorealistic rendering of 3D computer graphics. *Proceedings of the ACM/EG Expressive Symposium*, art. no. 11, 2018.

Ravi Teja Mullapudi, Steven Chen, Keyi Zhang, Deva Ramanan, and Kayvon Fatahalian. Online model distillation for efficient video inference. In *Proceedings of IEEE International Conference on Computer Vision*, pages 3572–3581, 2019.

Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. SNeRF: Stylized neural implicit representations for 3D scenes. *ACM Transactions on Graphics*, 41(4):142, 2022.

Lesley Northam, Paul Asente, and Craig S. Kaplan. Consistent stylization and painterly rendering of stereoscopic 3D images. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 47–56, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of Conference on Neural Information Processing Systems*, pages 8024–8035, 2019.

Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, 2003.

Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *ACM SIGGRAPH Conference Proceedings*, pages 581–586, 2001.

Linzi Qu, Jiaxiang Shang, Xiaoguang Han, and Hongbo Fu. ReenactArtFace: Artistic face image reenactment. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

Shyam Nandan Rai, Rohit Saluja, Chetan Arora, Vineeth N Balasubramanian, Anbumani Subramanian, and CV Jawahar. Fluid: Few-shot self-supervised image deraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3077–3086, 2022.

Carlos Rodriguez-Pardo and Elena Garces. Neural photometry-guided visual attribute transfer. *IEEE Transactions on Visualization and Computer Graphics*, 29(3):1818–1830, 2023.

Carlos Rodriguez-Pardo, Henar Domínguez-Elvira, David Pascual-Hernández, and Elena Garces. Umat: Uncertainty-aware single image high resolution material capture. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5764–5774, 2023a.

Carlos Rodriguez-Pardo, Konstantinos Kazatzis, Jorge Lopez-Moreno, and Elena Garces. NeuBTF: Neural fields for BTF encoding and transfer. *Computers & Graphics*, 114: 239–246, 2023b.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

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

Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In *ACM SIGGRAPH Conference Proceedings*, pages 401–406, 1997.

Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. A style-aware content loss for real-time hd style transfer. In *Proceedings of European Conference on Computer Vision*, pages 715–731, 2018.

Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. *ACM Transactions on Graphics*, 25(3):533–540, 2006.

Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. Overcoat: an implicit canvas for 3D painting. *ACM Transactions on Graphics*, 30(4):28, 2011.

Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics*, 35(4): 129, 2016.

Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4570–4580, 2019.

Eli Shechtman, Alex Rav-Acha, Michal Irani, and Steven M. Seitz. Regenerative morphing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 615–622, 2010.

Sumit Shekhar, Max Reimann, Moritz Hilscher, Amir Semmo, Jürgen Döllner, and Matthias Trapp. Interactive control over temporal consistency while stylizing video streams. *Computer Graphics Forum*, 42(4):e14891, 2023.

Xiaoyong Shen, Aaron Hertzmann, Jiaya Jia, Sylvain Paris, Brian L. Price, Eli Shechtman, and Ian Sachs. Automatic portrait segmentation for image stylization. *Computer Graphics Forum*, 35(2):93–102, 2016.

Yi-Chang Shih, Sylvain Paris, Connelly Barnes, William T. Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4):148, 2014.

Assaf Shocher, Nadav Cohen, and Michal Irani. "Zero-Shot" super-resolution using deep internal learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. InGAN: Capturing and remapping the "DNA" of a natural image. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4492–4501, 2019.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Proceedings of Conference on Neural Information Processing Systems*, pages 7135–7145, 2019a.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019b.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. arXiv:1409.1556.

Akhil Singh, Vaibhav Jaiswal, Gaurav Joshi, Adith Sanjeeve, Shilpa Gite, and Ketan Kotecha. Neural style transfer: A critical review. *IEEE Access*, 9:131583–131613, 2021.

Peter-Pike J. Sloan, William Martin, Amy Gooch, and Bruce Gooch. The Lit Sphere: A model for capturing NPR shading from art. In *Proceedings of Graphics Interface*, pages 143–150, 2001.

Noah Snavely, C. Lawrence Zitnick, Sing Bing Kang, and Michael F. Cohen. Stylizing 2.5-D video. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 63–69, 2006.

Efstathios Stavrakis and Margrit Gelautz. Image-based stereoscopic painterly rendering. In *Proceedings of the Eurographics Conference on Rendering Techniques*, pages 53–60, 2004.

Daniel Sýkora, John Dingliana, and Steven Collins. As-rigid-as-possible image registration for hand-drawn cartoon animations. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 25–33, 2009.

Daniel Sýkora, Ondřej Jamriška, Ondřej Texler, Jakub Fišer, Michal Lukáč, Jingwan Lu, and Eli Shechtman. StyleBlit: Fast example-based stylization with local guidance. *Computer Graphics Forum*, 38(2):83–91, 2019.

Yan Tang. Style transfer of chinese art works based on dual channel deep learning model. *Computational Intelligence & Neuroscience*, Art. ID. 4376006, 2022.

Krzysztof Templin, Piotr Didyk, Karol Myszkowski, and Hans-Peter Seidel. Perceptually-motivated stereoscopic film grain. *Computer Graphics Forum*, 33(7):349–358, 2014.

Aneta Texler, Ondřej Texler, Michal Kučera, Menglei Chai, and Daniel Sýkora. FaceBlit: Instant real-time example-based style transfer to facial videos. *Proceedings of the ACM in Computer Graphics and Interactive Techniques*, 4(1):14, 2021.

Ondřej Texler, David Futschik, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. Arbitrary style transfer using neurally-guided patch-based synthesis. *Computers & Graphics*, 87:62–71, 2020a.

Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics*, 39(4):73, 2020b.

Hideki Todo, Kunihiko Kobayashi, Jin Katsuragi, Haruna Shimotahira, Shizuo Kaji, and Yonghao Yue. Stroke transfer: Example-based synthesis of animatable stroke styles. *ACM SIGGRAPH 2022 Conference Proceedings*, art. no. 54, 2022.

Patrick Tresset and Frédéric F. Leymarie. Generative portrait sketching. In *Proceedings of International Conference on Virtual Systems and Multimedia*, pages 739–748, 2005.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of International Conference on Machine Learning*, pages 1349–1357, 2016a.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016b. arXiv:1607.08022.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016c. arXiv:1607.08022.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4105–4113, 2017.

Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Image shape manipulation from a single augmented training sample. In *Proceedings of IEEE International Conference on Computer Vision*, pages 13769–13778, 2021.

Liang Wang, Hailin Jin, Ruigang Yang, and Minglun Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

Miao Wang, Guo-Ye Yang, Ruilong Li, Runze Liang, Song-Hai Zhang, Peter M. Hall, and Shi-Min Hu. Example-guided style-consistent image synthesis from semantic labeling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1495–1504, 2019a.

Nannan Wang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Jie Li. Transductive face sketch-photo synthesis. *IEEE Transactions on Neural Networks and Learning Systems*, 24(9):1364–1376, 2013.

Nannan Wang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Jie Li. A comprehensive survey to face hallucination. *International Journal of Computer Vision*, 106(1):9–30, 2014.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of Conference on Neural Information Processing Systems*, pages 1144–1156, 2018a.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018b.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018c.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proceedings of Conference on Neural Information Processing Systems*, pages 1152–1164, 2018d.

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Bryan Catanzaro, and Jan Kautz. Few-shot video-to-video synthesis. In *Proceedings of Conference on Neural Information Processing Systems*, pages 5014–5025, 2019b.

Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009.

Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2561–2571, 2019c.

Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 7178–7186, 2017.

Xinrui Wang, Zhuoru Li, Xiao Zhou, Yusuke Iwasawa, and Yutaka Matsuo. Realtime fewshot portrait stylization based on geometric alignment, 2022. arXiv:2211.15549.

Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):463–476, 2007.

Pierre Wilmot, Eric Risser, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses, 2017. arXiv:1701.08893.

Chunlei Wu, Fei Hu, Di Sun, Liqiang Zhang, Leiquan Wang, and Huan Zhang. Exemplar-guided sedimentary facies modeling for bridging pattern controllability gap. *Petrophysics*, 64(2):271–286, 2023a.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Hong Lin. AI-generated content (AIGC): A survey, 2023b. arXiv:2304.06632.

Xide Xia, Tianfan Xue, Wei-sheng Lai, Zheng Sun, Abby Chang, Brian Kulis, and Jiawen Chen. Real-time localized photorealistic video style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1089–1098, 2021.

Ming Yang, Shu Lin, Ping Luo, Liang Lin, and Hongyang Chao. Semantics-driven portrait cartoon stylization. In *Proceedings of International Conference on Image Processing*, pages 1805–1808, 2010.

Jordan Yaniv, Yael Newman, and Ariel Shamir. The face of art: Landmark detection and geometric style in portraits. *ACM Transactions on Graphics*, 38(4):60, 2019.

Fachao Zhang, Xiaoman Liang, Yaqi Sun, Mugang Lin, Jin Xiang, and Huihuang Zhao. Pofmakeup: A style transfer method for peking opera makeup. *Computers and Electrical Engineering*, 104:108459, 2022.

Jichao Zhang, Aliaksandr Siarohin, Hao Tang, Enver Sangineto, Wei Wang, Humphrey Sh, and Nicu Sebe. Controllable person image synthesis with spatially-adaptive warped normalization, 2023a. arXiv:2105.14739.

Shangzhan Zhang, Sida Peng, Yinji ShenTu, Qing Shuai, Tianrun Chen, Kaicheng Yu, Hujun Bao, and Xiaowei Zhou. Dyn-E: Local appearance editing of dynamic neural radiance fields, 2023b. arXiv:2307.12909.

Yong Zhang, Weiming Dong, Oliver Deussen, Feiyue Huang, Ke Li, and Bao-Gang Hu. Data-driven face cartoon stylization. *SIGGRAPH Asia Technical Briefs*, art. no. 14, 2014.

Yuechen Zhang, Zexin He, Jinbo Xing, Xufeng Yao, and Jiaya Jia. Ref-NPR: Reference-based non-photorealistic radiance fields for controllable scene stylization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2023c.

Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-world image variation by aligning diffusion inversion chain. In *Proceedings of Conference on Neural Information Processing Systems*, 2023d.

Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. ControlVideo: Adding conditional control for one shot text-to-video editing, 2023. arXiv:2305.17098.

Mingtian Zhao and Song-Chun Zhu. Portrait painting using active templates. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*, pages 117–124, 2011.

Hao Zhou, Zhanghui Kuang, and Kwan-Yee Kenneth Wong. Markov weight fields for face sketch synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1091–1097, 2012.

Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics*, 37(4):49, 2018.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2242–2251, 2017a.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Proceedings of Conference on Neural Information Processing Systems*, pages 465–476, 2017b.

Yufeng Zhu, Jovan Popović, Robert Bridson, and Danny Kaufman. Planar interpolation with extreme deformation, topology change and dynamics. *ACM Transactions on Graphics*, 36(6):213, 2017c.

Appendix A

Author's Publications

Publications Related to the Thesis

In Journals with Impact Factor

The following publications were co-authored by the author of this thesis and published in impacted journals indexed by ISI. These publications were presented earlier in the thesis.

Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. Interactive video stylization using few-shot patch-based training. *ACM Transactions on Graphics*, 39(4):73, 2020b (IF: 5.41)

Cited in:

Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. *Computer Graphics Forum*, 39(4):91–105, 2020.

Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. *Computer Graphics Forum*, 40(4):29–43, 2021.

Ific Goudé, Rémi Cozot, Olivier Le Meur, and Kadi Bouatouch. Example-based colour transfer for 3D point clouds. *Computer Graphics Forum*, 40(6):428–446, 2021.

Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics*, 40(6):210, 2021.

Xide Xia, Tianfan Xue, Wei-sheng Lai, Zheng Sun, Abby Chang, Brian Kulis, and Jiawen Chen. Real-time localized photorealistic video style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1089–1098, 2021.

Niraj Ramesh Dayama, Simo Santala, Lukas Brückner, Kashyap Todi, Jingzhou Du, and Antti Oulasvirta. Interactive layout transfer. In *26th International Conference on Intelligent User Interfaces*, pages 70–80, 2021.

Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Image shape manipulation from a single augmented training sample. In *Proceedings of IEEE International Conference on Computer Vision*, pages 13769–13778, 2021.

Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. *Computer Graphics Forum*, 41(4):117–128, 2022.

Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. SNeRF: Stylized neural implicit representations for 3D scenes. *ACM Transactions on Graphics*, 41(4):142, 2022.

Feng-Lin Liu, Shu-Yu Chen, Yukun Lai, Chunpeng Li, Yue-Ren Jiang, Hongbo Fu, and Lin Gao. Deepfacevideoediting: Sketch-based deep editing of face videos. *ACM Transactions on Graphics*, 41(4):167, 2022.

Shyam Nandan Rai, Rohit Saluja, Chetan Arora, Vineeth N Balasubramanian, Anbumani Subramanian, and CV Jawahar. Fluid: Few-shot self-supervised image deraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3077–3086, 2022.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil DB Bruce, Richard P Wildes, and Konstantinos G Derpanis. A deeper dive into what deep spatiotemporal networks encode: Quantifying static vs. dynamic information. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 13999–14009, 2022b.

Yuechen Zhang, Zexin He, Jinbo Xing, Xufeng Yao, and Jiaya Jia. Ref-NPR: Reference-based non-photorealistic radiance fields for controllable scene stylization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2023c.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Hong Lin. AI-generated content (AIGC): A survey, 2023b. arXiv:2304.06632.

Carlos Rodriguez-Pardo, Henar Domínguez-Elvira, David Pascual-Hernández, and Elena Garces. Umat: Uncertainty-aware single image high resolution material capture. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 5764–5774, 2023a.

Ruixiang Jiang, Can Wang, Jingbo Zhang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. AvatarCraft: Transforming text into neural human avatars with parameterized shape and pose control. In *Proceedings of IEEE International Conference on Computer Vision*, pages 14371–14382, 2023.

Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of IEEE International Conference on Computer Vision*, pages 23206–23217, 2023.

Carlos Rodriguez-Pardo and Elena Garces. Neural photometry-guided visual attribute transfer. *IEEE Transactions on Visualization and Computer Graphics*, 29(3):1818–1830, 2023.

Jichao Zhang, Aliaksandr Siarohin, Hao Tang, Enver Sangineto, Wei Wang, Humphrey Sh, and Nicu Sebe. Controllable person image synthesis with spatially-adaptive warped normalization, 2023a. arXiv:2105.14739.

Carlos Rodriguez-Pardo, Konstantinos Kazatzis, Jorge Lopez-Moreno, and Elena Garces. NeuBTF: Neural fields for BTF encoding and transfer. *Computers & Graphics*, 114:239–246, 2023b.

Shangzhan Zhang, Sida Peng, Yinji ShenTu, Qing Shuai, Tianrun Chen, Kaicheng Yu, Hujun Bao, and Xiaowei Zhou. Dyn-E: Local appearance editing of dynamic neural radiance fields, 2023b. arXiv:2307.12909.

Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-world image variation by aligning diffusion inversion chain. In *Proceedings of Conference on Neural Information Processing Systems*, 2023d.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil D. B. Bruce, Richard P. Wildes, and Konstantinos G. Derpanis. Quantifying and learning static vs. dynamic information in deep spatiotemporal networks, 2022a. arXiv:2211.01783.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *Proceedings of International Conference on Learning Representations*, 2024.

Akinobu Maejima, Hiroyuki Kubo, Seitaro Shinagawa, Takuya Funatomi, Tatsuo Yotsukura, Satoshi Nakamura, and Yasuhiro Mukaigawa. Anime character colorization using few-shot learning. *SIGGRAPH Asia 2021 Technical Communications*, art. no. 8, 2021.

Wanglong Lu, Xianta Jiang, Xiaogang Jin, Yong-Liang Yang, Minglun Gong, Tao Wang, Kaijie Shi, and Hanli Zhao. Grig: Few-shot generative residual image inpainting, 2023.

Loc Huynh, Bipin Kishore, and Paul Debevec. A new dimension in testimony: Relighting video with reflectance field exemplars, 2021. arXiv:2104.02773.

Sumit Shekhar, Max Reimann, Moritz Hilscher, Amir Semmo, Jürgen Döllner, and Matthias Trapp. Interactive control over temporal consistency while stylizing video streams. *Computer Graphics Forum*, 42(4):e14891, 2023.

Min Zhao, Rongzhen Wang, Fan Bao, Chongxuan Li, and Jun Zhu. ControlVideo: Adding conditional control for one shot text-to-video editing, 2023. arXiv:2305.17098.

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Proceedings of European Conference on Computer Vision*, pages 707–723, 2022.

Akhil Singh, Vaibhav Jaiswal, Gaurav Joshi, Adith Sanjeeve, Shilpa Gite, and Ketan Kotecha. Neural style transfer: A critical review. *IEEE Access*, 9:131583–131613, 2021.

Lei Ma, Yuhui Zheng, Zhao Zhang, Yazhou Yao, Xijian Fan, and Qiaolin Ye. Motion stimulation for compositional action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(5):2061–2074, 2023.

David Futschik, Michal Kučera, Michal Lukáč, Zhaowen Wang, Eli Shechtman, and Daniel Sýkora. STALP: Style transfer with auxiliary limited pairing. *Computer Graphics Forum*, 40(2):563–573, 2021 (IF: 2.36)

Cited in:

Yan Tang. Style transfer of chinese art works based on dual channel deep learning model. *Computational Intelligence & Neuroscience*, Art. ID. 4376006, 2022.

Hideki Todo, Kunihiko Kobayashi, Jin Katsuragi, Haruna Shimotahira, Shizuo Kaji, and Yonghao Yue. Stroke transfer: Example-based synthesis of animatable stroke styles. *ACM SIGGRAPH 2022 Conference Proceedings*, art. no. 54, 2022.

Sumit Shekhar, Max Reimann, Moritz Hilscher, Amir Semmo, Jürgen Döllner, and Matthias Trapp. Interactive control over temporal consistency while stylizing video streams. *Computer Graphics Forum*, 42(4):e14891, 2023.

Linzi Qu, Jiaxiang Shang, Xiaoguang Han, and Hongbo Fu. ReenactArtFace: Artistic face image reenactment. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

Aneta Texler, Ondřej Texler, Michal Kučera, Menglei Chai, and Daniel Sýkora. FaceBlit: Instant real-time example-based style transfer to facial videos. *Proceedings of the ACM in Computer Graphics and Interactive Techniques*, 4(1):14, 2021 (IF: 1.3)

Cited in:

Ruixiang Jiang, Can Wang, Jingbo Zhang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. AvatarCraft: Transforming text into neural human avatars with parameterized shape and pose control. In *Proceedings of IEEE International Conference on Computer Vision*, pages 14371–14382, 2023.

Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. *Computer Graphics Forum*, 41(4):117–128, 2022.

Linzi Qu, Jiaxiang Shang, Xiaoguang Han, and Hongbo Fu. ReenactArtFace: Artistic face image reenactment. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

Fachao Zhang, Xiaoman Liang, Yaqi Sun, Mugang Lin, Jin Xiang, and Huihuang Zhao. Pofmakeup: A style transfer method for peking opera makeup. *Computers and Electrical Engineering*, 104:108459, 2022.

Xinrui Wang, Zhuoru Li, Xiao Zhou, Yusuke Iwasawa, and Yutaka Matsuo. Realtime fewshot portrait stylization based on geometric alignment, 2022. arXiv:2211.15549.

Chunlei Wu, Fei Hu, Di Sun, Liqiang Zhang, Leiquan Wang, and Huan Zhang. Exemplar-guided sedimentary facies modeling for bridging pattern controllability gap. *Petrophysics*, 64(2):271–286, 2023a.

In Conference Proceedings

David Futschik, Menglei Chai, Chen Cao, Chongyang Ma, Aleksei Stoliar, Sergey Korolev, Sergey Tulyakov, Michal Kučera, and Daniel Sýkora. Real-time patch-based stylization of portraits using generative adversarial network. In *Proceedings of the ACM/EG Expressive Symposium*, pages 33–42, 2019

Cited in:

Fangzhou Han, Shuquan Ye, Mingming He, Menglei Chai, and Jing Liao. Exemplar-based 3D portrait stylization. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1371–1383, 2021.

Michal Kučera, David Mould, and Daniel Sýkora. StyleBin: Stylizing video by example in stereo. *SIGGRAPH Asia 2022 Conference Papers*, art. no. 15, 2022

Appendix B

Authorship Contribution Statement

This statement describes the specific contributions of the author of this thesis to the publications presented therein.

Real-Time Patch-Based Stylization of Portraits Using Generative Adversarial Network (15%)

I have performed the user study and evaluated the results and feedback gained. Following that, I have written a part of the paper discussing the insight gained from the study.

Interactive Video Stylization Using Few-Shot Patch-Based Training (25%)

In this paper, I have contributed to the research done regarding the training of the neural network, performed experiments on the loss function as well as introduced optimizations to it. I have created the input video processing component of the implementation and contributed to both its front-end and back-end. I have also assisted during the user study in the later part of the research process and with the demo for the *Real-Time Live!* show.

FaceBlit: Instant Real-time Example-based Style Transfer to Facial Videos (25%)

I have contributed significantly to the implementation and research of the algorithm, as well as introducing a method for hair and facial hair stylization and deformation. I generated outputs of our method to use in the paper itself and assisted during the writing of the paper.

STALP: Style Transfer with Auxiliary Limited Pairing (25%)

I have designed and performed the user study, as well as the system used. I have evaluated both the results obtained through the study and the method itself. I also co-wrote the section discussing the created algorithm and its future.

StyleBin: Stylizing Video by Example in Stereo (85%)

This method and the details of its implementation has been mostly my contribution. I have performed the experiments that guided the research to its final form and heavily contributed to related research. I have also performed the quantitative and qualitative evaluations, including the user study. Concurrently, I have helped with the creation of the paper itself, writing parts of the text and producing appropriate visualizations, followed by the in-person presentation on the SIGGRAPH ASIA 2022 conference where the paper has been published.