Meet-In-Style: Text-driven Real-time Video Stylization using Diffusion Models

David Kunz, *CTU in Prague, FEE, Prague, 121 35, Czech Republic*
Ondřej Texler, *Drip.Art, San Francisco, California, 94117, USA*
David Mould, *Carleton University, Ottawa, Ontario, K1S 5B6, Canada*
Daniel Sýkora, *CTU in Prague, FEE, Prague, 121 35, Czech Republic*

Abstract—
We present Meet-In-Style—a new approach to real-time stylization of live video streams using text prompts. In contrast to previous text-based techniques, our system is able to stylize input video at 30 fps on commodity graphics hardware while preserving structural consistency of the stylized sequence and minimizing temporal flicker. A key idea of our approach is to combine diffusion-based image stylization with a few-shot patch-based training strategy that can produce a custom image-to-image stylization network with real-time inference capabilities. Such a combination not only allows for fast stylization, but also greatly improves consistency of individual stylized frames compared to a scenario where diffusion is applied to each video frame separately. We conducted a number of user experiments in which we found our approach to be particularly useful in video conference scenarios enabling participants to interactively apply different visual styles to themselves (or to each other) to enhance the overall chatting experience.

Video stylization, a captivating intersection of art and technology, has been an active topic of research for the last decade. Representing efforts to convey a stylized look similar to handcrafted animations [1], [2], [3], it has been driven by researchers' curiosity and has gained the interest of the artistic community by offering various interactive workflows [4], [5], [6]. Recent efforts have been fuelled by significant attention from the general public caused by the rise of generative approaches [7], [8], [9].

Video stylization has seen considerable technical improvement. Beginning with traditional algorithmic example-based solutions [1], [3], machine learning approaches hae become ascendant, transitioning from the use of pre-trained VGG network [2] and custom U-net type image translation techniques [5], [6] towards recent applications of text-driven diffusion [7], [9]. Text-driven techniques enable users of any skill level to produce stylized content without depending on traditional artistic media or having any experience with digital image editing. However, these techniques have high computational expense, preventing real-time or interactive uses. Recently, speed-up techniques such as score distillation sampling [10] or latent consistency models [11] have been used to reduce the number of diffusion steps and thus deliver interactive responses. Nevertheless, even higher frame rates are needed for real-time video processing, and temporal consistency of the stylized sequence remains a challenge.

In this paper we present a text-driven framework that enables the use of diffusion models while delivering a consistently stylized video stream in real time on commodity graphics hardware. The input to our system is a continuous live video stream and a text prompt specifying the desired artistic style. Similarly to Yang et al. [7], we use InstructPix2Pix [12] to apply a style to keyframes taken from the input video stream, then propagate it to subsequent frames, ensuring fast and visually consistent stylization (see Fig. 1). A key difference and advantage of our approach is that instead of the computationally demanding style propagation of Jamriška et al. [3], we employ a fast patch-based

Digital Object Identifier 10.1109/XXX.0000.0000000

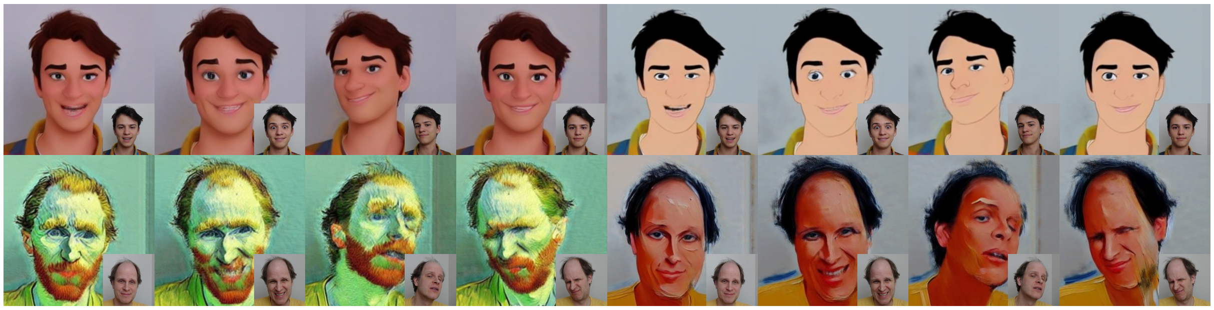

FIGURE 1. Our framework used to stylize live video stream in four different styles: 'make me as a Disney character', 'make him look like a cartoon character', 'as painted by Van Gogh', and 'oil painting, brush strokes'.

training strategy [5]. Both InstructPix2Pix and patch-based style propagation are essential: InstructPix2Pix allows a user to devise new styles on the fly, while patch-based style propagation makes real-time video processing possible.

To the best of our knowledge, our proposed system is the first that enables users to interactively experiment with the visual aesthetic of live video streams using text-driven diffusion models while delivering consistently stylized video streams at frame rates of 30 fps. Such a capability is particularly useful in live video conferencing scenarios (see Fig. 2) that were difficult to achieve using previous published approaches.

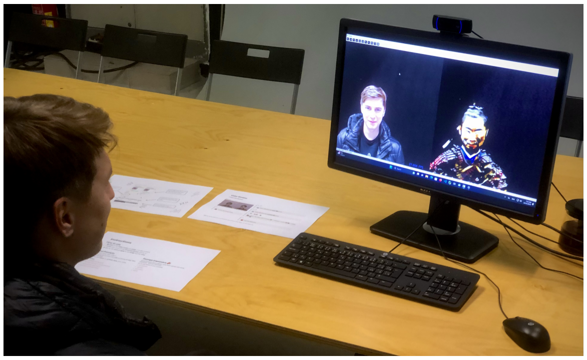

FIGURE 2. Participant using the real-time style transfer application during the Uroboros: Creative AI meet-up.

RELATED WORK

Stylization techniques aim to alter an input image so that it looks as if it had been created using a particular artistic style, while preserving the original content. This is usually achieved by changing colors and texture patterns to resemble a certain artistic medium, and also by applying various distortions or simplifications. During the last decades numerous algorithms were developed to achieve this goal, including procedural techniques, example-based methods, and recently also approaches based on neural networks and diffusion models.

Procedural methods rely on algorithmically manipulating images to mimic artistic effects through various hand-crafted rules and heuristics. Stroke-based approaches [13] produced convincing painterly, pen-and-ink, and hatching styles, among others. Filter-based approaches are a separate direction, also capable of producing a wide variety of styles; the versatile XDoG [14] is an example. Procedural methods can create beautiful images, but their expressive power of individual methods is limited and bespoke methods are needed for particular artistic styles.

Example-based methods try to mimic the style of an exemplar image *S* provided together with the target image *T* that needs to be stylized, i.e., performing style transfer from *S* to *T*. This can be achieved by copying and pasting small patches from the style images onto a different location in the target image to produce a coherent seamless mosaic that resembles content of the target image. Hertzmann et al. pioneered this approach in their Image Analogies framework [15], introducing an example-based approach where a pair of unstyled and stylized images serves as an example of the given artistic transformation and the task is to faithfully apply this transformation on a new unstyled input image. Building upon this foundation, Bénard et al. [1] and later Jamriška et al. [3] adapted this approach to video by incorporating various guidance channels derived from the underlying 3D animation or directly estimated from the input video that enable semantically meaningful and temporally coherent stylization. Despite remarkable quality, this approach remains computationally intensive, which can hinder real-time application scenarios such as video conferences.

Sýkora et al. [16] proposed a real-time example-based stylization algorithm for applications where accurate correspondences between the current video frame and the stylized keyframe can be estimated. This requirement is, however, difficult to fulfill in the context of real-time video stylization.

Neural style transfer pioneered by Gatys et al. [17] also employs the example-based approach. They employ a pre-trained convolutional network to separate and combine the content of the target image with the style in the exemplar image. Subsequent research, such as that of Ruder et al. [2], focused on making the stylization process consistent over time to enable style transfer to video sequences. While the neural approach of Gatys et al. produces impressive results on some inputs, it typically has difficulties with preservation of low-level style details and semantic context.

Image translation techniques can mitigate the drawbacks of neural methods. The image-to-image translation approach of Futschik et al. [18] can perform inference in real time, and the style transfer is consistent and semantically meaningful. It requires a larger set of paired stylized and unstyled images; appropriate pairs can be generated, e.g., using the patch-based approach of Fišer et al. [19]. Nevertheless, the dataset preparation and training times are still highly excessive. Texler et al. [5] address this drawback; their approach requires a smaller training dataset and reduced training time. They train on a set of randomly sampled patches cropped from a few stylized pairs. Nevertheless, their image pairs still require manual preparation.

Diffusion models such as Stable Diffusion [20] introduce a generative approach to neural style transfer; these models are trained to gradually perturb and denoise a content image over tens or hundreds of repeated steps, with the conditioning provided at each step steering the diffusion process toward the desired stylized look. While computationally more expensive than feedforward methods, diffusion models offer several advantages including the ability to capture complex styles and providing more control over the stylization process by adjusting the conditioning at intermediate steps. Diffusion models can also be utilized for image editing [12] and video stylization [21], [22], [23] where each video frame is generated independently, using a diffusion model with additional constraints and guidance to maintain temporal consistency across multiple frames. Hybrid approaches such as that of Yang et al. [7] combine the strengths of diffusion-based keyframe generation with example-based stylization techniques. However, despite their practical potential, these approaches are too computationally intensive for real-time applications. Recently, Parmar et al. [24] made significant progress towards the interactive setting by achieving one-step image translation with a text-to-image model; nevertheless, their performance (10 frames per second on A100 GPU at 512x512 resolution) is still far from real-time and the method lack any mechanism to enforce temporal coherence.

METHOD

Our approach combines the power of diffusion models with a patch-based training strategy to enable real-time stylization of live video streams, conditioned by a user-provided text prompt. Instead of applying the diffusion process on the entire sequence, we apply it only on a few selected keyframes. The stylized keyframes are then used to train a feed-forward image translation network that can be used immediately for real-time inference, generating a consistently stylized video sequence. Diffusion models alone cannot accomplish this task: they are too slow for real-time video processing, and the resulting stylized sequence would suffer from inconsistencies. Our hybrid system has the advantages of diffusion models (ability to create novel styles, arising from InstructPix2Pix) plus interactive video processing, enabling real-time applications (arising from patch-based style propagation).

The input to our framework is a live video stream I (Fig. 3a) and a text prompt P (Fig. 3c). I is typically obtained from a web camera that captures the user's face. While the capture is running, the user poses to take snapshots of a few keyframes I^k: usually one front-facing portrait, optionally accompanied by left and right side view to increase robustness (Fig. 3b). These keyframes are then stylized using InstructPix2Pix [12] (Fig. 3d)—a diffusion model that selectively applies edits to I^k based on P to produce stylized frames S^k. The stylized frames S^k are then used as training exemplars whereby the image translation network of Futschik et al. [18] (Fig. 3f) can learn the mapping between keyframes I^k and their stylized counterparts S^k. To speed up the learning process, a patch-based training strategy [5] is used to obtain a useful model within a couple of seconds, which can then be immediately applied to stylize newly incoming frames of the live video stream I in real time. The stylized video output O is displayed to the user at a rate of 30 frames per second while the quality of stylization improves over time.

Keyframe Stylization

For sequences where the movements happen mostly in the camera plane, a single keyframe is typically

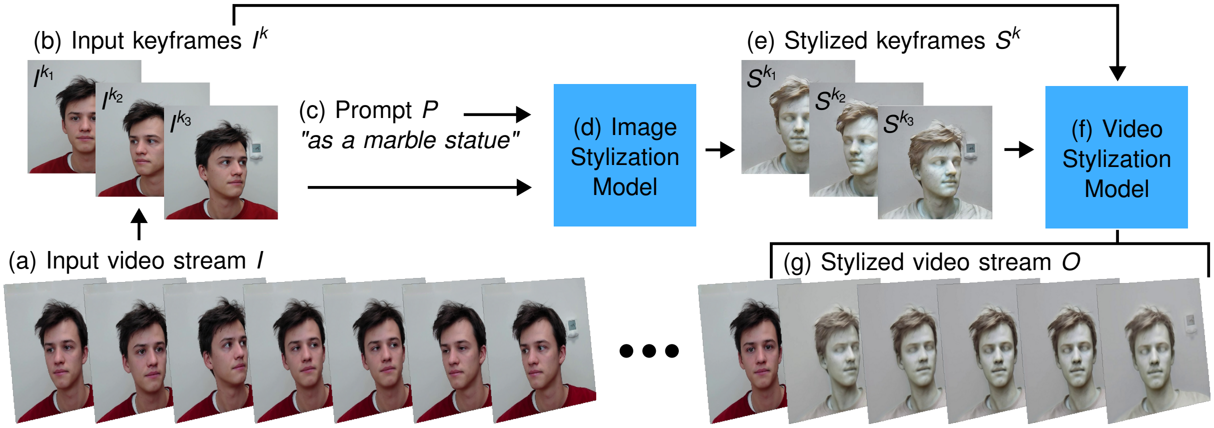
(b) Input keyframes I^k
I^{k_1}
I^{k_2}
I^{k_3}
(c) Prompt P
"as a marble statue"
(d) Image Stylization Model
(e) Stylized keyframes S^k
S^{k_1}
S^{k_2}
S^{k_3}
(f) Video Stylization Model
(a) Input video stream I
(g) Stylized video stream O

FIGURE 3. Stylizing a live video stream using using our framework that combines an image-to-image diffusion based approach InstructPix2Pix (d) with a video stylization technique of Texler et al. [5] (f).

sufficient. For more complex motions such as out-of-plane rotations, three keyframes (front-facing, left-facing, and right-facing) yield better stylization results. Since the keyframes are used to train an image translation network, it is vital for stylization to be consistent across them; e.g., if in S^{k_1} the hair region is painted with a specific color, keyframes S^{k_2} and S^{k_3} should also be painted using the same color. To ensure stylization consistency, we concatenate the keyframes into a single image before passing them to InstructPix2Pix. By applying diffusion process jointly to all keyframes, we obtain a notably more consistent stylization as compared to three independent InstructPix2Pix passes.

Video Stream Stylization

Given the stylized keyframes, we train the image translation network to learn the content-to-style mapping from I^k to S^k using Texler et al.'s patch-based strategy [5]. During training, we update the model every few seconds and in parallel we run an inference thread to convert newly incoming video frames I into stylized output O. The inference model is periodically updated using the weights from the training thread to reflect the new model improvements over time. Our parallel setup minimizes the stylization delay, allowing users to see the text-driven stylization of the incoming video stream interactively after entering a new prompt.

To further accelerate the training process, we apply foreground segmentation [25] and focus only on portions of the input keyframes that lie within the foreground region. Although conventional convolutional networks cannot be trained selectively, a patch-based training strategy [5] enables such an adaptive approach. In practice, the foreground typically accounts for about half of the pixels and thus the training convergence can be roughly two times faster.

Implementation

In order to meet real-time performance requirements and enable interactivity, we employ a client-server architecture and split the workload between a client machine and a server machine that communicate over a local network. This separation enables parallelization: the client handles the graphical user interface, camera capture, video display, and runs the style inference, while the server is responsible for more demanding tasks such as keyframe stylization using InstructPix2Pix and continuous training of the image translation model. During a live session, the user interacts with the application on the client side, providing text prompts and settings. These are then sent to the server, which is responsible for the synthesis of stylized keyframes and model training. The perpetually trained models are transferred back to the client to perform the stylization of the incoming video stream.

RESULTS

Results and Comparison

The results of our method on several subjects and on a variety of different photo-realistic and painterly styles are shown in Fig. 4 (see also our supplementary video). The initial style from the diffusion model is successfully propagated across the ongoing video sequence.

We compared our method with two alternative frameworks that are capable of video style transfer.

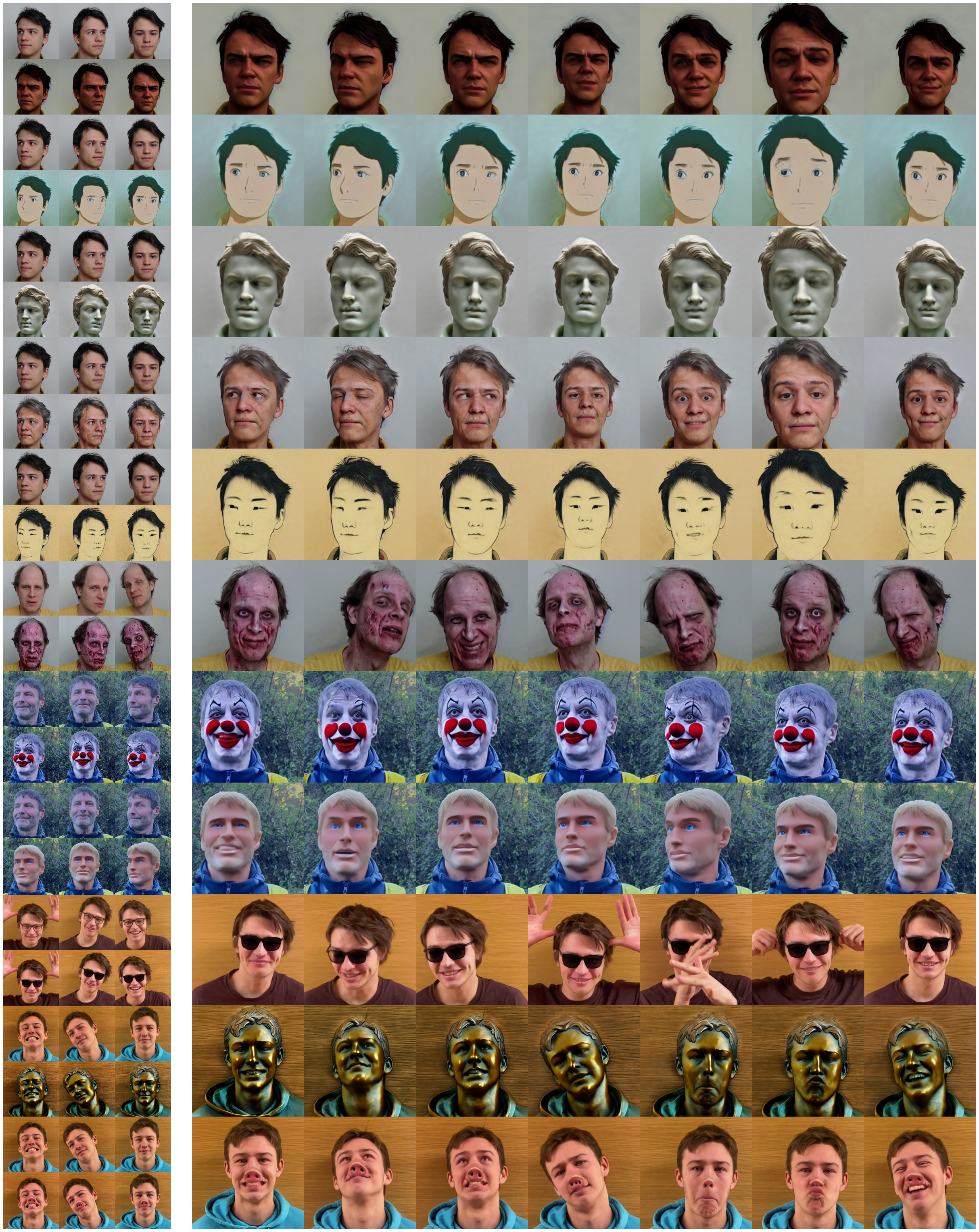

FIGURE 4. Our framework used to stylize a live video stream in several different styles. While it is possible to use only one or two keyframes, for maximal visual quality all results shown (right) were generated using three keyframes (left).

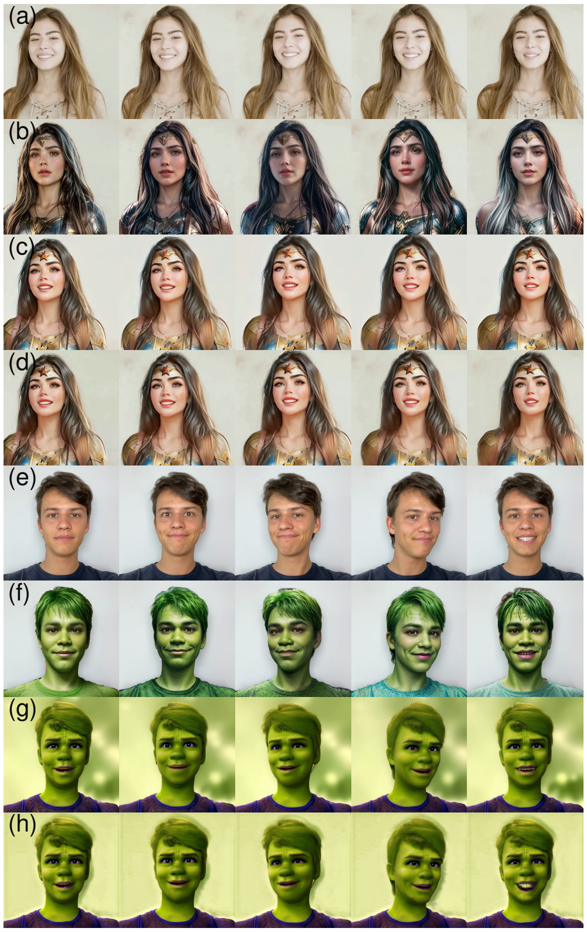

FIGURE 5. Comparison between Rerender A Video [7], fast per-frame generative approach based on distilled LCM [11], and our approach: (a, e) input frames; (b, f) distilled LCM; (c, g) Rerender A Video; (d, h) our approach (style references taken from the foreground of keyframes created by Rerender A Video).

Rerender A Video [7] (Fig. 5c,g), while providing satisfactory temporal consistency, sometimes suffers from error accumulation in longer sequences; moreover, it does not respect the fine geometry and structure (eyes, mouth, and other emotion-conveying facial nuances are sometimes not well preserved). Further, it is computationally intensive; with an average processing time of roughly 6 seconds per frame, it is impossible to use in real-time applications. We also consider a naive per-frame approach (Fig. 5b,f): a distilled LCM [11] model, capable of real-time inference and convincing stylization quality on a per-frame basis. However, even though the input sequence is perfectly consistent and the used prompt, seed, and other parameters do not change, the output sequence exhibits severe temporal inconsistencies. This result is unsurprising, as frames are generated independently and the method has no mechanism for enforcing temporal consistency. Our method (Fig. 5d,h) achieves high textural quality on individual frames, owing to the use of a diffusion model (InstructPix2Pix). Further, it has high temporal consistency and is capable of running in real time on a commonly available GPU, due to the patch-based online training. Note that our results, Fig. 5d and Fig. 5h, were generated using the foreground of 3 and 5 keyframes taken from style references Fig. 5c and Fig. 5g respectively. See also our supplementary video for comparison in motion.

Performance Analysis

To evaluate the interactive capabilities of our system, we analyzed the three metrics with greatest impact on overall user experience: frame rate, latency, and startup time.

Frame rate. In our implementation, the camera captures frames at 30 frames per second at the camera's native resolution of 800x600px. For simplicity and due to the GPU memory constraints, we resize the frames so that the shorter side is 448px, and then we crop the longer sides to obtain a square image of 448x448px. On this resolution, the inference requires approximately 25 milliseconds per frame on a commonly available GPU with performance comparable to Nvidia RTX 2080. Our processing speed is sufficient to achieve 40 frames per second but we limit the speed to the webcam frame rate.

Latency, the delay between the frame being captured and its stylized form being displayed to the user. Typical webcam lag is about 100–150 ms; image cropping, copying into GPU memory, and doing the inference add roughly an additional 30 ms. In our 30 fps setting the stylized video output is about one frame behind the input. The total lag then stays under 200 ms, which we observed to be an imperceptible delay during live streaming.

Startup time is defined as the waiting time between entering a new text prompt and seeing an initial version of the stylized output. Several steps in our pipeline contribute to this delay: (1) InstructPix2Pix requires approximately 9.2 seconds at 30 diffusion steps to generate the keyframes; (2) the stylized keyframes are subsequently passed to the training loop, which runs on a different GPU, resulting in an additional delay of roughly 130 ms; (3) the training loop releases a new model every 100 batches, roughly 2.8 seconds; (4) the model is then loaded on the GPU used by the client, taking approximately 270 ms. Following these four steps, the inference can begin and the user will start seeing stylized frames. In total, the delay between typing a new prompt and observing initial stylization results is roughly 12.4 seconds (for three keyframes). For a single keyframe, the total delay drops to 5.2 seconds. Note that the style transfer quality continues to improve as the training loop keeps learning and improving, in parallel with ongoing stylization of incoming video.

Stylization Fidelity Analysis

To assess the fidelity of the transferred style, we examine two key aspects. First, the *stylization coverage*, i.e., the ability to stylize all video frames in a similar quality, which depends mainly on the number of keyframes, the complexity of the desired style, and the complexity of the input sequence. Second, the stylization *convergence rate*, i.e., the rate at which the stylization improves over time. Convergence rate is influenced by style complexity and the number of pixels processed during training; we optionally apply a foreground mask to restrict the processed region and accelerate training.

Stylization coverage is largely determined by two factors: (1) the complexity of a given style; (2) how well the selected keyframes represent structural changes in the entire video sequence. Style complexity is mainly influenced by the quantity of high-frequency detail and the extent of geometric changes imposed on the original structure (see Fig. 4); e.g., caricatures may incorporate shape changes. Styles where only colors are changed or low-level structure is slightly altered (e.g., smoothing the skin, or changing hair or skin color) are usually easier to train. Styles that change the high-level structure (e.g., making a face more fat, skinny, or adding low-level structural details that were not present in the original keyframe) are usually more difficult to train. For simple styles, typically a single front-facing keyframe is sufficient to capture the desired style transformation. For more complex styles, additional keyframes can help to improve the stylization coverage, as demonstrated in Fig. 6; stylized frames coming from a model trained using only a single keyframe may exhibit artifacts, and a notable improvement can be obtained using a model trained on three different keyframes. However, note that the stylized keyframes should themselves be consistent; for example, minor differences in the injuries in the "zombie" example lead to inconsistent textures in the output. Matching elements in the keyframes reinforce these elements in the rendered frames.

Convergence rate is fast for simple styles; the stylization result is easily recognizable and convincing even with the first loaded model, roughly one or two seconds of training. Complex styles can take several seconds to become recognizable and full convergence takes even longer. Fig. 7 shows convergence profiles for various styles. The first row ("golden statue") shows a simple style that mainly changes the colors of the face. The second row ("James Bond") presents a harder style, changing skin tone and adding high-frequency details like wrinkles; while the skin tone converges quickly (under 10 s), the wrinkles and facial reflections need more time to fully converge (30 s). The third row presents a difficult painterly style ("van Gogh") that needs more than 40 seconds to properly converge.

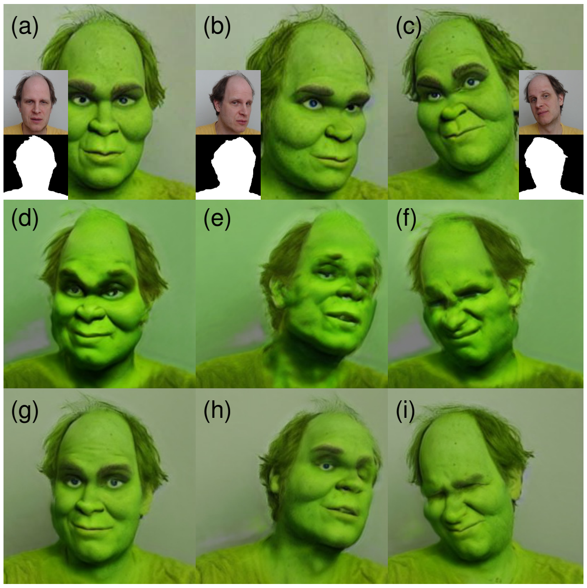

FIGURE 6. Comparison of training on single and multiple keyframes. The top row shows the stylized keyframes along with the original frames and foreground mask (inset). Results in the middle row were obtained using only single keyframe (a) and demonstrate how the resulting stylization may contain artifacts when the subject undergoes substantial head movement. The results in the bottom row were obtained using all three keyframes (a–c); in this case even significant head rotations are stylized convincingly.

Foreground masking. Convergence rate can be greatly improved by applying a foreground mask during the training phase, enabling the network to focus more on the important facial and torso details while paying less attention to pixels in the background. The impact of foreground masking on convergence speed is illustrated in Fig. 7: the third row presents the result where the foreground mask was applied, while the result on

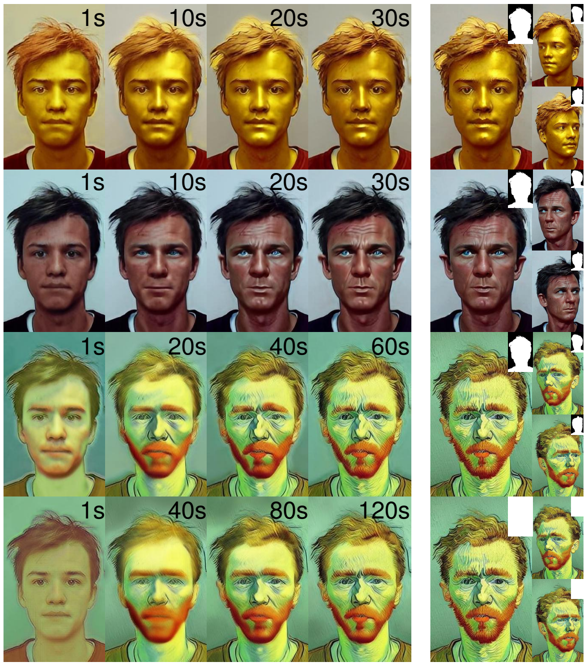

FIGURE 7. Convergence rate for various styles using foreground mask. Top row: a simple style that requires 10 s to give convincing results, and fully converges after 30 s. Second row: a harder style that takes 30 s to converge. Third row: a difficult painterly style that requires 40 seconds to properly converge. Fourth row: the same style as the third row; however, here the image translation model is trained on both foreground and background pixels and thus takes longer to converge. (Note: the images were cropped for demonstration purposes.)

the fourth row was trained on the entire frame, resulting in slower convergence. The Van Gogh style used in the third and fourth row represents a particularly challenging setting for patch-based training, since in the stylized keyframe there are numerous distinct brush strokes in the background, whereas the corresponding area is almost homogeneous in the target video sequence, i.e., highly ambiguous.

User Experience

To further evaluate our interactive prompt-based stylization framework we showcased it at various public events where we observed reactions of users, and gathered informal feedback (see Fig. 2). Approximately 40 participants actively tried our system and more than 100 observers saw and commented on the stylized videos. Those were a mixture of experts on generative AI as well as novice users without prior experience with video stylization techniques. Reactions were uniformly positive, with participants appreciating the intuitiveness of our user interface and its ability to deliver consistently stylized output in real time. The majority of experienced users confirmed that interactive video-to-video stylization systems they had tried previously were unable to preserve temporal consistency, and they felt this aspect was the most prominent novelty of our approach. Some users enjoyed pushing the limits of the system's capabilities by trying exaggerated poses and head movements.

The most common objection reported by the participants was the relatively slow startup phase mainly caused by the processing speed of the InstructPix2Pix diffusion model. We also noted that some users were sometimes not fully satisfied with the ability of InstructPix2Pix to create adequate stylization according to detailed instructions given in the text prompt. They would have appreciated a selection of existing preset styles as well as additional suggestions for preparing text prompts that produce visually interesting results.

Insights gained from the user study helped us validate the proposed interactive framework and identify areas for further development. One interesting improvement suggested by the users during the discussions was the idea to use voice input instead of text prompts. They also mentioned the possibility of using a large language model to analyze transcripts of ongoing conversations and then distill text prompts so that the stylization can be adjusted automatically to align with the topic being discussed.

LIMITATIONS AND FUTURE WORK

Our framework can deliver high-quality stylizations in an interactive prompt-based setting, something difficult to achieve using previous approaches. However, some limitations motivate potential future work. We discuss some of them in this section.

Keyframe quality and consistency. The underlying diffusion model has limited ability to produce consistent stylization across multiple keyframes. Inconsistencies across keyframes may have visible impact on the final stylization quality.

Since the original InstructPix2Pix [12] approach does not take consistency into account, we explicitly concatenate all keyframes into a single image to process them all in a single inference run. However, this strategy may not always guarantee precise consistency. Consider, e.g., the output shown in Fig. 8 (top left quadrant), where the carnival mask is placed slightly differently relative to the face in every stylized keyframe. Due to this ambiguity, the image translation network [5] may fail to capture the fine structure and geometry of the mask (see Fig. 8 top right quadrant for an example). The bottom part of Fig. 8 demonstrates a similar issue: in each stylized keyframe the number of horns differs, the horns are located in different positions, and they have inconsistent shapes, making it difficult for the image translation network to reproduce them fatefully.

The diffusion model may also occasionally produce a mismatch between the locations of facial features. This is shown in Fig. 8 (top right quadrant) where the shape of the mouth significantly differs between the input and the stylized keyframe. The resulting misalignment is clearly visible in the output sequence Fig. 8 (top right quadrant, rightmost result).

Keyframe coverage. The patch-based training strategy [5] generalizes well to unseen poses and expressions despite the limited training set (one to three keyframes). However, it may fail to provide convincing stylizations for more extreme head motions or facial expressions not captured by the keyframes. This is visible, e.g., in the last row of Fig. 4 where the “pig nose” is almost entirely missing from the frames with exaggerated facial expressions.

If we anticipate a range of movements or expressions that would be difficult to capture using only three keyframes, the solution could be to provide more of them. Nevertheless, by increasing the number of keyframes, the GPU memory and time required to process them increases as well and the the risk of introducing inconsistencies escalates.

Style boundary. Our framework also inherits a particular limitation from the style transfer technique [5], where stylistic elements extending beyond the subject’s silhouette are associated with the background during training and are often omitted. See an example in Fig. 8 (bottom part), where the horns are depicted on the background and since there is no structure in the original input frame that would identify their new location in the subsequent frames, the portion of the horns that extends beyond the receptive field of the used image translation network is missing in the results.

Sensitivity to global color changes. Another limitation inherited from the patch-based training strategy [5] is a sensitivity to global color and illumination changes. Significant global changes may visibly and dramatically decrease the stylization quality. In future work, we plan to employ dense visual descriptors that are invariant to illumination changes. However, doing so would necessitate changes to the network architecture as it was originally designed to accept an image as input, not high-dimensional feature vectors.

Manual keyframe selection. Finally, our implementation relies on manual selection of keyframes, anticipating the range of motions and expressions for the full input video. This task can be challenging for novice users who might struggle to choose adequate keyframes. In future work we plan to obtain a dense feature representation and then select frames with the most distant features as keyframe candidates.

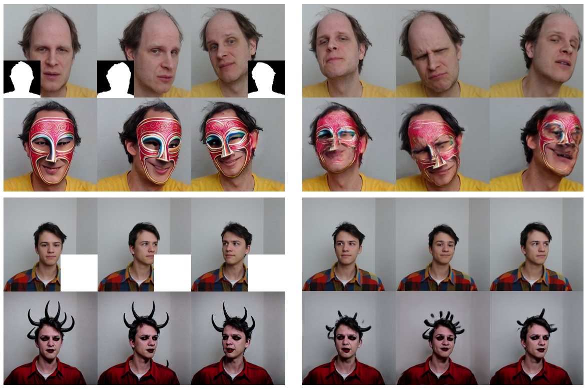

FIGURE 8. Limitations of our framework. Left: input frames with corresponding masks as insets and stylized keyframes; right: selected input frames and corresponding stylized frames. See the main text for discussion.

CONCLUSION

We presented a novel hybrid approach that enables real-time stylization of live video streams via text prompts, while maintaining high visual quality and performance. Our approach combines the strengths of text-driven diffusion-based stylization with a patch-based training strategy. Besides fast performance, we offer an intuitive user interaction with interactive style propagation, tailored to video conferencing use cases. Our method can deliver visually pleasing and diverse stylizations across different subjects, motions, and expressions. Compared to existing video stylization techniques, our framework enables previously unreachable real-time creative self-expression through text prompts. User feedback from public demonstrations confirmed intuitive interaction and a sufficiently expressive range for real-world applications. Limitations were identified regarding keyframe consistency and coverage of exaggerated motions beyond typical video call scenarios. In future work we plan to focus on improving keyframe consistency, automating keyframe selection, and improving robustness to global illumination changes.

ACKNOWLEDGMENT

This research was partially supported by Drip.Art, an NSERC Discovery Grant, the Grant Agency of the Czech Technical University in Prague, grant No. SGS22/173/OHK3/3T/13, and by the Research Center for Informatics, grant No. CZ.02.1.01/0.0/0.0/16 019/0000765.

REFERENCES

1. P. Bénard, F. Cole, M. Kass, I. Mordatch, J. Hegarty, M. S. Senn, K. Fleischer, D. Pesare, and K. Breeden, “Stylizing animation by example,” *ACM Transactions on Graphics*, vol. 32, no. 4, p. 119, 2013.
2. M. Ruder, A. Dosovitskiy, and T. Brox, “Artistic style transfer for videos and spherical images,” *International Journal of Computer Vision*, vol. 126, no. 11, pp. 1199–1219, 2018.
3. O. Jamriška, Šárka Sochorová, O. Texler, M. Lukáč, J. Fišer, J. Lu, E. Shechtman, and D. Sýkora, “Stylizing video by example,” *ACM Transactions on Graphics*, vol. 38, no. 4, p. 107, 2019.
4. C. Lu, Y. Xiao, and C.-K. Tang, “Real-time video stylization using object flows,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 2051–2063, 2018.
5. O. Texler, D. Futschik, M. Kučera, O. Jamriška, S. Sochorová, M. Chai, S. Tulyakov, and D. Sýkora, “Interactive video stylization using few-shot patch-based training,” *ACM Transactions on Graphics*, vol. 39, no. 4, p. 73, 2020.
6. D. Futschik, M. Kučera, M. Lukáč, Z. Wang, E. Shechtman, and D. Sýkora, “STALP: Style transfer with auxiliary limited pairing,” *Computer Graphics Forum*, vol. 40, no. 2, pp. 563–573, 2021.
7. S. Yang, Y. Zhou, Z. Liu, , and C. C. Loy, “Rerender A Video: Zero-shot text-guided video-to-video translation,” in *SIGGRAPH Asia Conference Papers*, 2023, p. 95.
8. J. Z. Wu, Y. Ge, X. Wang, W. Lei, Y. Gu, Y. Shi, W. Hsu, Y. Shan, X. Qie, and M. Z. Shou, “Tune-A-Video: One-shot tuning of image diffusion models for text-to-video generation,” in *Proceedings of IEEE International Conference on Computer Vision*, 2023, pp. 7623–7633.
9. M. Zhao, R. Wang, F. Bao, C. Li, and J. Zhu, “ControlVideo: Adding conditional control for one shot text-to-video editing,” in *Proceedings of International Conference on Learning Representations*, 2024.
10. B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, “Dreamfusion: Text-to-3d using 2d diffusion,” in *Proceedings of International Conference on Learning Representations*, 2023.
11. S. Luo, Y. Tan, L. Huang, J. Li, and H. Zhao, “Latent consistency models: Synthesizing high-resolution images with few-step inference,” 2023, arXiv:2310.04378.
12. T. Brooks, A. Holynski, and A. A. Efros, “InstructPix2Pix: Learning to follow image editing instructions,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.
13. A. Hertzmann, “A survey of stroke-based rendering,” *IEEE Computer Graphics & Applications*, vol. 23, no. 4, pp. 70–81, 2003.
14. H. Winnemöller, J. E. Kyprianidis, and S. C. Olsen, “XDoG: An extended difference-of-Gaussians compendium including advanced image stylization,” *Computers & Graphics*, vol. 36, no. 6, pp. 740–753, 2012.
15. A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, “Image analogies,” in *SIGGRAPH Conference Proceedings*, 2001, pp. 327–340.
16. D. Sýkora, O. Jamriška, O. Texler, J. Fišer, M. Lukáč, J. Lu, and E. Shechtman, “StyleBlit: Fast example-based stylization with local guidance,” *Computer Graphics Forum*, vol. 38, no. 2, pp. 83–91, 2019.
17. L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.
18. D. Futschik, M. Chai, C. Cao, C. Ma, A. Stoliar, S. Korolev, S. Tulyakov, M. Kučera, and D. Sýkora, “Real-time patch-based stylization of portraits using generative adversarial network,” in *Proceedings of the ACM/EG Expressive Symposium*, 2019, pp. 33–42.
19. J. Fišer, O. Jamriška, D. Simons, E. Shechtman, J. Lu, P. Asente, M. Lukáč, and D. Sýkora, “Example-based synthesis of stylized facial animations,” *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.
20. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 674–10 685.
21. P. Esser, J. Chiu, P. Atighehchian, J. Granskog, and A. Germanidis, “Structure and content-guided video synthesis with diffusion models,” in *Proceedings of IEEE International Conference on Computer Vision*, 2023, pp. 7346–7356.
22. D. Ceylan, C.-H. Huang, and N. J. Mitra, “Pix2Video: Video editing using image diffusion,” in *Proceedings*

of IEEE International Conference on Computer Vision, 2023, pp. 23 206–23 217.

23. C. Qi, X. Cun, Y. Zhang, C. Lei, X. Wang, Y. Shan, and Q. Chen, "FateZero: Fusing attentions for zero-shot text-based video editing," in *Proceedings of IEEE International Conference on Computer Vision*, 2023, pp. 15 932–15 942.
24. G. Parmar, T. Park, S. Narasimhan, and J.-Y. Zhu, "One-step image translation with text-to-image models," 2024, arXiv:2403.12036.
25. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for perceiving and processing reality," in *Proceedings of Third Workshop on Computer Vision for AR/VR*, 2019.