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



Leveraging machine learning for artistic stylization

Ing. David Futschik

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: Electrical Engineering and Information Technology Branch of study: Information Science and Computer Engineering

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

March 2023

LEVERAGING MACHINE LEARNING FOR ARTISTIC STYLIZATION

Ing. David Futschik futscdav@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

Artistic style transfer and stylization have a rich and well-established history in the field of computer graphics, and have enjoyed broad popularity, especially as digital art becomes ever more prevalent. The aim of methods implementing automatic artistic style transfer or stylization is to take a source image, typically a photograph, and transform it in accordance with artists' vision, which is commonly expressed via a hand-painted style exemplar image. These techniques allow for the application of unique or personalized visual styles to new content or in other contexts, resulting in visually appealing effects that would be too time-consuming and impractical to recreate by hand. Thanks to the focus on aesthetics, the field has been uniquely popular among non-technical audiences, despite relatively high technical difficulty.

The field has undergone a remarkable fundamental paradigm shift in recent times – previously, state-of-the-art relied on tailored procedural solutions. However, the emergence of powerful large machine learning models, particularly in the field of computer vision, has prompted academic researchers to rethink the approaches to stylization tasks, and has allowed newer methods to become more data-driven, for example through the use of pretrained models. The shift brings significant advantages, individual contributions no longer need to be limited to a specific medium or visual style, and do not require specialized guidance designs. On top of that, they provide novel and stunning visual effects that have stronger impact on users. Since the early days of neural-based style transfer, the field has continued to advance, and today, many have started to notice that some computer-generated art can be indistinguishable from real artworks.

In this thesis, we focus on some of the ways machine learning, and deep learning specifically, can be leveraged towards style transfer and stylization. More precisely, we propose several methods for example-based style transfer, as well as touching on photorealistic stylization task, we discuss the applications and state-of-the-art solutions. In particular, we developed: (1) a method for distilling patch-based style transfer into a neural model, (2) a method for algorithmic refinement and upsampling of neural style transfer methods, (3) real-time technique for interactive video style transfer, (4) method to improve generalization on video style transfer tasks with emphasis on long-term correspondences; and (5) a system for photorealistic stylization and editing of real images. Finally, we look at possible future directions stemming from our work.

The thesis is presented as a collection of five research papers that were published in respected journals and presented at prestigious conferences.

Keywords

computer graphics, machine learning, artistic style transfer, stylization, example-based style transfer, painterly style transfer, neural style transfer, nonphotorealistic rendering, digital art, video style transfer, generative models, latent space inversion

Acknowledgements

There are many people who I met as a Ph.D. student that greatly influenced my development and helped me find the right research directions, teaching me enormous numbers of things in the process. Thanking all of them here would be close to impossible, so here I would like to thank the ones who were possibly the largest influences. Throughout the years, I have relied on the seasoned and excellent advice, support and mentorship provided by my advisor, Daniel Sýkora, without whom none of it would have been possible. He, therefore, deserves my greatest and deepest gratitude of everyone. I am also grateful to the mentors I have had the pleasure to learn from during my internship visits: Menglei Chai, Michal Lukáč, Rohit Pandey and Kelvin Ritland, as well as other collaborators, most importantly Eli Shechtman. Special thanks also goes out to the amazing artists who helped us with the projects that were primarily aimed at them, Zuzana Studená and Jakub Javora. Last but not least, I want to thank all the people who made the journey more enjoyable, my colleagues, friends and my close ones.

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.

Využití strojového učení v doméně umělecké stylizace

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

Abstrakt

Vzestup digitálního zpracování obrazu přinesl mnoho vylepšení, která dnešní umělci považují za nezbytná, nicméně automatizace přenosu výtvarného stylu po dlouhá léta zůstávala těžko dosažitelným cílem. Její snahou je změnit vizuální formu vstupního obrazu tak, aby byl zachován charakter původní výtvarné předlohy. Často se jedná o přenos osobitého vizuálního stylu na jiný cílový, typicky fotorealistický, materiál. Manuálně je tento proces časově velmi náročný a pracný.

Toto odvětví v poslední době prochází kompletní změnou paradigmatu—dřívější přístupy využívaly především na míru šitá algoritmická řešení, která byla omezena kontextem dané cílové domény. Po objevu hlubokých neuronových sítí se však ukázalo, že předtrénované modely z oblasti počítačového vidění mohou být velmi dobrým nástrojem i v oblasti stylizace obrazu. V posledních letech se literatura začala ubírat téměř výhradně tímto novým směrem. Metody vyvinuté za pomoci strojového učení již často nejsou limitované konkrétním výtvarným médiem nebo stylem a začínají vykazovat schopnost generalizace. Díky této charakteristice je lze využít i jako zdroj umělecké inspirace. Od nedávných prvních vlaštovek se odvětví stylizace pomocí strojového učení posunulo dopředu, a dnes už si i laická veřejnost začíná všímat, že automaticky vygenerovaná umělecká díla mohou být k nerozeznání od tvorby reálných umělců.

V této disertační práci představujeme soubor nových algoritmů, které demonstrují možnosti využití hlubokého strojového učení a neuronových sítí pro automatický přenos výtvarné předlohy a stylizaci. Konkrétně ukážeme navržené nástroje pro: (1) destilaci chování algoritmické stylizační metody za použití neuronové sítě, (2) přenos výtvarného stylu ve vysokém rozlišení, (3) interaktivní přenos uměleckého stylu do videosekvencí v reálném čase, (4) state-of-the-art metoda pro přenosu výtvarného stylu do videosekvenci kladoucí důraz na specifikaci menšího počtu klíčových snímků, a (5) realistickou stylizaci skutečných fotografií.

Tato práce je koncipována jako soubor pěti článků popisujících navržené přístupy, které byly publikovány v uznávaných časopisech a prezentovány na prestižních konferencích.

Klíčová slova

počítačová grafika, strojové učení, přenos výtvarného stylu, stylizace, styl podle předlohy, ručně kreslené předlohy, neuronové sítě, nefotorealistické vykreslování, digitální tvorba, přenos stylu na video, generativní modely, projekce do latentního prostoru

Contents

1	Intr	roduction	1
	1.1	Introduction to Example-based Style Transfer	4
		1.1.1 Approaches to Style Transfer	8
	1.2	Our contributions and structure of the thesis	11
2	Rel	ated Work and State-of-the-Art	17
	2.1	Procedural Methods	17
	2.2	Image Exemplar Based Methods	18
		2.2.1 Non-parametric guided synthesis	18
		2.2.2 Parametric synthesis	19
		2.2.3 Hybrid: Parametric guidance for patch-based synthesis	21
	2.3	Dataset Based Methods	22
3		estyleGAN: Real-Time Patch-Based Stylization of Portraits Using	
	Ger	nerative Adversarial Network	23
	3.1	Introduction	23
	3.2	Related Work	24
	3.3	Our Approach	26
		3.3.1 Training Objective	26
		3.3.2 Network Architecture	27
		3.3.3 Implementation Details	28
	3.4	Results	29
		3.4.1 Interactive Scenario	29
		3.4.2 Generalization \ldots	30
		3.4.3 Perceptual Study	31
		3.4.4 Comparisons	32
	3.5	Limitations and Future work	34
	3.6	Conclusion	34
4	Δrb	oitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis	
т	1 1 1 1	strary style fransier esting rearany-Guided Faten-Dased Synonesis	37
	4.1	Introduction	37
	4.2	Related Work	40
	4.3	Our Approach	42
		4.3.1 Neural-Based Style Transfer	42

		4.3.2 Guided Patch-Based Synthesis
		4.3.3 NNF Upscaling
	4.4	VGG-Based Guidance
	4.5	Results
	4.6	Limitations and Future Work
	4.7	Conclusion
5	Inte	eractive Video Stylization Using Few-Shot Patch-Based Training 55
	5.1	Introduction
	5.2	Related Work
	5.3	Our Approach
		5.3.1 Patch-Based Training Strategy
		5.3.2 Hyper-parameter Optimization
		5.3.3 Temporal Coherency
	5.4	Results
		5.4.1 Comparison $\ldots \ldots \ldots$
		5.4.2 Interactive applications
	5.5	Limitations and Future Work
	5.6	Conclusion
6	STA	ALP: Style Transfer with Auxiliary Limited Pairing 73
	6.1	Introduction
	6.2	Related Work
	6.3	Our Approach
	6.4	Results
		6.4.1 Perceptual study
	6.5	Limitations and Future Work
	6.6	Conclusion
7	Chu	nkyGAN: Real Image Inversion via Segments 89
	7.1	Introduction
	7.2	Related Work
	7.3	Our Approach
	7.4	Evaluation
		7.4.1 Fidelity of projected images
		7.4.2 Editability of projected images
		7.4.3 Comparison with current state-of-the-art
	7.5	Applications
	7.6	Limitations
	7.7	Conclusion
8	Cor	nclusion 103
	8.1	Summary
	8.2	Concurrent and Future Work

CONTENTS

Re	ferences	107
A	Author's Publications	121
в	Authorship Contribution Statement	125
\mathbf{C}	FacestyleGAN Supplementary Material	127
D	Video Stylization Using Few-Shot Patch-Based Training Supplementary Material D.1 Interactive applications	 129 129 130 131 131 131
Ε	ChunkyGAN Supplementary Material E.1 Projection Fidelity	$\begin{array}{c} 136\\ 147 \end{array}$

List of Figures

1.1	Examples of overpaint workflow – captured camera footage (a, c) is painted over frame by frame with stylized content (b, d) by an artist. This work- flow is very intuitive and natural for artists, and usually results in roughly registered images, although it can be style-dependent. The chosen styles may vary wildly depending on the artists' intention – some can be elabo- rate with lots of textural details, while others simplify many features, as can be seen in these examples taken from production usage, (a, b) is one frame from the Loving Vincent movie; example (c, d) is a frame from an animated video produced by channel Joel Haver on YouTube.	3
1.2	Diagram of example based style transfer. User provides target image and style exemplar, which the chosen style transfer method then uses to synethesize the result. Some methods require additional inputs such as guidance channels or have tunable parameter settings that provide further	4
1.3	Stylizing video sequences. Images (b, d, f, g) represent the sequence to be stylized. Artist takes one frame from this sequence (b) and creates its stylized version (a). The goal is then to propagate the artistic style from (a) to the remainder of the sequence (c, e, g).	5
1.4	Patch-based synthesis. We aim to transfer the painterly style of (a) onto a photograph of a person (c) by copying image chunks to produce an analogous image (b). To define the patch similarity, guidance channels need to be designed (d, e), in this case a version of distance field based on semantic segmentation. If the teal patches drawn over images (d) and (e) are nearest neighbors, we would expect the red patch in (b) to be copied from the corresponding location in (a)	7
1.5	Early method implementing artistic painterly stylization [Hertzmann 1998]. A source image (a) depicts the content to be transformed. Image (b) is the result obtained by algorithmically applying small radius brushes over the source image (a). The brushes are applied in multiple layers; large brushes are applied first, medium and small brushed follow, thus creating an illusion of a real painting	8
1.6	Result of Lu et al. [2013]. Brush stroke exemplars (a) are used to syn- thesize a painting (b). The foreground flower strokes (close-up c), use oil paint exemplars(a–left), while the background strokes (close-up d), use plasticine exemplars (a–right). The low level image features imitate real artistic media well	9

1.7	Image Analogies [Hertzmann et al. 2001]. An unfiltered image A together with its filtered-stylized version A' define a desired transformation. The goal is to perform the same transformation on an unfiltered image B to obtain its filtered version B' . Besides A, A' , and B , there is no other input to the framework, the transformation is learned from the pair of A and A' .	9
1.8	An example of neural network based approach to the style transfer task [Gatys et al. 2016]. Resulting image (b) depicts the same semantic content as source photograph (a) and at all scales appears visually similar to the style exemplar (c).	10
1.9	General example-based style transfer. The task is to transfer artistic style from the given style exemplar (a) to the given target (b) while preserving the appearance of (a) and the content of (b). Example of a possible result in (c).	11
1.10	Stylization of face photographs using our approach as compared to FaceS- tyle [Fišer et al. 2017]. Our approach achieves similar quality while re- taining more identity defining features, better generalization and runs 50 times faster	12
1.11	Given one keyframe (a) and a video sequence (in blue), our method, Inter- active Video Stylization Using Few-Shot Patch-Based Training, produces the stylized results for the rest of the frames (b, c, d)	13
1.12	Stylizing using a limited pairing: (a) and (c) express the desired trans- formation, in STALP we train for style consistency across multiple input images, in this case the constituent photographs used to create (b). The result is a consistent stylization that can be seamlessly stitched together into a panorama image (d)	14
1.13	Using ChunkyGAN allows artists to produce local layered edits, applied in sequence on a real photograph (a): changing gaze direction (b), adding smile (c), changing haircut and nose shape (d)	14
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. [Fišer et al. 2017], Liao et al. [Liao et al. 2017], Selim et al. [Selim et al. 2016], and	

Gatys et al. [Gatys et al. 2016]) while being able to run at interactive frame rates on a consumer GPU. Style exemplar: © Scary Zara Mary.

3.2	Ablation study. A demonstration of visual quality improvement achieved using modified VGG loss and our improved network architecture: (a) re- sult of our network trained without using VGG loss, (b) result gener- ated using all losses, however, without our improved network architec- ture, i.e., using the original architecture of Johnson et al. [2016], (c) our result, (d) result generated using FaceStyle algorithm [Fišer et al. 2017], (e) style exemplar. Note how our full-fledged approach better repro- duces the original style exemplar (see the avoidance of artificial repet- itive 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 exem- plar: © <i>Matthew Cherry</i> via http://matthewivancherry.com/home.html and https://www.instagram.com/matthewivancherry.artist (HAT, oil on canvas, 48" x 48", 2011).	27
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.	28
3.4	Exemplars of styles used in Figures 3.6, 3.7, and 3.8. See Figures 7.1, 3.2, and 3.9 for the remaining style exemplars. Style exemplars: (a–b) © Adrian Morgan, (c) Viktor Ivanovich Govorkov, (d) © Will Murray.	29
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 7.1 and 3.2.	30
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 out- put from FaceStyle [Fišer et al. 2017]. The corresponding style exemplars are visible in Figure 3.4.	31
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 out- put from FaceStyle [Fišer et al. 2017]. The corresponding style exemplars are visible in Figures 3.4 and 3.9.	32
3.8	Comparisons of our approach with current state-of-the-art in image-to- image transation: <i>pix2pixHD</i> [Wang et al. 2018b], <i>pix2pix</i> [Isola et al. 2017], and <i>starGAN</i> [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.	33
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: (C) Graciela Bombalova-Bogra.	33

- 4.1 An example of stylizing an extremely high-resolution image using our proposed method: (a) style exemplar of 26400×13100 px, (b) content image of the same resolution, (c) low resolution result of [Gatys et al. 2016] enhanced and enlarged by our method to the mentioned resolution. To the right, zoom-in patches of different parts of (c) up to zoom of $128 \times$ are shown; see all the individual brush strokes and its sharp boundaries. Also, notice how the structure of the original canvas and little cracks of the painting are preserved.
- 4.2 An example of enhancing the result of neural-based approach using our method: (a) target photograph, (b) style exemplar of the same size, (c) $6 \times$ zoom-in to the style exemplar, (d) the output of neural-based method DeepArt [Gatys et al. 2016] is capable to perform convincing stylization; nevertheless, the image contains artifacts caused by the parametric nature of the used neural network. High-frequency details like the structure of strokes and canvas are largely lost, sacrificing the visual quality of the original artistic medium. In contrast, our method (e) brings significant quality improvement, it restores the individual brush strokes and boundaries between them faithfully, the result better reproduces the used artistic medium as well as canvas' structure. Note how the cracks of the original artwork are preserved; although zoom-in patches are shown, we encourage the reader to zoom-in even further.
- 38

38

Simplified scheme of a patch-based, neural-based, and our hybrid style 4.3transfer method: The left column shows a patch-based approach [Fišer et al. 2016] with guidance based on blurred grayscale images as proposed in the original Image Analogies method [Hertzmann et al. 2001]. The resulting image has high texture quality and preserves artistic attributes and canvas structure well; however, the result does not properly respect the content semantics, causing water to become brown. The middle column shows a neural-based approach [Gatys et al. 2016], no guidance channels are needed and global style properties and image semantic are preserved well. However, the resulting image lacks high-frequency details of the original style exemplar, contains artifacts, and colors that are not present in the original style. The right column represents our method where lowresolution neural transfer result is used as a guidance channel for patchbased style transfer. Our result attenuates the neural artifacts and restores

xiv

- Proposed pipeline: (a) style exemplar and (b) content image are both sub-4.4 sampled α -times and processed by a neural-based style transfer method (Sec. 4.3.1) which results in low resolution image (c) where fine details are missing and artifacts are apparent (see green and purple checkerboard artifacts). Next, low resolution result (c) from the previous step, style image (a) in the same resolution as (c), and β -times subsampled style image (a) are used as an input to a patch-based synthesis algorithm (Sec. 4.3.2) which outputs dense nearest neighbor field (NNF) (f) from which the corresponding image (d) can be produced using voting step [Wexler et al. 2007]. Finally, in NNF upscaling step (Sec. 4.3.3) the low-resolution NNF (f) is upscaled β -times to the original resolution (g). Patch coordinates in NNF (f) and (g) are encoded as red and green color levels. Note subtle color gradients in (f), which indicate the presence of fine patch coordinates in upscaled NNF that points to the patches in the original high-resolution style exemplar (a). Given the upscaled NNF (g) and the style exemplar in its original resolution (a), high-resolution, and a perfectly sharp final result is created using voting step (e).
- 4.5 An overview of our VGG-guided style transfer pipeline: we start with a target image and a style exemplar, extract their VGG-19 features, normalize them, reduce their dimensionality using PCA, and use these as guidance for subsequent patch-based synthesis. Even though the proposed pipeline is straightforward, it yields convincing output.
 40
- 4.6 Demonstration of the problem when patch-based synthesis has to rely on ambiguous color guidance: (a) style exemplar, (b) target image, (c) output of Gu et al. [2018], (d) output of our basic algorithm with color-based guidance, (e) output of our style transfer algorithm with neural guidance. Note how our VGG-guided algorithm better preserves the semantics of the target photo, cf. details in (f) and (g).

4.9	Our method enhancing the results of five different neural-based approaches: The leftmost column-content images and style exemplars (with zoomed patches). Next, left-to-right, are the result of DeepArt [Gatys et al. 2016], DeepDream, Gu et al. [2018], Liao et al. [2017], and Li et al. [2017]. The top-left triangle shows the result of the underlying neural-based approach (bicubically up-sampled from a typical size of 600×400 px to the target resolution), while the bottom-right shows result enhanced by our method (top row-entire stylized images, bottom row-zoom-in). Our results not only have significantly higher resolution but also better preserve the original colors and canvas structure as well as brush strokes visible in the exemplar painting. Various artifacts caused by the neural approach are significantly suppressed. All images shown in this figure are of resolution ranging from 4000×2200 to 6000×4000 px	47
4.10	Performance of our method (full pipeline–Fig. 4.4, excluding the neural part) on images ranging from resolution of 1Mpx, (i.e. 1000×1000 px) to extremely large resolution of 256Mpix (i.e., 16000×16000 px). Orange, yellow, and green lines show a case where the parameter β was set such that the patch-based method was run on a resolution of 1Mpix, 4Mpix, and 8Mpix respectively. The measurement was done on a mid-range laptop with NVIDIA GTX 1050 graphics card	48
4.11	Results produced by our VGG-guided style transfer algorithm (from left to right): style exemplar, target image, and our result. Our method works well namely in cases when style and target images depict similar content, i.e., when they have compatible VGG activations.	49
4.12	A screenshot of our method running in Adobe Photoshop: (a) zoom of a target layer, (b) zoom of a style layer; the visible layer is the result of DeepDream enhanced by our method	50
4.13	Additional results produced by our VGG-guided style transfer algorithm (from left to right): style exemplar, target image, and our result	51
4.14	A limitation common to neural-based approaches: (a-b) content image, (c-d) style exemplar, (e-f) result of [Li et al. 2017] enhanced by our method. The content of the original image is not preserved well. In the first case, the similar mixture of colors is used to paint bushes, house, and also the sky. In the second case, all colors appearing in the style exemplar are used to stylize the target regardless of its content. However, high-frequency content is reproduced well. To address this limitation, we propose to incorporate a neural network trained for image segmentation into our pipeline	52
4.15	Large-scale artifact limitation: (a) content image, (b) style exemplar, (c) result of Gatys et al., distortions in eye region are visible, (d) ours, colors and high-frequency details are reproduced well; however, in our current pipeline, large-scale artifacts produced by the underlying neural	

approach are not fixed. Thus distortion in the eye region is still apparent. 52

- 5.1 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 (c) Zuzana Studená. 56
- Comparison of full-frame training vs. our patch-based approach: the orig-5.3inal 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. 5.6). 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)
- 5.4 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 this paper, we use a combination of L1 loss, adversarial loss, and VGG loss as described in [Futschik et al. 2019]. Video frame (left) and style exemplar (right) courtesy of © Zuzana Studená.

60

- 5.5 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.
- 5.6 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 \bigcirc Zuzana Studená.
- 5.7 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 (C) Zuzana Studená and style exemplar (b) courtesy of (C) Pavla Sýkorová. 64
- Influence of important hyperparameters on visual quality of results. The 5.8loss, 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

xviii

62

63

5.9	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 signifi-	
	cantly 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.	67

- 5.10 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.
- 5.11 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 C Adam Finkelstein and style exemplars (a) courtesy of C Pavla Sýkorová. 68

69

5.15	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 \bigcirc MAUR film	70
5.16	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.	70
6.1	An example of style transfer with limited auxiliary pairing—an artist pre- pares a stylized version (source style) of a selected video frame (source frame). Then an image-to-image translation network is trained to trans- fer 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 styl- ized counterpart are taken into account. Once the network is trained, the entire sequence can be stylized in real-time (our approach). In contrast to current state-of-the-art in example-based video stylzation (Jamriška et al. [Jamriška et al. 2019] and Texler et al. [Texler et al. 2020b]) our approach better preserves important visual characteristics of the style ex- emplar even though the scene structure changed considerably (head rota- tion). The advantage of having an auxiliary stylized pair is also visible in comparison with the output of Deep Image Analogies of Liao et al. [Liao et al. 2017]. Although the style's texture is preserved reasonably well, the transfer is not semantically meaningful	73

- 6.3 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.

6.4

An ablation study demonstrating the importance of individual terms in	
our objective function (6.1)—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 combin-	
ing L_1 loss and VGG loss in (6.1) we get the result which produces a good	
balance between identity and style preservation. Source style (C) Graciela	
Bombalova-Bogra, used with permission.	78
An illustration of a wash-out effect caused by adding an explicit content	

An illustration of a wash-out effect caused by adding an ex 6.5loss term [Kolkin et al. 2019] into our objective function (6.1). Target render stylized using model \mathcal{F} optimized on a stylized pair from Fig. 6.9 with low, medium, and high content loss weight. Note how style details deteriorate gradually with the increasing content loss. Source style (C) Štěpánka 79

- 6.6Video 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) (c) Zuzana Studená, and (bottom row) 80
- Example of video stylization with multiple keyframes—two keyframes $K_1 =$ 6.7 (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 , $Y_1, Y_2, Z_1 \& Z_2 \bigcirc$ Muchalogy, used with permission.
- A different sampling strategy for a selection of frames in Z—a source frame 6.8 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 adaptively 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) 82

- 6.9 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], and 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.
- 6.11 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] and 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. 6.12. Source style © Štěpánka Sýkorová, used with permission.
- 6.12 Panorama stylization results (cont.)—two additional artistic styles (source style) used to stylize the panorama shown in Fig. 6.11. Note how our approach (stylized panorama) handles also a higher level of abstraction (first row). Source style (top row) (C) Jolana Sýkorová, used with permission. 84
- 6.13 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] and 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.

83

84

6.14	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 lighting 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.	85
6.15	Illustration of common limitations of our method	87
6.16	The advantage of using style transfer with auxiliary pairing in visual at- tribute transfer scenario of Deep Image Analogy [Liao et al. 2017]. Al- though the style's texture and semantics (see source style in Fig. 6.1) 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	87
6.17	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.	88
7.1	Real image manipulation examples created interactively using our method. The left-most images are the original photographs, the remaining columns show following edits: changing gaze direction, opening mouth, growing a beard and aging. <i>Source images: Shutterstock</i>	89
7.2	ChunkyGAN flowchart—the output image O computed as a weighted com- bination of n images generated by a network G^{I} given a set of n latent codes X^{I} . Weights are specified by a set of n segmentation masks S that can be specified manually or generated automatically by a segmentation network G^{S} using a latent code X^{S} . Source image: Raimond Spekking / CC BY-SA 4.0 (via Wikimedia Commons)	92
7.3	Progression of the optimization. Images and color-coded segmentation maps for iterations 1, 5, 9, 15, 23, 37, 500. Source image: Adobe Stock $\$.	94

xxiii

7.4	Projection fidelity – scatter plots. Our method is compared with global projections $(\mathcal{W}, \mathcal{W}^+, \mathcal{S}\text{-space})$. X and Y axis represent the LPIPS loss between the original image and the image projected globally and projected by our method in \mathcal{W}^+ respectively. Each point corresponds to one image from the CelebA subset, in blue and in orange with and without the regularization respectively. The red line delineates the equal LPIPS losses. Our method improves projection for all images in all tested latent spaces. The regularization slightly decreases the projection fidelity, but remains still better than global methods.	95
7.5	Qualitative assessment of projection fidelity on hard examples. All im- ages were projected with regularization. For more examples refer to the supplementary material. <i>Source images: Adobe Stock</i>	96
7.6	Global edits with the same effective strength. For our methods the latent codes of all segments were manipulated equally. Source images: Mingle Media TV (Kate Winslet), Neil Grabowsky / Montclair Film (Ethan Hawke)	98
7.7	Challenging global edits. The first row depicts the original and the pro- jected images using our approach with and without regularization, Pivotal Tuning [Roich et al. 2021], StyleFlow [Abdal et al. 2021], \mathcal{W} and \mathcal{W}^+ [Ab- dal et al. 2019]. The remaining two rows show resulting global edits of age. <i>Source image: BlochWorld</i>	99
7.8	Projection fidelity of our method with respect to the current state-of- the-art in encoder-based techniques: HyperStyle [Alaluf et al. 2022], ReStyle [Alaluf et al. 2021], pSp [Richardson et al. 2021], and e4e [Tov et al. 2021]. Source images: Ayush Kejriwal (bindi), BlochWorld (face mask)	100
7.9	Examples of local layered edits applied subsequently on a real photo- graph (a): changing gaze direction (b), adding smile (c), changing haircut and nose shape (d)	100
7.10	Enforcing continuity of inconsistent edits—a photo of a person to which we would like to add glasses (a), user-specified segmentation mask S_1 with a projection X_1 matching the original image (b), manipulating X_1 generates glasses that do not fit the shape of S_1 (c), a new mask S_2 is marked encompassing two discontinuous parts (d), a composite with a projected region S_2 where the new latent code X_2 is refined from X_1 to produce the dark region inside S_2 (e)	101
C.1	A selection of results used in the perceptual study. Note that the results are comparable for both approaches with no obious failures.	127
D.1	Scenario No. 1: an artist is drawing over a stencil of a keyframe using traditional media. The stencil contains markers that allow us to perfectly align the frames to prevent shift in images	130
D.2	Scenario No. 2: an artist is stylizing an object as seen by the camera in	130
D.3	Scenario No. 3: an artist is stylizing an object as seen by the camera in real-time using a physical stencil.	130

D.4	The keyframe (a) was used to produce the sequence of 148 frames. While	
	the body part is faithfully represented in both [Jamriška et al. 2019] (b)	
	and ours (c), our approach better preserves the facial region; see the zoom-	
	in views [Jamriška et al. 2019] (d) and ours (e). Video frames (insets of a–c)	
	courtesy of © MAUR film and style exemplar (a) courtesy of © Jakub	
	Javora.	132

- E.4 Qualitative assessment of projection fidelity on challenging examples encoder-based methods. Face occlusions are not faithfully inverted by any of the encoder-based methods. The out-of-domain image in column 3 is hard to project, other methods fail to generate the tentacles. 140
- E.5 Further comparison on images which are challenging to invert accurately using existing methods **optimization-based results**. Pivotal Tuning does not faithfully reproduce features which are far from the domain of the original trained network (toothbrush handle in columns 1 and 2). . . 141

E.7	Qualitative assessment of projection fidelity on challenging examples - optimization-based methods . The identity is reliably preserved using our method, S -space inversion, and Pivotal Tuning; however, the Pivotal Tuning does not generate the tattoo in the first column in great detail,	
	and the \mathcal{S} -space inversion produces unrealistically-looking hand in column	
		143
E.8	Qualitative assessment of projection fidelity on challenging examples - encoder-based methods. Encoder-based methods all fail to generate	1 / /
E.9	the hand in column 4 and the tattoo in column 1	144
E.9	inverted images by all tested methods. Other rows display corresponding edited results along given semantic directions. For our methods, the edit- ing was done simply by manipulating the latent codes the same for all the	
	segments. The results of the inversion and editing are fully automatic. No	
	manual adjustments and no postprocessing were performed	145
E.10	Limitations of using our method to perform edits which change the ge-	
	ometry to a significant extent. In these examples, segment seams become	
	visible for larger yaw changes without any explicit treatment of the segments.	146
E.11	Effect of regularization. The projected (composed) images are on the	
	right. The left side depicts individual projections with the corresponding	110
E.12	Effect of regularization - out-of-domain example. The projected (com-	148
	posed) images are on the right. The left side depicts individual projections	149
E.13	Interpolation examples—our approach can be used to perform interpola-	149
	tion between two different identifies. The estimated latent code in each segment is linearly interpolated and the final image is then composed using Laplacian numeric. A law advantage here is that in our method identity	
	Laplacian pyramid. A key advantage here is that in our method identity is preserved better and thus the transition looks more believable	150
E 14	Editing using our method based on StyleGAN2 model trained on photos	100
12.11	with cars—original image (a), detail of the original image (b), local edits	
	of wheel disc design $(c-e)$.	150
E.15	Edits performed on famous painting using our approach with StyleGAN2	
	model trained on real faces—original Da Vinci's Mona Lisa (a), more	
	pronounced smile (b), change in the gaze direction (c), original Botticelli's	
	The Birth of Venus (d), change in the mouth expression (e), different shape	
	of eyes (f), original Rembrandt's Little Self-portrait (g), changing mouth	151
	expression (h), different shape of the nose (i)	151

Chapter 1

Introduction

Since time long before written history, humankind has been fascinated with visual forms of art. Going as far back as the age of the earliest cave wall paintings or simple carvings, these depictions tend to be abstract, simplifying and subjective in their nature – at first, this fact likely came about as an artifact of attempts to *accurately* represent the world around us. Even though in modern age we possess the technical ability to readily and very accurately capture and reproduce how our primary sense presents the material world to us, we continue to embrace the subjective, artistic form of expression and, in fact, pursue it with the purpose of "feeding the soul". The craft of creating pleasing imagery has lead to such artistic directions as cubism, impressionism, expressionism or informalism, which almost seem to aim to produce works that are as far from physical reality as possible. As it stands, visual forms of art remain as appealing and interesting as ever, and more people than ever before dedicate their time, or entire lives to creation of new works and directions.

For most of history, the access to methods of both serious art production and consumption remained limited; sometimes admiration of era-defining, iconic works was afforded only to the wealthiest members of society. It is clear that traditional visual art media have distinct character and allure, but are naturally uneconomical for widespread usage. On the other hand, ubiquitous adoption of computer image processing democratized the creation and distribution of visual art to a very significant extent, completely transforming the discipline, and granting unprecedented numbers of participants the ability to contribute. Not only that, many artists also quickly found that experimentation became easier when incorporating computers into their creative process, and digital editing opened up avenues for new, wonderful directions which would be otherwise very tedious, expensive, or near impossible to achieve without digital tools. Today, just three decades after the first release of the legendary Adobe Photoshop, traditional artists are seemingly harder to find than ones specializing in digital creation. Over those three decades, the field of digital image processing has undergone monumental evolution from the times of simple pixel-wise operations and hard brush painting, and as the complexity of tools increases, so do the expectations of users.

Developing new software techniques to provide artists with ever more flexibility and freedom of expression is thus an area of very active research and continues to present increasingly intriguing and unique challenges. One of the most difficult but crucial objectives when designing systems with artists in mind is striking the right balance between ease of use, quality of pleasing outcomes, and freedom of control over how the system operates. The difficulty is compounded by the fact that different groups of end-users have different tolerances for each category. While casual consumers might be very happy with an automatic system that is easy to use (ideally, everything happens with the press of a single button!) and produces amusing results, professional artists and advanced hobbyists generally favor greater control and are willing to give up ease of use if it means they can, in return, achieve exceptional quality and create distinguished works with their signature style. Combining such different demands in the same package often results in friction and confusion for everybody, and therefore, it becomes important to define the target group of a technique and build the design philosophy around it.

This concern is further complicated by another demanding factor when it comes to artist-facing algorithms: the broad requirement for interactivity. Instantaneous response increases artistic productivity and reduces frustration. More importantly, the best results can only be obtained when the user is given enough freedom to experiment and explore. Even skilled and experienced artists will find it difficult to effectively use tools that are fraught with long latency times or lengthy iteration cycles. While general improvements in the computing capabilities of consumer devices have been able to alleviate many of these issues, it also seems to perpetuate a cycle of increased capability being met with correspondingly increased expectations. For example, it is common to require 4K resolution images today, compared to Full HD just a couple of years ago. Delivering usable algorithms is as much of an engineering challenge as it is a design and research one, and breakthroughs in one or the other aspects enable the entire field to move forward.

Recent developments in general availability of computing power and the tremendous progress in the fields of machine learning and computer vision enabled many novel approaches to digital image processing. The communal effort to create open sourced, easy to use ML frameworks like PyTorch [Paszke et al. 2019] and foundational vision models such as VGG-19 [Simonyan and Zisserman 2014] propelled many research areas forward, and artist-facing algorithms are no exception. In this thesis, we explore several original algorithms for artistic digital image processing, more specifically, four of the presented algorithms provide tools for **artistic stylization**, which is a unique subcategory of digital image processing for artistic domain translation. In general terms, it can refer to any kind of artistic modification to existing work in order to incorporate artistic expression, but for our purposes, such content creation technique follows outlines of popular creative process – typically, the goal is to alter a photorealistic visual guide (photos, video) to exhibit attributes closer to a desired artistic style or intention where most of the high-level structure of the original image is retained. When done by hand, it is called overpainting, a process during which an artist creates their own rendition of the image by directly painting over of the visual guide such as shown in Fig. 1.1. Performing stylization brings many artistic advantages, as it allows the final material to help direct viewer's attention, better highlight select features (such as caricature images), abstract away visual noise or even just feature a more interesting color palette. When we attempt to mimic the stylization process algorithmically, it can be viewed as a form of non-photorealistic rendering, since the goal is rarely to translate between photorealistic domains.

Non-photorealistic rendering (NPR) is a distinct and established field of research and practical application within computer graphics. Although its prominence is comparatively diminished in relation to its photorealistic counterpart, NPR seeks to replicate the intricacies of hand-crafted artwork that were formerly only accessible through manmade abstractions. Whereas human artists are naturally adept at this process and do



Figure 1.1: Examples of overpaint workflow – captured camera footage (a, c) is painted over frame by frame with stylized content (b, d) by an artist. This workflow is very intuitive and natural for artists, and usually results in roughly registered images, although it can be styledependent. The chosen styles may vary wildly depending on the artists' intention – some can be elaborate with lots of textural details, while others simplify many features, as can be seen in these examples taken from production usage, (a, b) is one frame from the Loving Vincent movie; example (c, d) is a frame from an animated video produced by channel Joel Haver on YouTube.

not consider it to be a particularly difficult task, its computational emulation presents a more loosely defined problem in contrast to photorealistic rendering, for which there is often a known real-world ground truth example. However, the basic objectives of NPR are no less important: creating sketches, cartoons, or paintings. We are not limited to merely emulating the physical media used to create these types of artworks, but also the abstractions which are commonly employed to construct such images, like larger widths of strokes in sketch images representing less detailed areas. Style transfer is a higher level task, in which we often only have a single exemplar to follow, yet we would like to reproduce many more images using the same mental and physical processes which were used to create the style.

In this chapter, we first briefly introduce example-based style transfer and stylization as a standalone task, its basic and more advanced applications, and describe algorithms and techniques used to attempt to solve this task and their primary formative ideas. Furthermore, we define the scope of our research focus, and we concisely cover our contributions to the field so far, which are then individually described in more depth in the following chapters.

1.1 Introduction to Example-based Style Transfer

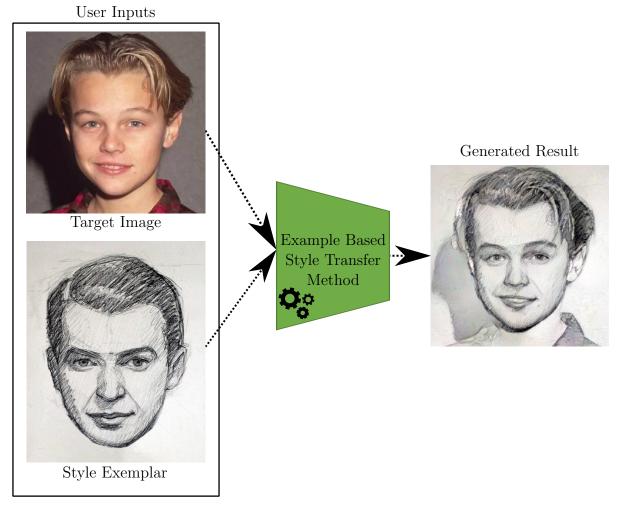


Figure 1.2: Diagram of example based style transfer. User provides target image and style exemplar, which the chosen style transfer method then uses to synethesize the result. Some methods require additional inputs such as guidance channels or have tunable parameter settings that provide further control over the process.

In the most basic sense, the goal of artistic style transfer is to create images that resemble artworks which could have been created by a person; in this context, transfer refers to applying artistic *style* defined by an existing artwork to an image depicting different content or belonging to a different image domain. This other image might be a photograph or another artistic image, and is commonly referred to as the *content* or *target* image. At the end of this process, we expect to obtain an image that retains contextually important, structural, or semantic information that comprise the content image, but assembled out of features extracted from the style image (for visual diagram of general setup see Fig. 1.2).



Figure 1.3: Stylizing video sequences. Images (b, d, f, g) represent the sequence to be stylized. Artist takes one frame from this sequence (b) and creates its stylized version (a). The goal is then to propagate the artistic style from (a) to the remainder of the sequence (c, e, g).

Naturally, we can extend the definition to video sequences. The ultimate goal is to propagate a given style from a still image into all frames of a given target video sequence where the content changes, in order to animate the stylized frame, like shown in Fig. 1.3. Current methods tend to work well on sequences without significant movement or abrupt changes, as whenever a part of an object that was not directly represented in the stylized exemplar appears, the desired outcome is inherently ambiguous. Besides this goal, methods that aim to map an entire video style guide onto a different video content target also exist [Jamriška et al. 2015; Yang et al. 2021]. This goal is even more ambitious because of the added difficulty of aligning nad reconciling appearances of different temporal behaviors between the input sequences, but less frequently desired by artists as preparing compatible sequences is a laborious process.

The description we offer is rather open-ended, and much like in most of art itself, it leaves as purely subjective whether an image successfully captures transferred artistic style. Style in art is composed of many different concepts, including color palette, lowlevel shapes, geometric arrangements, object composition, canvas material, and structure. With so many ill-defined variables, it is unclear whether an objective metric of style similarity can be created. Some unmistakable styles exist in the domain of visual art, such as those of Pablo Picasso or Vincent van Gogh. As we move to lesser-known artworks, the lines between different styles become blurrier. At the other end of the spectrum are generic styles without strong and unique features, often seen in digital painting. Ideally, we aim for such quality of style transfer that most people would agree was created by the same artist and with the same intention, in the form of Turing test [Salesin 2002]. Evaluating work using such tests is inherently costly and requires considerable effort; therefore, automated metrics have been proposed but have had limited success [Yeh et al. 2020]. Despite the difficulty in evaluation, style transfer techniques can be used in various applications and for a myriad of reasons, as the intended effect is automatically making images more enjoyable and interesting to look at, which is beneficial, for example, for advertising purposes or product design [Zhao et al. 2021]. Style transfer is also utilized to mask or hide unwanted artifacts in other computer generated imagery – consider as an example the task of changing the gaze direction of a person. Attempting full photorealism has been elusive in similar applications, and easily results in uncanny valley effects. If combined with a style transfer method to make the entire eye region look more painterly, it significantly increases the margin of error for most viewers, without stepping away from the original intention.

As it stands, large part of uses for still image style transfer revolve around facial stylization in particular, turning portrait photos into more painterly styles or even caricatures [Fišer et al. 2017; Futschik et al. 2019], and with popularity of social networks, style transfer has been welcomed in generating unconventional profile pictures and other consumer-facing content where differentiation is important and will no doubt play a big role in personalization as applications such as "metaverses" become more mainstream. Some still image style transfer algorithms have already seen conversion into commercial products for creating cheap, computer generated artistic images to use in place of traditional paintings, such as Deep Art [Gatys et al. 2016]¹ or Prisma [Johnson et al. 2016]².

Though applications for still images are interesting enough on their own, they are dwarfed by the possibilities when talking about artistic stylization applied to video sequences, for example in film production, where creation of traditional hand-drawn animations can become very labor-intensive, as every single frame of a sequence has to be individually painted or modified by an artist. To alleviate the effort, we can set up a scenario where an artist paints over e.g. a coarse 3D animation, such as in the approach of Bénard et al. [2013]. Some selected frames of the target sequence are painted by hand and serve as the input content-style mapping pair into a style transfer algorithm. This algorithm then should propagate the style into the remaining frames according to the underlying 3D geometry changes.

In similar vein, a common technique used in animation is rotoscoping; a live-action scene is shot, and artists then trace motion over every frame, creating the final handdrawn animation one frame at a time. The more artistic liberty the artist decides to take during this process, the more laborious it becomes. Interestingly, this technique was recently used in the production of at least one feature-length film, Loving Vincent³, in which every single frame was painted by hand. As can be expected for such undertaking, it took over 100 artists many years to complete the process, and even partial automation could mean extreme savings in time and resources. With the spread of digital over-painting, this technique is becoming more popular, for example in the animated series, Undone⁴, which still required enormous human effort too. Both works have recognizable style and received warm reception, indicating that hand-drawn style is very appealing to the public audiences and can add another layer of emotion into the material. Crucially, it has been difficult to define efficient professional pipelines that are suitable for creating

¹DeepArt: https://www.deeparteffects.com/

²Prisma: https://prisma-ai.com

³Loving Vincent: http://lovingvincent.com

⁴Undone: https://www.imdb.com/title/tt8101850

unique-looking products. Consequently, the few pipelines in use produce interchangeable visuals, significantly limiting the ability of their users to incorporate their artistic expression. Appropriately applied style transfer techniques (e.g., [Jamriška et al. 2019]) could not only make the production process significantly less time-consuming, freeing artists' time for more creative tasks, but also allow far more productions to create unique-looking results. However, propagating the style automatically in the general case is much harder than in the 3D case, since we have no convenient and precise knowledge of the geometry in the scene.

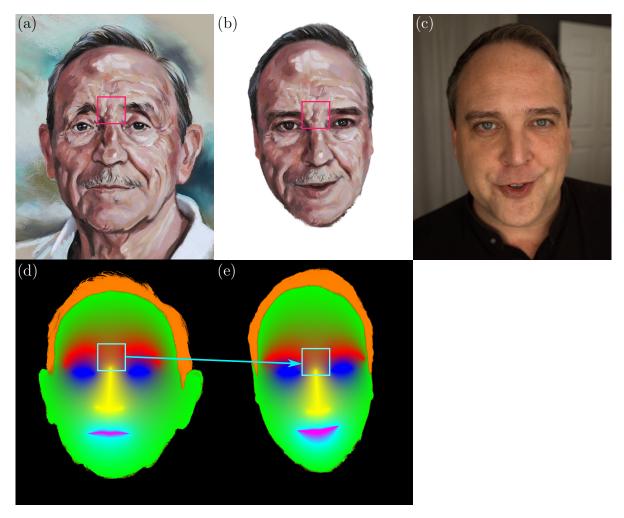


Figure 1.4: Patch-based synthesis. We aim to transfer the painterly style of (a) onto a photograph of a person (c) by copying image chunks to produce an analogous image (b). To define the patch similarity, guidance channels need to be designed (d, e), in this case a version of distance field based on semantic segmentation. If the teal patches drawn over images (d) and (e) are nearest neighbors, we would expect the red patch in (b) to be copied from the corresponding location in (a).

The process of video style transfer also presents a unique challenge in terms of ensuring temporal consistency, as it requires that neighboring frames be visually seamless with no perceptual flicker or inconsistencies. Achieving this can be difficult, as many artistic styles lose their appeal when applied to realistic motion, especially when the goal is to significantly alter the geometry of objects or exchange materials with vastly different physical properties, such as their interaction with light or their rigidity. As a result, it is necessary to exercise special care when designing video style transfer algorithms to address these issues. Even state-of-the-art techniques struggle with striking a balance between maintaining faithfulness to the input style and preserving natural object transformations during camera movement or as objects move in the scene, and it remains an important open problem.

1.1.1 Approaches to Style Transfer

The process of extracting the style information from the style exemplar and applying it to the desired target image varies greatly between different methods, but we can roughly group style transfer methods into three broad categories – (1) Procedural style transfer, (2) Non-parametric guided or patch-based style transfer and (3) Parametric or neural style transfer. Although this classification is based on the fundamental implementation ideas of the methods, it has become increasingly apparent that they each have unique properties, strengths and weaknesses, which translates into distinct and recognizable visual results. To a lesser extent the distinction is also influenced by historical factors, though there have been methods which attempt to combine the different approaches [Futschik et al. 2019; Texler et al. 2020a;b], and this hybrid approach shows promise as a potential direction for future development.



Figure 1.5: Early method implementing artistic painterly stylization [Hertzmann 1998]. A source image (a) depicts the content to be transformed. Image (b) is the result obtained by algorithmically applying small radius brushes over the source image (a). The brushes are applied in multiple layers; large brushes are applied first, medium and small brushed follow, thus creating an illusion of a real painting.

Early attempts at automatic style transfer were based on procedurally compositing the target style from a set of user-defined elements, e.g., brush-strokes and pens [Hertzmann et al. 2001; Bénard et al. 2010; Bousseau et al. 2006; Praun et al. 2001; Salisbury et al. 1997; Lu et al. 2013] or filtering kernels created through physical simulation [Curtis et al. 1997]. Example of an early style transfer result is shown in Fig. 1.5, a photograph composed exclusively from different brush strokes, while retaining the overall high level structure. More advanced approach based on the same underlying principles can be found in Fig. 1.6, where the strokes are no longer static, but given as exemplars and the algorithm's job is to use them to synthesize the final result.

An important milestone in digital style transfer was presented in *Image Analo*gies [Hertzmann et al. 2001]; an example-based approach where no specific, predefined

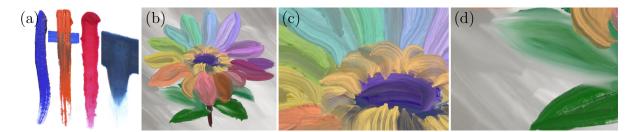


Figure 1.6: Result of Lu et al. [2013]. Brush stroke exemplars (a) are used to synthesize a painting (b). The foreground flower strokes (close-up c), use oil paint exemplars(a-left), while the background strokes (close-up d), use plasticine exemplars (a-right). The low level image features imitate real artistic media well.



Figure 1.7: Image Analogies [Hertzmann et al. 2001]. An unfiltered image A together with its filtered-stylized version A' define a desired transformation. The goal is to perform the same transformation on an unfiltered image B to obtain its filtered version B'. Besides A, A', and B, there is no other input to the framework, the transformation is learned from the pair of A and A'.

domain is assumed, and the method only expects an example of stylization – original image and its stylized version. The framework is then able to stylize other images in the same way as in the given stylization exemplar. Fig. 1.7 shows a typical application of this framework. Image A is the original exemplar and image A' is its stylized counterpart, and these two images define the transformation. The task is then to apply this transformation to another image B in order to get image B' stylized in the same way the image A' was changed. While this work was foundational and inspired many others, it suffers from several shortcomings, namely images A and A' need to be very carefully aligned, and the transformation function is only inferred from neighboring pixel locations, making any high level analogies impossible. Lastly, image B is expected to be from the same domain distribution as A (e.g. two photographs taken under roughly the same conditions, or both being a grayscale mask, or segmentation maps), an intuitive yet sometimes restrictive requirement.

Despite these limitations, the approach and its modifications can be used with great deal of success in some areas, such as video stylization with an overpaint image [Jamriška et al. 2019]. Methods based on fundamentals originating from the work of Hertzmann et al. [2001] are often called patch-based synthesis, as they principally function by copying patches of style source image into the result according to some metric. An advanced example of usage is shown in Fig. 1.4 which defines an analogy between the guidance domain (constructed from automatically extracted segmentation map) and the painterly artistic image. The required alignment between images (a) and (d) in this case is trivially satisfied, but limits the possible designs of the guidance channels.



Figure 1.8: An example of neural network based approach to the style transfer task [Gatys et al. 2016]. Resulting image (b) depicts the same semantic content as source photograph (a) and at all scales appears visually similar to the style exemplar (c).

The boom of deep learning research has opened up new directions for artistic stylization, and has in fact gained significant attention from both the academic and public spheres. With the emergence of novel methods that generate complex images previously only attributed to human artists, the subfield has seen a significant surge in popularity. In 2016, Gatys et al. [2016] proposed a novel approach to style transfer using deep learning models. While this approach can work well in limited cases, the abstraction of artistic style into a set of parametric statistics is oversimplifying and difficult to reason about, sometimes resulting in stylization results that are very far from the exemplars without any explanation. Compared to state-of-the-art patch-based methods, the results are of lower visual quality, suffer from blurriness and lack fine artistic details (see Fig. 1.8). Nonetheless, the work of Gatys et al. opened a completely new path in the field of style transfer and inspired many other researchers to employ deep learning models. Methods that rely on learning the probability distribution of the style exemplar instead of directly reusing it are referred to as neural-based methods.

A particularly promising subsection of neural-based methods revolves around the recent progress achieved in the field of text-driven generative models like Stable Diffusion [Rombach et al. 2021]. Not only do these new models significantly improve the visual fidelity of images produced by deep learning models, no longer producing the subpar image quality compared to patch based methods, they also open up the new possibilities of specifying or narrowing down the style component through textual input as well as image based one, which can include use cases such as describing a style for which there is no exemplar or using only certain features of a given exemplar. And although the legal story of computer generated images is far from decided, it is already clear that these models are a real game-changer to the mass production of artistic imagery, including stylization tasks. Further discussion about related work can be found in Chapter 2.



Figure 1.9: General example-based style transfer. The task is to transfer artistic style from the given style exemplar (a) to the given target (b) while preserving the appearance of (a) and the content of (b). Example of a possible result in (c).

1.2 Our contributions and structure of the thesis

In our view, artist-facing algorithms must consider several critical aspects, including freedom of expression, level of control over the output, and interactive response. When it comes to artistic stylization, two additional factors are particularly important. First, the algorithm should aim to faithfully replicate the original style of the artist, ensuring that visual similarities between the exemplar and output images are clear. Second, the algorithm should demonstrate semantic awareness in its transfer function. For instance, if the transfer is between a portrait painting and a portrait photograph, to synthesize eyes or hair the algorithm should construct image features that match the same region in the exemplar image. Overall, considering these factors is essential for developing effective artist-facing algorithms that can support the creative process while achieving high-quality results. Each of our contributions attempt to address these points and come up with a reasonable trade-off where necessary.

The primary objective of this thesis is to describe innovative algorithms and methods we developed which help artists, as well as casual users, create richer, more personalized content, help experiment effectively, and save time by automating repeated tasks. We further strived to enable experiences closer to hand-drawn workflow in applications where it was not previously possible, while focusing mainly on *example-based style transfer*, the task where an example of the artistic style is given in the form of a digital artwork. A smaller amount of attention is also given to photorealistic stylization for faces. Following is a brief description of our contributions to the field of style transfer and stylization, more in depth description and assessment of each method is then given in respective chapters.

In the previous section, we placed no restrictions on the kinds of images that can be used together as target and exemplar. However, this generality is not necessary to reach goals considered useful. In fact, in many cases, it is beneficial to incorporate some domain knowledge and limit the set of possible inputs – for example we can impose a limitation requiring the style exemplar and content image to have similar semantics, allowing us to increase the robustness and quality of the algorithms. For instance, we can restrict ourselves to images of human faces and expect results similar to Fig. 1.10. In such scenarios, we can typically implement methods that fare far better than general style transfer, by avoiding having to deal with corner cases or using simplified reasoning.

In Chapter 3 we present our proposed solution to the idea of real-time, high quality facial stylization: FacestyleGAN: Real-Time Patch-Based Stylization of Portraits Us-

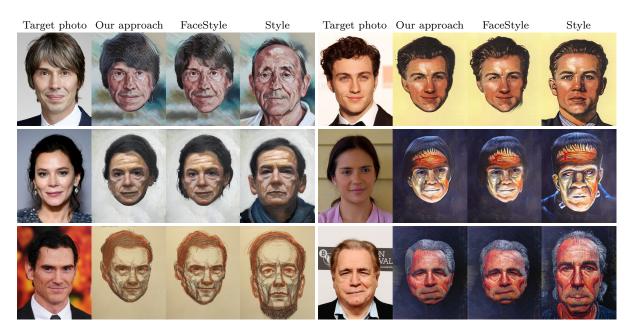


Figure 1.10: Stylization of face photographs using our approach as compared to FaceStyle [Fišer et al. 2017]. Our approach achieves similar quality while retaining more identity defining features, better generalization and runs 50 times faster.

ing Generative Adversarial Network [Futschik et al. 2019], which we achieve thanks to a marriage of patch-based example synthesis and neural-based machine learning style transfer. First, we use the algorithm of Fišer et al. [2017] to generate a large dataset of suitably stylized examples, and subsequently we apply an Image-to-image [Isola et al. 2017] machine learning pipeline to distill the dataset into a neural network model and generalize to unseen faces. Interestingly, our model learns to correct for certain common problems produced in the original method, at times even greatly improving the quality of the baseline style transfer.

However, for the cases where we do want to perform arbitrary, no domain assumed style transfer, for example as can be seen in Fig. 1.9, where a style from a painting of landscape is transferred to the photograph of a horse. In this very unconstrained case, it is usually hard to establish meaningful semantic correspondences between the style exemplar and target content; thus it is difficult to guide style transfer method in order produce results matched to the artist's intention (for instance part of the blue phone booth curiously appearing on the horse, which may or may not be desirable). In Chapter 4 we introduce our contribution towards this goal, Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis [Texler et al. 2020a]. This approach also combines neural and patch-based style transfer methods. We designed a framework for existing neural techniques to provide us adequate stylization at the global image feature level, and then use their output as an intermediate stage for subsequent patch-based Through this combination, our method keeps the high frequencies of the synthesis. original artistic media by directly copying patches, thereby dramatically increasing the fidelity of the resulting stylized imagery, but also receives the opportunity to exploit the ability of neural methods to semantically guide the stylization without associated work of creating explicit guidance. Furthermore, we show how to stylize extremely large images (e.g., 340 Mpix) without the need to perform the synthesis at pixel level, yet retaining the original high-frequency details, which is a notoriously hard task for neural methods.



Figure 1.11: Given one keyframe (a) and a video sequence (in blue), our method, Interactive Video Stylization Using Few-Shot Patch-Based Training, produces the stylized results for the rest of the frames (b, c, d).

As described previously, extending style transfer into video applications can be quite tricky due to the necessity of preserving temporal stability. To this end, we offer two novel methods that contribute to the state-of-the-art in the field of appearance transfer from stylized keyframe, where meaningful correspondences can be established – and propagated to the rest of the sequence. The first of these methods is Interactive Video Stylization Using Few-Shot Patch-Based Training [Texler et al. 2020b], described in Chapter 5. In this work, we again make use of the neural image-to-image paradigm to enable artists to perform real time stylization of a short video sequence, example of such sequence is shown in Fig. 1.11. We achieve the quick time-to-frame by training the transfer model on small image patches rather than entire images. Crucially, the use of our method promotes artistic experimentation and answers the interactivity problem for video keyframe based stylization, which was previously an infeasible workflow. Even though the stylization model offers acceptable results in seconds or minutes, the quality continues to improve with more iterations.

Though the method has a tremendous number of desirable properties, it can fall short in some important regards, especially as far as quality of image and longer-term correspondence propagation is concerned. To address some of these issues, we proposed a more heavy-weight approach in STALP: Style Transfer with Auxiliary Limited Pairing [Futschik et al. 2021a], presented in Chapter 6. Previous state-of-the-art video stylization methods tend to only focus on extracting the transfer function from the provided aligned correspondence pair (keyframe and stylized counterpart). We note that, despite not knowing the stylized counterparts, remaining frames of the sequence also contain valuable information that can improve the transfer function when applied to them. In fact, this scenario is not limited to just videos, Fig. 1.12 shows a case where we consistently stylize a panorama photo using a single stylized constituent part as exemplar.

While we observe notable improvements in visual quality and in general can stylize longer sequences with smaller number of input keyframes, we are forced to relax the real-time requirements in order to do so. However, thanks to the previous work, we can envision a workflow that utilizes Interactive Video Stylization Using Few-Shot Patch-Based Training during the experimental phase and then switches to STALP once the artist is satisfied with the broad features of her work.

Lastly, we take a look at a different angle to artistic stylization. Our previous work mostly deals with painterly or artistic media examples, thereby providing tools for inherently non-photorealistic rendering. In the last presented tool, we empower artists to

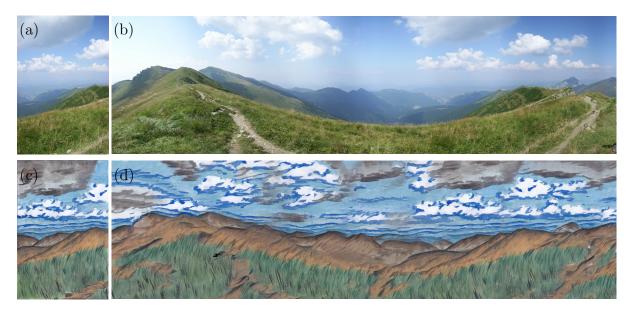


Figure 1.12: Stylizing using a limited pairing: (a) and (c) express the desired transformation, in STALP we train for style consistency across multiple input images, in this case the constituent photographs used to create (b). The result is a consistent stylization that can be seamlessly stitched together into a panorama image (d)

photorealistically alter, or stylize, real images. In our work, ChunkyGAN: Real Image Inversion via Segments, described in Chapter 7, we project real photographs into the latent space of a generative model and subsequently perform arithmetic operations inside the latent space to create plausible local modifications to the input image and use image space blending to reconstitute the entire image. We show the full power of our framework on facial images, using the pretrained network of StyleGAN v2 [Karras et al. 2020]. Using this framework, we are able to change a range of facial features, such as gaze direction, turning neutral pose into a smile or opening closed eyes, while retaining very close identity resemblance of the original subject and avoiding unnatural effects or artifacts. These local feature changes can be layered on top of one another, such as shown in Fig. 1.13.



Figure 1.13: Using ChunkyGAN allows artists to produce local layered edits, applied in sequence on a real photograph (a): changing gaze direction (b), adding smile (c), changing haircut and nose shape (d).

Besides, Chapter 8 gives a summary overview of the work and proposes possible directions for future work, Appendix A lists author's publications along with the full bibliography and a list of citations in other publications. Appendix B gives an overview about author's contributions to the individual papers used in this thesis. The remaining appendices contain supplementary material for their respective chapters – Appendix C shows additional results for Chapter 3, Appendix D contains supporting material for Chapter 5 and Appendix E consolidates further results and experiments for Chapter 7.

Chapter 2

Related Work and State-of-the-Art

Chapter 1 already touched on some important concepts and seminal works regarding stylization and example-based style transfer. In this chapter we describe different style transfer directions and recent developments in the field, the order in which we present the selected works is loosely chronological, though for the sake of coherency, there are exceptions. The earliest methods were largely based on procedurally compositing the result from a set of hand-crafted static elements (typically brush strokes). From there, state-of-the-art moved towards patch-based methods that seek to find suitable coherent image patches to copy from the style exemplar into the resulting image. As learning-based parametric or *neural* approaches took off with impressive results, we start seeing a decline in other directions. Lastly, we introduce emerging techniques, such as combinations of patch-and-neural based approaches or dataset-driven approaches.

2.1 Procedural Methods

The first and simplest methods to perform what could be classified as style transfer were akin to automatic painting with a preselected range of brushes and pens, or hand-designed kernel filters applied as post-processes. Some methods choose to explicitly simulate artistic media, others use brushes defined by images or analytic equations, and the resulting image would be a composition of many layers of such brush stamps, optionally with affine transformations applied to them. The composition itself would be performed according to the leading image processing principles of the time, along gradient direction or with edge detecting algorithms. Some devised algorithms include [Hertzmann 1998] for painterly style using kernel filters, [Salisbury et al. 1997] for pen and ink illustrations using textures, [Breslav et al. 2007] to use hatching for shading images, [Baxter et al. 2004] and [DiVerdi et al. 2010] to produce images resembling oil paint, or [Bousseau et al. 2006] and [Chu and Tai 2005] for watercolor simulation. Curiously, similar approaches could also be applied to 3D images with known depth, allowing for more complex decisions, such as automatic implicit layering [Schmid et al. 2011]. Later on, proposed algorithms would work with assumptions of exemplar brush textures rather than static ones, such as the work of Lu et al. [2013] (see Fig. 1.6) or Zheng et al. [2017]. The main strength of these methods is their simplicity – there are no complex parameters to tune, and their behavior is predictable, plus they can deliver impressive results when used correctly. However, setting up these methods is laborious, the results are quite limited

by the expectations adopted by the authors of how composition should work and are carefully tailored to the specific media or use-case they target, with little hope of easily extending them to other applications.

2.2 Image Exemplar Based Methods

Consequently, to address the limited nature of media-specific methods, following research directions shifted towards exemplar artwork based methods that allow usage of a more feature-rich guide – a full exemplar image. Within the provided image, there are local neighborhoods where the composition of brushes or other medium-level image features has already taken place, and we can attempt to reproduce it by directly copying pixels without having to deal with the problems of composition at all, which is a key advantage of such approach. However, it also means that decomposition back into e.g. individual strokes is nearly impossible, and thus we are limited to only the local features already present in the example image, without the possibility of synthesizing completely new content, even small deviations from content shown in the exemplar are difficult in practice.

Image Analogies by Hertzmann et al. [2001] was seminal work that defined a framework allowing for arbitrary style transfer using patch-based synthesis (see Fig. 1.7). Though the framework is still widely used today, competing paradigms also emerged, such as The Lit Sphere by [Sloan et al. 2001], which is a framework based on texture mapping and taking advantage of provided 3D geometry.

2.2.1 Non-parametric guided synthesis

The Image Analogies framework and its later improvements [Wexler et al. 2007; Kaspar et al. 2015; Fišer et al. 2016] rely on a guided process we that is also called patch-based synthesis. To perform transfer of style from one image to another, the methods directly copy *patches*, or *chunks* of the exemplar image into the resulting image. This process by construction ensures that the high frequency content from style image, while possibly scrambled, remains largely intact and recognizable. The area to be copied is optimized using a patch similarity metric function, commonly sum of squared errors, which is effectively user-defined by providing a set of guidance images or channels. The guidance channels spatially encode which patches between the images should be considered similar for the purposes of the optimization. The approach has been adopted for scenarios, such as fluid animation [Jamriška et al. 2015], 3D renders [Fišer et al. 2016; Sýkora et al. 2019], facial animations and image style transfer [Fišer et al. 2017; Bénard et al. 2013], or video style transfer [Fišer et al. 2014; Jamriška et al. 2019; Dvorožňák et al. 2018] with explicit temporal coherence.

Results produced by the framework are often of impressive visual quality and stay truthful to the original exemplar, and have trivially explainable behavior. On the other hand, these methods share one critical drawback – it is relatively labor-intensive and conceptually difficult to prepare tailored guidance channels. Some research effort was directed into deriving these channels automatically in restricted scenarios, through procedural analysis of the used images [Frigo et al. 2016; Fišer et al. 2017], for example by supposing that both the style exemplar and target content image are portrait images. The downside of this approach is that it is often non-trivial to design guidance that works well, even in the domain restricted case.

The patch-based approach works especially well when used on exemplars with ample high frequency content, where small patch seams are easy to hide. The sizes of copied chunks depend on the design of the guidance channels, and generally the aim is to keep them as large as possible. Conversely, when the image features are more structured and defined, it becomes harder to mosaic large patches into a new shape that preserves those essential characteristics, and visual quality degrades.

2.2.2 Parametric synthesis

The machine learning revolution spawned a completely different approach to style transfer and artistic image synthesis. Gatys et al. [2016] pioneered use of neural networks to replace the explicit guidance required by methods based on Image Analogies. The key advantage of neural methods is addressing the synthesis of completely new content. Unlike patch-based methods, the parametric nature of the synthesis process allows for emergence of content not explicitly shown by the style exemplar or perhaps not even found in the training dataset of the model. The other major advantage is the ability to automatically synthesize result based on perceived semantics and taking global context into account, which is notoriously hard with explicit guidance. Whereas patch-based methods would typically only look at small local pixel neighborhoods (possibly on multiple scales), neural models have comparatively huge receptive fields, and it is not unusual that they could span the entire input image. On the other hand, a major drawback common to most parametric techniques is that they are generally unable to exactly preserve high-frequency details of the style exemplar, producing blurry features and 'washed-out' look.

The state-of-the-art in parametric style transfer can be broadly split into two categories: optimization based methods and feed-forward networks. Feed-forward models, once trained, work in a look-once fashion, where the input image is fed through the network and the stylized result comes out at the end of that process. Optimization based approaches, on the other hand, extract some state information from the inputs once, and then carry out a number of optimization steps over some form of objective function. The optimization algorithm is commonly a variant of stochastic gradient descent, since backpropagation through the large feature-extracting model is required.

Both categories have pros and cons, perhaps the most obvious trade-off is compute time versus quality – feed-forward systems tend to be blazing fast, suitable for real-time applications but suffer from noticeable artifacts. Optimization-based methods are able to fix a lot of the failure cases presented by feed-forward networks; this is compensated by being far less interactive, requiring on the order of seconds or minutes per image. Indeed, it is possible to generate large datasets of images using an optimization method and train a feed-forward network to approximate the process, but such attempts seem to be subject to the trade-off nevertheless.

Lastly, a beneficial property of neural-based approaches is the natural extension to video, especially for feed-forward architectures, although explicit temporal coherence is required, since out-of-the-box solutions applied on a per-frame basis are prone to flickering or produce inconsistent results and otherwise unstable sequences. Some works attempt to include terms for temporal stability in the optimization criterion [Chen et al.

2017a; Gupta et al. 2017; Sanakoyeu et al. 2018; Ruder et al. 2018], while Blind Video Temporal Consistency approach of Lai et al. [2018] takes per-frame stylized video as input and outputs a temporally consistent video, which makes it invaluable as a post-processing step.

2.2.2.1 Optimization based methods

The optimization based approaches are predicated on using a pretrained feature extractor to reduce the input images into a set of feature maps, from which an initial state is produced. Then, the result is materialized by iteratively optimizing a loss function computed on the feature maps of the intermediate state, e.g. comparing statistical moments of the feature maps between the result and the style or target content features. Optimizing the image requires up to several backward passes through the feature extractor per iteration, and to get reasonable results, several hundred iterations are needed. This process is compute intensive, and worse, typically requires large amounts of memory that scales exponentially with the desired resolution of the images. At the same time, it is not clear how to meaningfully control these methods, and the results can be quite unpredictable. Initial conditions and optimization schedules are also tricky to tune properly. However, in many cases these methods produce appealing images, and are relatively easy to operate once a working setting is discovered.

Gatys et al. [2016] use pre-trained convolutional neural network (VGG-19 [Simonyan and Zisserman 2014]), trained on image classification task, for feature extraction from both the style exemplar and the target content images. Then, an optimization process matches the feature-wise statistics of both images to synthesize the result. Statistics related to style are extracted from different layers than content and are subject to a different optimization objective. Gatys et al. further make the observation that trained neural networks implicitly encode style information as correlation between channels of feature layers. The authors also further extended their idea to allow control over spatial location, color information, and scale of features [Gatys et al. 2017], in an attempt to mimic the explicit guidance offered by patch-based methods.

Gu et al. [2018] explore the possibility of reshuffling spatial locations of feature maps extracted from the style exemplar to form a structure closer to the content image. This method trivially minimizes the style loss proposed by Gatys et al. [2016], and they further observe that constraining the usage of the same feature patch and promoting feature diversity leads to improved results, an idea previously exploited in patch-based approaches. The optimization objective is then to invert these reshuffled feature maps back into an image. Despite directly using the known exemplar features, the inversion is not a trivial task and the visual quality can vary depending on the exemplar used.

Kolkin et al. [Kolkin et al. 2019] also make use of pretrained VGG features, but reimagine which operations are suitable for feature matching, opting to use optimal transport formulation instead of direct correlation matching or reshuffling, and achieve results of much higher quality. The authors also propose using self-similarity as the content conserving part of the objective function, which has desirable properties, such as preserving symmetry im the target image.

2.2.2.2 Feed-forward methods

Some of the issues observed with optimization methods can be alleviated by designing a feed-forward process for stylization. Namely, feed-forward networks complete an image stylization task around two to three orders of magnitude faster, thus enabling real-time interactive scenarios.

Johnson et al. [2016] noted that the style loss component of Gatys et al. [2016] can be looped into an image-to-image [Isola et al. 2017] framework to train a special purpose network to stylize according to a particular style exemplar. While this requires a large up-front investment and comes with the downside of having to retrain for every desired style exemplar, the trained model can run at interactive frames per second. This general concept was later improved by others, Ulyanov et al. [2016a] significantly improved the visual quality while keeping the compute requirements low, Dumoulin et al. [2016] attempt to sidestep the problem of retraining for each style by training a unified model with an internal dictionary for multiple style exemplars, and Chen et al. [2017a] investigated which architectures are naturally a good fit for a fully convolutional, real-time scenarios.

Even though the dictionary approach works well for a limited number of styles, it does not scale very well. To overcome necessity for additional training, encoder–decoder scheme was suggested by several authors [Li et al. 2017; Huang and Belongie 2017; Lu et al. 2017; Kotovenko et al. 2019]. This approach expects the style exemplar to be fed into the network alongside the target content input, and the feed-forward network then needs to perform the data-dependent style analysis in its encoder part. While the architecture of the encoder often builds off of the convolutional layers of VGG [Simonyan and Zisserman 2014], and is used to represent both the style and content image as two sets of feature channels, the job of the decoder is subsequently to combine the extracted features into a singular representation and, ultimately, resulting image. This removes both requirements of explicit optimization and necessity of additional training for each style exemplar, but it comes at the price of overall image quality.

2.2.3 Hybrid: Parametric guidance for patch-based synthesis

Given the mutual exclusivity of strengths of patch-based and neural methods, a natural thought for research direction is to combine them. There have been multiple efforts in the area, and while the resulting algorithms tend to be more complex, in many cases they have been shown to deliver superior results to either approach alone. Li et al. [2016b] search local neighborhoods of feature-space patches of the given style image in hopes of finding structures similar to ones found in the content image, and are thus able to achieve better reproduction of local textures than neural approach alone. The seminal work in this direction, neural version of Image Analogies [Hertzmann et al. 2001], called Deep Image Analogy [Liao et al. 2017] combines the idea of copying patches from exemplar into the result, but doing so in the domain of deep neural features and subsequently decoding the obtained mosaic. Because the network has implicit bias of looking for semantic information rather than textural structures, the method works best when applied to a pair of images that depict semantically similar objects, but crucially retains the high frequency features present in the style exemplar.

In Chapter 3, we propose a contribution that falls into this category – our work uses patch-based method of Fišer et al. [2017] to generate a dataset of stylized portrait pairs and then train adversarial neural network to perform the style transfer. Our second contribution into this category is Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis [Texler et al. 2020a], further described in Chapter 4.

2.3 Dataset Based Methods

A slightly extended version of the image exemplar based methods are *dataset* based methods. These approaches, rather than being provided a single image exemplar, expect the user to provide a dataset of styles which should be used to synthesize new results. Principally, we could use the Image Analogies framework for this approach by trivially creating an image mosaic of multiple images from the dataset merged into a single exemplar. However, such approach discards instance level information which can be leveraged for better results and would be computationally infeasible with larger datasets.

This area has become particularly interest in recent years, with many influential works published. CycleGAN [Zhu et al. 2017c] attempts to match the style between two dataset domains (e.g. stylized and photograph) by ensuring cycle consistency. In StarGAN [Choi et al. 2018] this idea is extended by changing it to multiple datasets and ensuring cycle consistency between each pair of them (e.g. many different styles). Contrastive Learning for Unpaired Image-to-Image Translation [Park et al. 2020] extracts patch-based statistics from patches across many instances and matches them together, thus producing a consistent stylization across the entire dataset. Interestingly, the large scale learning methods enable modes of style transfer which are otherwise very difficult, such as translating edge images into more photorealistic images or segmentations into paintings, as shown by Zhan et al. [2022]. The emergent properties of large models therefore seem to be relevant for stylization.

This large-data driven approach is quickly becoming popular with the advent of textconditioned generative models like Stable Diffusion [Rombach et al. 2021]. Channeling the exceptional power of general purpose generative models into stylization through limited fine-tuning of the models, shows very early but promising results, for example as shown by Dreambooth [Ruiz et al. 2022], but this is still an area under active exploration.

Chapter 3

FacestyleGAN: Real-Time Patch-Based Stylization of Portraits Using Generative Adversarial Network

3.1 Introduction

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



style exemplar target our approach Fišer et al. Liao et al. Selim et al. Gatys et al.

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. [Fišer et al. 2017], Liao et al. [Liao et al. 2017], Selim et al. [Selim et al. 2016], and Gatys et al. [Gatys et al. 2016]) while being able to run at interactive frame rates on a consumer GPU. Style exemplar: © Scary Zara Mary.

may be a source of potential failure (see Figures 3.5, 3.6, and 3.7, two right columns) and introduces algorithmic complexity which leads to huge computational overhead.

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.

In this paper, we demonstrate the benefits of combining state-of-the-art high-quality patch-based synthesis with the power of image-to-image translation networks. Thanks to the ability of patch-based method of Fišer et al. [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 patchbased 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.

3.2 Related Work

The stylization of head portraits is a long-standing challenge for non-photorealistic rendering (NPR) 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. 2002a;b; 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* [Wang and Tang 2009]), and then use multi-scale Markov Random Fields [Wang and Tang 2009; Li et al. 2011; Zhou et al. 2012; Wang et al. 2013a; 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 example-based approach can also be reduced to the level of individual brush strokes [Zhao and Zhu 2011; Berger et al. 2013; Wang et al. 2013b]. Although these techniques are compelling at delivering particular artistic looks (e.g., oil paint), they are difficult to apply on styles where the interaction between individual brush strokes cannot be modeled merely by blending operations.

Recently, neural network based style transfer becomes very popular thanks to the seminal work of Gatys et al. [2016]. The success of this method 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. Recently, patch-based techniques [Fišer et al. 2017; Lu et al. 2018] have been proposed that try to address this issue. Nevertheless, these require additional guiding channels to be prepared, which govern the synthesis process to transfer patches in a semantically meaningful way between the style exemplary and the target photo. Although such channels can be created automatically via a series of algorithmic detectors, this solution makes the system more fragile as an occasional failure of any individual unit may significantly affect the whole synthesis.

Li et al. [2016a] introduce a combination of neural- and patch-based synthesis. Their key idea is to use responses of a deep neural network trained on image classification [Simonyan and Zisserman 2014] to establish patch-wise correspondences between the style exemplar and the target 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] propose 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 in the image domain and preserve important pixel-level details.

Our approach bears a resemblance to techniques which can quickly perform certain image editing operations for which time-consuming algorithmic solutions exists [Xu et al. 2015; Chen et al. 2017b]. By training a feed-forward network on a pre-computed dataset they can achieve significant speed up as well as a level of generalization that sometimes outperforms quality of results produced by the original algorithm. A similar technique was also used in the context of neural-based style transfer by Johnson et al. [2016]. In their approach, the output from Gatys et al.'s algorithm [2016] was used to train the weights of a feed-forward neural network. However, as Gatys et al.'s method does not perform semantically meaningful transfer the 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. Recently, there were attempts to generalize neural-based stylization [Li et al. 2017; Huang and Belongie 2017] so that costly training nor optimization is required to perform fast style transfer from arbitrary exemplar. Nevertheless, those techniques are unable to perform semantically meaningful transfer and still suffer from visible pixel-level artifacts which decrease their ability to reproduce important visual characteristics of used artistic media.

3.3 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 $F : \mathbb{X} \to \mathbb{Y}$.

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.

3.3.1 Training Objective

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

$$\mathcal{L}_{GAN}(F, D_Y, \mathbb{X}, \mathbb{Y}) = \mathbb{E}_{y \sim p_{data}(y)} \left[\left(D_Y(y) - 1 \right)^2 \right] \\ + \mathbb{E}_{x \sim p_{data}(x)} \left[\left(D_Y(F(x))^2 \right) \right]$$
(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:

$$\mathcal{L}_{1}(F) = \mathbb{E}_{X, Y \sim p(X, Y)} ||Y - F(X)||_{1}$$
(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:

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

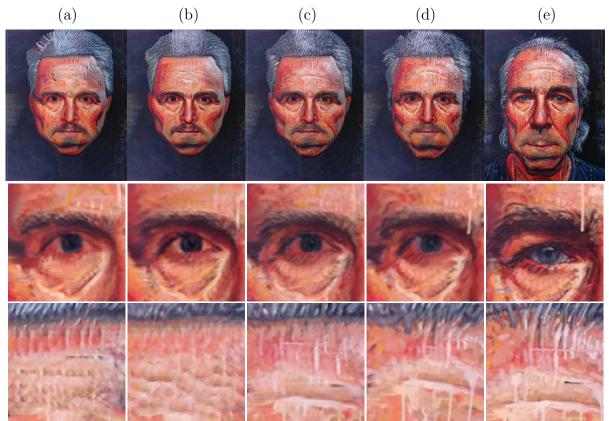


Figure 3.2: Ablation study. A demonstration of visual quality improvement achieved using modified VGG loss and our improved network architecture: (a) result of our network trained without using VGG loss, (b) result generated using all losses, however, without our improved network architecture, i.e., using the original architecture of Johnson et al. [2016], (c) our result, (d) result generated using FaceStyle algorithm [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).

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

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

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

3.3.2 Network Architecture

For our generator model we use the initial architecture from [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 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).

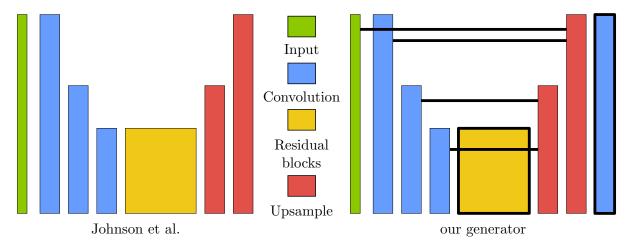


Figure 3.3: The original generator network architecture of Johnson et al. [2016] (left) followed by our improved architecture (right). Modifications are denoted with black color: added skip connections, increased the number of residual blocks, two upsampling layers are followed by additional transposed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

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. [2016a] and leaky ReLUs as activation. This helped us to lower the number of parameters and achieve a more stable gradient propagation.

3.3.3 Implementation Details

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

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

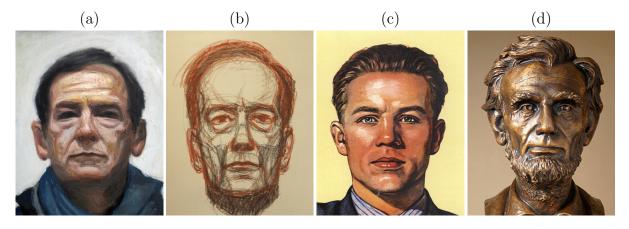


Figure 3.4: Exemplars of styles used in Figures 3.6, 3.7, and 3.8. See Figures 7.1, 3.2, and 3.9 for the remaining style exemplars. Style exemplars: $(a-b) \bigcirc$ Adrian Morgan, (c) Viktor Ivanovich Govorkov, $(d) \bigcirc$ Will Murray.

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.

3.4 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 highquality 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-ofthe-art.

3.4.1 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 highquality style transfer on live video streams (please refer to our supplementary video).



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 7.1 and 3.2.

We can downsize our network architecture to 256×256 resolution (along with reducing the number of filters in each layer) and also achieve interactive response on mobile devices without significant loss of visual quality.

3.4.2 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. 2017a; 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 3.6: 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.

3.4.3 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.

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



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.

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.

3.4.4 Comparisons

We compared the visual quality of our approach with current state-of-the-art in imageto-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

3.4. RESULTS



Figure 3.8: Comparisons of our approach with current state-of-the-art in image-to-image transation: 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.



style exemplar our approach Fišer et al. Liao et al. Selim et al. Gatys et al.

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.

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 7.1 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.

3.5 Limitations and Future work

We demonstrate that our approach brings comparable or even better visual quality within significantly lower computational overhead when compared to the current state-of-theart. However, there are still some limitations that can encourage future work.

One of the critical challenges is the accuracy and smoothness of head and hair segmentation masks. Although our method often outperforms FaceStyle algorithm 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 (see Figures 3.5, 3.6, and 3.7) 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.

3.6 Conclusion

We present a novel approach for example-based stylization of facial images. Our key idea is to combine a state-of-the-art patch-based synthesis algorithm with a new variant of conditional generative adversarial network. Such a fusion allows us to reach an equilibrium that retains or even improves the visual quality of results produced by the original patch-based approach while increasing its robustness. We compared our combined technique with current state-of-the-art in example-based image stylization as well as in learning-based image-to-image translation methods and reported a considerable quality improvement in both domains. Thanks to the ability to upload our trained generative network into a consumer graphics card we can present the first real-time by-example stylization engine that reaches the visual quality of state-of-the-art techniques tailored to offline processing.

Chapter 4

Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis

4.1 Introduction

In recent years, advances in neural style transfer and guided patch-based synthesis made the field of computer-assisted stylization very popular. Various publicly available software solutions (see, e.g., Prisma [Johnson et al. 2016], DeepArt [Gatys et al. 2016], StyLit [Fišer et al. 2016], FaceStyle [Fišer et al. 2017]) successfully brought the style transfer concepts to consumers. These applications enjoy popularity among casual users due to their novelty factors. However, they are not addressing the needs of professional users who demand high-resolution, high-quality output accurately preserving the textural details of the original artistic exemplar.

Though guided patch-based synthesis approaches [Fišer et al. 2016; 2017] can meticulously preserve fine-grained details, they require preparation of guidance channels. These guidance channels are important for establishing meaningful correspondences between the target image and the source style exemplar. Previous work designed guidance channels for specific use cases such as faces [Fišer et al. 2017], but designing meaningful guidance automatically in general case remains a difficult problem. On the other hand, neuralbased style transfer [Gatys et al. 2016; Gu et al. 2018] does not require explicit guidance to produce good stylization effects at a global level. Nevertheless, due to its convolutional nature, it usually fails to preserve low-level details such as brush strokes or canvas structure that are important to retain the fidelity of the underlying artistic media.

Neural techniques are also limited to work at lower resolutions (typically below 1K), which does not suit the need for FullHD, 4K or higher resolution used in real production settings. A similar limitation also holds for guided patch-based synthesis where the processing time grows significantly with increasing output resolution. Neural style transfer algorithms also have the problem of exhausting GPU memories where going beyond 4K resolution becomes impossible under current hardware constraints.

In this paper, we propose a straightforward approach which overcomes the aforementioned limitations by combining neural style transfer, patch-based synthesis, and dense correspondence field upscale. We first apply neural style transfer to obtain semantically



Figure 4.1: An example of stylizing an extremely high-resolution image using our proposed method: (a) style exemplar of $26400 \times 13100 \text{ px}$, (b) content image of the same resolution, (c) low resolution result of [Gatys et al. 2016] enhanced and enlarged by our method to the mentioned resolution. To the right, zoom-in patches of different parts of (c) up to zoom of $128 \times$ are shown; see all the individual brush strokes and its sharp boundaries. Also, notice how the structure of the original canvas and little cracks of the painting are preserved.

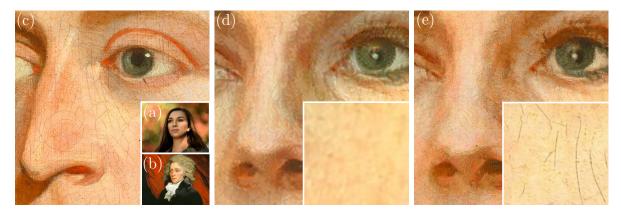


Figure 4.2: An example of enhancing the result of neural-based approach using our method: (a) target photograph, (b) style exemplar of the same size, (c) $6 \times$ zoom-in to the style exemplar, (d) the output of neural-based method DeepArt [Gatys et al. 2016] is capable to perform convincing stylization; nevertheless, the image contains artifacts caused by the parametric nature of the used neural network. High-frequency details like the structure of strokes and canvas are largely lost, sacrificing the visual quality of the original artistic medium. In contrast, our method (e) brings significant quality improvement, it restores the individual brush strokes and boundaries between them faithfully, the result better reproduces the used artistic medium as well as canvas' structure. Note how the cracks of the original artwork are preserved; although zoom-in patches are shown, we encourage the reader to zoom-in even further.

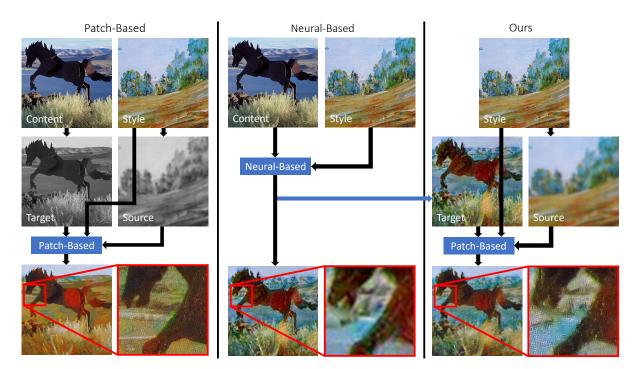
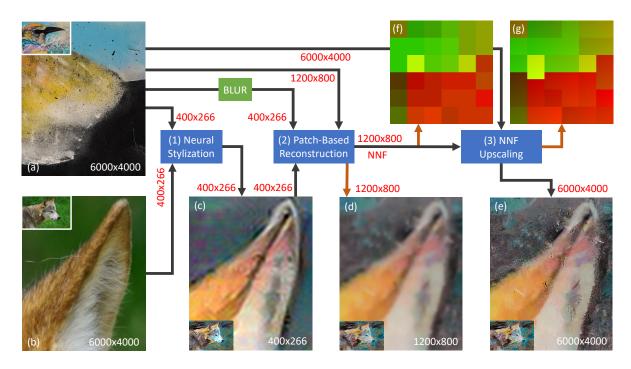


Figure 4.3: Simplified scheme of a patch-based, neural-based, and our hybrid style transfer method: The left column shows a patch-based approach [Fišer et al. 2016] with guidance based on blurred grayscale images as proposed in the original Image Analogies method [Hertzmann et al. 2001]. The resulting image has high texture quality and preserves artistic attributes and canvas structure well; however, the result does not properly respect the content semantics, causing water to become brown. The middle column shows a neural-based approach [Gatys et al. 2016], no guidance channels are needed and global style properties and image semantic are preserved well. However, the resulting image lacks high-frequency details of the original style exemplar, contains artifacts, and colors that are not present in the original style. The right column represents our method where low-resolution neural transfer result is used as a guidance channel for patch-based style transfer. Our result attenuates the neural artifacts and restores the original color and texture of the style exemplar.

meaningful stylization at a global level without the need of user intervention, and then use patch-based synthesis to remove low-level artifacts and restore the color and fine details to retain the fidelity of the original style, see Fig. 4.2. To significantly reduce computational overhead instead of running patch-based synthesis on the full resolution, we only upscale the dense correspondence field computed at a lower resolution level. We demonstrate that such a simple upscaling step can be performed quickly while still providing comparable visual quality as the full-fledged synthesis. This enables us to achieve high-quality stylization of extremely large images (see Fig. 4.1 where an image of 346Mpix is stylized). Our approach is generalized and can utilize any existing neural stylization method. We demonstrate this generality on a variant of our style transfer algorithm that directly uses the response of a neural network as a guide for patch-based synthesis. We developed a prototype of our method in the form of a Photoshop plug-in and put it into the hands of professional artists.



4.2 Related Work

Figure 4.4: Proposed pipeline: (a) style exemplar and (b) content image are both subsampled α -times and processed by a neural-based style transfer method (Sec. 4.3.1) which results in low resolution image (c) where fine details are missing and artifacts are apparent (see green and purple checkerboard artifacts). Next, low resolution result (c) from the previous step, style image (a) in the same resolution as (c), and β -times subsampled style image (a) are used as an input to a patch-based synthesis algorithm (Sec. 4.3.2) which outputs dense nearest neighbor field (NNF) (f) from which the corresponding image (d) can be produced using voting step [Wexler et al. 2007]. Finally, in NNF upscaling step (Sec. 4.3.3) the low-resolution NNF (f) is upscaled β -times to the original resolution (g). Patch coordinates in NNF (f) and (g) are encoded as red and green color levels. Note subtle color gradients in (f), which indicate the presence of fine patch coordinates in upscaled NNF that points to the patches in the original resolution style exemplar (a). Given the upscaled NNF (g) and the style exemplar in its original resolution (a), high-resolution, and a perfectly sharp final result is created using voting step (e).

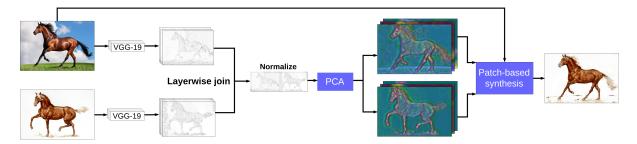


Figure 4.5: An overview of our VGG-guided style transfer pipeline: we start with a target image and a style exemplar, extract their VGG-19 features, normalize them, reduce their dimensionality using PCA, and use these as guidance for subsequent patch-based synthesis. Even though the proposed pipeline is straightforward, it yields convincing output.

One of the key tasks of non-photorealistic rendering [Kyprianidis et al. 2013] is to deliver stylized depictions of photos or synthetic scenes which preserve high-level information captured in the scene while on a detail level the resulting image resembles the artistic look.

Stroke-based approaches were one of the first techniques that enabled generation of stylized imagery. Rotated and translated brush strokes from a predefined set are placed according to some guiding information (e.g., the direction of image gradients). This technique is applicable both in 2D [Hertzmann 1998] and 3D [Schmid et al. 2011] environment producing quite compelling results. Nevertheless, the main drawback here is the restriction to a predefined set of strokes, which limit the variety and fidelity of the stylized output. Such a limitation can partly be alleviated by introducing example-based brushes [Lu et al. 2013; Zheng et al. 2017]; nevertheless, the final look is still limited to a subset of stryles that can be simulated by a composition of brush strokes.

To address this issue a more robust and general example-based approach called *Image* Analogies was pioneered by Hertzmann et al. [2001]. Given an arbitrary style exemplar and a set of guidance channels, the stylized image can be produced using guided patch-based synthesis [Wexler et al. 2007; Kaspar et al. 2015; Fišer et al. 2016]. This approach has been successfully applied to various stylization scenarios including fluid animations [Jamriška et al. 2015], 3D renders [Fišer et al. 2016; Sýkora et al. 2019], facial animations [Fišer et al. 2017] or video clips [Jamriška et al. 2019]. Nevertheless, a common drawback of this method is that it requires the preparation of custom-tailored guidance to deliver compelling stylization quality. Furthermore, an extensive computational overhead at higher resolutions makes those techniques difficult to use in production.

Neural-based style transfer approaches recently became popular due to advances made by Gatys et al. [2016], they successfully applied the pre-trained convolutional neural network VGG [Simonyan and Zisserman 2014] to the problem of style transfer. The core idea of their method is to match statistics in the domain of VGG [Simonyan and Zisserman 2014] features of both the content and style images. They further extended this idea in [Gatys et al. 2017] to introduce control over spatial location, color information, and scale of features. While these techniques produce impressive results for some particular style exemplars, they usually suffer from loss of high-frequency details of the style exemplar which is inevitably caused by the convolutional nature of the underlying neural network. Moreover, mentioned neural techniques usually have considerable computational overhead and memory footprint.

Although a feed-forward network can be pre-trained to speed up the stylization [Johnson et al. 2016; Ulyanov et al. 2016a; Dumoulin et al. 2016; Chen et al. 2017a], every new style requires additional costly training. Recently, adoption of encoder–decoder scheme was proposed [Li et al. 2017; Huang and Belongie 2017; Lu et al. 2017] to enable arbitrary style transfer in a feed-forward fashion. Here the encoder, usually convolution layers of the VGG, is used to get the feature representations (statistics) of the content and style, which are then combined, and a pre-trained decoder is used to turn the latent features back into the image. Nevertheless, all these techniques still suffer from convolutional artifacts leading to a lower quality of the synthesized imagery at a pixel level.

Recently, attempts to combine patch-based and neural-based techniques were proposed. Li et al. [2016b] search local neural patches from the style image concerning the structure of a content image, which leads to better reproduction of local textures. Liao et al. [2017] later extended this idea in their *Deep Image Analogy* framework which adapts the concept of *Image Analogies* [Hertzmann et al. 2001] in the domain of VGG features. Gu et al. [2018] recently proposed to perform reshuffle in spirit of [Kaspar et al. 2015] to reduce the overuse of particular features. Futschik et al. [2019] use patch-based method [Fišer et al. 2017] to generate a larger dataset of stylized portraits which is then used to train a generative adversarial network capable of reproducing similar quality results as those in the underlying dataset. Although these techniques can notably improve the stylization quality and better preserve high-frequency details, they still heavily rely on the space of VGG features and do not explicitly enforce textural coherence on a pixel level in color domain [Wexler et al. 2007] which is essential to retain the fidelity of the original style exemplar.

4.3 Our Approach

We propose an approach to combine patch-based synthesis with neural style transfer methods. The proposed pipeline overcomes three crucial obstacles which prevent existing stylization approaches from being used in real production: first, lower texture quality of neural-based techniques; second, the necessity of specific guidance for patchbased methods; and third, the resolution limitation which affects the usability of both approaches. Our framework allows easy switching to the newest future inventions in either neural-based or patch-based techniques.

As our first step, given the exemplar *Style* and the target image *Content*, we use an arbitrary neural-based style transfer method to synthesize an initial result (see Fig. 4.3 middle column). The resulting image on its own lacks high-frequency details of the style exemplar, and contains artifacts such as geometric distortions and colors that are not present in the original style. Also, the original contrast is usually artificially exaggerated, and edges are not sharp. However, on the other hand, it nicely preserves global style properties such as color distribution and respects the image semantics in general.

Our key idea is to use the low-resolution neural style transfer result as a guiding channel for patch-based synthesis. This enables us to combine the advantages of both techniques and to address the aforementioned limitations (see Fig. 4.3 right column). In particular, a pair of guidance channels *Source* and *Target* is needed for guided patch-based synthesis. We use blurred style exemplar as the *Source* guide and the low-resolution neural style transfer result as the *Target* guide. After running the guided patch-based synthesis, our result (Fig. 4.3 right column, bottom) effectively attenuates the neural artifacts and restores the color and texture of the original style exemplar.

Fig. 4.4 illustrates our entire pipeline which consists of three main parts: neural-based style transfer method, guided patch-based synthesis, and nearest neighbor field (NNF) upscaling method. Those individual steps are described in more detail in the following sections.

4.3.1 Neural-Based Style Transfer

Both *Style* (Fig. 4.4a) and *Content* (Fig. 4.4b) images are first subsampled by a coefficient α . This step is necessary not only to overcome the resolution restrictions but, more importantly, to suppress various high-frequency artifacts caused by neural-based techniques (α essentially defines the *working resolution* of a neural-based method). The

 α -times subsampled neural-based result (Fig. 4.4c) is then used as a guide for the patchbased synthesis method. Its resolution will be improved later in our pipeline.

4.3.2 Guided Patch-Based Synthesis

The output from the neural method (Fig. 4.4c) is used as a *Target* guide image in the patch-based method. Our pipeline does not assume any particular patch-based method; we used StyLit [Fišer et al. 2016] algorithm for synthesis, however, we adapt its original error metric for measuring patch similarity to our needs. Let S be a style exemplar, \mathcal{O} an output image, and G^S and G^T source and target guides, for matching two patches $p \in G^S$ and $q \in G^T$; we use the following error metric:

$$E(\mathcal{S}, \mathcal{O}, G^{\mathcal{S}}, G^{\mathcal{T}}, p, q) = ||\mathcal{S}(p) - \mathcal{O}(q)||^2 + \lambda_q ||G^{\mathcal{S}}(p) - G^{\mathcal{T}}(q)||^2$$
(4.1)

where λ_g is a weighting factor for guiding channel and the first term helps to preserve *texture coherence* by directly matching colors in patches of *Style* to those in the output image \mathcal{O} . Of all the images, only \mathcal{O} is iteratively updated during the optimization process described in StyLit [Fišer et al. 2016].

To obtain *Source* guide image, we use the already subsampled style image used in the previous step (Sec. 4.3.1), and upsample it back to its original resolution. To encourage the patch-based synthesis to find good correspondences for the style transfer, equivalent subsampling followed by upsampling needs to be done for both the *Source* and *Target* images. In spirit of *Color Me Noisy* [Fišer et al. 2014], an additional low-pass filter can be applied on the *Source* image to let the synthesis algorithm deviate more from the initial solution, thus making the final result more abstract.

In Fig. 4.4d the result of patch-based synthesis is depicted in color for clarity, nevertheless, internally in our processing pipeline we use only the resulting nearest neighbor field (Fig. 4.4f) which is subsequently upsampled (Fig. 4.4g) and turned into a high-resolution image in the next step.

4.3.3 NNF Upscaling

Given the computed NNF-nearest neighbor field (Fig. 4.4f) and the style exemplar in its original resolution (Fig. 4.4a), a *voting step* (c.f. [Wexler et al. 2007]) needs to be performed in order to reconstruct the final image. To reduce the computational overhead, we perform the patch-based synthesis (Sec. 4.3.2) at β -times lower resolution than the original target resolution (thus β essentially defines the *working resolution* of a patchbased method). Next, the resulting **nnf** (Fig. 4.4f) is upscaled by a factor of β to obtain the **NNF** (Fig. 4.4g) of the same resolution as the target image as follows:

$$\mathbf{NNF}(x,y) = \mathbf{nnf}(x/\beta, y/\beta) \cdot \beta + (x \mod \beta, y \mod \beta)$$
(4.2)

Finally, we perform a voting step using **NNF** to produce the final high-resolution result precisely preserving the characteristics of the canvas and the original artistic medium (Fig. 4.4e).

4.4 VGG-Based Guidance

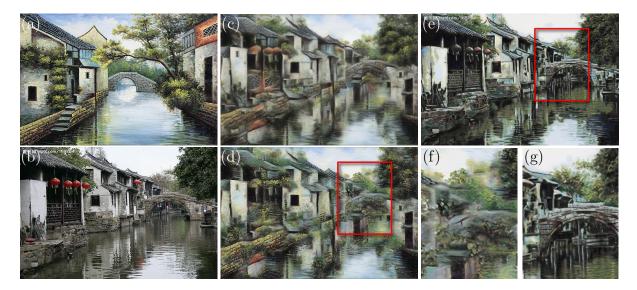


Figure 4.6: Demonstration of the problem when patch-based synthesis has to rely on ambiguous color guidance: (a) style exemplar, (b) target image, (c) output of Gu et al. [2018], (d) output of our basic algorithm with color-based guidance, (e) output of our style transfer algorithm with neural guidance. Note how our VGG-guided algorithm better preserves the semantics of the target photo, cf. details in (f) and (g).

One of the limitations of the proposed base algorithm introduced in the previous section is that it relies on color information to establish correspondences between style exemplar and the target image. This drawback could lead to an ambiguity that may introduce visible stylization artifacts (see Fig. 4.6).

In this section, we introduce a variant of our style transfer pipeline that uses features extracted by the convolutional layers of a classification network for guidance directly rather than relying on a neural style transfer algorithm to produce initial color domain stylization. The aforementioned neural responses provide more discriminative guidance than colors and thus can preserve global semantics of the target while still keeping the benefits of patch-based optimization.

Our approach is inspired by modern optimization-based neural style transfer techniques of Liao et al. [2017] and Gu et al. [2018] that rely on computationally demanding global descent through a complicated loss function using an optimizer like L-BFGS. Although this approach is conceptually similar to the patch-based optimization framework, in our case expensive global descent is approximated by a highly efficient approximate nearest-neighbor matching.

The algorithm first extracts neural features for both the source and target image in multiple scales (see Fig. 4.5). Specifically, we run the input images through the neural network on four resolutions: 1344×1344 , 896×896 , 448×448 and 224×224 . This set was chosen to capture a broader range of neural features.

For this purpose, we use VGG-19 network architecture trained on the ImageNet dataset [Simonyan and Zisserman 2014]. After running a feed-forward pass on the input image, features are extracted from 6 different layers of the network. The layers used are

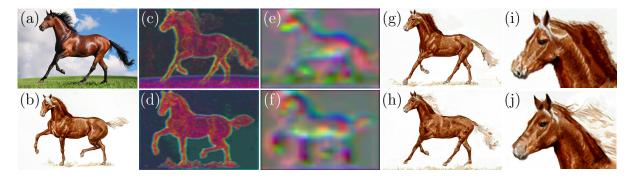


Figure 4.7: An example result from our VGG-guided style transfer algorithm: (a) target image, (b) style exemplar, corresponding compressed VGG-responses of low- (c, d) and high-level (e, f) features used as a guide for patch-based synthesis, (g) output of Liao et al. [2017], (h) output of our style transfer framework with neural guidance, note how our method can deliver comparable visual quality, cf. details in (i) and (j).

conv2_2, conv3_1, conv3_4, conv4_1, conv4_4, and conv5_1. Features are extracted after applying the ReLU activation.

These neural features capture localized semantic similarities found in both images and can be used to guide the patch-based synthesis. However, the high dimensionality of these per-pixel features might significantly compromise both the performance and the quality of the patch-matching step. To avoid this, we reduce the feature dimension using PCA [Turk and Pentland 1991]. In particular, we treat each feature vector as an independent point and process feature maps in groups of the same resolution. The number of principal components we extract varies by feature map resolution. We use top 3 components at 1344 × 1344, top 6 components at 896 × 896, and finally top 12 components for the two remaining resolutions. We normalize the resulting values to [0, 255] interval and resample them to the required resolution using bicubic upsampling. This can either be lower resolution, typically used in neural techniques, or full resolution of the target image. Lastly, we run the patch-based synthesis algorithm of Fišer et al. [2016] to produce the final stylized image. The output is visually comparable to the state-of-the-art [Liao et al. 2017; Gu et al. 2018] (see Fig. 4.7).

4.5 Results

We implemented our method both for CPU and GPU, using C++ and CUDA, respectively.



Input

Gatys et al.

DeepDream

Figure 4.8: Portrait on a wall: (a) target content of resolution $4000 \times 3000 \text{ px}$, (b) style exemplar of a painting on a wall having the same resolution, (c) 10x zoom-in to the (b) to show fine artistic attributes and structure of the canvas-wall/plaster. Our method is entirely independent of the used artistic medium as well of a canvas the style exemplar is presented on. The results are presented in the same fashion as in Fig. 4.9.

The parameter α is set to make the input images to the neural-based method approximately 400–500 pixels wide. In the case when the input images are already of low-resolution, we set α to be at least 2—to ensure the patch-based synthesis will have enough freedom to fix some of the artifacts caused by the neural-based approach. The α —sub-sampling allows us to get the result from a neural-based approach much faster or use a method that does not support high-resolution input. Moreover, it allows us to significantly suppress some of the artifacts of neural approaches. The parameter β allows us to stylize images of size 346Mpix or even larger, and to get the final result much faster (see an extreme-resolution result in Fig. 4.1 and our supplementary material). We observed that if the parameter β is in range 1–4, the perceived loss in the quality is almost negligible. If the parameter β is in range 6–10, when zooming closely, one can observe some repetition artifacts, however, the image is sharp and the overall quality is still satisfactory.

We measured run-time and memory performance. For detailed run-time measurement on mid-range laptop see graph in Fig. 4.10. On a desktop PC, the computational overhead is even lower, e.g., on NVIDIA Quadro M2000, stylizing the image of size 160Mpix takes between 3–30 seconds depending on the selection of the parameter β . Increasing the parameter β causes a linear increase in the computational time, while the number of pixels grows exponentially. Our method requires a few hundred MBs of RAM/GPU

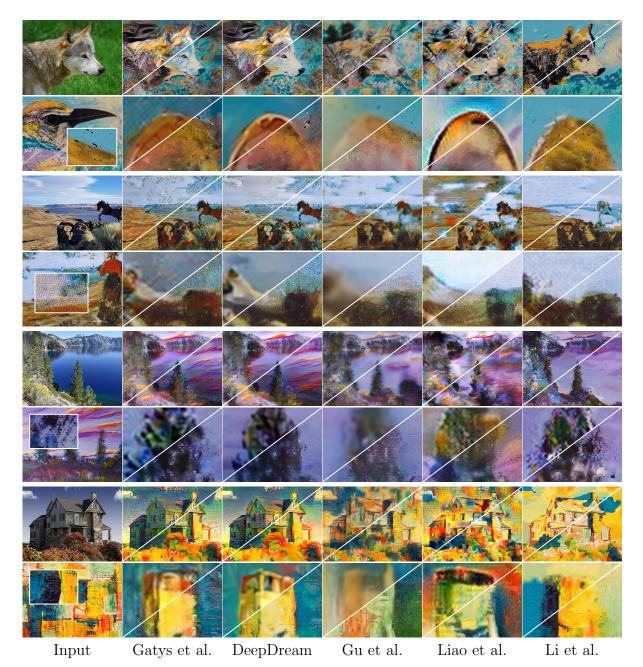


Figure 4.9: Our method enhancing the results of five different neural-based approaches: The leftmost column–content images and style exemplars (with zoomed patches). Next, left-to-right, are the result of DeepArt [Gatys et al. 2016], DeepDream, Gu et al. [2018], Liao et al. [2017], and Li et al. [2017]. The top-left triangle shows the result of the underlying neural-based approach (bicubically up-sampled from a typical size of 600×400 px to the target resolution), while the bottom-right shows result enhanced by our method (top row–entire stylized images, bottom row–zoom-in). Our results not only have significantly higher resolution but also better preserve the original colors and canvas structure as well as brush strokes visible in the exemplar painting. Various artifacts caused by the neural approach are significantly suppressed. All images shown in this figure are of resolution ranging from 4000×2200 to 6000×4000 px.

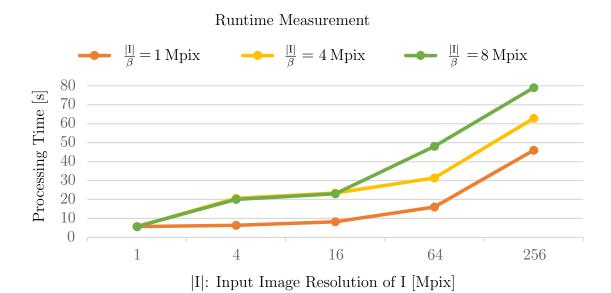


Figure 4.10: Performance of our method (full pipeline–Fig. 4.4, excluding the neural part) on images ranging from resolution of 1Mpx, (i.e. $1000 \times 1000 \ px$) to extremely large resolution of 256Mpix (i.e., $16000 \times 16000 \ px$). Orange, yellow, and green lines show a case where the parameter β was set such that the patch-based method was run on a resolution of 1Mpix, 4Mpix, and 8Mpix respectively. The measurement was done on a mid-range laptop with NVIDIA GTX 1050 graphics card.

memory. The exact amount depends on the resolution of the input images and the value of the parameter β .

The performance of the neural-based step depends on a particular method. However, because the input is of very low resolution, 400–500 px wide, the run-time typically ranges between hundreds of milliseconds and several seconds. Most neural-based approaches cannot stylize images larger than 4K-by-4K due to GPU memory constraints. Although there is a possibility to decompose the synthesis into a set of tiles that are processed separately and stitched together, the resulting image would still suffer from the convolutional nature of used neural network introducing disturbing high-frequency artifacts and colors not present in the original style exemplar.

We plugged several different state-of-the-art neural-based style transfer techniques into our framework (see Fig. 4.9 and 4.8). In all cases, applying patch-based synthesis with neural transfer output as guidance produces better results than using the neural-based approach alone. The most noticeable differences are visible in (1) the original colors (e.g., saturated pixels that do not appear in the original style exemplar are removed), (2) suppression of checkerboard artifacts caused by deconvolution [Odena et al. 2016], and (3) results are sharper containing important high-frequency details of the original brush strokes and underlying canvas structure. Fig. 4.1 demonstrates stylization of a 346Mpix image. Despite the huge resolution, the result is still perfectly sharp and preserves well characteristics of the original artistic media.



Figure 4.11: Results produced by our VGG-guided style transfer algorithm (from left to right): style exemplar, target image, and our result. Our method works well namely in cases when style and target images depict similar content, i.e., when they have compatible VGG activations.

To demonstrate the benefit of using the output of the neural approach to guide the patch-based synthesis, we compared our method to the guidance based only on blurred grayscale images (Fig. 4.3 left column) as proposed in the original Image Analogies method [Hertzmann et al. 2001], the result does not properly respect the content semantics, causing trees to become pink.

In Fig. 4.11 and 4.13, we present additional results of our VGG-guided style transfer algorithm. These demonstrate the proposed method can produce convincing stylization without the need to use existing neural techniques as a preprocess.

Finally, in Fig. 4.12, we demonstrate a UI prototype of our method running in Photoshop.



Figure 4.12: A screenshot of our method running in Adobe Photoshop: (a) zoom of a target layer, (b) zoom of a style layer; the visible layer is the result of DeepDream enhanced by our method.

4.6 Limitations and Future Work

Although in most cases, our approach is capable of delivering significantly better and visually more pleasing results than the underlying neural technique itself, it still relies on the neural result as the initial solution. Due to this reason, we cannot fix large-scale artifacts produced by the neural-based method (see Fig. 4.15). In the current pipeline, only high-frequency artifacts can be suppressed. When zooming in, the improvement in the texture quality is immediately visible, nevertheless, looking from a distance, high-resolution image obtained by our method may appear almost identical as the result of the underlying neural approach.

4.6. LIMITATIONS AND FUTURE WORK

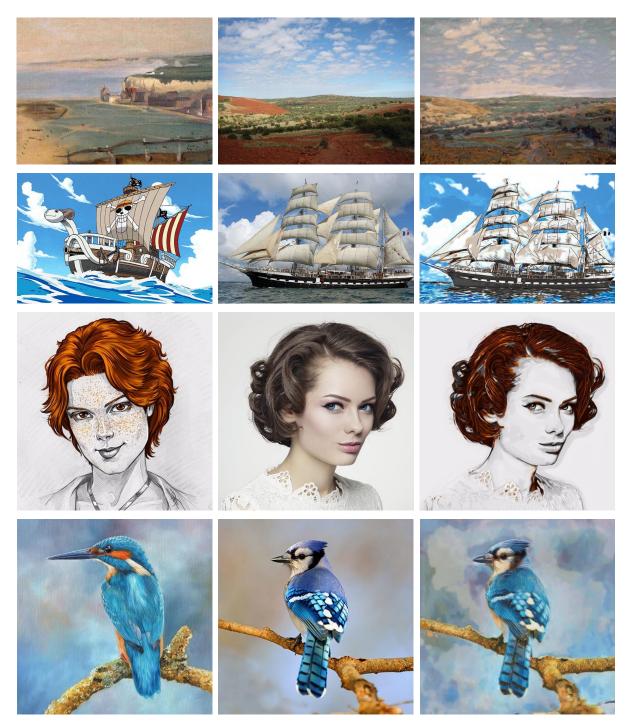


Figure 4.13: Additional results produced by our VGG-guided style transfer algorithm (from left to right): style exemplar, target image, and our result.



Figure 4.14: A limitation common to neural-based approaches: (a-b) content image, (c-d) style exemplar, (e-f) result of [Li et al. 2017] enhanced by our method. The content of the original image is not preserved well. In the first case, the similar mixture of colors is used to paint bushes, house, and also the sky. In the second case, all colors appearing in the style exemplar are used to stylize the target regardless of its content. However, high-frequency content is reproduced well. To address this limitation, we propose to incorporate a neural network trained for image segmentation into our pipeline.



Figure 4.15: Large-scale artifact limitation: (a) content image, (b) style exemplar, (c) result of Gatys et al., distortions in eye region are visible, (d) ours, colors and high-frequency details are reproduced well; however, in our current pipeline, large-scale artifacts produced by the underlying neural approach are not fixed. Thus distortion in the eye region is still apparent.

4.7. CONCLUSION

As future work, we would like to tackle the issue commonly seen in neural techniques, i.e., many different colors are mixed together within a single coherent region or when the same mixture of colors is used to stylize semantically different regions (see an example in Fig. 4.14). To address this problem, we see two promising solutions. First, extending our pipeline in a way that patch-based synthesis is guided by a neural network trained for segmentation on both natural and artistic images to encourage more semantically correct matching of patches. Second, incorporate mask-based loss function as described in [Reimann et al. 2019]. Although, this might not be feasible for all neural-network approaches we use or in a case when it is desired to treat an underlying neural-network as a black box.

Our technique helps to restore high-frequency details and essential attributes of used artistic media; however, in some cases, this process might destroy some of the important content details. We see a promising solution in the work of Calvo [Calvo et al. 2019], where they introduce a technique to intensify or reduce the stylization strength locally.

Another interesting follow-up of our work could be an extension to videos. This might seems straightforward, but even if the video delivered by the underlying neural-based style transfer method is stable in time, randomness in the patch-based step of our pipeline will most likely introduce disturbing temporal inconsistency. To solve this, one could use techniques described in [Jamriška et al. 2019] or [Fišer et al. 2017].

Another area for future work worth exploring would be adding interactions to control the result. Also, some of the neural-based approaches support multiple style exemplars; we suggest to explore possibilities of using multiple styles in our enhancing scenario.

4.7 Conclusion

We have presented a new approach that combines neural and patch-based style transfer techniques, and proposed a way to utilize the generality of the former, while achieving the texture quality of the latter. We introduced a computationally inexpensive algorithm for upscaling the synthesis output to obtain its high-resolution version and a new approach to neural-based style transfer that can use responses of the neural network directly as a guide for patch-based synthesis. Thanks to those advances, we can produce style transfer results with notably larger resolutions than previous neural-based techniques and significantly reduce the computational overhead while retaining comparable visual quality. We believe our method could enable broader applicability of style transfer methods in commercial practice. To that end, we integrated our approach into Adobe Photoshop in the form of a plug-in.

Chapter 5

Interactive Video Stylization Using Few-Shot Patch-Based Training

5.1 Introduction

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. 2019]. 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

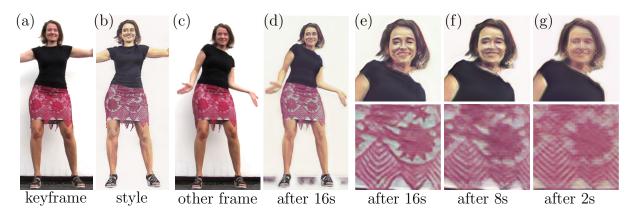


Figure 5.1: An example of a sequence stylized using our approach. One frame from the original sequence is selected as a keyframe (a) and an artist stylizes it with acrylic paint (b). We use this single style exemplar as the only data to train a network. After 16 seconds of training, the network can stylize 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 \bigcirc Zuzana Studená.

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.

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 this paper, 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-toimage 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.

5.2 Related Work

A straightforward approach to propagate the stylized content from a painted keyframe to the rest of the sequence could be to estimate dense correspondences between the painted keyframe and all other video frames [Wang et al. 2019c; Li et al. 2019] or compute an optical flow [Chen et al. 2013] between consecutive frames, and use it to propagate the stylized content from the keyframe. However, as shown in Jamriška et al. [2019] this simple approach may lead to noticeable distortion artifacts as the textural coherence is not maintained. Moreover, even when the distortion is small the texture advection effect leads to an unwanted perception that the stylized content is painted on the surface.

A more sophisticated approach to keyframe-based video stylization was pioneered by Bénard et al. [2013] who use guided patch-based synthesis [Hertzmann et al. 2001] to maintain textural coherence. In their approach a 3D renderer is used to produce a set of auxiliary channels, which guides the synthesis. This approach was recently extended to arbitrary videos by Jamriška et al. [2019]. In their framework, guiding channels are reconstructed automatically from the input video. Jamriška et al. also offer a post-processing step that merges the content stylized from multiple possibly inconsistent keyframes. Although patch-based techniques prove to deliver convincing results, their crucial drawback is that they can stylize the video only sequentially and require an explicit merging step to be performed when multiple keyframes are provided. Those limitations hinder random access, parallel processing, or real-time response, which we would like to preserve in our video stylization framework.

When considering fast video stylization, appearance translation networks [Isola et al. 2017] could provide a more appropriate solution. Once trained, they can perform semantically meaningful appearance transfer in real-time as recently demonstrated on human portraits [Futschik et al. 2019]. Nevertheless, a critical drawback here is that to learn such a translation network a large training dataset is required. That 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 [Wang et al. 2018a; Chan et al. 2019] which is completely intractable.

Recently, few-shot learning techniques were introduced [Wang et al. 2019a;b] to perform appearance translation without the need to have a large dataset of specific style translation pairs. However, to do that a domain-specific dataset is required (e.g., facial videos, human bodies in motion, etc.) to pre-train the network. Such a requirement impedes the usage of previous few-shot methods in a general context where the target domain is not known beforehand.

In our method, we relax the requirement of domain-specific pre-training and show how to train the appearance translation network solely on exemplars provided by the user. Our approach bears resemblance to previous neural texture synthesis techniques [Li and Wand 2016a; 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. A key idea here is to leverage the fully convolutional nature of the neural net. Even if the network is trained on smaller patches it can be used to synthesize larger images.

Recently, the idea of patch-based training was further explored to accelerate training [Shocher et al. 2018] or to maintain high-level context [Zhou et al. 2018; Shocher et al. 2019; Shaham et al. 2019]; however, all those techniques deal only with a singe image scenario and are not directly applicable in our context. Also, they do not use a deliberately smaller batch of randomly cropped patches as a means of overfitting avoidance which is one of our key contributions.

Handling temporal consistency is a central task of video stylization methods. When individual frames are stylized independently, the resulting stylized animation usually contains intense temporal flickering. Although this effect is natural for traditional handcolored animations [Fišer et al. 2014] it may become uncomfortable for the observer when watched for a longer period of time. Due to this reason, previous video stylization methods, either patch-based [Bénard et al. 2013; Fišer et al. 2017; Jamriška et al. 2019; Frigo et al. 2019] or neural-based [Chen et al. 2017a; Sanakoyeu et al. 2018; Ruder et al. 2018], try to ensure temporal stability explicitly, e.g., by measuring the consistency between previous and a newly generated video frame. Alternatively, blind temporal coherency [Lai et al. 2018] could be used in the post-processing step. Yet, these approaches introduce data-dependency to the processing pipeline, which we would like to avoid in order to enable random access and parallel processing.

Our approach bears also a resemblance to a just-in-time training recently proposed by Mullapudi et al. [2019]. In their approach, labelling is provided for a subset of frames by a more accurate predictor and then propagated to 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 we demonstrate could suffer from strong overfitting artifacts and thus is not applicable in our scenario where a detailed texture needs to be propagated.

5.3 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;

5.3. OUR APPROACH

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. 5.2 for details).

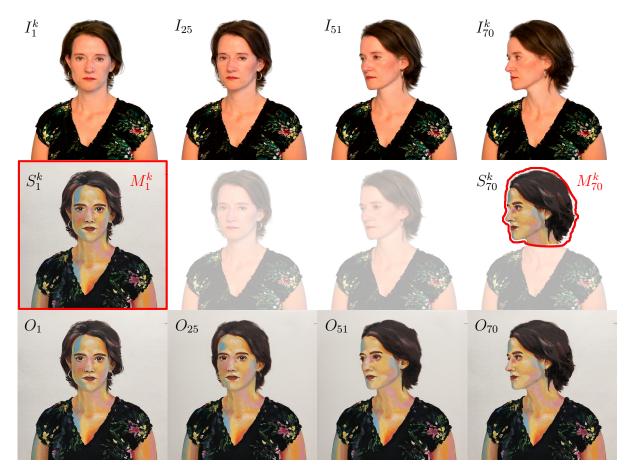


Figure 5.2: The setting of video stylization with keyframes. The first row shows an input video sequence I. There are two keyframes painted by the user, one keyframe is painted fully (S_1^k) and the other is painted only partially (S_{70}^k) . Mask M_1^k denotes that the entire keyframe is used; mask 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 \bigcirc Zuzana Studená.

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



Figure 5.3: 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. 5.6). 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 \bigcirc Zuzana Studená.

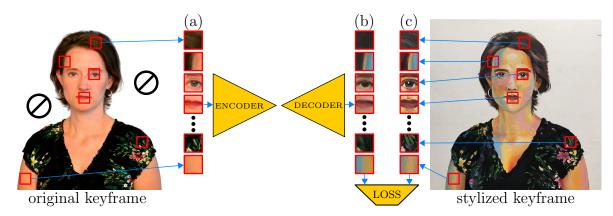


Figure 5.4: Training strategy: we randomly sample a set of small patches from the masked area of the original keyframe (a). These patches are then propagated through the network in a single batch to produce their stylized counterparts (b). We then compute the loss of these stylized counterparts (b) with 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 this paper, we use a combination of L1 loss, adversarial loss, and VGG loss as described in [Futschik et al. 2019]. Video frame (left) and style exemplar (right) courtesy of \bigcirc Zuzana Studená.

parallel, or any single frame on demand. It would also be beneficial if the learning phase 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 current stateof-the-art (see Fig. 5.3c 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 5.3 and 5.9. Furthermore, the resulting sequence also 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 current state-of-theart, 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.

5.3.1 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 [Futschik et al. 2019], we randomly sample smaller rectangular patches from all stylized keyframes S^k (see Fig. 5.4) 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. 5.5). 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].

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.

5.3.2 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. 5.8). 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.

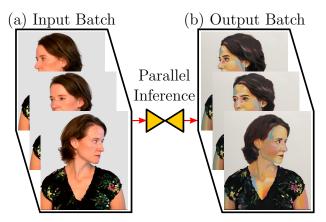


Figure 5.5: Inference: thanks to the fully convolutional nature of the network, we can perform the inference on entire video frames, even though the training is done on small patches only. Since the inference does not depend on other stylized frames, all video frames can be stylized in parallel or in 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 \bigcirc Zuzana Studená.

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 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. 5.6). In-depth elaboration can be found in Section 5.4.

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.

5.3.3 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.

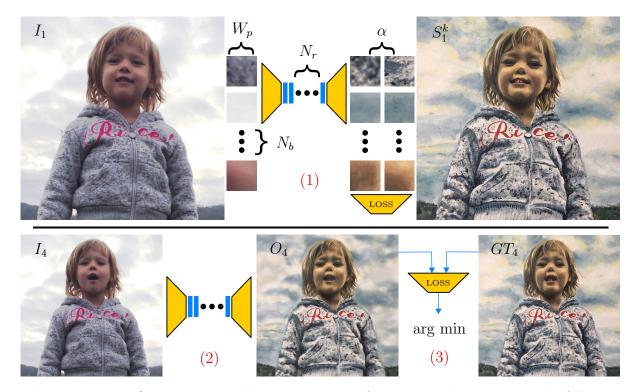


Figure 5.6: To fine-tune critical hyperparameters of our network, we propose the following optimization scheme. We tune batch size N_b , patch size W_p , number of ResNet blocks N_r , and learning rate α . Using the grid search method we sample 4-dimensional space given by these hyperparameters and for 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 (O₄) and result of [Jamriška et al. 2019] (GT₄) - 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 \bigcirc Zuzana Studená.

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. 5.7). To compensate for the motion in the input video,

Gaussians are treated as points attached to a grid, which is deformed using as-rigid-aspossible (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.

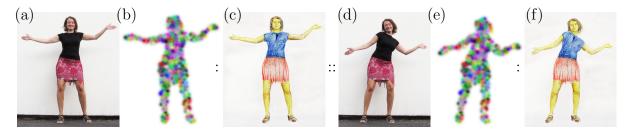


Figure 5.7: To suppress visual ambiguity of the dark mostly homogeneous T-shirt in (a) an auxiliary input layer is provided that contains a mixture of randomly distributed and colored Gaussians (b). The translation network is trained on patches of which input pixels contain those additional color components. 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 (C) Zuzana Studená and style exemplar (b) courtesy of (C) Pavla Sýkorová.

The reason why the mixture of Gaussians is used instead of directly encoding pixel coordinates as done, e.g., in [Liu et al. 2018; 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).

5.4 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].

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

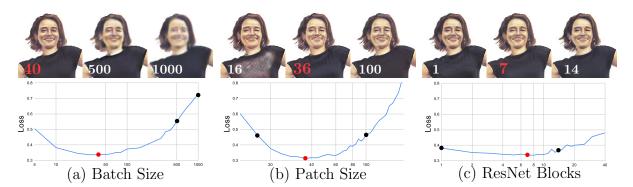


Figure 5.8: Influence of important hyperparameters on visual quality of results. The loss, y-axes, is computed w.r.t. the output of Jamriška et al. [2019]. The best setting for each hyperparameter is highlighted in red: (a) The loss curve for the batch size N_b —the number of patches in one training 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.

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. 5.8 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. 5.8a). 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. 5.8b) 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. 5.8c) 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. 5.1).

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 5.1, 5.3, 5.7, 5.11, and 5.12) while the more complex ones have multiple (ranging from two to seven, see Figures 5.14, 5.13, 5.15, and 5.16). 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.

5.4.1 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. 5.9).

We compared our approach with the current state-of-the-art in keyframe-based video stylization [Jamriška et al. 2019]. For the results see Figures 5.10, 5.12, 5.14, 5.15, 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



Figure 5.9: 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.

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.

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

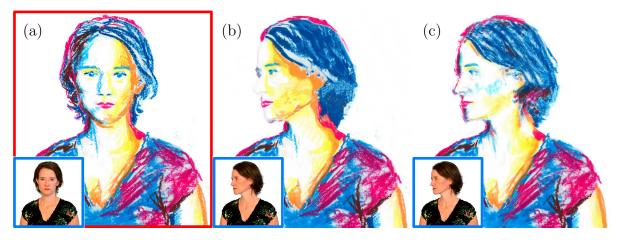


Figure 5.10: When the target subject undergoes a substantial appearance change, the results of both Jamriška et al. [2019] (b) and our method (c) exhibit noticeable artifacts. The parts that were not present in the keyframe are reconstructed poorly—see the face and hair regions where [Jamriška et al. 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 (C) Zuzana Studená.

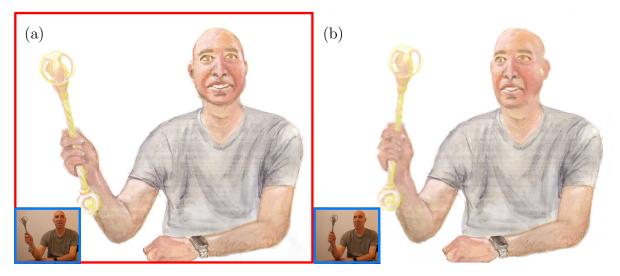


Figure 5.11: Given one keyframe (a) and a video sequence (in blue), our method produces the stylized result (b). Video frames (insets of a, b) courtesy of \bigcirc Adam Finkelstein and style exemplars (a) courtesy of \bigcirc Pavla Sýkorová.

and stylize using our approach, whereas the method of Jamriška et al. requires around 15 minutes.

5.4.2 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.

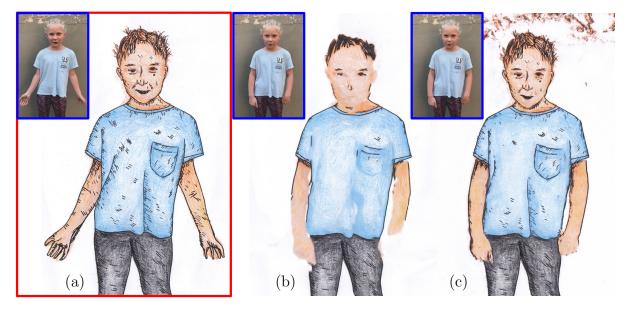


Figure 5.12: For the state-of-the-art algorithm of [Jamriška et al. 2019], contour based styles (a) present a particular challenge (b). Using our approach (c), the contours are transferred with finer detail and remain sharp even as the sequence undergoes transformations. Video frames (insets of a-c) and style exemplar (a) courtesy of (C) Štěpánka Sýkorová.



Figure 5.13: 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 \bigcirc Jakub Javora.

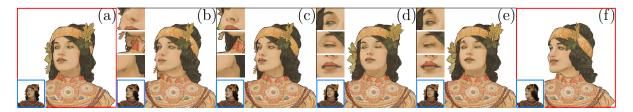


Figure 5.14: Keyframes (a, f) were used to stylize the sequence of 154 frames. See the qualitative difference between Jamriška et al. [2019] (b) and our result (c). Focusing mainly on zoom-in views, our approach better preserves contour lines around the nose and chin; moreover, the method of Jamriška 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 \bigcirc 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,

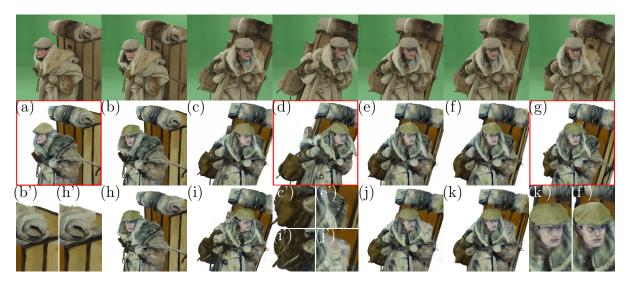


Figure 5.15: A complex input sequence (the first row) with seven keyframes, three of them are shown in (a, d, g). Here we compare our approach to the approach of Jamriška et al. [2019]. See our result (b) and theirs (h) along with the close-ups (b', h'); due to their explicit handling of temporal coherence, 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 \bigcirc MAUR film.



Figure 5.16: 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 \bigcirc Muchalogy.

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.

5.5 Limitations and Future Work

Although our framework 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 current state-of-the-art [Jamriška et al. 2019] (deep convolutional network vs. guided patchbased 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 patchbased 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. 5.10. 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 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.

5.6 Conclusion

We presented a neural approach to keyframe-based stylization of arbitrary videos. With our technique, one can stylize the target sequence using only one or a few hand-drawn keyframes. In contrast to previous neural-based methods, our method does not require large domain-specific datasets nor lengthy pre-training. Thanks to our patch-based training scheme, optimized hyperparameters, and handling of temporal coherence, a standard appearance translation network can be trained on a small set of exemplars. Once trained, it can quickly deliver temporally coherent stylized videos with a visual quality comparable to the current state-of-the-art in keyframe-based video stylization, which uses guided patch-based synthesis. A key benefit of our technique is that it can work in a frameindependent mode, which is highly beneficial for current professional video editing tools that rely heavily on random access and parallel processing. It also does not require the explicit merging of stylized content when slightly inconsistent keyframes are used.

Moreover, since the network in our framework can be trained progressively, and the inference runs in real-time on off-the-shelf GPUs, we can propose several new video editing scenarios that were previously difficult to achieve. Those include stylization of a live video stream using a physical hand-drawn exemplar being created and captured simultaneously by another video camera. We believe interactive scenarios such as this will empower the creative potential of artists and inspire them with new creative ideas.

Chapter 6

STALP: Style Transfer with Auxiliary Limited Pairing

source frame target frames our approach Jamriška et al. Texler et al. Liao et al.



source style

Figure 6.1: 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 current state-of-the-art in example-based video stylzation (Jamriška et al. [Jamriška et al. 2019] and Texler et al. [Texler et al. 2020b]) 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. [Liao et al. 2017]. Although the style's texture is preserved reasonably well, the transfer is not semantically meaningful.

6.1 Introduction

In recent years, methods for performing automatic style transfer from an exemplar image to a target image or a video have gained significant popularity. Although state of the art in this field progresses quickly and produces ever more believable artistic images, there are still aspects in which most methods tend to have fundamental shortcomings. One such crucial element is defining the semantic intent while still preserving visual characteristics of the used artistic media.

A seminal work in this direction was the Image Analogies framework introduced by Hertzmann et al. [2001], which requires the user to provide a set of guidance channels [Bénard et al. 2013; Fišer et al. 2016; 2017; Jamriška et al. 2019] that encourage the synthesis algorithm to transfer smaller patches of the style exemplar onto desired spatial locations in the target image. Those channels, however, need to be prepared explicitly by the user or generated algorithmically for a limited target domain, e.g., 3D renders [Fišer et al. 2016], facial images [Fišer et al. 2017], or a sequence of video frames close to the stylized keyframe [Jamriška et al. 2019]. Deriving consistent semantically meaningful guidance in the general case remains an open problem.

Neural approaches to style transfer [Gatys et al. 2016; Li et al. 2017; Kolkin et al. 2019] rely on the assumption that one can encode semantic similarity using the correspondence of statistics of neural features extracted from responses of the VGG network [Simonyan and Zisserman 2014]. Although such an assumption holds in some cases, it is not easy to amend when it fails. Moreover, in contrast to patch-based methods, neural techniques tend to produce noticeable visual artifacts due to their statistical nature. One can partially alleviate this drawback by applying patch-based synthesis in the neural domain [Li and Wand 2016c; Liao et al. 2017]. However, since in this scenario neural features are transferred explicitly, the requirement of knowledge of accurate correspondences is still inevitable.

Another possibility of preserving semantically meaningful transfer is using the imageto-image translation principle pioneered by Isola et al. [2017]. This approach can encode semantic intent and retain high-quality output. However, it has a fundamental limitation of requiring a relatively large dataset of image pairs (original image plus its stylized counterpart), which is rarely easy to obtain when considering artistic applications. Lastly, a group of unpaired image translation algorithms could be used [Zhu et al. 2017a; Park et al. 2020], however, since it can be difficult to incorporate intent into these methods, they are not as suitable for tasks where the artist needs greater control.

In this paper, we present a novel approach to neural style transfer that allows 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 contrast to previous neural techniques, in our framework, the user explicitly encodes the semantic intent by specifying a stylized counterpart for a selected image from the set that needs to be stylized. Using this single style exemplar, we then train an image-to-image translation network that stylizes the remaining images. Our approach bears a resemblance to the recent keyframe-based video stylization framework of Texler et al. [2020b], where a similar workflow is used. A key difference in our 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 auto-completion of a panorama painting, stylization of 3D renders, or different portraits captured under similar illumination conditions.

6.2 Related Work

Despite the renewed interest and broader impact, image stylization algorithms date back decades. Traditionally, they were based on predefined, hand-designed transformations limited to a subset of styles, and possibly target domains as well. One example of such transformation approach was shown by Curtis et al. [1997], running a physical simulation to produce watercolor filter effect. Other research directions focused on composing images from static or procedurally generated brush strokes or pens [Bénard et al. 2010; Bousseau et al. 2006; Praun et al. 2001; Salisbury et al. 1997]. These conventional algorithmic approaches can create very appealing results, but they have the added difficulty of requiring the style filters to be designed on an individual basis. Therefore, the act of creating a new style or even slight modifications of existing styles tends to necessitate considerable amounts of effort. These methods do not require a style exemplar, but instead contain a prior given by the design of the filter.

The framework of Image Analogies proposed by Hertzmann et al. [2001] trades designing elements of the output image directly for designing a set of guidance channels which form a loss function. Optimizing over pixel locations and directly copying patches of an exemplar image guarantees that features found in the exemplar will be represented exactly in the resulting image. This framework became the basis of numerous style transfer methods [Bénard et al. 2013; Fišer et al. 2016; 2017; Dvorožňák et al. 2018; Jamriška et al. 2019]. A key advantage over traditional algorithmic methods lies the fact that this framework allows for transfer of arbitrary style.

However, creating the guidance channels is cumbersome, and in some potential applications it might not be always clear how to design algorithms for obtaining them automatically, and still, the task of preparing a framework that would work with arbitrary images remains seemingly impossible. To sidestep this issue, methods of general style transfer have been formulated. Frigo et al. [2016] attempts to re-imagine the problem of guiding channels by splitting the image into partitions and matching these to their counterparts. More commonly known, Gatys et al. [2016] uses responses of a neural network to generate global style statistics which an optimization process sees to reproduce in the result while incorporating a content constraint to prevent the overall structure from diverging too far from the target image. Refining these ideas to a video domain and employing a more sophisticated loss functions, others [Chen et al. 2017a; Li et al. 2017; Ruder et al. 2018; Kolkin et al. 2019] manage to produce results which are coherent in time and more faithful to the style. While they produce impressive results on some inputs, these methods generally take all the control out of the artists' hands and are notoriously difficult to steer in different directions, as their mechanisms are non-intuitive and unpredictable.

A different view of the problem is offered by the image-to-image framework, which aims to translate images from one domain to another, which is directly applicable to style transfer. While the original image translation methods [Isola et al. 2017; Johnson et al. 2016] require relatively large dataset to work reliably, by their combination with generative adversarial models [Goodfellow et al. 2014; Zhu et al. 2017a], this requirement can be relaxed. Unlike techniques based on image analogies, these methods tend to require substantial amount of model training. And although patch-based synthesis [Fišer et al. 2017] can be used to generate a large number of image pairs on which one can train the image-to-image translation network [Futschik et al. 2019], the problem of having meaningful guidance remains.

Few-shot learning techniques [Liu et al. 2019; Wang et al. 2019b], as well as approaches based on deformation transfer [Siarohin et al. 2019a;b] require only a single style exemplar. However, they still need pre-training on large dataset of specific target domains and thus are not applicable in general case. Moreover, these techniques capture only the target subject's coarse deformation characteristics; its structure or identity is omitted. A similar drawback also holds for approaches based on generative adversarial networks such as StyleGAN v2 [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's features. However, this process is inaccurate, leading to imprecise alignment that hinders the network's ability to preserve the target subject's structure or identity.

6.3 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. 6.2). 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. 6.8). 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. 6.3), 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 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. Texler et al. [2020b] uses the network architecture of \mathcal{F} as well in a similar setting as ours, i.e., small number of keyframes K, however, their 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. 6.2, which we minimize as we train \mathcal{F} :

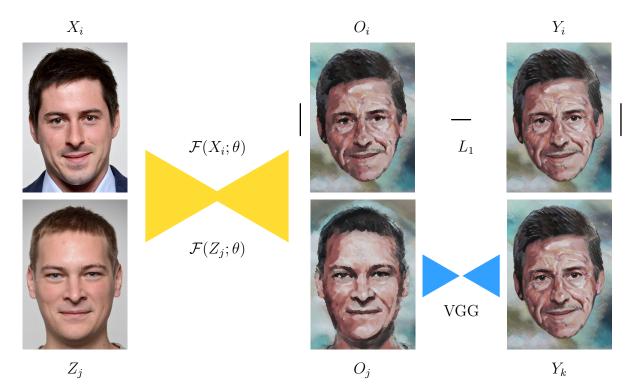


Figure 6.2: An overview of our approach—we optimize weights θ of a translation network \mathcal{F} which accepts images from a source domain X or Z and produces output images O with a similar appearance as those in the target domain Y. The high-frequency details are preserved well, thanks to the L_1 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 (C) Graciela Bombalova-Bogra, used with permission.

- 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:

$$\sum_{i} |\mathcal{F}(X_{i};\theta) - Y_{i}| + \lambda \sum_{j,k} \sum_{l} ||\mathcal{G}^{l}(\mathcal{F}(Z_{j};\theta)) - \mathcal{G}^{l}(Y_{k})||^{2}$$
(6.1)

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 [Simonyan 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. 6.4, where if we

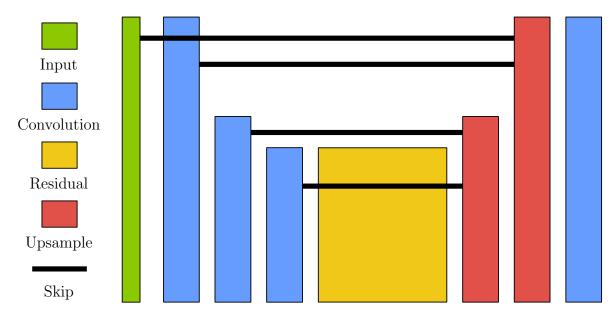


Figure 6.3: A network architecture used for our model \mathcal{F} : input layer (green), one 7×7 and two 3×3 convolution blocks (blue), nine 3×3 residual blocks (yellow), two 3×3 upsampling blocks (red), and one additional block with 7×7 convolutions (blue). Skip connections (black) are used to connect downsampling and upsampling layers.

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.



Figure 6.4: An ablation study demonstrating the importance of individual terms in our objective function (6.1)—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 (6.1) we get the result which produces a good balance between identity and style preservation. Source style \mathcal{O} Graciela Bombalova-Bogra, used with permission.

By minimizing the objective (6.1) 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. 6.5). 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 semantic structure (X to Y), we posit that this acts as an implicit content preservation technique.

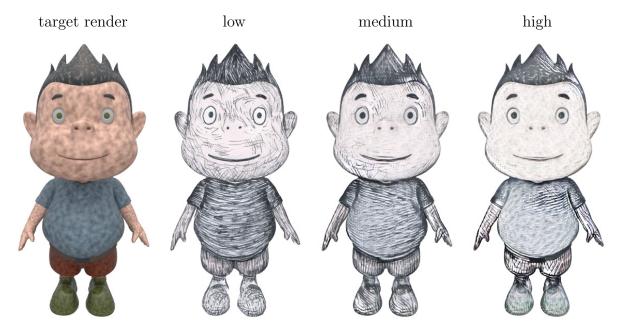


Figure 6.5: An illustration of a wash-out effect caused by adding an explicit content loss term [Kolkin et al. 2019] into our objective function (6.1). Target render stylized using model \mathcal{F} optimized on a stylized pair from Fig. 6.9 with low, medium, and high content loss weight. Note how style details deteriorate gradually with the increasing content loss. Source style \bigcirc Štěpánka Sýkorová, used with permission.

6.4 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. 2016b] instead of batch normalization. All layers used for Gram matrix computation are post-activated with ReLU to better incorporate nonlinearity. 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.

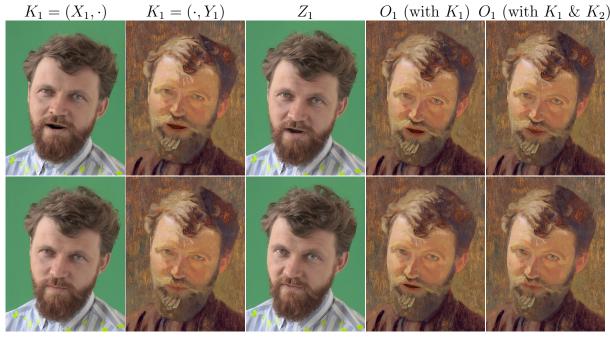
source frame source style target frame our approach Jamriška et al. Texler et al.



Figure 6.6: 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.

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 6.1 and 6.6 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. 6.7 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, 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. 6.8).

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.



 $K_2 = (X_2, \cdot)$ $K_2 = (\cdot, Y_2)$ Z_2 O_2 (with K_1) O_2 (with $K_1 \& K_2$)

Figure 6.7: 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 $X_1, X_2, Y_1, Y_2, Z_1 \ & Z_2 \ & O$ Muchalogy, used with permission.

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. 6.9). 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 6.9 and 6.10 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. 6.9) and implicitly maintains temporal consistency.

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

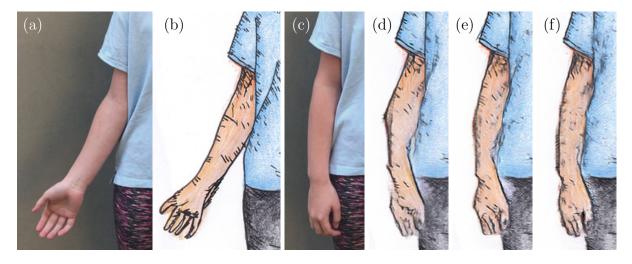


Figure 6.8: A different sampling strategy for a selection of frames in Z—a source frame from a sequence V (a) and its stylized counterpart (b) are used as K. Then weights of \mathcal{F} are optimized with K and Z, where Z contains all frames from V (d), 10% of uniformly sampled frames from V (e), and 10% of adaptively 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) (c) Štěpánka Sýkorová, used with permission.

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. 6.11 and 6.12 from the comparisons with [Liao et al. 2017; 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. 6.13 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 [Gatys et al. 2016; Kolkin et al. 2019] tend to preserve identity better, but lose style details. On the other hand, patch-based technique [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. Then, during the next video call \mathcal{F} is used to stylize captured video frames in real-time. See Fig. 6.14 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.

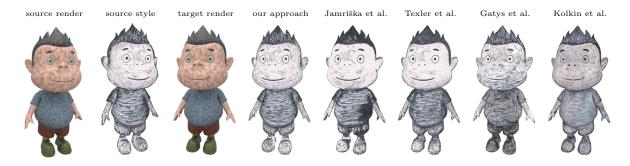
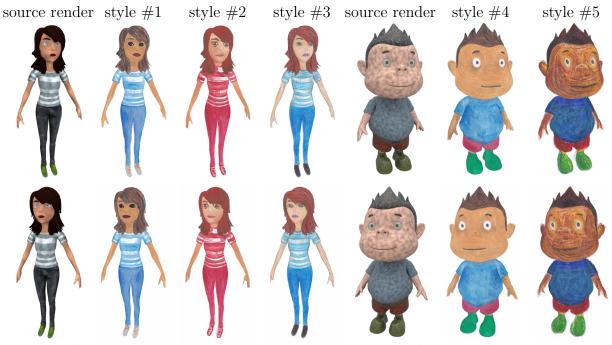


Figure 6.9: Stylization of 3D renders—a colored 3D model enhanced with an artificial noisy texture to avoid large flat regions (source render) is stylized at a selected viewpoint by an artist (source style). The network is then trained using the stylized pair and a set of additional renders of the same 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], and 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 (\bigcirc Štěpánka Sýkorová, used with permission.



target render output #1 output #2 output #3 target render output #4 output #5

Figure 6.10: 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) \mathbb{C} Štěpánka Sýkorová, used with permission.

6.4.1 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





our approach

Kolkin et al.

Figure 6.11: 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] and 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. 6.12. Source style \bigcirc Štěpánka Sýkorová, used with permission.



Figure 6.12: Panorama stylization results (cont.)—two additional artistic styles (source style) used to stylize the panorama shown in Fig. 6.11. Note how our approach (stylized panorama) handles also a higher level of abstraction (first row). Source style (top row) \bigcirc Jolana Sýkorová, used with permission.

(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

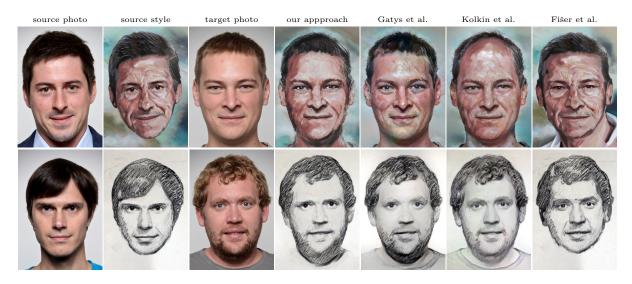
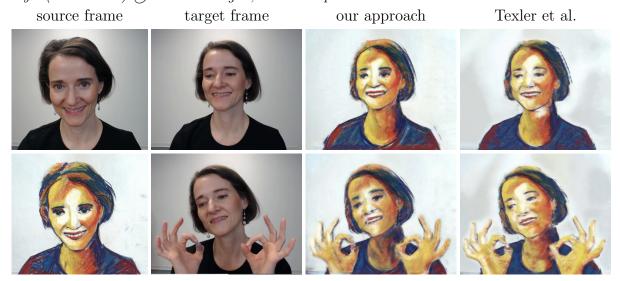


Figure 6.13: Stylization of portraits—a portrait photo (source photo) taken from a set of portraits captured under similar lighting conditions is stylized by an artist (source style). The network is then trained on the stylized pair and other portraits from the original set (target photo). Once trained 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] and Kolkin et al. [2019]) or have difficulties to retain identity (Fišer et al. [2017]). Source style (top row) \bigcirc Graciela Bombalova-Bogra and style (bottom row) \bigcirc Adrian Morgan, used with permission.



stylized frame

Figure 6.14: 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 lighting 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.

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.

The measured preference scores of our method compared to other techniques can be seen in Fig. 6.17. 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 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.

6.5 Limitations and Future Work

While our approach improves on current state-of-the-art in example-based stylization, we have observed some limitations in how it can be applied.

The most important limitation 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 of Texler et al. [2020b] with the computation of VGG loss which needs to be performed in a full-frame setting.

Due to the usage of relatively computationally expensive neural network model, the maximum resolution is limited. While we are able to generate output images with resolutions greater than method of Texler et al. (e.g. 768×768 vs. 512×512), it is still significantly lower than what patch-based methods [Jamriška et al. 2019] are capable of. As a future work we envision to alleviate this drawback by combining our neural approach with patch-based technique of [Texler et al. 2020a].

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.

A key advantage of our approach over current state-of-the-art in example-based video stylization [Jamriška et al. 2019; Texler et al. 2020b] 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. 6.1). 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. 6.15ii). 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. 6.16).

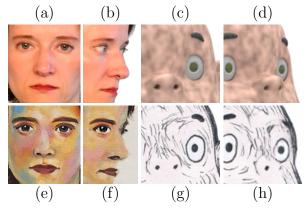
Most significantly, the method does not seem to generalize very well for completely generic use cases, for example in Fig. 6.15i, where input images are sampled from different

6.6. CONCLUSION



target photo stylized output target photo stylized output

(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. 6.13.



(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 6.15: Illustration of common limitations of our method.



Figure 6.16: 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. 6.1) 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.

underlying distributions. Thus, the set of potential applications is limited to groups of images of visually similar settings created under comparable conditions.

6.6 Conclusion

We presented an approach of semantically meaningful style transfer that can leverage a limited number of paired exemplars to stylize a broader set of target images having similar content to the examples. We optimize weights of an existing image-to-image translation network by minimizing a novel kind of objective function that considers the consistency among the provided stylized pairs as well as discrepancy between VGG features of style exemplars and a subset of stylized target images.

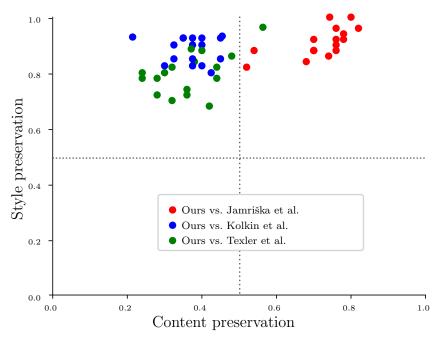


Figure 6.17: Results of perceptual study—each point represents aggregated votes over a group of 10 participants. On the x axis we depict the percentage of answers in favor of content preservation of our method while on the y axis we show the style reproduction percentage. Comparisons were performed with 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.

Thanks to this combination, our approach can better preserve style details even when the target images' content differs significantly from the style exemplar. Moreover, our method implicitly maintains temporal consistency in the video stylization scenario, which needs to be treated explicitly in previous techniques. We demonstrated the benefits of our approach in numerous practical use cases, including style transfer to videos and faces, auto-painting of panorama images, and real-time stylization of 3D models and video calls.

Chapter 7

ChunkyGAN: Real Image Inversion via Segments

7.1 Introduction

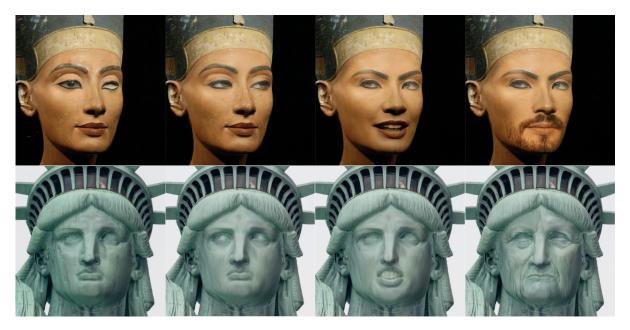


Figure 7.1: Real image manipulation examples created interactively using our method. The left-most images are the original photographs, the remaining columns show following edits: changing gaze direction, opening mouth, growing a beard and aging. Source images: Shutter-stock

The increasing ability of GANs to generate images virtually indistinguishable from real photographs [Karras et al. 2020; 2021], has created a new paradigm for image editing. In this paradigm, one first estimates a latent code for the network that best reconstructs the input image [Karras et al. 2019; Wu et al. 2021], and then manipulates this latent code in specific ways to create particular variations of the input image. With a knowledge of which directions in latent space of a particular generator encode which properties of the output image, it is possible to perform high-level semantic editing of the appearance of

the input photo while retaining the original visual features, e.g., adding more hair to a bald person while retaining their identity [Tov et al. 2021; Patashnik et al. 2021].

Due to the nature of adversarial training, a well-trained generator transforms any latent code drawn from the trained distribution into a plausible output, but mapping of an arbitrary in-domain image to a latent code might be difficult or even not possible. Existing methods address this by instead projecting into deeper spaces which makes accurate reconstruction easier, but weakens the original guarantee that every code maps to a plausible output, meaning that manipulated results may be out of domain and visually appear broken. This means there is an inherent trade-off between ease and accuracy of reconstruction, and quality of edited outputs [Tov et al. 2021], and existing methods perform on the spectrum of this trade-off. For example in StyleGAN2 [Karras et al. 2020], the original input code $z \in \mathbb{R}^{512}$ is transformed into a latent vector $\mathcal{W} \in \mathbb{R}^{512}$ which is easy to edit but difficult to reconstruct, whereas Abdal et al. [2019] use $\mathcal{W}^+ \in \mathbb{R}^{18 \times 512}$ that has enough degrees of freedom to provide good reconstruction, but is more difficult to manipulate.

This issue becomes much more apparent when we examine examples that are indomain, but far from typical. For example in the case of StyleGAN trained on a dataset of faces, we may consider human faces with unique features or accessories that do not appear in training datasets such as CelebA [Lee et al. 2020] or FFHQ [Karras et al. 2019], such as bindis, unusual glasses, heavy occlusions, etc. In these cases even techniques that have greater flexibility such as S-space [Wu et al. 2021] usually fail.

The source of much of these difficulties are two underlying assumptions: that there exists a single latent code that exactly or almost exactly reconstructs the target image, and that the manifold of representative images is nearly convex with respect to finding such a latent code. But because the number of output pixels is much higher than the number of degrees of freedom in the latent space, we may view the reconstruction problem as overdetermined, and although the aggregated reconstruction loss has local minima that can be found, a minimum for the entire image is not necessarily a minimum for all its regions. In practice, this means that the code retrieval problem is difficult and the solutions we arrive at are in effect suboptimal. In this paper we propose to resolve this difficulty by relaxing exactly these assumptions. We search not for a single latent code to represent the entire image, but rather a vector of latent codes, each corresponding to a segment of the image, such that when assembled they resemble the original image as closely as possible (see Fig. 7.2).

Since each latent code is then estimated for a much lower dimensional target, each of the regional subproblems become less overdetermined, which makes for an easier optimization problem. This in turn means that we can achieve much lower total error and thus more accurate reconstruction of the original. Besides superior accuracy and greater ability to generalize to the out-of-domain features, the segment-based nature of our method also allows for strictly localized edits, either based on segmentation generated automatically as a by-product of our method, or based on user-specified segments. Thanks to that property, visual content in different segments remains intact and thus helps retain the fidelity of the original photo. This leads to an interesting novel interactive scenario where the user adaptively applies individual local modifications in sequence to achieve a desired output that would normally be difficult to obtain using global manipulation techniques (see examples in Fig. 7.1). We demonstrate the power of our approach in various use cases that would be difficult to achieve using current state of the art. Moreover, a great advantage of our approach is that it does not replace previous methods but rather serves as a complementary part that, when plugged in, enables even better results than those produced by the technique applied in isolation.

7.2 Related Work

State-of-the-art approaches to finding suitable latent codes for the input image can be broadly split into two major categories: direct optimization and encoder-based techniques.

The first category takes into account the fact that the generator network is differentiable function on its own and thus gradient descent can be used to move from a real image into its latent code [Lipton and Tripathi 2017; Huh et al. 2020; Kang et al. 2021; Xu et al. 2021]. This typically leads to an inversion which is close to the original, however, since constraining the optimization to search across the manifold of naturally looking latent codes is nontrivial, the resulting projection is usually difficult to manipulate.

The other category relies on training an encoder which predicts the specific latent code given an image, using generated samples as training data [Zhu et al. 2016; Bau et al. 2019]. Tov et al. [2021] show that the encoder can learn to embed the real image into the natural manifold much closer than optimization methods, it does, however, often come at the cost of overall reconstruction quality, even considering multi-pass iterative techniques [Alaluf et al. 2021] or a modulation of StyleGAN weights [Alaluf et al. 2022; Dinh et al. 2022].

Both of these approaches, therefore, are characterized by an important trade-off between faithfulness to the original image and the ability to perform editing operations on the projected latent code. Hybrid approach has also been proposed, such as the one by Zhu et al. [2020], in which the direct optimization method is initialized by latent code proposed by a trained encoder, striking a better balance on the trade-off chart, however, the final result is far from ideal in either axis.

The trade-off itself is also not one dimensional. As the representation of the latent code turns into the final image via operations inside the generator network, it becomes easier to invert images into intermediate representations, at the cost of increased dimensionality, making editing more difficult. Recent work [Zhu et al. 2021; Yao et al. 2022; Kang et al. 2021] tries to exploit this knowledge by imposing constraints like segmentation on relatively high-level, spatial representations, leading to solutions that can create highquality inversions at the cost of restricting the set of possible edits.

Ling et al. [2021] presented EditGAN that enables to edit images by altering their segmentation masks. In contrast to our technique EditGAN can only change shape and relative position of selected regions. There is no control over the content generated inside the edited area, and it is also challenging to perform global edits. Moreover, EditGAN uses only a single latent code with lower expressive power while relying on a pre-trained DatasetGAN model [Zhang et al. 2021] that jointly generates images and their corresponding semantic segmentations. In our approach, each region have its own latent code, can be added on the fly at arbitrary locations and subsequently edited.

In StyleFlow, Abdal et al. [2021] use continuous normalizing flows in the latent space that are conditioned by various attribute features. This enables edit disentanglement comparable to our approach that is, however, redeemed by lower reconstruction quality. Moreover, StyleFlow also requires pre-trained classifiers to find the disentangled attributes along which the edits are performed.

Roich et al. [2021] propose that it is possible to fine-tune the generator network itself to improve the reconstruction quality while retaining the editability offered by a natural latent code. While their technique provides a well-rounded solution to both inversion accuracy and latent code editability, it requires fitting and storing per-image generator network, making it more resource-intensive and less suitable for downstream tasks.

In the earlier version of our method [Futschik et al. 2021b], segmentation-based inversion was developed for user-assisted local editing. In this extended version, we introduce joint optimization framework that enables automatic projection of the entire image while refining the shape of individual segments.

7.3 Our Approach

Our method accepts a real image I and reconstructs it as a vector of segmentation masks $S = \{S_i\}_{i=1}^n$, where pixel values range continuously from 0 to represent fully outside and 1 fully inside, and a vector of corresponding per-segment latent codes $X^I = \{X_i^I\}_{i=1}^n$. The masks are constrained so that they per-pixel sum up to 1. These latent codes are interpreted as images using a shared image generator G^I and the output image is obtained by pixel-wise linear blending, visualized in Fig. 7.2:

$$O(X^{I}, S) = \sum_{i=1}^{n} G^{I}(X_{i}^{I}) \cdot S_{i}.$$
(7.1)

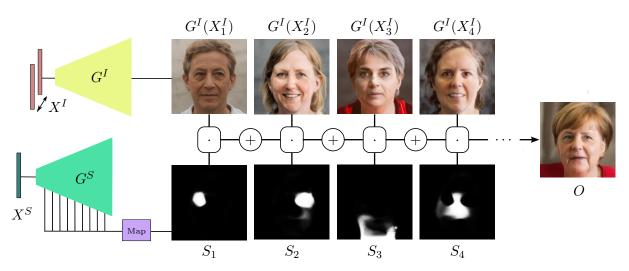


Figure 7.2: ChunkyGAN flowchart—the output image O computed as a weighted combination of n images generated by a network G^I given a set of n latent codes X^I . Weights are specified by a set of n segmentation masks S that can be specified manually or generated automatically by a segmentation network G^S using a latent code X^S . Source image: Raimond Spekking / CC BY-SA 4.0 (via Wikimedia Commons)

This expression is trivially differentiable with respect to both S and X, and is optimized with respect to some dissimilarity measure between I and the composite O just like in a single-segment reconstruction scenario. Unless otherwise specified, in this paper we optimize with respect to the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ of Zhang et al. [Zhang et al. 2018].

7.4. EVALUATION

Because the semantic segmentation is not universal and can vary dramatically between individual faces, it is necessary to optimize the masks as well. Optimizing them on a per-pixel basis would be memory intensive and would not take advantage of the domain knowledge we have for the problem. Therefore, we use a mask generator G^S to generate them from a segment latent code X^S , i.e., $S_i = G^S(X^S)_i$. In this work, we use a segment generator network based on DatasetGAN [Zhang et al. 2021]. It consists of StyleGAN2 generator and a mapping network trained on a modest dataset (a few tens of images) of randomly generated StyleGAN2 images annotated by example based synthesis [Fišer et al. 2017], using a single manually annotated image as exemplar.

To this end, the canonical form of our optimization problem is as follows:

$$\min_{X^S, X^I} \mathcal{L}_{\text{LPIPS}} \left(I, \ \sum_{i=1}^n G^I(X_i^I) \cdot G^S(X^S)_i \right) + \lambda_{reg} \sum_{i=1}^n \|X_i^I - X_\mu^I\|_2^2, \tag{7.2}$$

where the first term measures reconstruction loss and the second term penalizes dispersion among the latent codes, measured as sum of squared deviations from the mean code X^{I}_{μ} . Such regularization helps avoid mutually distant latent codes that do not produce realistic images. This is not typically a problem in the projection step, but during manipulation distant codes may diverge in appearance more quickly. This is caused by limitations in visual coherence in the pre-trained editing directions.

Our approach is orthogonal to the choice of the latent space of the X codes. In general, it can be any combination of common latent spaces that allows compact encoding of the input image. In the case of StyleGAN [Karras et al. 2019; 2020], we consider \mathcal{W} , \mathcal{W}^+ [Abdal et al. 2019], and \mathcal{S} -space [Wu et al. 2021], however, any previously published, potentially newly developed or a mixture of methods can be used. In fact, our method is a complementary extension that could help achieve better results regardless of the selected projection method.

In Fig. 7.3, we show an example of the optimization (per Equation 7.2) progression, starting from mean latent codes until convergence. Note that the segments tend to align with semantic facial features.

The processing speed of the optimization process relies on the number of segments and the number of optimization steps. When a joint multi-segment optimization with the DatasetGAN is performed the projection can take several minutes. However, during the interactive editing (as seen in our supplementary video), where segments are specified by the user one-by-one, the method runs at interactive rates on the GPU (a few seconds).

7.4 Evaluation

To validate our approach we performed two quantitatively and qualitatively evaluated experiments. In the first experiment we validate whether the projections produced by our method can reproduce target photos with greater fidelity when compared to standard projection techniques. In the second experiment we demonstrate the ability of our approach to edit projected images by manipulating estimated latent codes and compare the fidelity of the resulting edits with standard techniques. Finally, we compare our approach with current optimization-based and encoder-based projection techniques.



Figure 7.3: Progression of the optimization. Images and color-coded segmentation maps for iterations 1, 5, 9, 15, 23, 37, 500. Source image: Adobe Stock

7.4.1 Fidelity of projected images

To quantitatively evaluate fidelity of projected images we took the first 100 images from CelebA dataset [Lee et al. 2020] excluding blurred images and those with people wearing additional props such as hats or glasses. We then projected all those images globally into \mathcal{W} , \mathcal{W}^+ , \mathcal{S} -space, and also locally using our method. When using \mathcal{W}^+ , we show both cases, with ($\lambda_{reg} = 1$) and without ($\lambda_{reg} = 0$) the regularization. For all projections we measured the LPIPS, identity (measured as cosine distance between ArcFace descriptors [Deng et al. 2019]), and L_2 loss with respect to the original target photos.

Projection	LPIPS	Identity	L_2
\mathcal{W}	0.4190 ± 0.0363	0.1745 ± 0.1328	0.0725 ± 0.0699
Ours in \mathcal{W}	0.3697 ± 0.0396	0.1384 ± 0.1117	0.0481 ± 0.0289
\mathcal{W}^+	0.3675 ± 0.0387	0.1195 ± 0.1047	0.0436 ± 0.0623
Ours in \mathcal{W}^+	0.3194 ± 0.0365	0.0937 ± 0.0855	0.0207 ± 0.0151
Ours in \mathcal{W}^+ reg.	0.3330 ± 0.0350	$\textbf{0.0894} \pm \textbf{0.074}$	0.0217 ± 0.0130
S	0.3577 ± 0.0397	0.1070 ± 0.0965	0.0328 ± 0.0188
Ours in \mathcal{S}	0.3572 ± 0.0401	0.1053 ± 0.0928	0.0319 ± 0.0187

Table 7.1: Projection fidelity. Losses were measured between the projected and the original image for each of the projection methods. Each cell reports the loss averaged over the CelebA subset along with the standard deviation. Our method significantly outperforms the baseline methods in all latent spaces for all losses.

The resulting numbers are shown in 7.1 which shows losses averaged over all 100 images with corresponding standard deviations. Those confirm that on average our method outperforms global projection methods significantly. This fact is visually apparent from scatter plots shown in Fig. 7.4 where each point corresponds to an image and its coordinates encode the LPIPS losses for the global and the segmented projection respectively. Red line depicts the margin where losses for both projection methods are equal.

Since the best projection is achieved by our method in \mathcal{W}^+ , we select \mathcal{W}^+ as the default space for our method. The regularization slightly decreases the projection fidelity in terms of LPIPS, but improves the identity and editability, which is discussed in Sec. 7.4.2.

Because differences between the evaluated methods are difficult to observe in a typical case, we have for the purposes of qualitative evaluation of projection fidelity deliberately

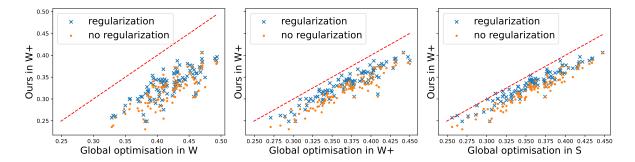


Figure 7.4: Projection fidelity – scatter plots. Our method is compared with global projections $(\mathcal{W}, \mathcal{W}^+, \mathcal{S}\text{-space})$. X and Y axis represent the LPIPS loss between the original image and the image projected globally and projected by our method in \mathcal{W}^+ respectively. Each point corresponds to one image from the CelebA subset, in blue and in orange with and without the regularization respectively. The red line delineates the equal LPIPS losses. Our method improves projection for all images in all tested latent spaces. The regularization slightly decreases the projection fidelity, but remains still better than global methods.

pre-selected a subset of hard-to-project images. Specifically, these were images that contain features uncommon in the standard datasets, e.g. bindis, face masks, asymmetric glasses, or occluded faces. For those examples all compared methods were initialized equally (using mean latent vector) and the corresponding projection results are presented in Figure 7.5. It is apparent that thanks to greater flexibility of our approach, more realistic projections can be achieved when compared to standard techniques. Moreover, a workable inversion can be obtained even on out-of-domain images as shown in Fig. 7.5 (two bottom rows).

7.4.2 Editability of projected images

Quantitative evaluation of editability was performed on the same set of CelebA images used for evaluation of projection fidelity. We pre-selected 4 semantic directions (gender, smile, age, and beard), changed all latent codes X in the same direction with the same magnitude, and finally measured the effect of the edits on identity.

Since the effect of unit strength manipulation along a pre-trained semantic direction can differ among latent spaces and the use of global/local projection, we calibrate the changes to make sure the effect on the manipulated image is equal. To do that we use an image classifier for each semantic direction. For each space and method, we measure image classifier responses while spanning the latent edit strength along a semantic direction for the entire dataset. We use linear regression to find the rate of change of the classifier response to the edit strength, and adjust the edit strength to be equal for all tested methods.

Table 7.2 shows a quantitative evaluation of the identity loss between the projected and edited images. It is apparent that the identity losses are the best for our method with the regularization engaged since regularization pushes the codes of all segment images towards latent areas where the linear latent manipulation works better. The results confirm that our method keeps the identity consistent during editing.

Regarding the reconstruction-editability trade-off [Tov et al. 2021], latent code regularization is essential in order to perform realistic edits. While our method without

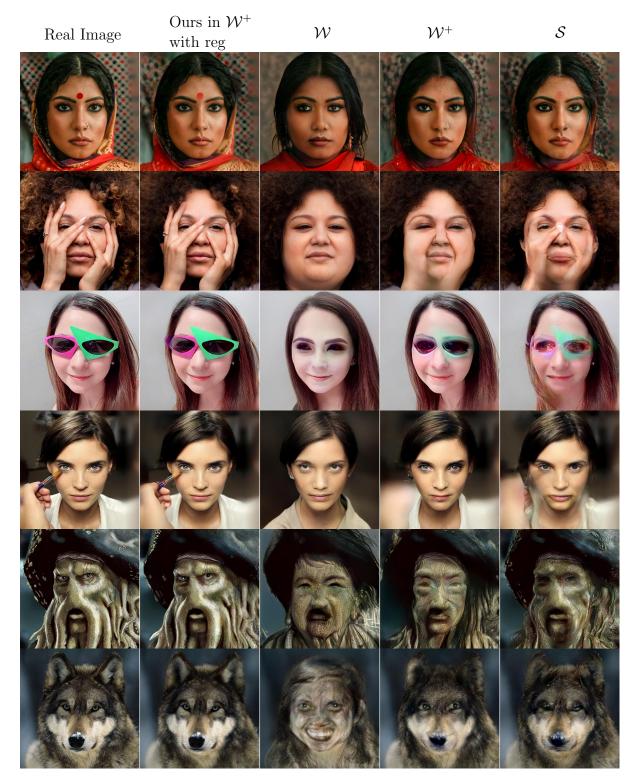


Figure 7.5: Qualitative assessment of projection fidelity on hard examples. All images were projected with regularization. For more examples refer to the supplementary material. Source images: Adobe Stock

		(a)				(b)		
	gender	\mathbf{smile}	age	beard	gender	\mathbf{smile}	age	beard
${\mathcal W}$	0.169	0.022	0.07	0.279	0.249	0.18	0.191	0.328
\mathcal{W}^+	0.209	0.02	0.095	0.296	0.256	0.128	0.171	0.325
Ours in \mathcal{W}^+	0.298	0.049	0.151	0.312	0.325	0.125	0.203	0.333
Ours in \mathcal{W}^+ reg.	0.126	0.018	0.069	0.091	0.169	0.099	0.129	0.144

Table 7.2: Identity preservation during editing. Identity loss was computed between the projected and the edited images (a), and between the original and the edited images (b). Our method with regularization outperforms all other methods.

regularization achieves better results in projection fidelity it performs poorly during editing. By adding the regularization term, projection fidelity slightly deteriorates, but the identity preservation during edits improves by a large margin. The editability can be observed during the classifier-based calibration; methods without regularization need much stronger edits in order to achieve the same editing effect.

For the qualitative evaluation we pre-selected images and directions (age and yaw) that would cause difficulties to standard techniques, i.e., the identity is not well preserved during editing. During the yaw manipulation using our method the segmentation masks were edited as well (the segmentation latent code was manipulated automatically in the same way as the images) to adjust the segments geometrically. Results are presented in Fig. 7.6 and 7.7. It is clearly visible that our method keeps the identity better. Fig. 7.7, a man wearing a mask is especially challenging. The global techniques are unable to project the image properly. Our method projects the image faithfully and moreover, the global edits still work. Note that these results were achieved fully automatically, neither manual adjustment of the segmentation partitioning nor any post-processing were applied for images in Fig. 7.6 and 7.7.

7.4.3 Comparison with current state-of-the-art

To demonstrate how our approach compares to current state-of-the-art in the optimizationbased and encoder-based techniques we performed various qualitative experiments seen in Figures 7.7 and 7.8. When compared to current best approaches based on optimization (Pivotal Tuning [Roich et al. 2021] and StyleFlow [Abdal et al. 2021]), our method achieves better or comparable projection quality while still being able to deliver compelling edits (c.f. Fig. 7.7). Our method also outperforms encoder-based techniques (HyperStyle [Alaluf et al. 2022], ReStyle [Alaluf et al. 2021], pSp [Richardson et al. 2021], and e4e [Tov et al. 2021]) with respect to the projection fidelity namely thanks to its ability to reproduce small details that are usually omitted by encoders (c.f. Fig. 7.8).

7.5 Applications

Aside from the fully automatic solution proposed in Section 7.3, our framework can also be extended to allow for interactive step-by-step manipulation in a few different ways. To facilitate this, we define the notion of a static mask S^X which defines an area of the image which is not changed during the optimization. In terms of our objective function,

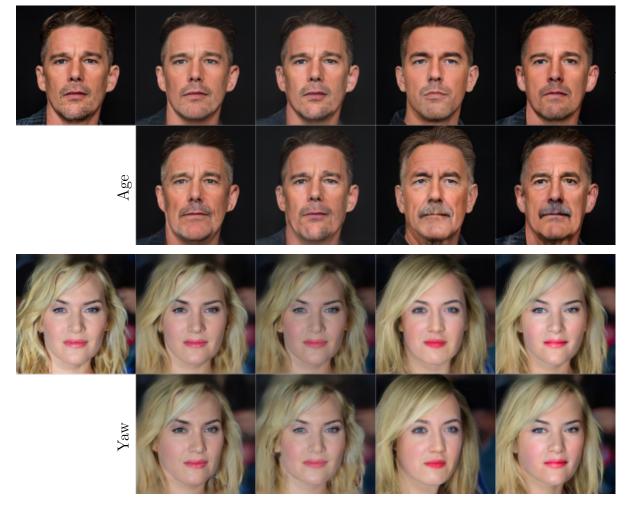


Figure 7.6: Global edits with the same effective strength. For our methods the latent codes of all segments were manipulated equally. Source images: Mingle Media TV (Kate Winslet), Neil Grabowsky / Montclair Film (Ethan Hawke)

this creates a mixed composite:

$$O(X^{I}, S, S^{X}, I) = S^{X} \cdot I + (1 - S^{X}) \cdot \sum_{i=1}^{n} G^{I}(X_{i}^{I}) \cdot S_{i}$$
(7.3)

In practice, for edits with small spatial extent it is often sufficient to reduce the number of segments being optimized to one, in which case there is no need to optimize S_i .

Using this static mask, instead of generating segment masks automatically, we allow the user to manually specify the region of interest. The user then runs the projection, edits the latent code, and produces an intermediate composite O which can then become a new I for next iteration. This user-driven iterative scheme is shown in Fig. 7.9. Such a workflow is intuitive for users as they can specify what they want to change, overview the resulting composition, and then possibly go back and revise their requirements by making additional changes in different regions.

When making the composite O from edited image, even when edits of X are consistent, continuity around boundaries may no longer be guaranteed. Small discrepancies are suppressed automatically thanks to blending with soft masks. When the edit produces more notable global color shift we use Poisson image editing [Pérez et al. 2003] to alleviate them. In most challenging scenario segment boundaries may start to interfere with newly synthesized salient features. In this case continuity can be enforced using a slightly modified version of our segmentation-based approach that will act as semantically meaningful hole-filling as illustrated in Fig. 7.10.



Figure 7.7: Challenging global edits. The first row depicts the original and the projected images using our approach with and without regularization, Pivotal Tuning [Roich et al. 2021], StyleFlow [Abdal et al. 2021], W and W^+ [Abdal et al. 2019]. The remaining two rows show resulting global edits of age. Source image: BlochWorld

Suppose we have a photo of a person (Fig. 7.10a) and the aim is to add glasses. We select a loose region S_1 around eyes (Fig. 7.10b) and run the local projection to get latent code X_1 that reproduces the original image within S_1 (Fig. 7.10b). Then we manipulate X_1 to add glasses, however, as visible in Fig. 7.10c the shape of S_1 is insufficient to encompass newly added content. To fix this discrepancy we let the user specify correction mask S_2 with two connected components (Fig. 7.10d) and refine X_1 to obtain a new code X_2 that will match the content within S_2 (green region). From the image generated by X_2 we then use the dark part that lies inside S_2 to make the final composite (Fig. 7.10e). The X_2 code in fact generates a semantically meaningful hole-filling that completes the missing part of glasses.

7.6 Limitations

While the multi-segment reconstruction is remarkably robust, and segmented editing produces superior results for spatially limited edits, we can experience incoherence between segments for global edits (e.g. age, yaw) with high strength. The reason for this is that the editing directions are local linear approximations of the property of interest on the latent manifold, and for higher edit strength this linearity assumption no longer applies. This issue is present also in single-code editing, where it may cause loss of identity which may be in some scenarios more tolerable. With multiple segments however, this is highlighted as a greater change resulting in individual segments to lose identity in different ways and therefore gives rise to incoherence. It only occurs in editing and not in reconstruction because in reconstruction the input image provides effective supervision to maintain coherence between segments.

The incoherence does not usually occur for easy-to-invert images and moderate edits, as seen in Fig. 7.6, but can be spotted in harder examples with a challenging global



Figure 7.8: Projection fidelity of our method with respect to the current state-of-the-art in encoder-based techniques: HyperStyle [Alaluf et al. 2022], ReStyle [Alaluf et al. 2021], pSp [Richardson et al. 2021], and e4e [Tov et al. 2021]. Source images: Ayush Kejriwal (bindi), BlochWorld (face mask)



Figure 7.9: Examples of local layered edits applied subsequently on a real photograph (a): changing gaze direction (b), adding smile (c), changing haircut and nose shape (d).

edit, as e.g., in Fig. 7.7 in Age+ of our method with regularization. Nevertheless, the small artifact on the mask shape, can be interactively removed by the hole-filling method demonstrated in Fig. 7.10.

As another option, this issue could be addressed by formulating and imposing an explicit segment coherence measure during editing, which can be done either locally, by measuring agreement between segments in their regions of overlap, or globally by e.g. an adversarial loss. Alternatively, instead of linear directions, one might train a separate model to explicitly encode a higher-order approximation of identity-preserving edit direction, which has the potential to also benefit vanilla methods under high edit strength.

7.7 Conclusion

We presented a new technique for image reconstruction and editing based on generative adversarial networks that subdivides the input image into a set of segments for which the corresponding latent vectors are retrieved separately. By so decomposing the problem,

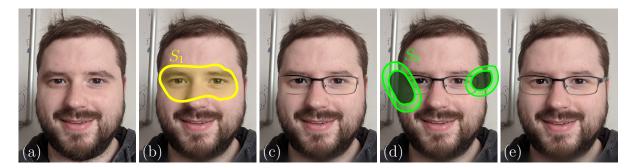


Figure 7.10: Enforcing continuity of inconsistent edits—a photo of a person to which we would like to add glasses (a), user-specified segmentation mask S_1 with a projection X_1 matching the original image (b), manipulating X_1 generates glasses that do not fit the shape of S_1 (c), a new mask S_2 is marked encompassing two discontinuous parts (d), a composite with a projected region S_2 where the new latent code X_2 is refined from X_1 to produce the dark region inside S_2 (e).

we facilitate more accurate reconstructions that better preserve the identity and visual appearance of facial images, especially in more challenging cases that are difficult to handle using state-of-the-art techniques.

We demonstrated the utility of this technique for both the base project-and-edit scenario and novel interactive sequential editing applications. As our approach provides measurable improvements while being easily combined with other techniques, we anticipate it will find a place in modern image editing tools.

Chapter 8

Conclusion

This thesis has introduced five new approaches that push the boundaries of machine learning in the field of artistic stylization, with a particular emphasis on style transfer. These approaches serve as fundamental building blocks that make it possible for digital artists to complete tasks that were previously tedious or even impossible, especially in real-time video style transfer and realistic face editing. In this chapter, we summarize our contributions and highlight the key achievements of our work. We also discuss concurrent work and explore potential directions for future research and development.

8.1 Summary

In Chapter 3, we presented our contribution for real-time facial stylization using a learning based approach, *FacestyleGAN: Real-Time Patch-Based Stylization of Portraits Using Generative Adversarial Network* [Futschik et al. 2019], work we presented as a technical paper at the Expressive 2019 conference. We have combined several existing techniques in this work to achieve our results. First, we have selected a large dataset of faces and automatically generated masks to remove unwanted elements from the images. We then utilized the method proposed by Fišer et al. [2017] to transfer texture from selected artistic facial paintings onto extracted face landmarks. This helped us to preserve identity as a weak prior. In addition, we have developed a state-of-the-art neural framework to extract the transfer function from the created pairs, which consists of a registered photo and its stylized counterpart in a given style. Our system runs in real-time and can achieve sufficient frames per second to be highly interactive. It can be used in various settings, such as online avatar stylization or as general entertainment.

In the domain of more general style transfer, Chapter 4 presented our work Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis [Texler et al. 2020a] published in Computers & Graphics journal. We have developed a novel approach to artistic stylization by combining the strengths of neural and patch-based methods. Neural techniques provide high-quality stylization at the global level, which we use as prior information for subsequent patch-based synthesis. By doing so, we are able to preserve the high frequencies of the original artistic media, resulting in stylized images with dramatically increased fidelity. Additionally, our method allows for the stylization of extremely large images with high visual quality, up to 340 Mpix. In our work, we also introduce a novel algorithm that directly uses responses from a pre-trained neural network to guide patch-based synthesis. This approach yields comparable visual quality to state-of-the-art neural style transfer, but with resolutions previously unachievable using such methods.

In Chapter 5, we introduced our publication Interactive Video Stylization Using Few-Shot Patch-Based Training [Texler et al. 2020b] presented at the SIGGRAPH 2020 conference and published in ACM Transactions on Graphics journal. In this publication, we introduce a refined training strategy for image-to-image translation networks, and we successfully apply it to the problem of keyframe based video stylization. We were able to develop a framework in which, using a single training pair, the model is trained from scratch on the order of seconds or minutes on a consumer-grade GPUs, and model inference runs in real-time. Our framework appears to preserve temporal coherency without the need to process previous frames, and thus allows for arbitrary order or parallel processing. Furthermore, it implicitly handles multiple keyframes and produces consistent results without any explicit blending operation, unlike previous patch-based approaches. This allowed us to come up with various interactive scenarios that were not possible before, e.g., a real-time style transfer to a live video stream that uses an exemplar that is being simultaneously painted on a captured canvas.

Chapter 6 describes *Style Transfer with Limited Auxiliary Pairing*, or STALP for short, presented at Eurographics 2021 and published in Computer Graphics Forum. Our method addresses the same issue of keyframe based video stylization, but sets out to achieve two distinct objectives: enhancing the visual quality of style transfer and addressing the challenge of temporal stability. As in previous approaches, STALP represents the style transfer function as a neural network model that is specifically trained from scratch for each transfer task. However, we introduce a crucial weak loss term during training, which promotes style consistency across multiple input images. This innovation allows us to demonstrate remarkable temporal stability in the resulting sequences, even over complex transformations such as occlusions, rotations, or large-scale changes. Our method sets a new standard in terms of reducing the number of keyframes required for consistent stylization of video sequences. Furthermore, we show that our approach has potential for broader applications, such as consistently stylizing panoramic images or a dataset of portrait images taken under similar conditions.

Lastly, Chapter 7 presents our most recent work, *ChunkyGAN: Real Image Inver*sion via Segments [Šubrtová et al. 2022], which departs from the painterly focus of our stylization work, and veers into the world of photorealistic stylization. This work was presented at the European Conference on Computer Vision 2022, and offers an answer to the problem of changing localized features in real photographs using pretrained GANs. We demonstrate this ability on human faces using StyleGAN [Karras et al. 2020]. Compared to similar work, our approach is able to very closely retain the identity of the person found in the image, thanks to a clever combination of automatic segmentation and image space blending of per-segment projected parts of the original image, and provides muchdesired localized control for GAN-based editing. Although this approach is not ideal for large scale edits such as excessive aging, geometric transformations or drastic hairstyle changes, we argue that artists who make use of such systems are much more likely to want to have control over more localized changes such as fixing skin imperfections or slightly changing facial expressions.

8.2 Concurrent and Future Work

Stylization remains an attractive topic, from the point of view of research and products alike. It should therefore not be surprising that many novel methods and techniques are introduced every year. In the domain of general purpose style transfer, the recently proposed method of Kolkin et al. [2022] has shown significant improvements on still images, most notably in terms of textural quality, achieved by improving the fundamental ideas of Liao et al. [2017] of replacing neural features of the content image by the features produced from the style image and then optimizing the result to produce such arrangement of neural features. Many other works present modifying techniques applicable to existing methods, in order to, for example, stabilize style transfer [An et al. 2021] through reversible flows or move the content preservation problem into the feature space of style transfer models via attention maps [Park and Lee 2019].

The other areas seeing particularly exciting developments concurrently with our work include, for instance, applications of NeRF-like frameworks for stylization of videos or complete 3D environments. Even though they rarely preserve the planarity of style exemplars, the methods produce very appealing results that manage to reconcile the disconnect between the reality of our 3D world and artistic renditions, as shown by Nguyen et al. [2022]. In the work of Huang et al. [2022] the authors show remarkable consistency in stylizing 3D scenes using such a method, allowing arbitrary user-guided exploration of artistically stylized spaces. This line of work could e.g., have future applications in making virtual worlds more approachable to more sensitive users by making any visual artifacts easier to accept.

Furthermore, text-conditioned image generators like Stable Diffusion [Rombach et al. 2021], Imagen [Saharia et al. 2022] or DALLE-2 [Ramesh et al. 2022] have shown impressive ability to generate countless types of content, and many have noticed tendency to generate more believable images when the model is conditioned for artistic outputs, certainly in part due to more forgiving nature of some styles. It is then unsurprising that a large amount of effort is currently being put into leveraging these general purpose models for artistic stylization, either by finding a conditioning paired with a given style (trial and error, image inversion, image encoder), or by fine-tuning the model on a small set of images that are close to the desired style [Ruiz et al. 2022]. Although it is possible, the approaches using these general generative models will typically have difficulty applying the chosen style onto an existing image, which is commonly the end goal for stylization tasks. That is why method of Brooks et al. [2022] combines the written instructions prompt with the given input image to perform an editing operation that attempts to preserve the structure of the input image, which is much closer to the traditional formulation of artistic stylization presented in this thesis.

As past and contemporary efforts illustrate, there is little doubt that future work in stylization and style transfer will involve leveraging machine learning and incorporating novel results from other fields such as computer vision – making use of automatic segmentation, object recognition, or classification to improve results by exploiting enormous datasets that would be difficult to design for stylization specifically. Moreover, combining the image input with a textual specification is a promising direction, thanks to the fact that text prompt, unlike image editing, is so convenient and straight-forward to experiment with if the system runs at interactive speeds. At the same time, there are yet untapped possibilities for style transfer and stylization; as virtual worlds become more common, stylization will unquestionably find its place to enable user customization and unlock creating unique-looking content material, especially in the subsection of *avatars* that aside from representing the values of a user, are meant to have sentimental value that is more achievable with artistic visuals. Furthermore, as more features become possible in real-time either through new breakthroughs or hardware acceleration, style transfer will surely be a natural fit for creating distinctive appearances for computer games.

In conclusion, stylization has garnered significant attention from both artists and researchers alike, and serves as a remarkable gateway towards computer-assisted or generated imagery. Our thesis has contributed a set of methods that aid artists in becoming more efficient, creative, and comfortable with interactive workflows, while also fostering an environment for experimentation. This field is currently in an exciting state of rapid progress, and artists are already embracing these tools as an integral part of their creative process. We believe our contributions further the state-of-the-art in this area, and are confident in the positive impact it will have on creating future captivating experiences.

References

- Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to embed images into the StyleGAN latent space? In *Proceedings of IEEE International Conference on Computer Vision*, 2019.
- Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. StyleFlow: Attributeconditioned exploration of stylegan-generated images using conditional continuous normalizing flows. ACM Transactions on Graphics, 40(3):21, 2021.
- Adéla Subrtová^{*} and David Futschik^{*}, Jan Cech, Michal Lukáč, Eli Shechtman, and Daniel Sýkora. ChunkyGAN: Real image inversion via segments. In *Proceedings of European Conference on Computer Vision*, pages 189–204, 2022. *equal contribution.
- Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. ReStyle: A residual-based StyleGAN encoder via iterative refinement. In *Proceedings of IEEE International Conference on Computer Vision*, pages 6711–6720, 2021.
- Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. HyperStyle: StyleGAN inversion with hypernetworks for real image editing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 18511–18521, 2022.
- Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 862–871, 2021.
- David Bau, Jun-Yan Zhu, Jonas Wulff, William S. Peebles, Bolei Zhou, Hendrik Strobelt, and Antonio Torralba. Seeing what a GAN cannot generate. In *Proceedings of IEEE International Conference on Computer Vision*, pages 4501–4510, 2019.
- William Baxter, Jeremy Wendt, and Ming C. Lin. IMPaSTo: A realistic, interactive model for paint. In Proceedings of International Symposium on Non-Photorealistic Animation and Rendering, pages 45–56, 2004.
- 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.
- Itamar Berger, Ariel Shamir, Moshe Mahler, Elizabeth J. Carter, and Jessica K. Hodgins. Style and abstraction in portrait sketching. *ACM Transactions on Graphics*, 32(4):55, 2013.
- 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.
- 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.
- Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022.
- Matthew Brown and David G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.
- J. R. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. ACM Transactions on Graphics, 2(4):217–236, 1983.
- Santiago Calvo, Ana Serrano, Diego Gutierrez, and Belen Masia. Structure-preserving style transfer. In *Proceedings of Spanish Computer Graphics Conference*, pages 25–30, 2019.
- Kaidi Cao, Jing Liao, and Lu Yuan. Carigans: Unpaired photo-to-caricature translation. ACM Transactions on Graphics, 37(6):244:1–244:14, 2018.
- 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, 2017a.
- Hong Chen, Lin Liang, Ying-Qing Xu, Heung-Yeung Shum, and Nan-Ning Zheng. Example-based automatic portraiture. In *Proceedings of Asian Conference on Computer Vision*, pages 171–178, 2002a.
- 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, 2002b.

REFERENCES

- 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.
- Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2516–2525, 2017b.
- 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-toimage translation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 8789–8797, 2018.
- Nelson Chu and Chiew-Lan Tai. Moxi: Real-time ink dispersion in absorbent paper. ACM Transactions on Graphics (TOG), 24:504–511, 07 2005. doi: 10.1145/1073204.1073221.
- Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. In 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.
- Jiankang Deng, J. Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 4685–4694, 2019.
- Valentin Deschaintre, George Drettakis, and Adrien Bousseau. Guided fine-tuning for large-scale material transfer. In *Computer Graphics Forum*, volume 39, pages 91–105. Wiley Online Library, 2020.
- R Dinesh Kumar, E Golden Julie, Y Harold Robinson, S Vimal, Gaurav Dhiman, and Murugesh Veerasamy. Deep convolutional nets learning classification for artistic style transfer. *Scientific Programming*, 2022:1–9, 2022.
- Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. HyperInverter: Improving stylegan inversion via hypernetwork. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 11389–11398, 2022.
- 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.

- Stephen DiVerdi, Aravind Krishnaswamy, and Sunil Hadap. Industrial-strength painting with a virtual bristle brush. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, page 119–126, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450304412. doi: 10.1145/1889863. 1889889. URL https://doi.org/10.1145/1889863.1889889.
- Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. CoRR, abs/1610.07629, 2016.
- 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.
- 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: Examplebased 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, 2021a.
- David Futschik, Michal Lukáč, Eli Shechtman, and Daniel Sýkora. Real image inversion via segments. In *arXiv*, number 2110.06269, 2021b.
- 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.
- Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3985–3993, 2017.

REFERENCES

- 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. In *Computer Graphics Forum*, volume 40, pages 428–446. Wiley Online Library, 2021.
- 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.
- 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.
- 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. Painterly rendering with curved brush strokes of multiple sizes. In SIGGRAPH Conference Proceedings, pages 453–460, 1998.
- Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In SIGGRAPH Conference Proceedings, pages 327–340, 2001.
- Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. In *Computer Graphics Forum*, volume 41, pages 117–128. Wiley Online Library, 2022.
- 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.
- Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. 2022.
- Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 17–34, 2020.

- 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.
- 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, Sárka 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.
- 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.
- Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. GAN inversion for out-of-range images with geometric transformations. In *Proceedings of IEEE International Conference on Computer Vision*, pages 13941–13949, 2021.
- 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.
- Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In Proceedings of Conference on Neural Information Processing Systems, 2021.
- 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 (TOG), 40(6):1–12, 2021.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.
- Nicholas Kolkin, Michal Kucera, Sylvain Paris, Daniel Sykora, Eli Shechtman, and Greg Shakhnarovich. Neural neighbor style transfer. *arXiv e-prints*, pages arXiv–2203, 2022.
- 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. Pointbased neural rendering with per-view optimization. In *Computer Graphics Forum*, volume 40, pages 29–43. Wiley Online Library, 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, 2019.
- 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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13999–14009, 2022.
- 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, volume 80, 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. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *Proceedings of European Conference on Computer Vision*, pages 702–716, 2016b.
- 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, 2016c.
- 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.

- Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. EditGAN: High-precision semantic image editing. In *Proceedings of Conference* on Neural Information Processing Systems, 2021.
- Zachary C. Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. In *Proceedings of International Conference on Learning Representations*, 2017.
- 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.
- Jingwan Lu, Connelly Barnes, Stephen DiVerdi, and Adam Finkelstein. RealBrush: painting with examples of physical media. *ACM Transactions on Graphics*, 32(4):117, 2013.
- 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.
- Ming Lu, Feng Xu, Hao Zhao, Anbang Yao, Yurong Chen, and Li Zhang. Exemplar-based portrait style transfer. *IEEE Access*, 6:58532–58542, 2018.
- 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.
- 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), 2022.
- Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.
- Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5880–5888, 2019.

- Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.
- 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.
- Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Style-CLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of IEEE International Conference on Computer Vision*, pages 2085–2094, 2021.
- Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. ACM Transactions on Graphics, 22(3):313–318, 2003.
- Adam Platkevič, Cassidy Curtis, and Daniel Sýkora. Fluidymation: Stylizing animations using natural dynamics of artistic media. In *Computer Graphics Forum*, volume 40, pages 21–32. Wiley Online Library, 2021.
- Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In SIGGRAPH, pages 581–586, 2001.
- 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.
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- Dongyu Rao, Xiao-Jun Wu, Hui Li, Josef Kittler, and Tianyang Xu. Umfa: a photorealistic style transfer method based on u-net and multi-layer feature aggregation. *Journal* of *Electronic Imaging*, 30(5):053013–053013, 2021.
- Max Reimann, Mandy Klingbeil, Sebastian Pasewaldt, Amir Semmo, Matthias Trapp, and Jürgen Döllner. Locally controllable neural style transfer on mobile devices. *The Visual Computer*, pages 1–17, 2019.
- Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: A stylegan encoder for image-to-image translation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 2288–2296, 2021.
- Daniel Roich, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. In *arXiv*, number 2106.05744, 2021.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

- 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.
- Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. 2022.
- David Salesin. Non-photorealistic animation & rendering: 7 grand challenges. Keynote talk at NPAR, 2002.
- Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In 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.
- 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.
- 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.
- 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 Con*ference on Computer Vision and Pattern Recognition, pages 2377–2386, 2019b.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for largescale image recognition. *CoRR*, abs/1409.1556, 2014.
- 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.
- 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, 2022, 2022.
- 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, Sárka 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. In ACM SIGGRAPH 2022 Conference Proceedings, pages 1–10, 2022.
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for StyleGAN image manipulation. *ACM Transactions on Graphics*, 40(4): 133, 2021.
- Patrick Tresset and Frédéric F. Leymarie. Generative portrait sketching. In Proceedings of International Conference on Virtual Systems and Multimedia, pages 739–748, 2005.
- Matthew Turk and Alex Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, 1991.

- Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016a.
- Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016b.
- Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Image shape manipulation from a single augmented training sample. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13769–13778, 2021.
- Adéla Šubrtová, David Futschik, Jan Čech, Michal Lukáč, Eli Shechtman, and Daniel Sýkora. ChunkyGAN: Real image inversion via segments. In Proceedings of European Conference on Computer Vision, pages 189–204, 2022.
- 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, 2013a.
- 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.
- Tinghuai Wang, John P. Collomosse, Andrew Hunter, and Darryl Greig. Learnable stroke models for example-based portrait painting. In *Proceedings of British Machine Vision Conference*, 2013b.

REFERENCES

- 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. pages 2561–2571, 2019c.
- 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.
- Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for StyleGAN image generation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 12863–12872, 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.
- Li Xu, Jimmy S. J. Ren, Qiong Yan, Renjie Liao, and Jiaya Jia. Deep edge-aware filters. In *JMLR Workshop and Conference Proceedings*, pages 1669–1678, 2015.
- Yangyang Xu, Yong Du, Wenpeng Xiao, Xuemiao Xu, and Shengfeng He. From continuity to editability: Inverting GANs with consecutive images. In Proceedings of IEEE International Conference on Computer Vision, pages 13910–13918, 2021.
- Ming Yang, Shu Lin, Ping Luo, Liang Lin, and Hongyang Chao. Semantics-driven portrait cartoon stylization. In *Proceedings of International Conference on Image Pro*cessing, pages 1805–1808, 2010.
- Shuai Yang, Zhangyang Wang, and Jiaying Liu. Shape-matching gan++: Scale controllable dynamic artistic text style transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3807–3820, 2021.
- Xu Yao, Alasdair Newson, Yann Gousseau, and Pierre Hellier. Feature-style encoder for style-based GAN inversion. In *arXiv*, number 2202.02183, 2022.
- Mao-Chuang Yeh, Shuai Tang, Anand Bhattad, Chuhang Zou, and David Forsyth. Improving style transfer with calibrated metrics. pages 3149–3157, 2020.
- Fangneng Zhan, Yingchen Yu, Rongliang Wu, Jiahui Zhang, Shijian Lu, and Changgong Zhang. Marginal contrastive correspondence for guided image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10663–10672, June 2022.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- Yong Zhang, Weiming Dong, Oliver Deussen, Feiyue Huang, Ke Li, and Bao-Gang Hu. Data-driven face cartoon stylization. In SIGGRAPH Asia Technical Briefs, page 14, 2014.

- Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient labeled data factory with minimal human effort. In *Proceedings of IEEE Conference on Computer Vision* and Pattern Recognition, pages 10145–10155, 2021.
- Lihuan Zhao, Silu Liu, and Xiaoming Zhao. Big data and digital design models for fashion design. Journal of Engineered Fibers and Fabrics, 16:155892502110190, 01 2021. doi: 10.1177/15589250211019023.
- 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.
- Ming Zheng, Antoine Milliez, Markus H. Gross, and Robert W. Sumner. Example-based brushes for coherent stylized renderings. In *Proceedings of International Symposium* on Non-Photorealistic Animation and Rendering, page 3, 2017.
- 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.
- Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In Proceedings of European Conference on Computer Vision, pages 592–608, 2020.
- Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference* on Computer Vision, pages 597–613, 2016.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-toimage 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.
- Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. Barbershop: GAN-based image compositing using segmentation masks. ACM Transactions on Graphics, 40(6): 215, 2021.
- 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, 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 (IF: 1.94)

Cited in:

- Dongyu Rao, Xiao-Jun Wu, Hui Li, Josef Kittler, and Tianyang Xu. Umfa: a photorealistic style transfer method based on u-net and multi-layer feature aggregation. *Journal of Electronic Imaging*, 30(5):053013–053013, 2021.
- Adam Platkevič, Cassidy Curtis, and Daniel Sýkora. Fluidymation: Stylizing animations using natural dynamics of artistic media. In *Computer Graphics Forum*, volume 40, pages 21–32. Wiley Online Library, 2021.
- R Dinesh Kumar, E Golden Julie, Y Harold Robinson, S Vimal, Gaurav Dhiman, and Murugesh Veerasamy. Deep convolutional nets learning classification for artistic style transfer. *Scientific Programming*, 2022:1–9, 2022.
- Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. In *Computer Graphics Forum*, volume 41, pages 117–128. Wiley Online Library, 2022.

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. In *Computer Graphics Forum*, volume 39, pages 91–105. Wiley Online Library, 2020.
- Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, volume 40, pages 29–43. Wiley Online Library, 2021.
- Ific Goudé, Rémi Cozot, Olivier Le Meur, and Kadi Bouatouch. Example-based colour transfer for 3d point clouds. In *Computer Graphics Forum*, volume 40, pages 428–446. Wiley Online Library, 2021.
- Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. ACM Transactions on Graphics (TOG), 40(6):1–12, 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 the IEEE/CVF 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. In *Computer Graphics Forum*, volume 41, pages 117–128. Wiley Online Library, 2022.
- Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. ACM Transactions on Graphics, 41(4), 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 the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13999–14009, 2022.

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, 2021a (IF: 2.36)

Cited in:

- Yan Tang. Style transfer of chinese art works based on dual channel deep learning model. Computational Intelligence & Neuroscience, 2022, 2022.
- Hideki Todo, Kunihiko Kobayashi, Jin Katsuragi, Haruna Shimotahira, Shizuo Kaji, and Yonghao Yue. Stroke transfer: Example-based synthesis of animatable stroke styles. In ACM SIGGRAPH 2022 Conference Proceedings, pages 1–10, 2022.

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. Exemplarbased 3d portrait stylization. *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1371–1383, 2021.

Adéla Šubrtová* and David Futschik*, Jan Čech, Michal Lukáč, Eli Shechtman, and Daniel Sýkora. ChunkyGAN: Real image inversion via segments. In *Proceedings of European Conference on Computer Vision*, pages 189–204, 2022. *equal contribution

Appendix B Authorship Contribution Statement

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

FacestyleGAN: Real-Time Patch-Based Stylization of Portraits Using Generative Adversarial Network (55%)

For this work, I introduced the image-to-image paradigm and key ideas to make it efficient and effective at reasonable resolutions. I implemented the training pipeline, complete with visualizations, custom loss functions and configurable settings. I designed the overall architecture of the solution and later converted it to run at interactive speeds for realtime stylization. With the help of my collaborators, I created most of the results shown in the paper and supplementary material, and helped with preparation of the manuscript.

Arbitrary Style Transfer Using Neurally-Guided Patch-Based Synthesis (15%)

My main contribution was performing a deep exploration of how guiding the patch-based synthesis with neural features extracted from VGG network might work, which formed the basis of the extension we presented in the final paper and that was published as part of the journal submission. This involved multiple technical subtasks, such as extending internal framework for patch-based synthesis to work on floating-point data and arbitrary number of guidance channels, as well as developing a configurable way of extracting and normalizing the VGG feature responses to convert them into usable guidance channels. With this setup, I generated all the results we showed for our extension methods and provided the discussion on it.

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

I carried out a series of initial experiments based on the framework I developed and maintained for previous work, implemented the patch-based sampling and improved efficiency of the training. Later, I made the system work for the interactive painting scenario shown in the paper, video and supplementary material. I worked on the design and implementation for the server-client architecture to run the training job on a remote machine while the inference ran on edge at the same time, so two distinct machines and GPUs could be utilized without having to construct a multi-GPU node to provide to the artist. I also helped prepare and finalize the paper manuscript.

STALP: Style Transfer with Auxiliary Limited Pairing (60%)

For STALP, I designed the method in its entirety, from used architecture, model selection and specific customizations, to the data pipelines including augmentations, to used loss function and the technical implementation of the method. I conducted the majority of the experiments shown in the paper, including fine-tuning hyperparameters to get the best results, and helped with preparation of the manuscript. I presented this work at the Eurographics 2021 conference.

ChunkyGAN: Real Image Inversion via Segments (40%)

In this paper, as an equal-contribution (joint first) author, I proposed and explored the idea of projecting segmented chunks of images in isolation and manipulating them separately. Later on, I created a proof-of-concept framework incorporating these ideas with Poisson image-editing based blending, and wrote a draft that formed the basis of the manuscript. I wrote a significant portion of the final code and produced some of the results shown in the paper and performed some of the comparisons with related work.

Appendix C

FacestyleGAN Supplementary Material

To confirm that 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 at the first glance with no obvious failures on both sides Fig. C.1. During the time the questionnaire was open, we collected 194 responses for the full 12 question test.

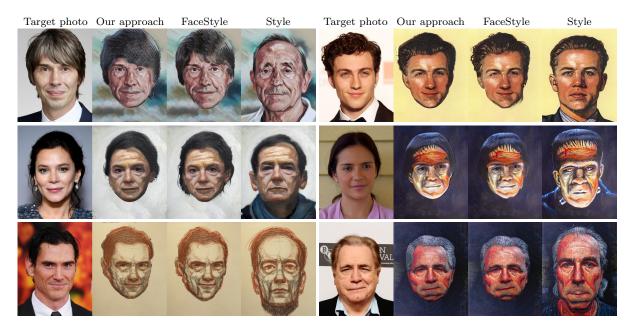


Figure C.1: A selection of results used in the perceptual study. Note that the results are comparable for both approaches with no obious failures.

We set out with the null hypothesis stating there is no statistically significant difference between the quality of the output of both tested methods, which we attempt to reject based on the collected data using the Student's t-test. In the case of identity preservation, our data shows we can reject the null hypothesis with a probability of only 49%, which suggests 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 one another. Regarding the style reproduction study, 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, amounting to 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, as judged by our users.

Appendix D

Video Stylization Using Few-Shot Patch-Based Training Supplementary Material

In this supplementary material we describe the interactive applications of our framework in more detail, presenting the overall architecture of the solution as well as mentioning the specific hardware we used. Furthermore, we show example photos of our framework during real-time stylization sessions with artists (see our supplementary video for live recordings from those sessions) and discuss feedback we received during our informal user study. Lastly, we show additional results produced by our framework, and additional experiments with hyperparameter setting.

D.1 Interactive applications

To demonstrate interactive applications, we provide artists with a setup of our framework in a few variations. Each scenario involves working with a workstation PC, equipped with a consumer-grade GPU (we use Nvidia RTX 2080), on which the artists perform a task. This machine runs our framework executable, which displays visual feedback for the artist. Training of the model is done off-site on a server with an Nvidia Tesla V100 GPU. The client machine sends necessary training data to this server and the training server in turn periodically sends back models trained with the new data. The training data is replaced every time the server receives a new version of a frame. Our training process quickly adapts the model to the new data.

Trained models are used on the artist's PC to generate stylized video frames. Our approach allows us to display an acceptable result in as little as 5 seconds, which improves with time as better models arrive. In practice, the potentially lengthy process of art creation amortizes training time, largely masking the downside of this delay.

Note that inference could also be performed on the server but we do it locally to reduce delay during live-feed stylization.

We devise the following real-time style transfer tasks:

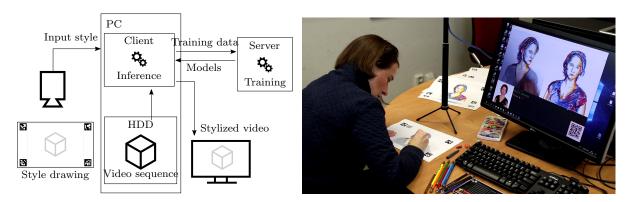


Figure D.1: Scenario No. 1: an artist is drawing over a stencil of a keyframe using traditional media. The stencil contains markers that allow us to perfectly align the frames to prevent shift in images.



Figure D.2: Scenario No. 2: an artist is stylizing an object as seen by the camera in real-time using image editing software.

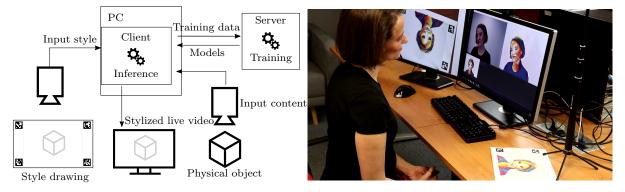


Figure D.3: Scenario No. 3: an artist is stylizing an object as seen by the camera in real-time using a physical stencil.

D.1.1 Pre-recorded video + live style capture (traditional)

The artist is provided with (or creates) a pre-recorded video sequence and selects one or more keyframes which they will paint over. These keyframes are printed in low contrast on a stencil with markers. These markers allow us to perfectly match and align the contents of the stencil with the input sequence frames, so as to avoid misalignment of the training data and achieve the best performance possible. In case of multiple keyframes, we differentiate stencils using additional markers so that the artist is free to swap between them during the session.

As the artist starts painting the first keyframe, the server recognizes which keyframes are ready and only uses previously seen keyframes to train on. Unfinished or unseen parts will likely produce poor visual results which will indicate spots which need to be fixed in current or other keyframes. The artist may also wish to create masks for each keyframe, to prevent introducing ambiguity of different appearances for identical content or to save repetitive work, especially if the keyframes are relatively similar. Diagram for this setup and an example photograph are shown in Fig. D.1.

D.1.2 Live video capture + live style capture (digital)

This scheme is different from the previous in that there is no pre-recorded video sequence, instead, we arrange a camera, capturing a scene in real-time. Our framework allows the artist to export a still image of the scene into image editing software of their choice. This image can then be edited or painted over to achieve an artistic look. Its modified version is periodically sent to the training server, where it serves as the current style exemplar used for training.

During the session, the artist is free to change the scene, while observing the stylization in real-time. If the scene contains some object, a common modification of the scene would be rotating or moving the object. Once the artist is satisfied with the result, they can export additional still images to fix any issues in the scene. This could be, for example, one image for the front of an object and another image for the back of the object. Diagram for this scenario and an example photograph of a session are shown in Fig. D.2.

D.1.3 Live video capture + live style capture (traditional)

We design our framework to also let us combine the two previous scenarios. When a still image of a live scene is exported, it can be printed on a stencil. Artist draws on that stencil, and we set up a second camera to capture it. The framework automatically aligns it to the still image and sends it to the training server again. Defining multiple keyframes is then as simple as printing multiple different stencils with identifying markers.

Although working with a digital image is often faster, this setup is useful due to the preference of some artists to work with traditional artistic media. Our framework is well suited for capturing real strokes and stylizing the video frames in a way similar to traditional animation. This scenario is visually explained in Fig. D.3.

D.1.4 User study

We asked the artists for their comments on using our framework. Although our user study was informal, we believe it still presents an interesting insight into the contribution of this work.

One of the very first impressions was the moment of surprise and awe whenever a new model arrived at the client machine and a better stylization started appearing on the screen. Thanks to this effect, the artists felt engaged throughout the whole session, some even asked us for further sessions so they could explore the implications of our framework more.



Figure D.4: The keyframe (a) was used to produce the sequence of 148 frames. While the body part is faithfully represented in both [Jamriška et al. 2019] (b) and ours (c), our approach better preserves the facial region; see the zoom-in views [Jamriška et al. 2019] (d) and ours (e). Video frames (insets of a-c) courtesy of \bigcirc MAUR film and style exemplar (a) courtesy of \bigcirc Jakub Javora.

Generally, artists tended to describe the proposed system as a completely new tool to approaching artistic animation, thanks to the real-time feedback and continuous improvement. The other aspect that makes using our framework easy and entertaining, according to the comments, is using the photo stencils, as painting over a photograph using brushes is much easier than creating art from scratch. This also makes it suitable for children, who are largely familiar with using stencils from coloring books.

Lastly, artists appreciated the fact that no explicit masking needs to be done during the creation process (e.g., background masking). The model we use seems good at representing identity transformation, thus leaving parts of the image unstylized means that the original background just propagates to the output.

While the overwhelming majority of the comments we received were positive, the one negative remark was that the result image quality is somewhat lower than well-optimized sequence created by Jamriška et al. [2019]. However, compared to the inability of their method to deliver such a real-time experience, we feel our framework makes for a reasonable trade-off.

D.1.5 Additional Results and Experiments

In this section, we first present an additional result of our approach compared to the result of Jamriška et al. [2019], see Fig. D.4.

Second, as already primarily covered in the main text, we discuss hyperparameter optimization on one more example. As it is a common practice to reduce the network size to prevent overfitting, in Fig. D.5, we demonstrate that in the task of style transfer, certain network capacity is necessary to achieve high-quality results.

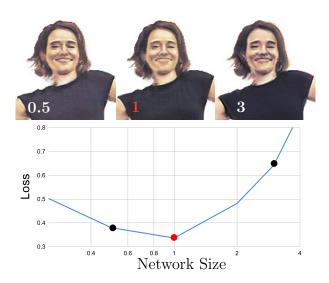


Figure D.5: Impact of network size on the visual quality of results. The loss, y-axes, is computed w.r.t. the output of Jamriška et al. [2019]. The x-axes shows the network size (i.e., number of filters) relative to the best setting we found via hyper-parameter search. Other hyperparameters are fixed. The middle image (1) depicts the best setting, the left image (0.5) represents setting with half number of filters, and the right image (3) represents setting with three times more filters compared to the middle image. The difference in the visual quality of images, as well as the loss curve, clearly show that there exists a saddle point.

Appendix E

ChunkyGAN Supplementary Material

Here we present additional experiments that provide further insight and evaluations. Namely, in Sections 7.4.1 and 7.4.2, we quantitatively compare with additional competing methods: two encoders pSp [Richardson et al. 2021], e4e [Tov et al. 2021], and Pivotal Tuning [Roich et al. 2021]. We show more challenging examples and qualitative results of all tested methods. REGULARIZATION shows the automatic segmentation and corresponding images for both cases of our method, with and without the regularization. Finally, in APPS we demonstrate additional examples of interactive image editing and application of our method to the image interpolation task.

E.1 Projection Fidelity

Projection	LPIPS	Identity	L_2
$\overline{\mathcal{W}}$	0.4190 ± 0.0363	0.1745 ± 0.1328	0.0725 ± 0.0699
Ours in \mathcal{W}	0.3697 ± 0.0396	0.1384 ± 0.1117	0.0481 ± 0.0289
\mathcal{W}^+	0.3675 ± 0.0387	0.1195 ± 0.1047	0.0436 ± 0.0623
Ours in \mathcal{W}^+	0.3194 ± 0.0365	0.0937 ± 0.0855	0.0207 ± 0.0151
Ours in \mathcal{W}^+ reg.	0.3330 ± 0.0350	$\textbf{0.0894} \pm \textbf{0.074}$	0.0217 ± 0.0130
S	0.3577 ± 0.0397	0.1070 ± 0.0965	0.0328 ± 0.0188
Ours in \mathcal{S}	0.3572 ± 0.0401	0.1053 ± 0.0928	0.0319 ± 0.0187
[Tov et al. 2021]	0.4444 ± 0.0418	0.1912 ± 0.1343	0.0468 ± 0.0165
[Richardson et al. 2021]	0.4433 ± 0.0418	0.1706 ± 0.1182	0.0351 ± 0.0135
[Alaluf et al. 2021]	0.4444 ± 0.0430	0.1900 ± 0.1318	0.0433 ± 0.0162
[Roich et al. 2021]	0.3332 ± 0.0353	0.0936 ± 0.0616	0.0135 ± 0.0071
[Alaluf et al. 2022]	0.4297 ± 0.0404	0.1420 ± 0.1003	0.0247 ± 0.0115

Table E.1: Projection fidelity (extended)—losses were measured between the projected and the original image for each of the projection methods. Each cell reports the loss averaged over the CelebA subset along with the standard deviation. Recommended value of 5 iterations was used for methods of [Alaluf et al. 2021] and [Alaluf et al. 2022].

The experiment measures the average LPIPS, Identity, and L_2 losses between the original and inverted images on CelebA subset of 100 images. Table E.1 is extended by three rows with other methods compared to Table 7.1 in the main text. Notably, both fast encoder-based approaches e4e [Tov et al. 2021] and pSp [Richardson et al. 2021] produce lower fidelity images. In the case of Pivotal Tuning [Roich et al. 2021] we started refining the StyleGAN2 model from W^+ codes as pivots. Our method performs better when measuring LPIPS and Identity. Pivotal Tuning is superior by a small margin only in the case of L_2 metric, which is known to be uncorrelated with the human perception. Moreover, the major drawback of Pivotal Tuning is that it requires to store the entire StyleGAN2 model for each image together with the corresponding latent code. See qualitative results in Fig. E.2–Fig. E.6 to compare differences among the methods visually.

		(a)				(b)		
	gender	\mathbf{smile}	age	beard	gender	smile	age	beard
${\mathcal W}$.169	.022	.07	.279	.249	.18	.191	.328
\mathcal{W}^+	.209	.02	.095	.296	.256	.128	.171	.325
Ours in \mathcal{W}^+	.298	.049	.151	.312	.325	.125	.203	.333
Ours in \mathcal{W}^+ reg.	.126	.018	.069	.091	.169	.099	.129	.144
[Tov et al. 2021]	.088	.024	.054	.239	.26	.242	.245	.351
[Richardson et al. 2021]	.153	.026	.126	.074	.282	.223	.258	.248
[Alaluf et al. 2021]	.097	.030	.081	.213	.417	.409	.399	.453
[Roich et al. 2021]	.135	.037	.089	.329	.237	.176	.200	.388
[Alaluf et al. 2022]	.107	.12	.135	.107	.15	.163	.166	.157

E.2 Editability

Table E.2: Identity preservation during editing (extended)—identity loss was computed between the projected and the edited images (a), and between the original and the edited images (b). Recommended value of 5 iterations was used for methods of [Alaluf et al. 2021] and [Alaluf et al. 2022].

We tested the three extra methods for identity preservation during editing. Table E.2 extends the same table in the main text. The calibrated edits were made and the angular identity loss was measured. From Table E.2 it is apparent that with regularization our method compares very favorably when examining the projected and edited images (a). Nevertheless, it outperforms by a large margin all competitors when comparing the *original* and edited images (b). This is caused by the fact that the projection quality of previous approaches is not very faithful to the original image as can be seen quantitatively in Table 7.1 and qualitatively in Fig. E.9. Pivotal Tuning gives a fair inversion quality, however, it can be seen that the facial mask of the man is still blurred and the bindi of the Indian woman was not reconstructed at all, unlike in our methods. The editing for Pivotal Tuning is rather similar to W^+ , not very convincing.

Concerning limitations of our method, they occur in case of extreme edits that significantly change the geometry, such as yaw. Then segments do not match and visible seam artifacts are produced, see Fig. E.10. We shortly discussed in the main paper in Section 6 possible future options to resolve the issue.

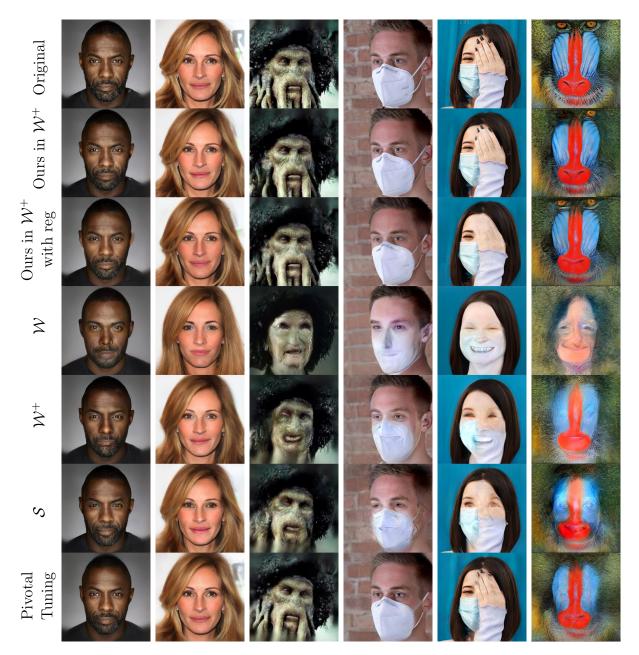


Figure E.1: Qualitative assessment of projection fidelity on challenging examples optimization-based methods. The out-of-domain images are especially hard to project for the existing methods. The best results are produced by our method. In the case of Pivotal Tuning the in-domain images are projected faithfully, but for the out-of-domain images are still missing important features (such as eyes of the mandrill, hand in front of the face, details of the face mask). Our method is flexible enough to generate the unusual features of the original image.

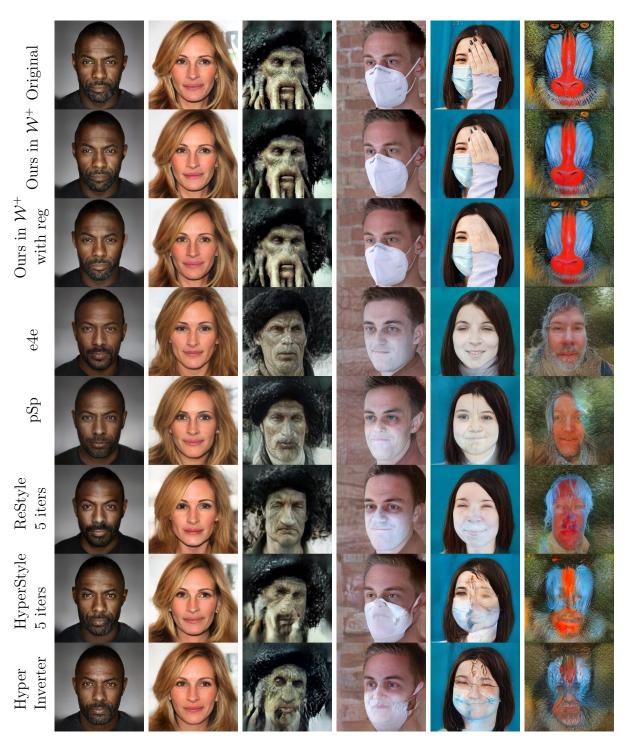


Figure E.2: Qualitative assessment of projection fidelity on challenging examples - **encoderbased methods**. The out-of-domain images are especially hard to project for the existing methods. The best results are produced by our method. HyperStyle produces good results for the in-domain images, but the out-of-domain images contain artifacts. Our method is flexible enough to generate the unusual features of the original image.

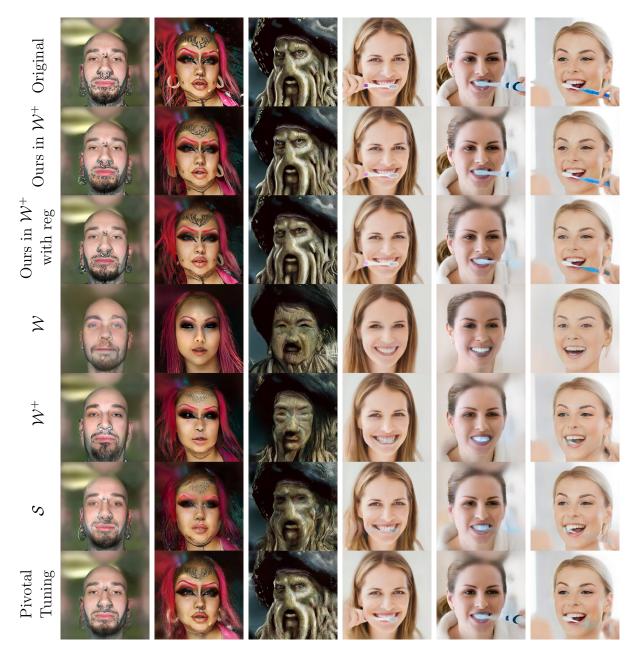


Figure E.3: Qualitative assessment of projection fidelity on challenging examples - **optimization-based methods**. Pivotal Tuning faithfully synthesizes the out-of-domain image in column 3, but the images in column 1 and 2 lack details of the piercings.

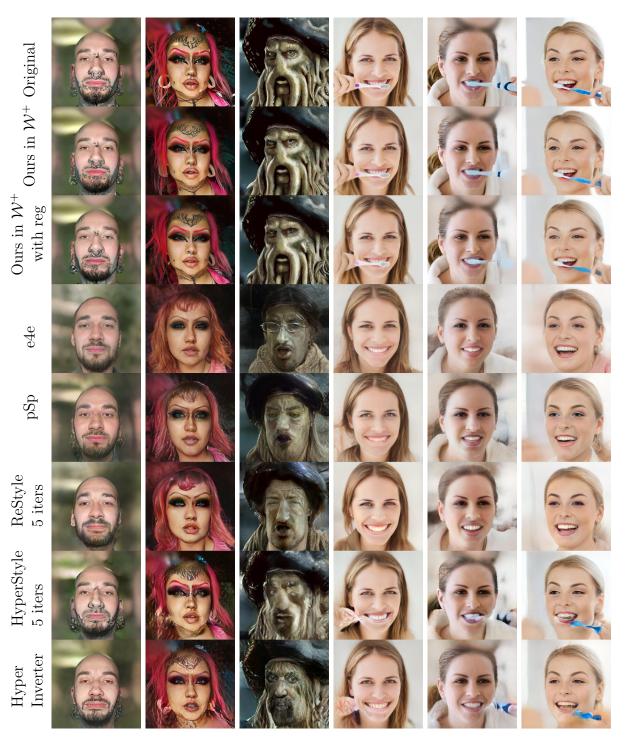


Figure E.4: Qualitative assessment of projection fidelity on challenging examples - **encoderbased methods**. Face occlusions are not faithfully inverted by any of the encoder-based methods. The out-of-domain image in column 3 is hard to project, other methods fail to generate the tentacles.



Figure E.5: Further comparison on images which are challenging to invert accurately using existing methods - **optimization-based results**. Pivotal Tuning does not faithfully reproduce features which are far from the domain of the original trained network (toothbrush handle in columns 1 and 2).



Figure E.6: Further comparison on images which are challenging to invert accurately using existing methods - **encoder-based methods**. Encoder based methods fail to reproduce major expression features (tongue in columns 3, 4 and 5) and the hands occluding the face in column 6.

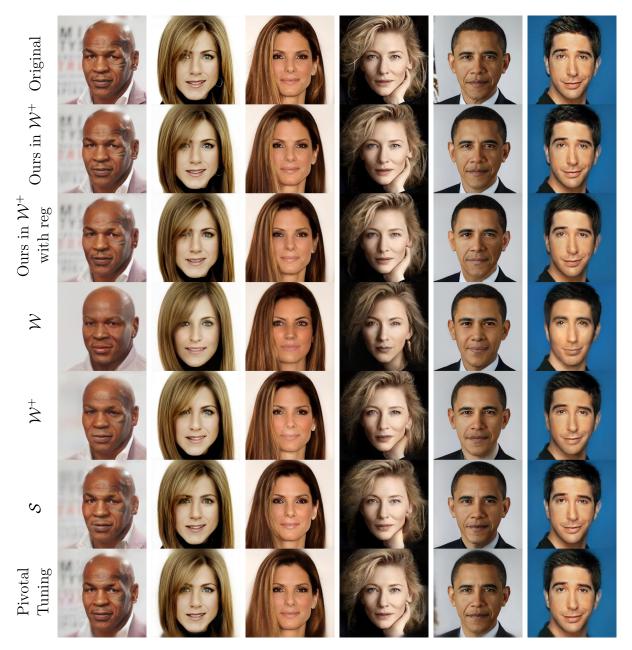


Figure E.7: Qualitative assessment of projection fidelity on challenging examples - optimization-based methods. The identity is reliably preserved using our method, S-space inversion, and Pivotal Tuning; however, the Pivotal Tuning does not generate the tattoo in the first column in great detail, and the S-space inversion produces unrealistically-looking hand in column 4.

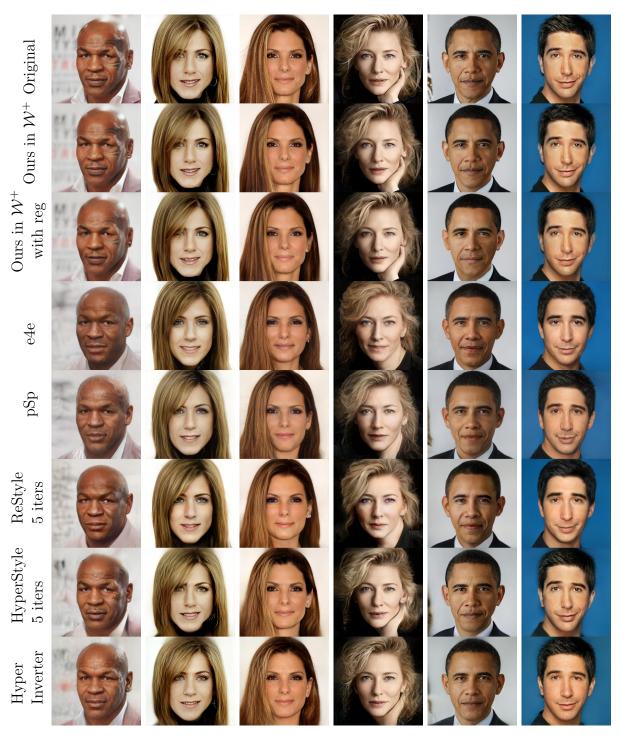
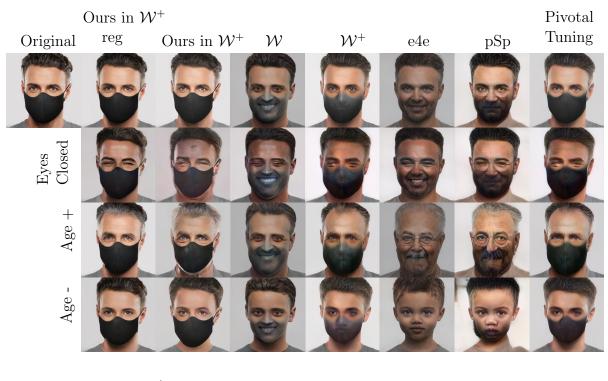


Figure E.8: Qualitative assessment of projection fidelity on challenging examples - **encoderbased methods**. Encoder-based methods all fail to generate the hand in column 4 and the tattoo in column 1.



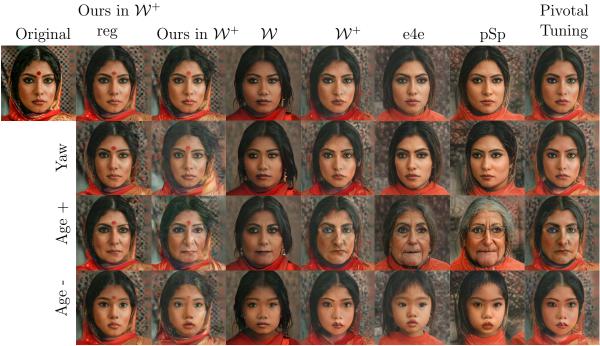


Figure E.9: ***Challenging global edits. The row besides the original image shows inverted images by all tested methods. Other rows display corresponding edited results along given semantic directions. For our methods, the editing was done simply by manipulating the latent codes the same for all the segments. The results of the inversion and editing are fully automatic. No manual adjustments and no postprocessing were performed.

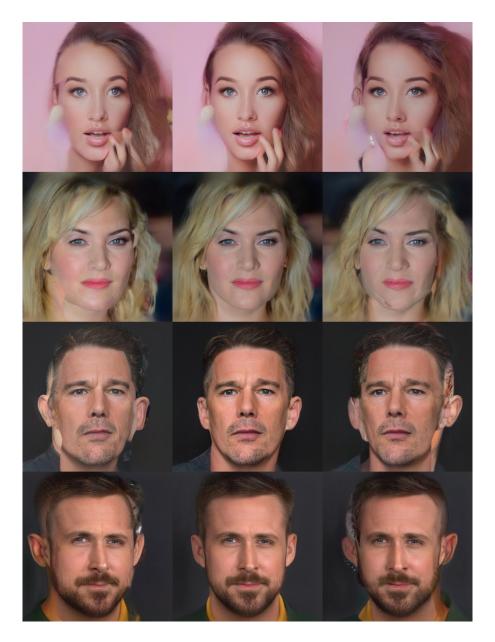


Figure E.10: Limitations of using our method to perform edits which change the geometry to a significant extent. In these examples, segment seams become visible for larger yaw changes without any explicit treatment of the segments.

E.3 Regularization

In Table E.2, it is seen that the regularization has a positive impact on the identity preservation during editing. We believe the reason is that the non-regularized projection may generate unrealistic images with codes far from the mean. The regularization encourages the codes to be closer to the mean, producing in-domain images for which the editing by latent code manipulation along pre-trained semantic directions works better. The effect of the regularization is demonstrated in Fig. E.11. In both cases, the composed image is very faithful to the original, and the composed images are hard to distinguish. However, the component images are notably more realistic when the regularization is on. Out-of-domain example is shown in Fig. E.12.

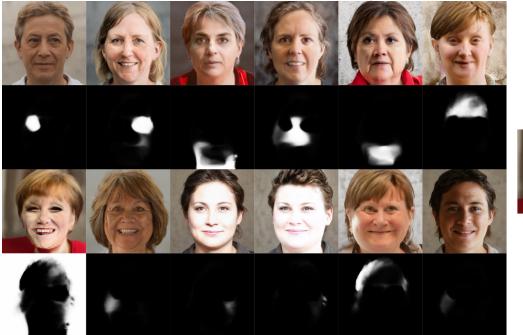
E.4 Additional Applications

Our approach can be used for image interpolation task (see Fig. E.13). In this application two estimated latent codes in each segment are linearly interpolated to produce partial inbetween image which is then composed with other inbetween images using Laplacian pyramid [Burt and Adelson 1983]. Since our method preserves identity better and the corresponding facial features are naturally mapped on each other due to the interpolation in the latent space the resulting transition looks convicing.

In Fig. E.14 we present an editing example produced using our method where a Style-GAN2 model trained on cars was used as a backbone. This example demonstrates that our approach is agnostic to the domain on which the model was trained. The only requirement for the used model is that it has a sufficient number of pre-trained directions for latent code manipulation.

As demonstrated in the main paper our method is flexible enough to make a projection of out-of-domain images, i.e., images that were not considered during the training of StyleGAN2 model. In Fig. E.15 we edit various famous paintings using our method with StyleGAN2 model trained exclusively on photographs of real faces. Thanks to the accurate projection the subsequent edits look like if they were produced by a StyleGAN2 model trained on paintings.

(a) Regularization on $(\lambda_{reg} = 1)$





(b) Regularization off $(\lambda_{reg} = 0)$





Figure E.11: Effect of regularization. The projected (composed) images are on the right. The left side depicts individual projections with the corresponding segmentation masks underneath.

(a) Regularization on $(\lambda_{reg} = 1)$





(b) Regularization off $(\lambda_{reg} = 0)$





Figure E.12: Effect of regularization - out-of-domain example. The projected (composed) images are on the right. The left side depicts individual projections with the corresponding segmentation masks underneath.

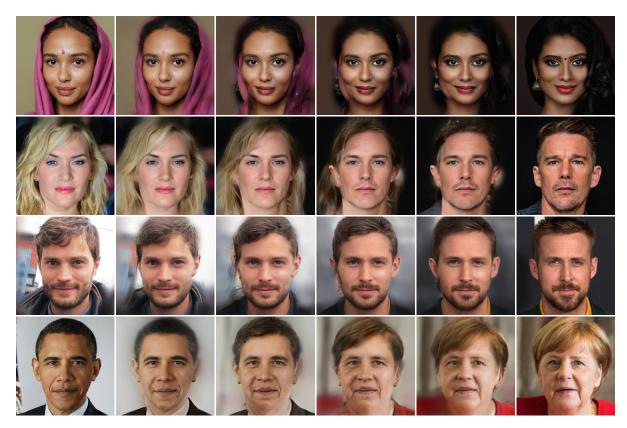


Figure E.13: Interpolation examples—our approach can be used to perform interpolation between two different identifies. The estimated latent code in each segment is linearly interpolated and the final image is then composed using Laplacian pyramid. A key advantage here is that in our method identity is preserved better and thus the transition looks more believable.



Figure E.14: Editing using our method based on StyleGAN2 model trained on photos with cars—original image (a), detail of the original image (b), local edits of wheel disc design (c-e).

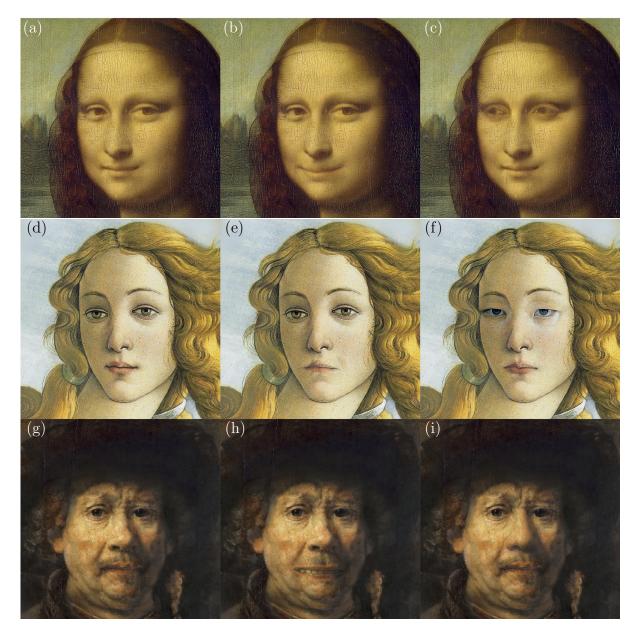


Figure E.15: Edits performed on famous painting using our approach with StyleGAN2 model trained on real faces—original Da Vinci's Mona Lisa (a), more pronounced smile (b), change in the gaze direction (c), original Botticelli's The Birth of Venus (d), change in the mouth expression (e), different shape of eyes (f), original Rembrandt's Little Self-portrait (g), changing mouth expression (h), different shape of the nose (i).