StyleBin: Stylizing Video by Example in Stereo

Michal Kučera
CTU in Prague, FEE
Prague, Czech Republic
kucerm22@fel.cvut.cz

David Mould
Carleton University, SCS
Ottawa, Canada
mould@scs.carleton.ca

Daniel Sýkora
CTU in Prague, FEE
Prague, Czech Republic
sykorad@fel.cvut.cz

ABSTRACT

In this paper we present StyleBin—an approach to example-based stylization of videos that can produce consistent binocular depiction of stylized content on stereoscopic displays. Given the target sequence and a set of stylized keyframes accompanied by information about depth in the scene, we formulate an optimization problem that converts the target video into a pair of stylized sequences, in which each frame consists of a set of seamlessly stitched patches taken from the original stylized keyframe. The aim of the optimization process is to align the individual patches so that they respect the semantics of the given target scene, while at the same time also following the prescribed local disparity in the corresponding viewpoints and being consistent in time. In contrast to previous depth-aware style transfer techniques, our approach is the first that can deliver semantically meaningful stylization and preserve essential visual characteristics of the given artistic media. We demonstrate the practical utility of the proposed method in various stylization use cases.

CCS CONCEPTS

• **Computing methodologies** → **Non-photorealistic rendering**; **Virtual reality**.

KEYWORDS

video style transfer, example-based, stereo

ACM Reference Format:
Michal Kučera, David Mould, and Daniel Sýkora. 2022. StyleBin: Stylizing Video by Example in Stereo. In *SIGGRAPH Asia 2022 Conference Papers (SA '22 Conference Papers), December 6–9, 2022, Daegu, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3550469.3555420

1 INTRODUCTION

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

Given an input video and one or more stylized keyframes accompanied by information about depth in the scene, we synthesize a stylized output sequence for each eye. Our approach is a patch-based synthesis process where patch selection is informed by a family of guidance channels seeking to match aspects of the images, including color, position, and edges; the method is similar to that of Jamriška et al. [2019], though we must contend with the added difficulty of adapting the guidance channels to right and left eye views and then synthesizing both views consistently in time and space. Our use of patches guarantees accurate reproduction of important planar structures in the style exemplar and the disparity-adapted guidance channels ensure their semantically meaningful transfer.

This paper's main contribution is its versatile framework for stereo stylization, able to reliably create stereoscopic video with semantically-meaningful stylization from an input monocular video and sparse style/depth keyframes. It extends the works of Jamriška et al. [2019] and Luo et al. [2015] to the stereo stylization setting. Its key technical contribution is the joint synthesis of stereo and temporal consistency. We demonstrate the effectiveness of the approach with several examples and a qualitative user study.

SA '22 Conference Papers, December 6–9, 2022, Daegu, Republic of Korea

ACM ISBN 978-1-4503-9470-3/22/12…$15.00
https://doi.org/10.1145/3550469.3555420

2 RELATED WORK

Our work builds on advances in style transfer and draws on past efforts to create and modify stereoscopic video. We discuss these topics in turn.

Recent improvements in example-based style transfer have been based on guided texture synthesis [Fišer et al. 2016; Jamriška et al. 2015; Sýkora et al. 2019] and on neural methods [Gatys et al. 2016; Liao et al. 2017]. The problem of example-based stylization of videos has received considerable attention [Fišer et al. 2017; Futschik et al. 2021; Jamriška et al. 2019; Ruder et al. 2018; Texler et al. 2020]. Stylization of 3D models has been studied [Bénard et al. 2013] and continues to be an area of interest, with Hauptfleisch et al. [2020] providing a recent example.

The above-cited work, however, little considers stereo images, which is a specialized topic with its own literature. Stavrakis and Gelautz [2004] were the first to consider computer-generated stylized stereo images and identified many of the challenges in stylized stereo, including the need for planarity of style elements in the output. They used a stroke-based rendering system, ensuring consistency by enforcing similar stroke placement across right and left views. Northam et al. [2012] propose a more general framework for stylized stereo images which uses multiple discrete disparity layers and a separate stylization for each layer. While this approach was effective for still images, the discretization of layers is problematic for application to video.

Considerable effort has also been directed towards synthesizing stereo line drawings. Kim et al. [2013] laid the groundwork for this area, working with 3D geometry as an input. They note that the simple approach of detecting and rendering silhouettes separately from each eye creates incoherent collections of lines. By rendering only matched pairs of lines and excluding lines that cannot be fused in stereo, they are able to create a high-quality experience of 3D stereo line drawings from geometry. Later work [Bukenberger et al. 2018; Istead and Kaplan 2018; Istead et al. 2021] produced stylized line drawings from stereo depth images. Other researchers have considered also specialized systems for particular effects and scenarios, such as film grain in stereo [Templin et al. 2014] or stylization of lightfields [Egan et al. 2021].

Application of neural style transfer to stereo images or to generation of novel views has enjoyed some success recently [Chen et al. 2018; Gong et al. 2018; Huang et al. 2021]. Such systems incorporate estimates of stereo or multi-view consistency into the loss function. However, the resulting stylization does not guarantee semantically meaningful transfer and also distorts visually important features seen in the original exemplar such as individual brush strokes or a canvas structure.

The work most similar to ours was undertaken by Luo et al. [2015], who use a patch-based approach for coherence-preserving modification of stereo images. However, they do not consider stereo-consistent example-based stylization of videos, which remains an open problem.

3 OUR APPROACH

The input to our method is a target sequence T and a selection of one or more keyframes $K \subset T$ for which the user will provide (i) a stylized counterpart S_k and (ii) a disparity map D_k (see Fig. 1) that can be obtained manually or automatically. In our experiments we employ boosted monocular depth estimation [Miangoleh et al. 2021], and when applicable, also use Attention Mesh [Grishchenko et al. 2020] with Poisson image editing [Pérez et al. 2003] to improve disparity in facial regions. The precise choice of method is unimportant; any other depth estimation technique or additional depth sensor can be used to obtain D_k. The user may also decide to refine D_k manually to achieve the desired disparity.

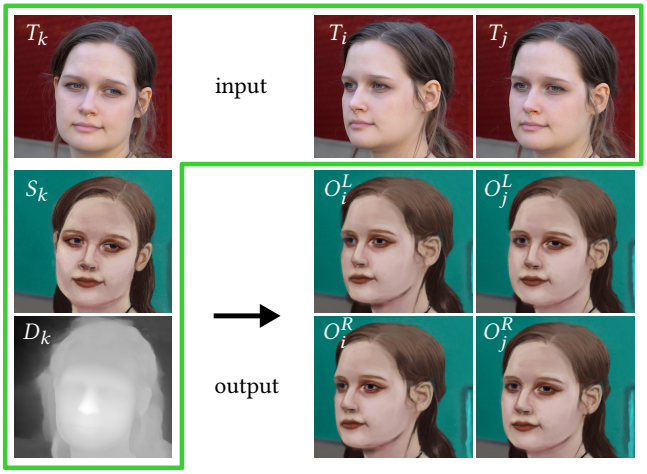

Figure 1: An overview of the inputs and outputs of our method. The user provides a target sequence T in which one or more keyframes $T_k \in K$ are stylized S_k and contain information about disparity D_k. We propagate the disparity from D_k to the entire sequence T and transfer the style from S_k to T such that two stylized sequences O^L and O^R are produced, each of which can then be viewed by the corresponding eye to achieve a stereoscopic effect. Video frames T and style exemplar S_k © Jana Kyllerová.

The goal of our method is to produce two temporally coherent output sequences, a left sequence O^L and a right sequence O^R (see Fig. 1), in which the target sequence T will be stylized according to the style exemplar S_k such that when the frames from O^L and O^R are displayed to the corresponding eyes, the viewer will see a stereo effect driven by the disparity map D_k. This also means that O^L and O^R need to be consistent both in space and time to avoid ghosting and flickering artifacts.

We first describe the general approach to produce O^L and O^R from T using S_k and D_k. Later, in Section 4, we demonstrate that the individual building blocks of our method can be applied in different scenarios: for example, we may have a target sequence T that is already fully stylized, or we may know D for each frame beforehand, perhaps because T was generated by 3D rendering or captured using a depth sensor.

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

both stylized views are consistent in time and space according to the motion in the scene and the disparity given by D_k.

3.1 Disparity propagation

As an initial step, we need to propagate the disparity stored in D_k from each keyframe $T_k \in K$ to the rest of the target sequence T (see Fig. 2). To do that, we employ the guided patch-based synthesis of Jamriška et al. [2019], providing the disparity map D_k as the style exemplar. In the case of multiple keyframes K, we propagate disparity to T from each keyframe separately and then blend the resulting frames using a weight proportional to the distance in time between the currently blended frame and the keyframe.

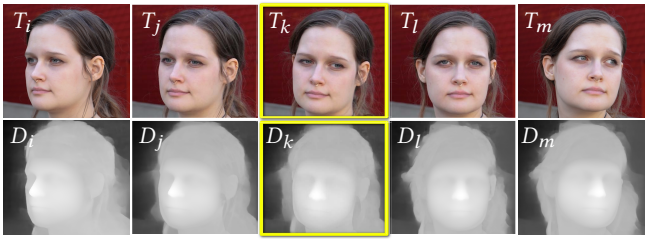

Figure 2: An example of disparity map D_k propagation from a keyframe T_k to the rest of the sequence T. An output of this process is a sequence of disparity maps D aligned with every frame in T. Video frames T © Jana Kyllerová.

3.2 Disparity shifting

As a byproduct of the previous disparity propagation step, a set of auxiliary channels $C = \{F, G_{\text{color}}, G_{\text{edge}}, G_{\text{pos}}\}$ is produced for each frame in T (see Fig. 3 and c.f. Jamriška et al. [2019]). Here F is the optical flow computed between the consecutive frames in T using the method of Kroeger et al. [2016]. During the synthesis, F is used to help enforce temporal consistency. The channel G_{color} is a color guide that stores copies of individual frames of T. It helps to ensure that the style from S_k is transferred to locations where T_i has similar colors to those in T_k. G_{edge} denotes an edge guide that encourages salient features in T_i to be stylized consistently with those stored in T_k. G_{edge} is computed as follows: $G_{\text{edge}}(T_i) = T_i - \mathcal{N}_\sigma \circ T_i$, where $\mathcal{N}_\sigma$ is a Gaussian filter with standard deviation σ and $\circ$ denotes convolution. Finally, G_{pos} is a positional guide that encourages transfer of style pixels from keyframe T_k to the corresponding positions in the current frame T_i. G_{pos} is computed by accumulating a series of consecutive optical flows $F_{i-k} \in F$ between frames T_i and T_k.

The sequence of optical flows F plus the above-mentioned guiding channels G are sufficient to perform style transfer using the original method of Jamriška et al. In our setting, however, we need to produce a binocular sequence, for which we need a left C^L and right C^R view for each channel in C. Those new views can be obtained by shifting the content in C using the disparities stored in D; see Fig. 3.

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

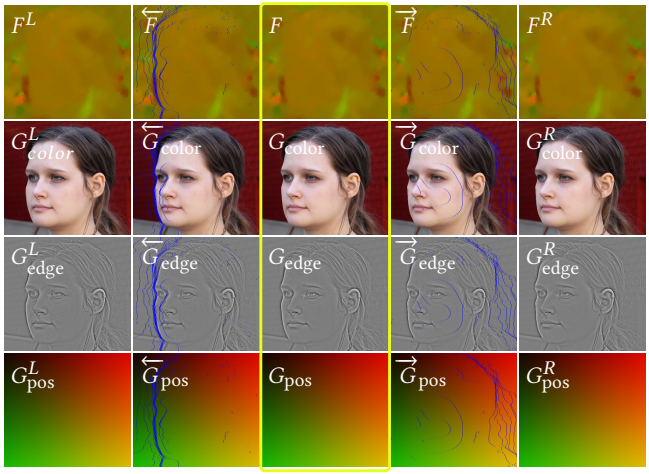

Figure 3: An example of shifting and completion of auxiliary channels $C = \{F, G_{\text{color}}, G_{\text{edge}}, G_{\text{pos}}\}$: optical flow F as well as guiding channels G are first shifted to the left $\overleftarrow{C}$ and to the right $\overrightarrow{C}$ using disparities stored in D, and then disoccluded areas are filled using disparity-guided patch-based synthesis to obtain complete properly aligned auxiliary channels C^L and C^R for the left and right views. Video frame G_{color} © Jana Kyllerová.

3.3 Handling disocclusion

After the shifting phase, a subset of the pixels in channels $\overleftarrow{C}$ and $\overrightarrow{C}$ may remain untouched due to disocclusion (see blue areas in Fig. 3). To fill those gaps, we first apply the disparity completion approach of Wang et al. [2008] to obtain consistent left D^L and right D^R disparity maps. Once D^L and D^R are available, we can employ disparity-guided patch-based synthesis, similar to that used by Luo et al. [2015]. Here the goal is to minimize the following:

$$\mathrm{E}_D(C^S, C^V) = \sum_{\hat{t} \in C^V} \min_{\hat{s} \in C^S} \mathrm{Q}(\hat{s}, \hat{t}), \tag{1}$$

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

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

Here s and t are individual pixels from patches $\hat{s}$ and $\hat{t}$, and w_{dis} is the weight of a disparity term that compares the disparity of the source pixel s stored in D^S with the disparity of the target pixel t stored in D^V (substituting here either for D^R or D^L). Note that D^S was obtained in the disparity propagation phase (Section 3.1) while D^V

originates from the preceding disparity completion step. The following disparity-dependent dissimilarity term helps to control the smoothness of the synthesized channel C^V (here C is one of F, G_{color}, G_{edge}, or G_{pos}, and V stands for L or R). By setting

$$w_{\text{val}}(s,t) = \exp(-|D^S(s) - D^V(t)|^2/\sigma^2) \tag{3}$$

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

3.4 Final synthesis

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

$$\mathrm{E}_S(S_k, O_i^L, O_i^R) = \sum_{\hat{t}^L \in O_i^L} \min_{\hat{s}^L \in S_k} \mathrm{M}^L(\hat{s}^L, \hat{t}^L) + \sum_{\hat{t}^R \in O_i^R} \min_{\hat{s}^R \in S_k} \mathrm{M}^R(\hat{s}^R, \hat{t}^R), \tag{4}$$

which is a sum of two partial energies computed over the left O_i^L and right O_i^R stylized views. The aim is to find a source patch $\hat{s}^L \in S_k$ for each target patch $\hat{t}^L \in O_i^L$ in the left view and a source patch $\hat{s}^R \in S_k$ for each target patch $\hat{t}^R \in O_i^R$ in the right view that minimizes the following patch dissimilarity metric (see Fig. 4):

$$\begin{aligned}\mathrm{M}^V(\hat{s},\hat{t}) = \sum_{s\in\hat{s}, t\in\hat{t}} & w_{\text{tex}}\mathrm{M}^V_{\text{tex}}(s,t) + w_{\text{color}}\mathrm{M}^V_{\text{color}}(s,t) + \\ & w_{\text{pos}}\mathrm{M}^V_{\text{pos}}(s,t) + w_{\text{edge}}\mathrm{M}^V_{\text{edge}}(s,t) + \\ & w_{\text{temp}}\mathrm{M}^V_{\text{temp}}(s,t) + w_{\text{uni}}\Omega(s).\end{aligned} \tag{5}$$

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

$$\mathrm{M}^V_{\text{tex}}(s,t) = |S_k(s) - O_i^V(t)|^2 + w_{\text{stereo}}|S_k(s) - O_i^{\neg V}(t \pm D_i^V(t))|^2, \tag{6}$$

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

$$\mathrm{M}^V_{\text{guide}}(s,t) = |G_k^S(s) - G_i^V(t)|^2 + w_{\text{stereo}}|G_k^S(s) - G_i^{\neg V}(t \pm D_i^V(t))|^2. \tag{7}$$

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

$$\mathrm{M}^V_{\text{temp}}(s,t) = |S_k(s) - F_i^V[O_{i-1}^V](t)|^2. \tag{8}$$

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

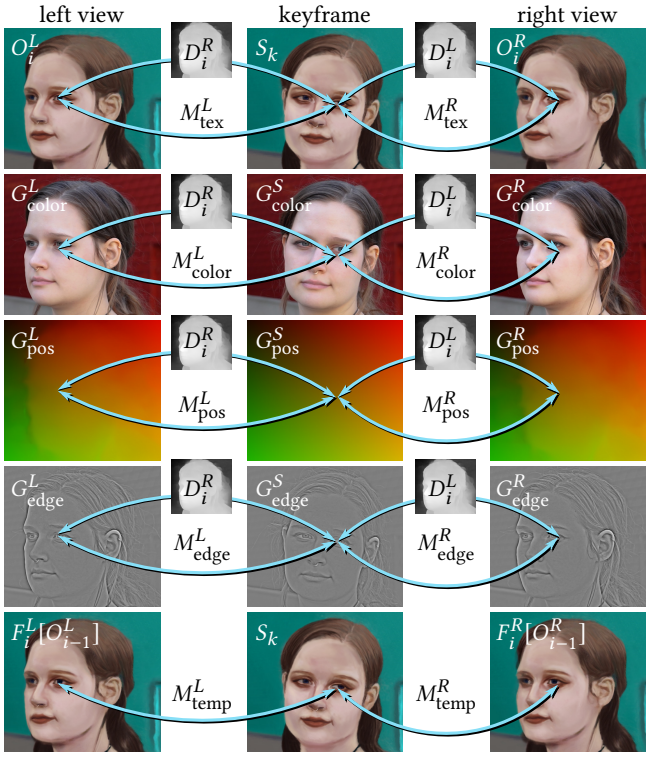

Figure 4: An overview of terms consisting of patch dissimilarity metrics $\mathbf{M}^L$ and $\mathbf{M}^R$ and their dependence on auxiliary channels. See the text for the detailed explanation. Video frame G^S_{color} and style exemplar S_k © Jana Kyllerová.

3.5 Optimization

To minimize E_D and E_S, we use the EM-like optimization scheme proposed by Wexler et al. [2007] and later refined by Kaspar et al. [2015] to update the patch occurrence term. During the optimization of E_D, only patches whose central pixel lies within the

disoccluded area are modified. All others remain unchanged and serve as boundary conditions to encourage the synthesis to produce seamless transitions between the original shifted pixels and those being synthesized to fill in dissocluded areas. In the case of E_S, the optimization runs over all target pixels since the style S needs to be consistently propagated to the entire frames. In the case of multiple keyframes, we transfer the style from each exemplar S separately and then perform linear blending to obtain the final merged sequence. Alternatively, a more advanced merging based on a screened Poisson equation can be used as described in [Jamriška et al. 2019].

4 RESULTS

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

To demonstrate the versatility of our framework, we prepared a selection of testing sequences with a variety of input data. These include one or more stylized keyframes in different styles with depth information obtained via boosted monocular depth estimation [Miangoleh et al. 2021] or rendered from a 3D model aligned with the target scene [Grishchenko et al. 2020]. We also demonstrate a use case when the target sequence is partly or entirely stylized and where keyframes are produced using a different style transfer method or contain only information about the depth in the scene. All results are presented in Figures 5 and 6 where the stereo effect can be seen using red-cyan anaglyph glasses. The full stylized sequences are also presented in the supplementary video, rendered both in red-cyan anaglyph and side-by-side mode. The latter is suitable for a cardboard or a VR headset, where the resulting stereo effect is most apparent.

The *Lili* sequence (see Fig. 5.1) contains subtle head motions. We prepared a single keyframe with an oil painting as a style exemplar and obtained depth information by combining boosted monocular depth estimation [Miangoleh et al. 2021] and a rendering of a 3D face model aligned with the head pose in the keyframe [Grishchenko et al. 2020]. We merged those two sources using Poisson image editing [Pérez et al. 2003]. The rest of the sequence was stylized using our approach, i.e., depth was propagated to the remaining frames, left and right auxiliary channels were produced, and finally the synthesis was executed to obtain the final stylized views.

In the *Jana* sequence (see Fig. 5.2) with its more dramatic head motion, a single keyframe was digitally painted by hand and then three other keyframes were generated using STALP [Futschik et al. 2021]—a neural style transfer method that handles more dramatic changes in the scene. For those additional keyframes, depth information was estimated using the same approach as for the *Lili* sequence, i.e., we combined estimated and rendered depth maps. We used our approach to propagate depth and stylize the sequence from each keyframe and then we blended them to produce the final output.

The *Selfie* sequence (see Fig. 6.1) shows a human head with moving body and the *Lynx* sequence (see Fig. 6.2) depicts an animal in motion. For each of these sequences, two keyframes were digitally painted and depth was estimated using [Miangoleh et al. 2021]. Our approach was used to propagate depth and stylize the sequence using both keyframes. The final output was produced by blending.

Finally, sequences *Knights* and *Alchemist* (see Figures 5.3 and 6.3) were created in monocular view by an artist using a combination of hand-painted layers that undergo parallax motion and the video style transfer method of Jamriska et al. [Jamriška et al. 2019]. Depth for those two sequences was obtained by generating eight keyframes using [Miangoleh et al. 2021]. Our method was then used to propagate the depth from the keyframes, construct auxiliary channels, and perform the synthesis to produce the resulting stereo pairs.

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

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

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

Figure 5: A collection of three different sequences stylized using our approach—*Lili* Fig. 5.1, *Jana* Fig. 5.2, and *Knights* Fig. 5.3. From *Lili*'s and *Jana*'s input sequences (1d & 2d) a single keyframe was selected (1a & 2a) for which a stylized counterpart was prepared by an artist (1b & 2b) and also a depth map specified (1c & 2c). Our method then produced the final binocular sequences (1e & 2e) of which anaglyph examples are shown in (1f & 2f). In the case of *Knights*. the input sequence (3d) was already stylized by an artist, and the aim here is to add a stereoscopic effect (3e). To do that, our method propagates depth information (3b) from a set of keyframes (3a) to the entire sequence and synthesizes the stylized stereo view (3f). See also our supplementary video for a side-by-side version of this result. Video frames (1a) & (1d) © Michal Dvořák, video frames (2a) & (2d) and style exemplar (2b) © Jana Kyllerová, stylized video frames (3a) & (3d) © Jakub Javora.

5 DISCUSSION

In the previous section, we showed a variety of examples with different styles and arising from different input scenarios. The corresponding videos can be seen in the supplementary material. As the anaglyph view provides only an approximation of the stereo effect, We strongly recommend viewing the resulting sequences in VR, whether with a full headset or with a cardboard viewer.

Our method extends video stylization from monocular video to stereo. The stereo output is largely free of objectionable stereo inconsistencies while maintaining coherence of style elements across views and conveying the sense of observing a 3D world while still preserving the 2D essence of the style.

Our method shares some limitations with the approach of Jamriška et al. [2019]. Both techniques are sensitive to significant changes in the input video (e.g., viewpoint, pose or illumination) and can find it difficult to propagate high-frequency details from the style exemplar through the full video sequence. This drawback can be mitigated by providing additional corrective keyframes, either manually or using a more advanced style transfer technique such as STALP [Futschik et al. 2021] as we demonstrated in the *Jana* example.

There is also some dependence on the quality of input depth. While monocular depth estimation is outside the scope of our contribution, inaccurate depth maps may impose problems on us, sometimes manifesting as inconsistent halo effects or a lack of depth perception. We demonstrated how to partially mitigate this by fitting a 3D mesh [Grishchenko et al. 2020] into the input sequence to obtain higher-quality depth values in facial regions, but a more general solution remains an open problem.

Our method may encounter difficulties in scenarios where more accurate reconstruction of disocclusions is necessary. Our expectation is that holes are relatively small and thus there is no need to handle continuation of semantically meaningful structures in the scene. For larger holes or a complex configuration of occluders (e.g., a dense forest with leaves and branches blowing in the wind), more elaborate methods would be required.

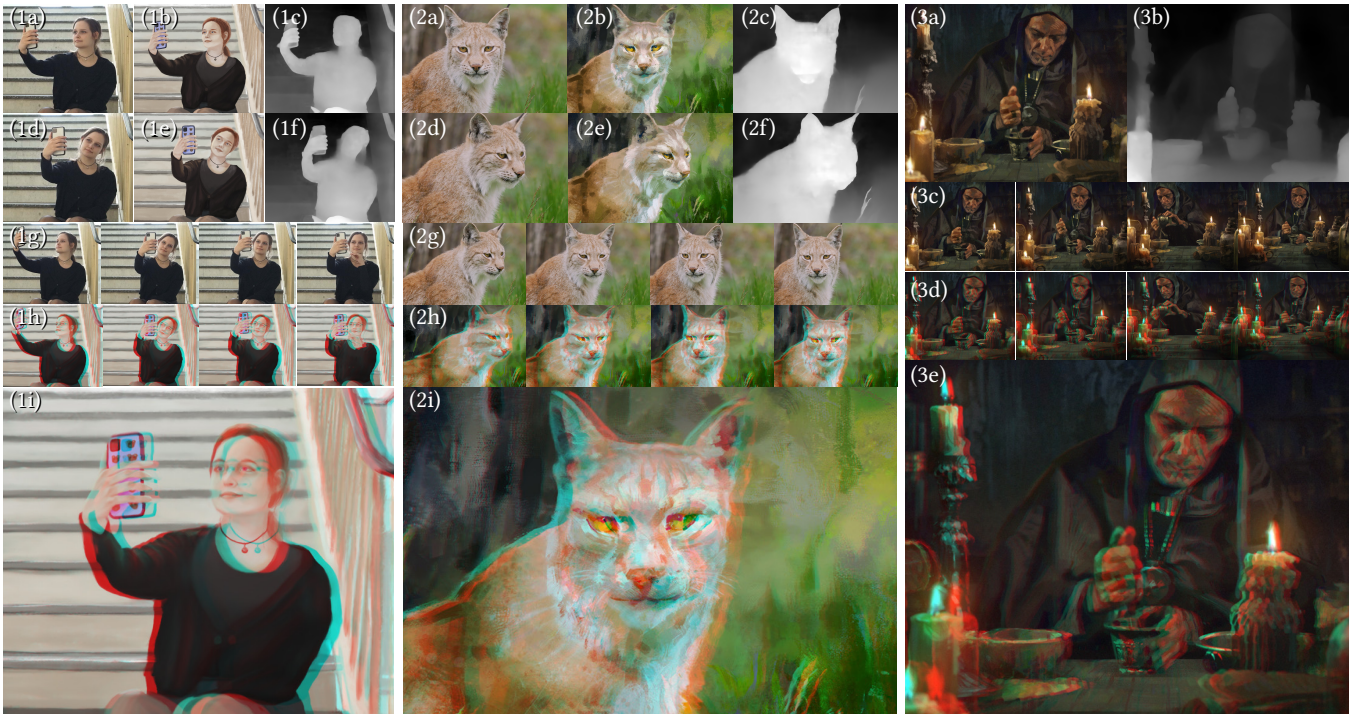

Figure 6: StyleBin applied to three different sequences—*Selfie* Fig. 6.1, *Lynx* Fig. 6.2, and *Alchemist* Fig. 6.3. From *Selfie*'s and *Lynx*'s input sequences (1g & 2g) the user will pick two keyframes (1a, 1d, 2a, 2d), prepare their stylized variants (1b, 1e, 2b, 2e), and provide an estimate of depth in the scene (1c, 1f, 2c, 2f). Our method then transfers the style from those keyframes onto the rest of the video (1g & 2g) producing a consistent stereo sequence (1h & 2h) of which one frame is displayed here as a red-cyan anaglyph (1i & 2i). In the case of *Alchemist*, the input video (3c) was already stylized by an artist. A set of depth maps (3b) is provided for a selection of keyframes (3a). Our algorithm then propagates the information about depth to the entire stylized video and synthesizes a stereo sequence (3d). An anaglyph close-up of one frame from our stereoscopic output is shown in (3e). See also our supplementary video for a side-by-side version of this result. Video frames (1a), (1d) & (1g) and style exemplars (1b) & (1e) © Jana Kyllerová, style exemplars (2b) & (2e) and stylized video frames (3a), (3c) & (3d) © Jakub Javora.

6 CONCLUSION

We present an example-based approach for style transfer in the stereo image setting. Our input is a monocular video sequence plus one or more stylized keyframes with information about depth. We extrapolate the provided style and depth across all frames and produce right and left stylized views. Our method uses a guided patch-based synthesis, choosing patches from the style exemplar such that an energy function is minimized, measuring similarity according to multiple guiding channels (as done by Jamriška et al. [Jamriška et al. 2019]) and enforcing consistency across the video frames as well as between the corresponding patches in right and left views.

We showed results in a variety of styles and with varied input conditions. The patch-based synthesis allows us to preserve the 2D aspects of the style while the joint optimization of temporal and stereo consistency produces a comfortable viewing experience. In our user study, no participants experienced discomfort or expressed concerns about their 3D interpretation of the scene. We believe our approach could simplify creation of stylized stereoscopic videos.

Some scope remains for future work. Depth estimation is an ongoing area of research, and our method would benefit from advances here. Further, while our method deals adequately with simple disocclusions, complex scenes may impose difficulties; future work can attempt to extrapolate the style exemplar across more dramatic changes and disocclusions.

Our method could also help to bootstrap machine learning methods for stereo stylization. Training data for still images is abundant; stylized video is rarer, and stylized stereo video rarer still. Our method could be used in data amplification in this setting.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback and insightful comments. We are also grateful to Jana Kyllerová, Jakub Javora, and Michal Dvořák for creating style exemplars and input video sequences. This research was supported by the Research Center for Informatics, grant No. CZ.02.1.01/0.0/0.0/16_019/0000765, by the Grant Agency of the Czech Technical University in Prague, grant No. SGS19/179/OHK3/3T/13, and by a Discovery Grant provided by NSERC, grant No. RGPIN-06298.

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.

Dennis R. Bukenberger, Katharina Schwarz, and Hendrik P. A. Lensch. 2018. Stereo-Consistent Contours in Object Space. *Computer Graphics Forum* 37, 1 (2018), 301–312.

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

Dónal Egan, Martin Alain, and Aljosa Smolic. 2021. Light Field Style Transfer with Local Angular Consistency. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. 2300–2304.

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), 155.

David Futschik, Michal Kučera, Michal Lukáč, Zhaowen Wang, Eli Shechtman, and Daniel Sýkora. 2021. STALP: Style Transfer with Auxiliary Limited Pairing. *Computer Graphics Forum* 40, 2 (2021), 563–573.

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.

Xinyu Gong, Haozhi Huang, Lin Ma, Fumin Shen, Wei Liu, and Tong Zhang. 2018. Neural Stereoscopic Image Style Transfer. In *Proceedings of European Conference on Computer Vision*. 56–71.

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

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

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

Lesley Istead and Craig S. Kaplan. 2018. Stylized Stereoscopic 3D Line Drawings from 3D Images. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 20.

Lesley Istead, Andreea Pocol, Craig S. Kaplan, Isaac Watt, Nick Lemoing, and Alicia Yang. 2021. Generating Rough Stereoscopic 3D Line Drawings from 3D Images. In *Proceedings of Graphics Interface*. 178–185.

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.

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

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

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

Nicholas I. Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2019. Style Transfer by Relaxed Optimal Transport and Self-Similarity. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 10051–10060.

Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast Optical Flow Using Dense Inverse Search. In *Proceedings of European Conference on Computer Vision*. 471–488.

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.

Sheng-Jie Luo, Ying-Tse Sun, I-Chao Shen, Bing-Yu Chen, and Yung-Yu Chuang. 2015. Geometrically Consistent Stereoscopic Image Editing Using Patch-Based Synthesis. *IEEE Transactions on Visualization and Computer Graphics* 21, 1 (2015), 56–67.

S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. 2021. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 9685–9694.

Lesley Northam, Paul Asente, and Craig S. Kaplan. 2012. Consistent Stylization and Painterly Rendering of Stereoscopic 3D Images. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering*. 47–56.

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

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.

Efstathios Stavrakis and Margrit Gelautz. 2004. Image-Based Stereoscopic Painterly Rendering. In *Proceedings of the Eurographics Conference on Rendering Techniques*. 53–60.

Daniel Sýkora, Ondřej Jamriška, Ondřej Texler, Jakub Fišer, Michal Lukáč, Jingwan Lu, and Eli Shechtman. 2019. StyleBlit: Fast Example-Based Stylization with Local Guidance. *Computer Graphics Forum* 38, 2 (2019), 83–91.

Krzysztof Templin, Piotr Didyk, Karol Myszkowski, and Hans-Peter Seidel. 2014. Perceptually-motivated Stereoscopic Film Grain. *Computer Graphics Forum* 33, 7 (2014), 349–358.

Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. 2020. Interactive Video Stylization Using Few-Shot Patch-Based Training. *ACM Transactions on Graphics* 39, 4 (2020), 73.

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

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.