Stylizing Video by Example

ONDŘEJ JAMRIŠKA, Czech Technical University in Prague, Faculty of Electrical Engineering
ŠÁRKA SOCHOROVÁ, Czech Technical University in Prague, Faculty of Electrical Engineering
ONDŘEJ TEXLER, Czech Technical University in Prague, Faculty of Electrical Engineering
MICHAL LUKÁČ, Adobe Research
JAKUB FIŠER, Adobe Research
JINGWAN LU, Adobe Research
ELI SHECHTMAN, Adobe Research
DANIEL SÝKORA, Czech Technical University in Prague, Faculty of Electrical Engineering

Fig. 1. An example of a stylized sequence produced by our approach. One frame from the sequence is selected as a keyframe (a) and a corresponding style exemplar is painted using watercolor (b). Then, for the rest of the sequence (c, e, g) our technique produces stylized output (d, f, h) which preserves the artistic attributes of the specified style exemplar, reflects structural changes in the target video, and maintains temporal coherence. Video frames (a, c, e, g) courtesy of © MAUR film, style exemplar (b) courtesy of © Pavla Sýkorová, used with permission.

We introduce a new example-based approach to video stylization, with a focus on preserving the visual quality of the style, user controllability and applicability to arbitrary video. Our method gets as input one or more keyframes that the artist chooses to stylize with standard painting tools. It then automatically propagates the stylization to the rest of the sequence. To facilitate this while preserving visual quality, we developed a new type of guidance for state-of-art patch-based synthesis, that can be applied to any type of video content and does not require any additional information besides the video itself and a user-specified mask of the region to be stylized. We further show a temporal blending approach for interpolating style between keyframes that preserves texture coherence, contrast and high frequency details. We evaluate our method on various scenes from real production setting and provide a thorough comparison with prior art.

CCS Concepts: • **Computing methodologies** → **Motion processing**; **Image processing**.

Additional Key Words and Phrases: style transfer

ACM Reference Format:
Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Stylizing Video by Example. *ACM Trans. Graph.* 38, 4, Article 107 (July 2019), 11 pages. https://doi.org/10.1145/3306346.3323006

Authors' addresses: Ondřej Jamriška, Czech Technical University in Prague, Faculty of Electrical Engineering, jamriond@fel.cvut.cz; Šárka Sochorová, Czech Technical University in Prague, Faculty of Electrical Engineering, sochosar@fel.cvut.cz; Ondřej Texler, Czech Technical University in Prague, Faculty of Electrical Engineering, texleond@fel.cvut.cz; Michal Lukáč, Adobe Research, lukac@adobe.com; Jakub Fišer, Adobe Research, fiser@adobe.com; Jingwan Lu, Adobe Research, jlu@adobe.com; Eli Shechtman, Adobe Research, elishe@adobe.com; Daniel Sýkora, Czech Technical University in Prague, Faculty of Electrical Engineering, sykorad@fel.cvut.cz.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
0730-0301/2019/7-ART107 $15.00
https://doi.org/10.1145/3306346.3323006

1 INTRODUCTION

In the past decades, advances in computer graphics led to a revolution in the art of animation, giving birth to an entirely new branch of animation which is three-dimensional, and includes photorealistic lighting effects and physically accurate simulation. Together with lighting, material, and performance capture, the production pipelines of animated video now resemble live-action production more closely than traditional animation. An unfortunate side effect of this is that, due to production and technical considerations, there is a "style gap" between traditional and 3D animation, where the latter has its own distinct look, and it has so far been impossible to convincingly reproduce the look of the former using the aforementioned production pipelines. Currently, there are no automated methods that could use live-action performance capture to produce

the look of traditional animation. Although artists have attempted to bridge this gap for, e.g., abstract stylization (*A Scanner Darkly*[1]) or painterly look (*Loving Vincent*[2]), these were monumentally laborious efforts that had to be created manually frame-by-frame.

One possible way to overcome this could be to employ example-based style transfer techniques to transfer artistic style from a traditionally created style exemplar to a synthetic or live action target. This approach recently became popular thanks to advances in neural [Gatys et al. 2016; Ruder et al. 2018] and patch-based transfer [Frigo et al. 2016, 2019] techniques. These approaches can alter the global appearance of the target to roughly resemble the given visual style, but the effectiveness of stylization relies solely on the internal representation of style and content of the respective algorithm. They do not offer users any explicit controls and cannot fulfill the need of artists to precisely express their artistic intent.

In an effort to provide this sort of control over style transfer, Hertzmann at el. [2001] pioneered an Image Analogies framework where the style exemplar, as well as the target, are extended with additional guiding channels which provide spatial control of how the style is transferred. This ensures that particular features of the style exemplar will appear at desired locations in the target. It was shown recently [Bénard et al. 2013; Fišer et al. 2016] that this additional control allows for a more semantically meaningful transfer and results in higher visual quality than the generic methods [Frigo et al. 2019; Gatys et al. 2016].

The main drawback of this approach is that the guidance channels need to be generated first. Much research was done into algorithmic solutions for specific scenarios (e.g., rendering attributes from known geometry [Bénard et al. 2013; Fišer et al. 2016] or using landmark detectors and face segmentation [Fišer et al. 2017]), but guidance generation for the general case of arbitrary images remains an open problem. Efforts into neural-based guidance by Liao et al. [2017] and later Gu et al. [2018] demonstrated that the response of VGG net – a deep neural network trained for object classification [Simonyan and Zisserman 2014] – can be used as a guide to automatically control the transfer in certain cases. Unfortunately, this approach is able to reliably discriminate features only on the type of images VGG was trained on (faces, animals, objects, etc.). In a more general scenario, the accuracy is insufficient and may lead to obvious inconsistencies (see, e.g., transfer of facial patterns to the legs of the target subject in Fig. 2). Moreover, those techniques do not address temporal coherence which is crucial for video synthesis.

In this paper, we formulate an alternative analogy-based approach for the artistically controlled stylization of video, which (a) addresses the style gap by facilitating free-form artistic stylization of synthetic or live-action video sequences, (b) gives artists control by allowing them to explicitly specify local styles using traditional painting techniques that are familiar to them, and (c) does not require "insider knowledge" of the target content, such as segmentation, landmarks, 3D or rendering information.

We build on the keyframe stylization paradigm proposed by Benard et al. [2013], where the artist paints one or more keyframes in a preferred style, and the algorithm then propagates the specified style to the rest of the sequence. Our key difference from the aforementioned approach is that we do not require any knowledge of the underlying 3D structure of the target scene. Instead, we obtain semantically meaningful transfer by using the original color information from the input video together with approximate positional and temporal guidance generated using optical flow estimation. We further show that when used in conjunction with a state-of-art patch-based synthesis algorithm [Fišer et al. 2016], this guidance results in superior visual quality while preserving the artistic intent. Finally, we demonstrate the practical utility of the proposed approach on examples from real production settings.

Fig. 2. Common neural stylization artifacts. Style transfer guided by the response of VGG network might transfer the facial pattern onto the leg region. (a) target frame, (b) result of Gu et al. [2018], (c) our approach. Video frame (a) courtesy of © MAUR film, used with permission.

2 RELATED WORK

Traditional image and video stylization methods employ algorithmic filters hand-crafted to transform an input image or video to a particular style. These can be based on a physical simulation of a given artistic medium [Curtis et al. 1997; Haevre et al. 2007; Lu et al. 2012], procedural techniques [Bénard et al. 2010; Bousseau et al. 2006, 2007; Montesdeoca et al. 2018], or compositing predefined pen [Praun et al. 2001; Salisbury et al. 1997; Snavely et al. 2006] or brush strokes [Hays and Essa 2004; Litwinowicz 1997; Schmid et al. 2011; Zhao and Zhu 2011]. While these approaches give impressive results on the respective domains that they are designed for, they are invariably limited to a single style or a small set of styles, and suffer from unintuitive controls that make it difficult to express artistic intent.

A more modern take on this problem are methods based on generative adversarial networks [Goodfellow et al. 2014], which can be trained to perform image-to-image [Isola et al. 2017; Zhu et al. 2017a,b] as well as video-to-video [Tulyakov et al. 2018; Wang et al. 2018] translation, including stylization. Researchers have also introduced neural network based approaches that target artistic stylization specifically [Johnson et al. 2016; Ulyanov et al. 2016a,b, 2017; Wang et al. 2017; Wilmot et al. 2017], training one network per style. These methods cannot reproduce styles that they are not trained on,

[1]https://en.wikipedia.org/wiki/A_Scanner_Darkly_(film)
[2]http://lovingvincent.com

and for the styles they support, the results typically do not accurately reproduce fine textural details. Sanakoyeu et al. [2018] attempted to improve the stylization quality by introducing a style-aware content loss, but the results still have some semantic inconsistencies (see supplementary material). Researchers have also introduced stylization techniques that transfer arbitrary visual styles to content images using a single network at the expense of limited faithfulness to the target styles [Huang and Belongie 2017; Li et al. 2017]. In general, neural approaches require time-consuming and arcane training process and offer limited user control [Gatys et al. 2017].

Example-based approaches naturally support stylization using arbitrary style imagery, and no training is needed. The most widespread approach formulated the concept of Image Analogies [Hertzmann et al. 2001], where guidance channels are added to both the style exemplar and the target photo to guide a patch-based synthesis algorithm [Fišer et al. 2016; Kaspar et al. 2015; Wexler et al. 2007] which decides how different features of the style should be transferred to various regions of the target. The remaining problem is finding appropriate guidance channels, which can be generated algorithmically in certain cases [Bénard et al. 2013; Fišer et al. 2016; Jamriška et al. 2015] or for particular content (e.g., faces [Fišer et al. 2017]). Creating the guiding channels manually is possible but unintuitive and highly laborious in the case of video.

To circumvent this problem, generic approaches which do not require specific guidance [Frigo et al. 2016; Gatys et al. 2016] were formulated. More recent neural-based techniques [Gu et al. 2018; Li and Wand 2016; Liao et al. 2017] achieve this by using responses of the VGG network trained on object classification [Simonyan and Zisserman 2014] to guide the synthesis. These latter approaches produce impressive results when used on images structurally similar to those in ImageNet – natural photographs with a single identifiable foreground object or scene – but are difficult to control and behave unpredictably when generalizing to different types of images such as complex natural scenes or paintings of abstract styles.

Stylization of video offers the additional challenge of handling temporal coherence. This was itself a topic of previous research, where coherence was formulated as an additional constraint for patch-based synthesis together with the control over the amount of visible temporal flickering [Dvorožňák et al. 2018; Fišer et al. 2017, 2014]. Similarly, for generic style transfer not requiring specific types of guidance, explicit temporal coherence was incorporated into neural-based [Chen et al. 2017; Gupta et al. 2017; Ruder et al. 2018; Sanakoyeu et al. 2018] as well as patch-based [Frigo et al. 2019] techniques. Lai et al. [2018] introduced a blind temporal coherency approach that takes per-frame stylized video as input and outputs a temporally consistent video as post-processing.

We based our approach on the image analogies framework that offers both precise control as well as the ability to handle arbitrary style. We combine keyframe-based user control as in the method of Bénard et al. [2013] with a synthesis process similar to that used in the approach of Fišer et al. [2017]. A key added value of our solution is that we overcome two significant drawbacks of these previous methods: (1) dependence on a specific target domain (3D computer-generated animation and facial video) and (2) inability to handle challenging scenario when multiple inconsistent keyframes are used to stylize the target sequence. To do that we design a new set of domain-independent guidance channels and formulate a corresponding error metric for the subsequent patch-based synthesis. To combine content from multiple keyframes, we propose a solution that prefers high-frequency details according to their relevance and avoids loss of contrast.

3 OUR APPROACH

The input to our method is a *target* video sequence T and one or more stylized keyframes or *style exemplars*, S. To create keyframes, artists can paint digitally or physically using their preferred artistic media over arbitrarily-selected frames of a video. Similar to the physical painting process used in StyLit [Fišer et al. 2016], we print a low-contrast version of the frame with registration marks, which allows accurate re-digitization and registration of stylized artwork. The output of our method is a temporally coherent video sequence O, in which every frame is stylized analogically to the style exemplar S, i.e., various semantic parts of the input sequence are stylized the same way as in the example frame.

One possible approach to example-based video stylization is to estimate dense correspondences between the target keyframe T_i and all other frames in the sequence [HaCohen et al. 2011; Yücer et al. 2012], and then use the resulting deformation field to warp the style exemplar. However, this naive approach would introduce undesirable texture distortion to the example style and generate artifacts in situations like disocclusions or lighting changes in the target sequence. To address this fundamental drawback we formulate our problem as a guided patch-based synthesis similarly to [Bénard et al. 2013; Fišer et al. 2017]. However, since in our scenario we do not have any prior knowledge of the underlying scene we need to design a new set of guiding channels that can be computed solely based on the input video.

For clarity, we first explain the stylization process with just one keyframe and then show how it extends to multiple keyframes.

3.1 Guidance for a single keyframe

Our new set of guiding channels consists of the original video frames G_{col}, mask G_{mask}, positional G_{pos}, edge G_{edge}, and temporal G_{temp} guides (see Fig. 3). These will be explained next.

Color guide. G_{col} corresponds to the original color frames of the target sequence T (see Fig. 3a). It captures appearance changes, e.g., facial gestures, subtle cloth deformations, varying illumination, etc.

Mask guide. G_{mask} highlights the objects of interest. It helps the algorithm distinguish object boundaries to handle occlusion and also allow for layered stylization if preferred by artists. When there is no strong occlusion in T or no need to accurately delineate object boundaries, addition of the mask guide is optional otherwise G_{mask} can be obtained using, e.g., green screen matting (see Fig. 3b), color separation or other semi-automatic segmentation method [Li et al. 2016].

Positional guide. G_{pos} helps the algorithm maintain the overall structure of the stylized keyframe for meaningful transfer (see Fig. 3c). It serves to resolve ambiguity between distinct features which have similar appearance, but need to be stylized differently as artist desired. In Fig. 5a,b the result of synthesis without using G_{pos} is visible.

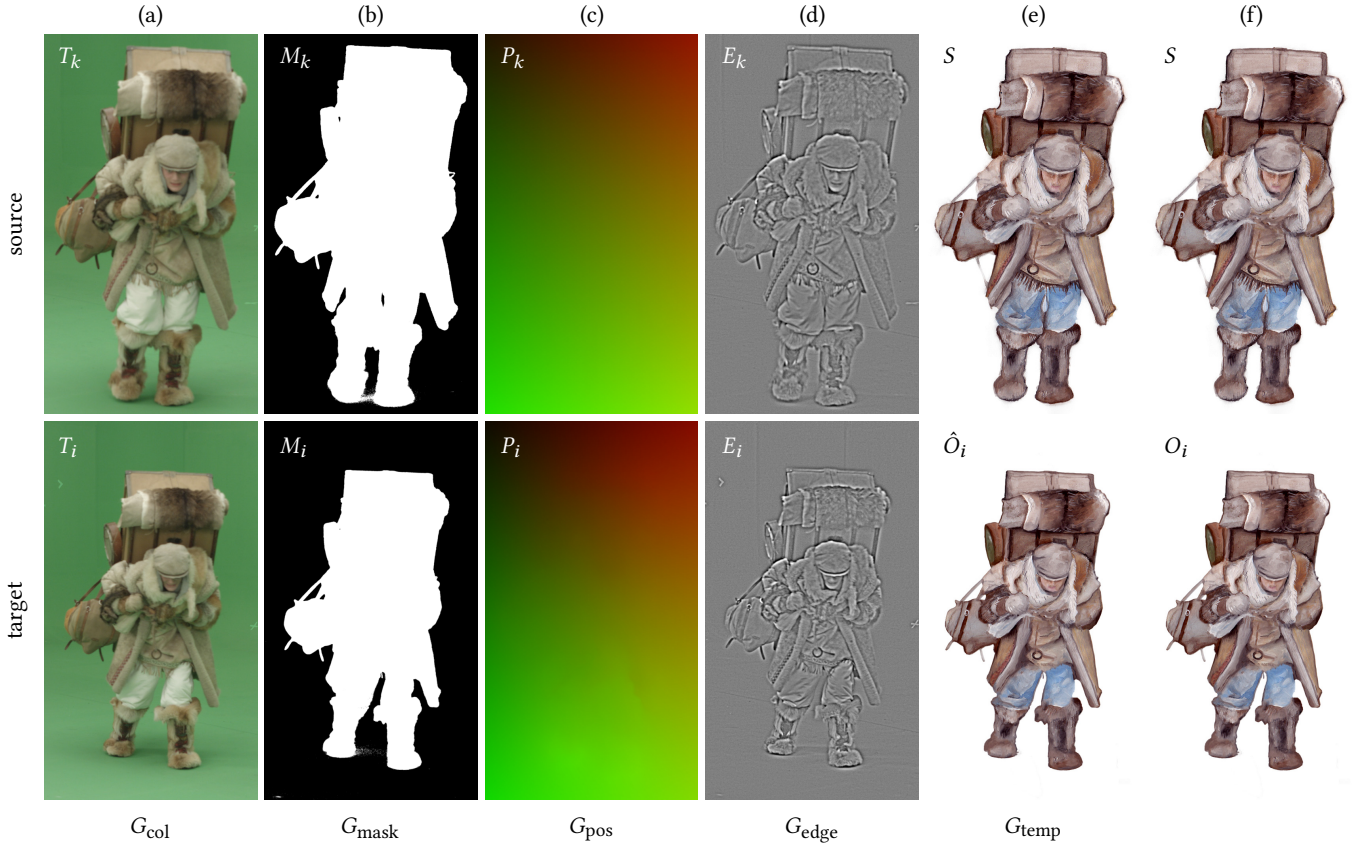

Fig. 3. The set of guidance channels used by our method. G_{col} is essential to preserve the target appearance, G_{mask} helps to preserve sharp object boundaries, G_{pos} is important to maintain overall structure, G_{edge} makes synthesis more robust to illumination changes and improves positioning of stylized features, G_{temp} is essential for temporal coherence. For further details, please refer to the text. Video frames (T_k and T_i) courtesy of © MAUR film, style exemplar (S) courtesy of © Pavla Sýkorová, used with permission.

Note, how the light brown wood texture from behind the subject shows up on the leather bag. We define G_{pos} as a dense correspondence map between the current frame T_i and the keyframe T_k. We compute this map by first estimating optical flow between consecutive frames of T using SIFT Flow [Liu et al. 2011]. This yields a sequence of inter-frame motion fields D_i, which we use to incrementally propagate the original pixel coordinates encoded in a coordinate map P_k (see Fig. 4a). We only perform this advection on the pixels inside the object mask M_k (Fig. 4b), and use diffusion [Orzan et al. 2008] to smoothly fill in the remaining values (Fig. 4c,d). The resulting map could introduce considerable texture distortion if used directly to warp stylized keyframes (see supplementary material). However, when used as a guide for patch-based synthesis, it encourages transfer of correct style features to the intended locations. Singularities and distortions that would result from direct advection are prevented by the other guiding terms, c.f., error metric (1).

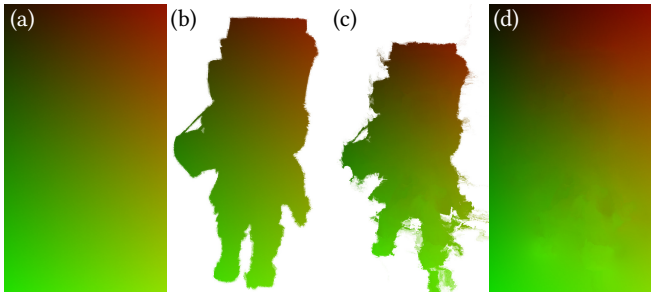

Fig. 4. Generating the G_{pos} guiding channel. Red and green color channels denote x and y coordinates. G_{pos} corresponding to the keyframe is constructed as a linear gradient in x - red, and y - green (a). Mask G_{mask} is then applied to the G_{pos} of the keyframe (b). Masked values are then propagated through the sequence according to the motion field D (c). Values outside of the mask are filled using diffusion [Orzan et al. 2008] (d).

Edge guide. G_{edge} highlights the object edges and salient features in the target sequence (see Fig. 3d), making the result less volatile with respect to color variation in G_{col} caused especially by changes in illumination. Because many artistic styles emphasize edges, this term has the additional benefit of "anchoring" appropriate style features (see Fig. 5c,d). We define $G_{\text{edge}}(T_i) = T_i - \mathcal{N}_\sigma \circ T_i$, where $\mathcal{N}_\sigma$ is a Gaussian filter with standard deviation σ.

Fig. 5. Importance of the G_{pos} and the G_{edge} guidance channels. (a) without the G_{pos}, some features with similar appearance cannot be fully distinguished. Note that the light brown wood texture from box on the subject's back appears on the leather bag at the bottom left corner. The G_{pos} term in (b) helps preserve the overall structure of the different features well. The G_{edge} (c-without, d-with) makes the synthesis less sensitive to illumination changes (see differences in stylization on top of the box) and helps preserving boundaries between the individual style features, thus making the result sharper.

Temporal guide. G_{temp} is designed to encourage temporal coherence by penalizing the synthesis from diverging too much from a previously synthesized frame [Fišer et al. 2017; Jamriška et al. 2015] (see Fig. 3e). We compute G_{temp} by advecting the stylization result of the previous frame O_{i-1} using the motion field D_i computed previously for G_{pos}. The advection produces a stylization prediction $\hat{O}_i$ which is not a satisfactory result on its own due to texture distortion, but as a guide, encourages temporally coherent stylization.

Error metric. The set of guiding channels we discussed thus far $\mathbb{G} = \{\text{col}, \text{mask}, \text{pos}, \text{edge}, \text{temp}\}$, defines a patch error measure that is plugged into the original StyLit algorithm [Fišer et al. 2016]. We use superscript $\mathcal{S}$ to denote the source part and $\mathcal{T}$ the target part of each guiding channel. The error metric for matching two patches $p \in \mathcal{S}$ and $q \in \mathcal{T}$ is then computed as follows:

$$E(S, O_i, G^{\mathcal{S}}, G^{\mathcal{T}}, p, q) = ||S(p) - O_i(q)||^2 + \sum_{g \in \mathbb{G}} \lambda_g ||G_g^{\mathcal{S}}(p) - G_g^{\mathcal{T}}(q)||^2 \tag{1}$$

where λ_g is a weighting factor for each individual guiding channel and the first term helps to preserve *texture coherence* by directly matching colors in patches of stylized keyframe S to those in the output frame O_i (see Fig. 3f). The style S and all guiding channels remain unchanged during the synthesis. Only O_i is iteratively updated. See [Fišer et al. 2016] for more details about the optimization.

3.2 Handling multiple keyframes

In many cases, it is sufficient to have only one keyframe. If, however, a sequence has new content appearing which did not exist in the keyframe and was not stylized, the artist may choose to specify a new keyframe to precisely control the stylization of the new content. The use of multiple keyframes introduces difficulty to the algorithm, since manually created keyframes will inevitably have subtle inconsistency in structure and colors. Previous approaches [Browning et al. 2014; Darabi et al. 2012; Shechtman et al. 2010] either suffer from detail clutter or produce temporal artifacts such as unnatural "boiling" or "pumping".

We propose a different solution which keeps the keyframe stylization unchanged while producing smooth and seamless transitions between keyframes. We first stylize the sequence using keyframes at the beginning (S_k^a) and at the end (S_l^b) to produce two separately stylized sequences O^a and O^b. To produce the final frame of index i, we blend the corresponding frames O_i^a and O_i^b. Now the question becomes what blending technique should we use.

A trivial approach would be to perform a linear blend: $O_i = (1-\alpha)O_k^a + \alpha O_l^b$, where $\alpha = (i-k)/(l-k)$. Such a solution flattens the original contrast and introduces ghosting artifacts (see Fig. 7b). In addition, linear blending implicitly assumes that the content of the frame changes smoothly in time; such an assumption is violated when there is disocclusion in the sequence, which suddenly introduces new local content that exists in keyframe S_l^b but not in S_k^a. In this case, we should stylize the new content using S_l^b exclusively, without blending in any features from S_k^a. To achieve this, we take advantage of the fact that our algorithm gives a patch matching error (1) for each pixel p in every frame for both O^a and O^b. Our intuition is that between two patches from O^a and O^b located at pixel p, the one with lower matching error will lead to "better" result and thus should be locally preferred.

Error-based gradient domain fusion. In order to merge the best content from the two stylized sequences, we use gradient domain fusion similar to that used in Image Melding [Darabi et al. 2012] (see Fig. 6), where a screened Poisson equation [Bhat et al. 2008] is applied to perform this task. Our solution, differs in how we select the gradient and how the screening value for reconstruction is computed. For the gradient, we select $\nabla O_i^a(p)$ or $\nabla O_i^b(p)$ according to a pixel selection mask Z_i (see Fig. 6) where white pixels indicate the state where the synthesis error $E_i^a(p)$ is lower than $E_i^b(p)$ and thus $\nabla O_i^a(p)$ is selected, while the opposite holds for black pixels. This ensures the blending result borrows the structure and high-frequency content from the synthesis result that most closely matches its respective keyframe.

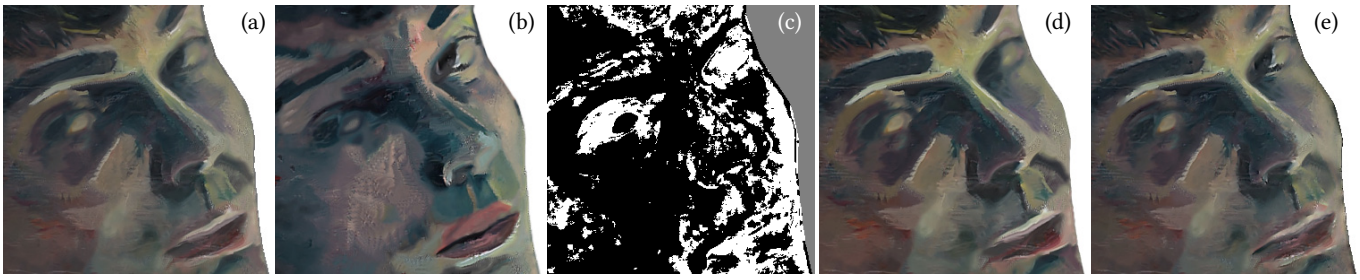

Fig. 6. Gradient domain mixing. Two stylized images O_i^a (a) and O_i^b (b) are synthesized at frame i using two different keyframes S_k and S_l. We compute a pixel selection mask Z_i (c) where black pixels indicate the locations where synthesis error E_i^a is lower than E_i^b and white pixels vice versa (gray indicates background). We then pick gradients according to Z_i (∇O_i^a for the black pixels and ∇O_i^b for the white ones) and run a screened Poisson solver on the contrast-preserving blend O_i^{ab} of O_i^a and O_i^b (d). Note that the resulting image O_i (e) contains high-frequency details according to Z_i.

Fig. 7. Comparison of different blending methods. (a) Regenerative Morphing [Shechtman et al. 2010] exhibits some detail clutter and loss of detail. (b) Linear blend leads to contrast loss and ghosting is visible. (c) The contrast-preserving linear blend [Heitz and Neyret 2018] has higher contrast, but the ghosting is still apparent. (d) Our approach has high contrast and ghosting is significantly suppressed.

Preserving color histogram. To ensure the global color histogram varies smoothly over time, we use a blended sequence O^{ab} for screening. Instead of linear blending, we use contrast-preserving blending from Heitz and Neyret [2018], which blends two images O_i^a and O_i^b and produces an image O_i^{ab} with a prescribed histogram H which is constructed by tabulating the colors of pixels according to pixel selection mask Z_i, i.e., we count the colors from pixels which have lower synthesis error. Though in O_i^{ab} ghosting artifacts still exist (see Fig. 7c), they are suppressed by the screened Poisson reconstruction in the resulting frame O_i. The screening value O_i^{ab} only serves to regularize the color histogram of the result.

Temporal coherence of pixel selection mask. Although the synthesis error usually increases as the target frame gets further away from the keyframe, the increase in error might not be monotonous in some local regions. This behavior may introduce visible flickering since the matching error constraint may cause the algorithm to frequently alternate between choosing contents stylized from different keyframes S_a and S_b. To avoid such temporal instability, we explicitly enforce temporal coherence of the pixel selection mask Z (see our supplementary material for illustrative figure). We store pixel selection mask Z_{i-1} from the previous frame and use estimated inter-frame motion field D_i to produce an initial $\hat{Z}_i$ which indicates existing pixel selection advected from the previous frame. We then update $\hat{Z}_i$ using the lower error constraint as described previously. However, we prevent the situation where a pixel that has already been assigned to take color and gradient information from an image stylized using the later keyframe S_b from switching back to the earlier keyframe S_a.

After applying this refinement, the resulting fused image has better contrast and less ghosting artifacts (Fig. 7d).

4 RESULTS

We pre-process guiding channels off-line on the CPU. This includes green screen matting, optical flow estimation, advection of content from the previous frame, and hi-pass filtering of the target video frame (using $\sigma = 6$). For a one-megapixel frame this process takes less than 20 seconds with the most time-consuming part being the computation of optical flow using SIFT flow method [Liu et al. 2011]. We use the following default setting of weights for individual guiding channels: $\lambda_{\text{col}} = 6$, $\lambda_{\text{pos}} = 2$, $\lambda_{\text{edge}} = 0.5$, $\lambda_{\text{mask}} = 1$, $\lambda_{\text{temp}} = 0.5$.

The actual synthesis then runs on the GPU (with CUDA) using the StyLit algorithm [Fišer et al. 2016] with the following settings: 5×5 patches, 6 pyramid levels, 12 search-vote iterations, and 6 PatchMatch sweeps [Barnes et al. 2009]. We also use the optimization described in [Fišer et al. 2017], i.e., the nearest neighbor field

Fig. 8. Eskimo sequence: digitally painted keyframe (a) was used to stylize the 148 frames long sequence (b, d, f), stylized frames (c) and (e). Video frames (b, d, f) courtesy of © MAUR film, stylized keyframe (a) courtesy of © Jakub Javora, used with permission.

Fig. 9. Lynx sequence: digitally painted keyframes (a) and (g) were used to stylize the 100 frames long sequence (b, d, f, h). Keyframe (a) was painted entirely while in keyframe (d) only few strokes were added on top of the synthesis result, stylized frames (c) and (e). Video frames (b, d, f, h) courtesy of © kjekol / Adobe Stock, stylized keyframes (a, g) courtesy of © Jakub Javora, used with permission.

propagation is executed only on patches that lower the matching error in previous search step. With this fine-tuning, we can synthesize one-megapixel frame in 9 seconds using GeForce GTX 1070.

For the fusion of sequences stylized from different keyframes we implemented the method of Heitz and Neyret [2018] as well as screened Poisson solver [Bhat et al. 2008] where we set the screening parameter $\lambda_d = 0.1$. The computation runs on the CPU and time for merging two one-megapixel frames is on average 10 seconds.

We validate our approach on multiple sequences from real production (please refer to our supplementary video) with varying complexity using different styles including physical, artistic media such as oil paint, watercolor, pencil drawing, and digital paint. The number of keyframes used for synthesis depends on the shot complexity. One keyframe is typically sufficient for shots where objects move mostly in the camera plane without occlusion or significant changes in illumination (see Figures 1, 8, 11, and 12). For more complex shots with out-of-plane rotation and illumination changes, two (Figures 9 and 14) or more keyframes (Figures 10 and 13) are necessary. The keyframes painted by an artist are highlighted with red rectangles in the figures. In a fully digital pipeline, not all keyframes need to be prepared from scratch. Instead, one can stylize the entire shot using one painted keyframe, and then manually fix deteriorated regions when needed. Frames with corrections become new keyframes (see Figures 9 and 10). For shots with frequent occlusions, we separate each frame into multiple layers for best synthesis quality and lowest number of keyframes (see our supplementary material).

Fig. 15. Comparison with patch-based techniques: (a) Frigo et al. [2019], (b) Benard et al. [2013], (c) our approach.

We compared our approach with the stylization framework proposed by Bénard et al. [2013] (see Fig. 15b). Although the original method does not support generic video stylization, we prepared the necessary guiding channels using our technique and provide

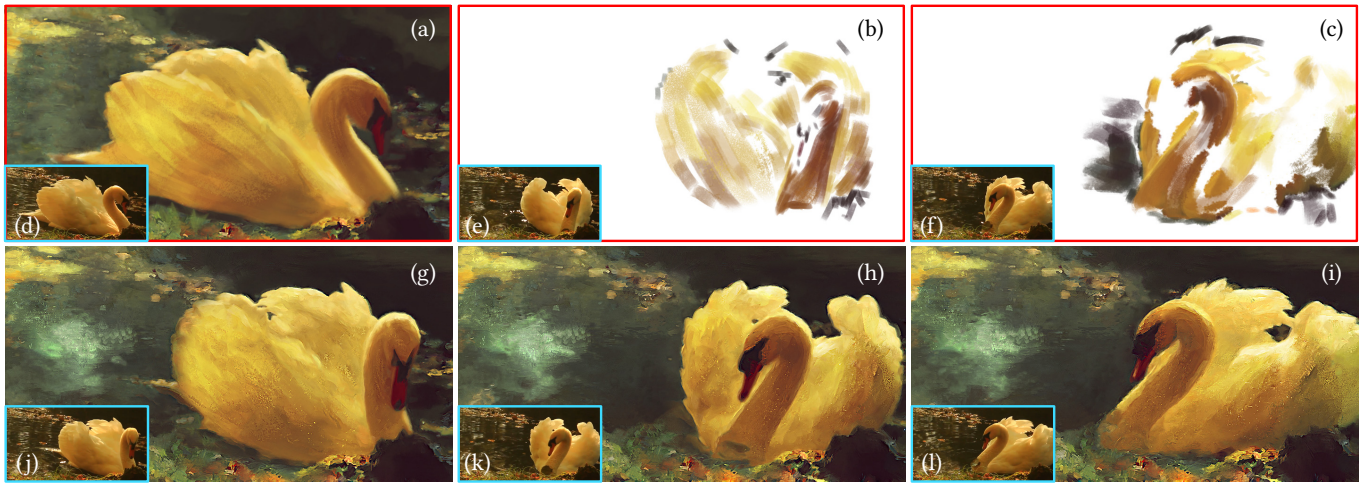

Fig. 10. Swan sequence: five digitally painted keyframes out of which three are shown in this figure (a, b, c) were used to stylize the 437 frames long sequence (d, e, f, j, k, l). Keyframe (a) was painted entirely while in keyframes (b) and (c) only few strokes were added on top of the synthesis result, stylized frames (g, h, i). Video frames (d, e, f, j, k, l) courtesy of © Primus1 / Adobe Stock, used with permission.

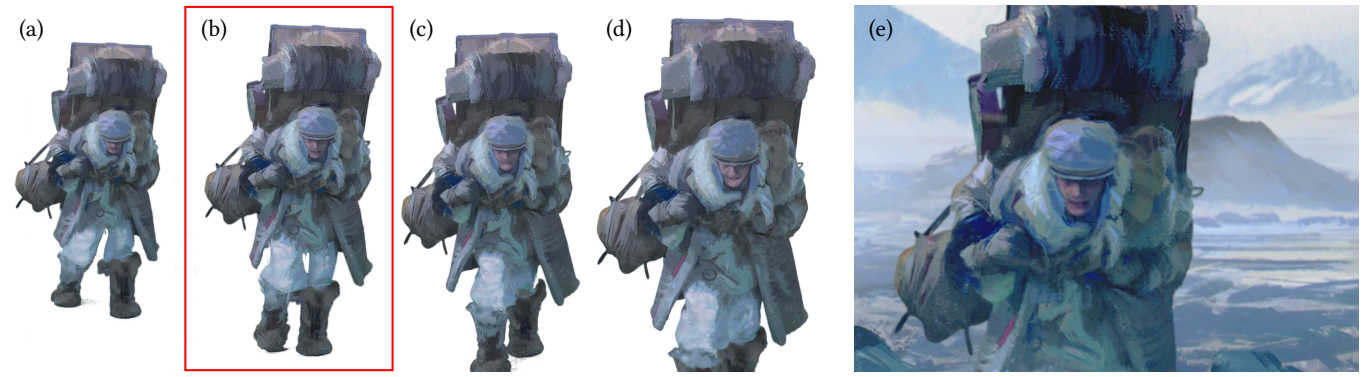

Fig. 11. Snowstorm composition: only one keyframe (b) was used to stylize video sequence with 521 frames including frames (a, c, d). The final composition (e). Stylized keyframe (b) and the final composition (e) courtesy of © Jakub Javora, used with permission.

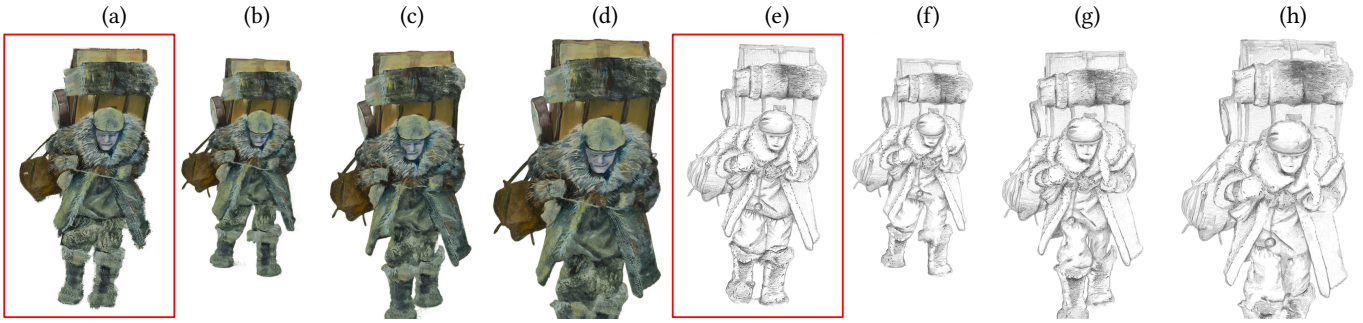

Fig. 12. Two different style exemplars—oil paint (a) and pencil drawing (e) were used to stylize the same set of target video frames as in Fig. 1. Style exemplars courtesy of © MAUR film, Václav Švankmajer (a), © Pavla Sýkorová (e), used with permission.

them as an input to their algorithm. We also compared with another patch-based technique that supports temporal coherence [Frigo et al. 2019] (see Fig. 15a). See our supplementary video for comparison on the entire sequence. We were interested in how well each algorithm preserves the quality of the original style exemplar and handles temporal coherence. From the results, Bénard et al.'s method has

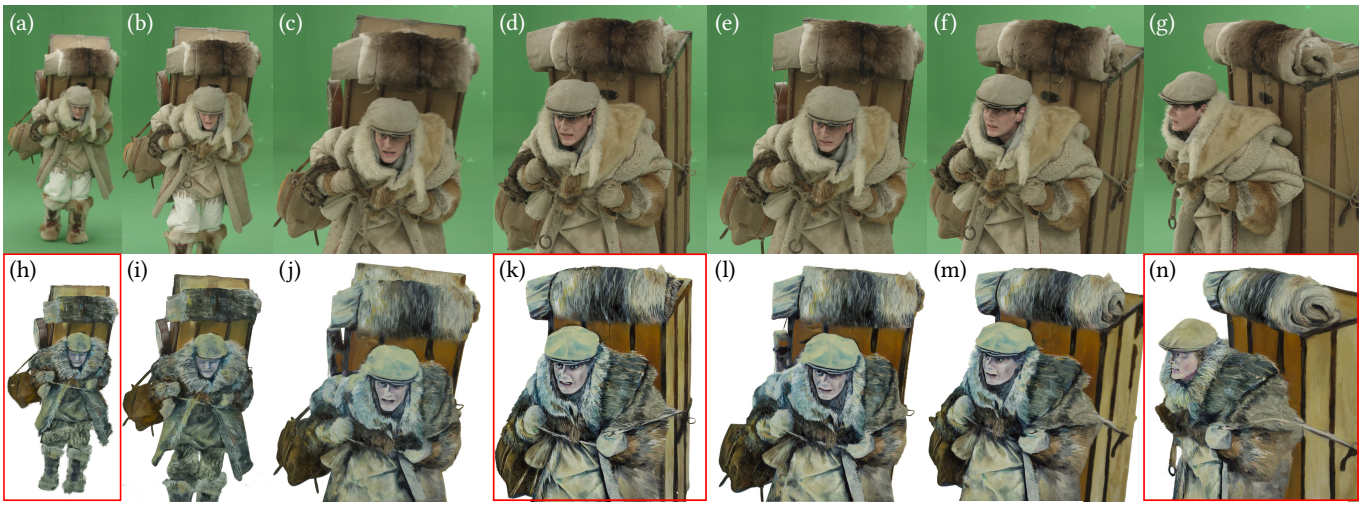

Fig. 13. Long video sequence with multiple keyframes: (a–g) are the target frames, (i, j, l, m) are resulting synthesized frames using two respective nearest keyframes (h, k, n). In total, the sequence contains 889 frames and 8 keyframes. Video frames (a–g) and stylized keyframes (h, k, n) courtesy of © MAUR film, Václav Švankmajer, used with permission.

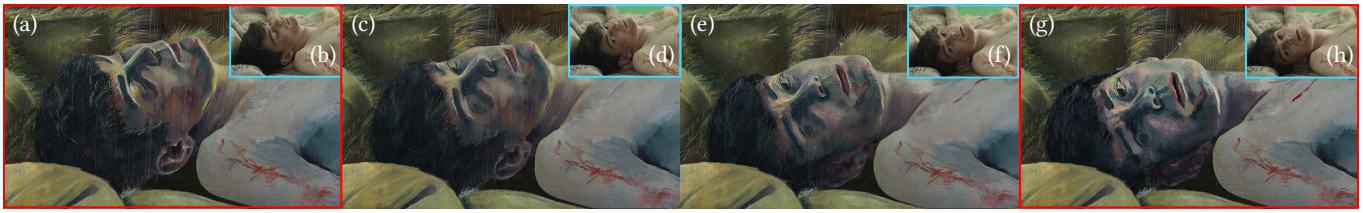

Fig. 14. Stylization between two keyframes: target video sequence (b, d, f, h) is first stylized using keyframe (a), then the same sequence is stylized using keyframe (g), and finally, the two resulting stylized sequences are then fused together (c, e). To stylize 1545 frames, only two keyframes were used. Video frames (b, d, f, h) and stylized keyframes (a, g) courtesy of © MAUR film, Václav Švankmajer, used with permission.

difficulty in preserving high-frequency details of the original style exemplar and tends to produce visible drifting chunks resulting in more temporal noise. Frigo et al.'s method also fails to preserve sharp details and cannot reproduce the colors in the style exemplar.

We also performed a comparison with state-of-the-art neural-based approaches (see Fig. 16 and our supplementary video). The method of Ruder et al. [2018] preserves temporal coherence, but does not fully transfer the details of the style exemplar. The method of Li et al. [2017] has similar appearance problems, and does not support temporal coherence. The approach of Liao et al. [2017] better reproduces the style, but introduces visible misalignment of salient features and again, does not preserve temporal coherence. The approach of Gu et al. [2018] can avoid the misalignment at the cost of smoothing out important high-frequency details of the original style exemplar. In addition, the last three methods suffer from severe temporal flickering when applied on video. We tried to post-process all three with the blind temporal consistency method of Lai et al. [2018]. Although the results were a bit temporally smoother, they exhibited additional loss of contrast and detail. See supplementary video.

5 LIMITATIONS AND FUTURE WORK

Although our new technique improves visual quality over the state-of-the-art and enables considerable reduction of manual labor in the creation of stylized videos, there are still some limitations that can motivate further research.

One of the key drawbacks of our approach is sensitivity to more substantial illumination changes in the target video. This may happen, e.g., when a part of the stylized object is originally in light, and then it enters a shadow. In this case, the inconsistent colors of G_{col} can be misleading. Although the use of G_{edge} and the diffuse studio lighting may suppress this behavior (see Fig. 5c,d and our supplementary material), a more advanced appearance matching technique would be helpful. Fišer et al. [2017] used the method of [Shih et al. 2014] that is, however, tailored to facial images. A more generic approach is needed in our scenario.

As a related problem, structural changes between nearby keyframes harm the synthesis quality. In some cases, separation into layers may help to reduce clutter and preserve content coherence. However, the appearance of target objects may change considerably if they contain dynamic high-frequency structures (e.g., distinct texture or wrinkles on clothing, see our supplementary material). This

Fig. 16. Comparison with recent neural-based methods: (a) Ruder et al. [2018], (b) Li et al. [2017], (c) Liao et al. [2017], (d) Gu et al. [2018], (e) ours. Fig. 13k was used as the style exemplar for all methods.

change will lead to inconsistencies in G_{col}. In these scenarios, more appropriate clothing or an additional detail-removing filtering [Bi et al. 2015; Xu et al. 2011] may help improve the synthesis quality.

Although our technique for mixing two stylized sequences does well in preserving contrast and suppressing ghosting, excessively large structural changes may still lead to subtle ghosting effect due to usage of blended screening target (see Fig. 7c, eyebrow in Fig. 14, and our supplementary material). Though solutions exist that can deform local features for better structural matching [Liao et al. 2014; Ruiters et al. 2010], we cannot apply them since we need to avoid free-form deformations that may destroy the structure of the original paint texture. A better warping scheme that preserves local high-frequency structure could potentially improve our method's tolerance to these large structural changes.

6 CONCLUSION

We presented a new approach to temporally coherent artistic stylization of video. Our two primary design considerations were (1) to allow direct and free-form artistic control in the form of keyframes painted in any desired traditional medium and (2) to support stylization of arbitrary input videos. Our approach enables a practical pipeline in real production shots for creating traditional-style animation from live-action performance capture. It further provides an easier artistic video creation workflow eliminating the need for a tedious frame-by-frame painting process while preserving the unique and rich visual qualities of traditional artistic media. We hope this will help bridge the gap between live action, 3D animation, and traditional hand-painted animation.

ACKNOWLEDGMENTS

We would like to thank Pavla Sýkorová, Markéta Kolářová, Jakub Javora, and MAUR film for providing testing video sequences and style exemplars. We are also grateful to all anonymous reviewers for their fruitful comments and suggestions. This research has 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, grant No. SGS19/179/OHK3/3T/13 (Research of Modern Computer Graphics Methods 2019–2021), by Research Center for Informatics, grant No. CZ.02.1.01/0.0/0.0/16_019/0000765, and by Adobe.

REFERENCES

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. 2009. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics* 28, 3 (2009), 24.

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

Pierre Bénard, Ares Lagae, Peter Vangorp, Sylvain Lefebvre, George Drettakis, and Joëlle Thollot. 2010. A Dynamic Noise Primitive for Coherent Stylization. *Computer Graphics Forum* 29, 4 (2010), 1497–1506.

Pravin Bhat, Brian Curless, Michael Cohen, and C. Lawrence Zitnick. 2008. Fourier Analysis of the 2D Screened Poisson Equation for Gradient Domain Problems. In *Proceedings of European Conference on Computer Vision*. 114–128.

Sai Bi, Xiaoguang Han, and Yizhou Yu. 2015. An L1 Image Transform for Edge-preserving Smoothing and Scene-level Intrinsic Decomposition. *ACM Transactions on Graphics* 34, 4 (2015), 78.

Adrien Bousseau, Matthew Kaplan, Joëlle Thollot, and François Sillion. 2006. Interactive Watercolor Rendering with Temporal Coherence and Abstraction. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 141–149.

Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. 2007. Video Watercolorization Using Bidirectional Texture Advection. *ACM Transactions on Graphics* 26, 3 (2007), 104.

Mark Browning, Connelly Barnes, Samantha Ritter, and Adam Finkelstein. 2014. Stylized Keyframe Animation of Fluid Simulations. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering*. 63–70.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *Proceedings of IEEE International Conference on Computer Vision*. 1114–1123.

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

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. 2012. Image Melding: Combining Inconsistent Images Using Patch-Based Synthesis. *ACM Transactions on Graphics* 31, 4 (2012), 82.

Marek Dvorožňák, Wilmot Li, Vladimir G. Kim, and Daniel Sýkora. 2018. ToonSynth: Example-Based Synthesis of Hand-Colored Cartoon Animations. *ACM Transactions on Graphics* 37, 4 (2018), 167.

Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. 2016. StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. *ACM Transactions on Graphics* 35, 4 (2016), 92.

Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. 2017. Example-Based Synthesis of Stylized Facial Animations. *ACM Transactions on Graphics* 36, 4 (2017).

Jakub Fišer, Michal Lukáč, Ondřej Jamriška, Martin Čadík, Yotam Gingold, Paul Asente, and Daniel Sýkora. 2014. Color Me Noisy: Example-Based Rendering of Hand-Colored Animations with Temporal Noise Control. *Computer Graphics Forum* 33, 4 (2014), 1–10.

Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. 2016. Split and Match: Example-Based Adaptive Patch Sampling for Unsupervised Style Transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 553–561.

Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. 2019. Video Style Transfer by Consistent Adaptive Patch Sampling. *The Visual Computer* 35, 3 (2019), 429–443.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2414–2423.

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3730–3738.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. 2018. Arbitrary Style Transfer with Deep Feature Reshuffle. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 8222–8231.

Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2017. Characterizing and Improving Stability in Neural Style Transfer. In *Proceedings of IEEE International Conference on Computer Vision*. 4087–4096.

Yoav HaCohen, Eli Shechtman, Dan Goldman, and Dani Lischinski. 2011. Non-rigid Dense Correspondence with Applications for Image Enhancement. *ACM Transactions on Graphics* 30, 4 (2011), 70.

William Van Haevre, Tom Van Laerhoven, Fabian Di Fiore, and Frank Van Reeth. 2007. From Dust Till Drawn: A Real-Time Bidirectional Pastel Simulation. *The Visual Computer* 23, 9-11 (2007), 925–934.

James Hays and Irfan A. Essa. 2004. Image and Video Based Painterly Animation. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 113–120.

Eric Heitz and Fabrice Neyret. 2018. High-Performance By-Example Noise Using a Histogram-Preserving Blending Operator. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 2 (2018), 31.

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

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *Proceedings of IEEE International Conference on Computer Vision*. 1510–1519.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. 5967–5976.

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

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Proceedings of European Conference on Computer Vision*. 694–711.

Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. 2015. Self Tuning Texture Optimization. *Computer Graphics Forum* 34, 2 (2015), 349–360.

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *Proceedings of European Conference on Computer Vision*. 179–195.

Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis.. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2479–2486.

Wenbin Li, Fabio Viola, Jonathan Starck, Gabriel J. Brostow, and Neill D. F. Campbell. 2016. Roto++: Accelerating Professional Rotoscoping Using Shape Manifolds. *ACM Transactions on Graphics* 35, 4 (2016), 62.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal Style Transfer via Feature Transforms. In *Advances in Neural Information Processing Systems*. 386–396.

Jing Liao, Rodolfo Lima, Diego Nehab, Hugues Hoppe, Pedro Sander, and Jinhui Yu. 2014. Automating Image Morphing Using Structural Similarity on a Halfway Domain. *ACM Transactions on Graphics* 33, 5 (2014), 168.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer Through Deep Image Analogy. *ACM Transactions on Graphics* 36, 4 (2017), 120.

Peter Litwinowicz. 1997. Processing Images and Video for an Impressionist Effect. In *SIGGRAPH Conference Proceedings*. 407–414.

Ce Liu, Jenny Yuen, and Antonio Torralba. 2011. SIFT Flow: Dense Correspondence across Scenes and Its Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 5 (2011), 978–994.

Cewu Lu, Li Xu, and Jiaya Jia. 2012. Combining Sketch and Tone for Pencil Drawing Production. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 65–73.

Santiago E Montesdeoca, Hock Soon Seah, Amir Semmo, Pierre Bénard, Romain Vergne, Joëlle Thollot, and Davide Benvenuti. 2018. MNPR: A Framework for Real-Time Expressive Non-Photorealistic Rendering of 3D Computer Graphics. In *Proceedings of The Joint Symposium on Computational Aesthetics and Sketch Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*. 11.

Alexandrina Orzan, Adrien Bousseau, Holger Winnemöller, Pascal Barla, Joëlle Thollot, and David Salesin. 2008. Diffusion Curves: A Vector Representation for Smooth-Shaded Images. *ACM Transactions on Graphics* 27, 3 (2008), 92.

Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. 2001. Real-Time Hatching. In *SIGGRAPH Conference Proceedings*. 581–586.

Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2018. Artistic Style Transfer for Videos and Spherical Images. *International Journal of Computer Vision* 126, 11 (2018), 1199–1219.

Roland Ruiters, Ruwen Schnabel, and Reinhard Klein. 2010. Patch-Based Texture Interpolation. *Computer Graphics Forum* 29, 4 (2010), 1421–1429.

Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. 1997. Orientable Textures for Image-Based Pen-and-Ink Illustration. In *SIGGRAPH Conference Proceedings*. 401–406.

Artsiom Sanakoyeu, Dmytro Kotovenko, Sabine Lang, and Björn Ommer. 2018. A Style-Aware Content Loss for Real-Time HD Style Transfer. In *Proceedings of European Conference on Computer Vision*. 715–731.

Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert Sumner. 2011. OverCoat: An Implicit Canvas for 3D Painting. *ACM Transactions on Graphics* 30, 4 (2011), 28.

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

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

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

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

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1526–1535.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S. Lempitsky. 2016a. Texture Networks: Feed-Forward Synthesis of Textures and Stylized Images. In *ICML*, Vol. 48. 1349–1357.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016b. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR* abs/1607.08022 (2016).

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2017. Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 4105–4113.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Nikolai Yakovenko, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *Advances in Neural Information Processing Systems*. 1152–1164.

Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. 2017. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 7178–7186.

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

Pierre Wilmot, Eric Risser, and Connelly Barnes. 2017. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. *CoRR* abs/1701.08893 (2017).

Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. 2011. Image Smoothing via L0 Gradient Minimization. *ACM Transactions on Graphics* 30, 6 (2011), 174.

Kaan Yücer, Alec Jacobson, Alexander Sorkine-Hornung, and Olga Sorkine-Hornung. 2012. Transfusive Image Manipulation. *ACM Transactions on Graphics* 31, 6 (2012), 176.

Mingtian Zhao and Song-Chun Zhu. 2011. Portrait Painting Using Active Templates. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 117–124.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017a. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of IEEE International Conference on Computer Vision*. 2242–2251.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems*. 465–476.