Pacific Graphics 2020
E. Eisemann, A. Jacobson, and F.-L Zhang
(Guest Editors)

Volume 39 (2020), Number 7

StyleProp: Real-time Example-based Stylization of 3D Models

F. Hauptfleisch[1,3], O. Texler[2], A. Texler[2], J. Křivánek[1,3], D. Sýkora[2]

[1]Chaos Czech, Czech Republic
[2]Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic
[3]Charles University in Prague, Czech Republic

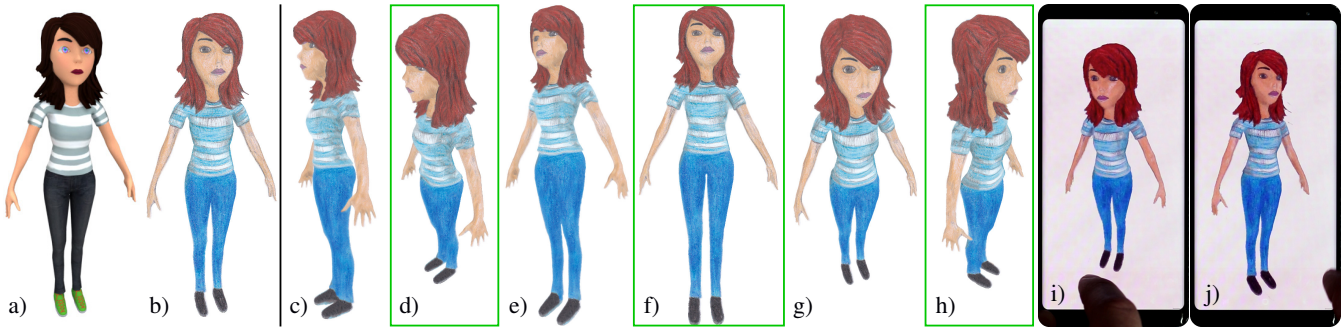

Figure 1: *StyleProp in action: a hand-drawn style is transferred to a given 3D model (a) from a single exemplar created using color pencils (b). A novel variant of guided patch-based synthesis is used to pre-calculate a sparse set of samples (d, f, h) from which the model can be rendered in real-time at arbitrary location within available interaction space (c, e, g) even on a mobile phone (i, j) while maintaining consistency when the viewing direction is changed. Style exemplar (b) courtesy of © Štěpánka Sýkorová.*

Abstract

We present a novel approach to the real-time non-photorealistic rendering of 3D models in which a single hand-drawn exemplar specifies its appearance. We employ guided patch-based synthesis to achieve high visual quality as well as temporal coherence. However, unlike previous techniques that maintain consistency in one dimension (temporal domain), in our approach, multiple dimensions are taken into account to cover all degrees of freedom given by the available space of interactions (e.g., camera rotations). To enable interactive experience, we precalculate a sparse latent representation of the entire interaction space, which allows rendering of a stylized image in real-time, even on a mobile device. To the best of our knowledge, the proposed system is the first that enables interactive example-based stylization of 3D models with full temporal coherence in predefined interaction space.

CCS Concepts

• ***Computing methodologies*** → ***Non-photorealistic rendering;***

1. Introduction

With the rapid evolution of physically-based rendering and the ability to reproduce natural materials' appearance, artists nowadays produce breathtaking animated movies and video games that are quickly reaching a state of absolute visual perfection. Although the audience highly appreciates this convergence, artists start to feel that with the prevalence of realism, the visuals they continue producing become less and less unique. It is usually challenging for an uninformed observer to recognize an animated movie's authorship or a video game by its visuals. Due to this reason, artists start to seek techniques that can automatize repetitive tasks while still being able to retain their unique style. The ability to draw by hand either physically or digitally and reproduce the look of traditional artistic media has recently become increasingly attractive (see, e.g., games such as *Cuphead*, *Memories Retold*, *Dreams*, *Machinarium*, or *Dordogne* and recently released animated features *Spider-Man: Into the Spider-Verse* and *Loving Vincent*, Disney shorts *Just A Thought* and *Jing Hua* or Riot Games' *Annie*).

Besides the production of animated movies and video games, a similar trend also emerges in other fields where visual uniqueness plays an important role. For instance, in the architecture design, when a studio participates in a competition to realize a developer project, photo-realistic visualizations are usually considered a disadvantage. They prevent the committee from recognizing the unique style of a particular studio, which indirectly serves as a quality certificate. Nevertheless, creating such a distinctive presentation is a tedious task; thus, there is a high demand for tools that could help automatize the creative process while still retaining the original aesthetic quality.

An ideal tool that would help artists to simplify the creative process in the sense mentioned above would take a stylized example (e.g., an initial view on a 3D model), distill its distinct visual properties, and transfer them on the target content (e.g., the same model in different viewpoint or pose) so that the resulting stylized counterpart reproduces the look and the feel of the original artwork.

Such a setting is in line with the current research efforts on automatic style transfer that became popular thanks to significant advances made by neural techniques [GEB16, KSS19, KSLO19]. Despite the impressive results those approaches can produce, their fundamental limitation is that they are trying to reproduce only the given artistic style's statistical properties. There is no guarantee that a specific local stylization choice made by an artist (e.g., a carefully crafted stroke depicting an eye region) will retain in the stylized counterpart.

A concurrent approach to neural style transfer uses guided patch-based synthesis [HJO*01, BCK*13, FJL*16, FJS*17, JvST*19], which focuses more on local textural details and semantic meaningfulness of the transferred style instead of global statistics. By taking into account those essential properties, the results produced by those techniques are sometimes difficult to distinguish from the original artwork. However, their drawback is a significant computational overhead that hinders their applicability in interactive applications. Although real-time approximative solutions exist [FCC*19, SJT*19], those impose various restrictions on the content being stylized (e.g., faces only) and the type of guidance that can be used (e.g., sufficient spatial variation).

In this paper, we introduce a novel solution to guided patch-based synthesis that enables real-time response with temporal coherence while being agnostic to the stylized content and guidance. We sparsely sample the space of possible interaction states (e.g., camera rotations) and compute each state's stylization coherently with nearby samples. Then for each stylized state, we store only its latent representation (the nearest-neighbor field) from which we can quickly reconstruct the intermediate states and render the final stylized image. Moreover, since the sampled set is relatively compact, we can transfer it swiftly via the network and deliver a smooth interactive 3D viewing experience even on a mobile device.

2. Related Work

Early approaches to non-photorealistic rendering [KCWI13] use hand-crafted algorithmic solutions to paint an input image or video in a particular style. Some employ physical simulation [CAS*97, HLFR07, LXJ12] or a hand-crafted shader [BKTS06, BNTS07, BLV*10, MSS*18] to mimic given artistic medium; others compose the result from a library of predefined pen [SWHS97, PHWF01, SZKC06], hatch [BSM*07], or brush strokes [Lit97, HE04, SSGS11, ZZ11]. Although these techniques can deliver convincing results, they work only on their respective domain; they are limited to a single style or a certain artistic tool.

Sloan et al. [SMGG01] tried to address this lack of control over the appearance in their technique called The Lit Sphere (a.k.a. MatCap). They allow the user to prepare a hand-drawn exemplar that depicts a stylized counterpart of an illuminated sphere and use it to stylize the illumination of an arbitrary target 3D model. To do that, they employ environment mapping [BN76]—a particular variant of texture mapping where vertex normals are used for texture lookup instead of UV coordinates. Nevertheless, MatCap cannot be directly applied in our scenario since it assumes the stylization of illumination. Debevec et al. [DTM96] proposed a similar technique that can re-project photographs on 3D models. Although their method is directly applicable in our scenario, it cannot handle more extensive viewpoint changes and it distorts the planar structures in the original style exemplar due to texture re-projection (c.f. Fig. 2).

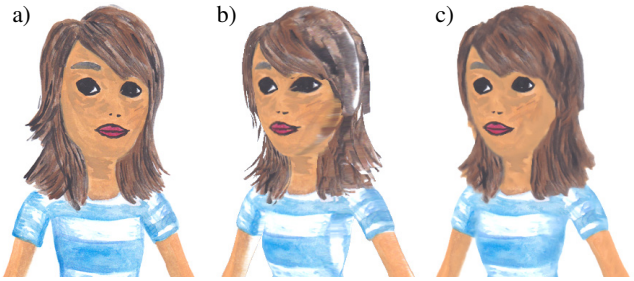

Figure 2: *A simple approach to our problem would be to employ the technique of Debevec et al. [DTM96], i.e., to use the original style exemplar (a) as a texture and re-project it on the new pose (b). While this method enables real-time rendering and can provide sufficiently good results when the camera position does not change considerably, it is prone to disturbing artifacts in our scenario. A key issue here is that texture mapping does not preserve the original style exemplar's planarity, i.e., it deforms strokes to respect the shape of the underlying geometry and thus makes the visual system believe the painting was created on the surface and not in the image plane. It is also apparent that the re-projection cannot correctly handle model parts with a normal almost parallel to the original image plane. Since the re-projection is limited to individual triangles, the resulting image may suffer from a misalignment of sharp geometric details with fluffy structures painted in the original style exemplar. Our approach alleviates all mentioned issues (c). Style exemplar (a) courtesy of © Štěpánka Sýkorová.*

Hertzmann et al. [HJO*01] proposed an image analogies framework to alleviate the mentioned drawbacks. In their technique, they employ patch-based synthesis [WSI07, KNL*15, FJL*16] to preserve the planarity of structures in the original style exemplar while still maintaining meaningful style transfer using additional guiding channels. Others extended this concept to handle style transfer to fluid animations [JFA*15], 3D renders [FJL*16], or facial animations [FJS*17]. However, obtaining high-resolution stylized

images using patch-based synthesis is a computationally expensive task even on the GPU; thus, these methods are hardly accessible when a low computational budget is available, e.g., on a mobile device. Recently, Sýkora et al. [SJT*19] introduced a real-time variant of guided patch-based synthesis that is, however, limited only to a specific type of guidance containing sufficient spatial variation such as surface normal or texture coordinates.

Our setting bears a resemblance to a stylization scenario where the aim is to propagate the appearance of a single stylized keyframe to the remaining animation frames or a video sequence. This approach was pioneered by Bénard et al. [BCK*13], who extended the patch-based method of Hertzmann et al. [HJO*01] by a set of auxiliary guiding channels provided by a 3D renderer and by a new optimization scheme that enables the generation of temporally coherent sequences. Recently, Jamriška et al. [JvST*19] proposed a video stylization framework where necessary guiding channels are extracted automatically from the video. Moreover, Jamriška et al. offer a post-processing step to merge content stylized from different keyframes. However, a fundamental limitation of these techniques is that they are not interactive and can preserve coherency only in one dimension—in time.

A popular example-based approach to style transfer pioneered by Gatys et al. [GEB16] uses the response of the VGG-19 network [SZ14] to measure the similarity of the stylized image and the target content. Based on this measurement, they refine the output stylized image using back-propagation. This approach, however, requires costly optimization. Others used this technique to generate a larger dataset and train a feed-forward network that can reproduce a particular artistic style notably faster [JAFF16, UVL16, ULVL16, WOZW17, UVL17, WRB17]. However, those approaches suffer from two significant drawbacks: (1) they often fail in reproducing fine textural details presented in the original style exemplar, and (2) they do not guarantee that the transfer is semantically meaningful, e.g., that the strokes used to stylize an eye in the original style exemplar are used to stylize an eye region in the target image.

One can solve the problem of appearance transfer by employing generative adversarial networks [GPAM*14]. Those can be trained to perform so-called image-to-image [IZZE17, ZPIE17, ZZP*17] as well as video-to-video [TLYK18, WLZ*18] translation. However, this approach relies on a huge dataset of translation pairs, which is not available in our scenario. Some techniques utilize an encoder-decoder scheme to enable the transfer of an arbitrary style to a content image using a single network trained on unpaired exemplars [HB17, LFY*17, LZY*17]. The encoder, usually a set of convolutional layers of the VGG-19, extracts feature representation from both style and content image. The features are then combined, and a pre-trained decoder turns them back into the image space. Recently, Kotovenko et al. [KSM*19, KSLO19] proposed complex encoder-decoder systems that can deliver impressive results nicely reproducing even lower-level details. Nevertheless, their transfer is still not semantically meaningful as they measure only statistical correlations between the stylized image and the original style exemplar.

Various methods combine aspects of patch-based synthesis and neural-style transfer to achieve semantically meaningful transfer while maintaining neural networks' ability to generalize. To better reproduce local features, Li et al. [LW16] search for neural patches in a style image while following the structure of a content image. Liao et al. [LYY*17] extended this idea into a deep image analogy framework. Instead of image patches, they compute dense correspondences in the feature space of responses given by the VGG-19 network. Although this technique delivers impressive results, it is computationally expensive and does not support coherence when considering animation. Futschik et al. [FCC*19] approximate the patch-based method of Fišer et al. [FJS*17] by training a feed-forward network on a large dataset produced by the mentioned method. Recently, Texler et al. [TFF*20] combined neural style transfer with patch-based synthesis to enable the generation of high-resolution stylized imagery. Although their approach can deliver notably better stylization quality, it still relies on the network's capability to provide meaningful results respecting scene semantics.

When performing style transfer to animation or video, the temporal consistency has to be taken into account. Although a certain amount of temporal flicker is natural for traditional hand-colored animations [FLJ*14], it can be visually demanding when the resulting sequence is observed for a longer period and can cause dizziness we would like to avoid. Various methods, both patch-based [BCK*13, FJS*17, DLKS18, JvST*19, FSDH19] and neural-based [CLY*17, GJAFF17, SKLO18, RDB18], allow for enforcing temporal consistency explicitly by considering relations between individual animation/video frames. Alternatively, one can employ a blind temporal coherency [LHW*18] to stabilize the arbitrary input video sequence. Although all mentioned methods can help to suppress or entirely remove temporal flicker, they consider temporal coherency only in one dimension—in time. In our scenario, we need to solve the problem of temporal consistency in two or more dimensions.

Our approach also resembles image-based rendering that can produce impressive novel views from a sparse set of input photographs [STB*19, MSC*19]. A key difference in our scenario is that we have only a single input image and we aim to preserve planar structures of the original style exemplar, i.e., to retain scale and orientation of individual brush strokes and specific canvas patterns or paper grain. Those features are usually distorted by out-of-plane deformations, which are desirable when generating novel views under perspective projection. These deformations are, however, unwanted in our style transfer scenario.

3. Our Approach

Our method's input is a 3D model M with a texture T that highlights semantically essential details. To prepare a style exemplar, we first produce a render of M: R_i (see Fig. 3) at a specific location $i \in \mathcal{I}$, where $\mathcal{I}$ is an interaction space through which the user can explore the model M (e.g., a set of all possible camera rotations or zooming in/out). We assume R_i contains all important structures that would appear when exploring $\mathcal{I}$. In our current implementation i is chosen manually by the user. However, we envision an automatic estimation of the optimal location as future work. Finally, we print R_i on a paper and provide it to the artist as a stencil to prepare a stylized hand-drawn exemplar S_i (also denoted as S). Optionally, the artist can paint over the stencil digitally using a tablet.

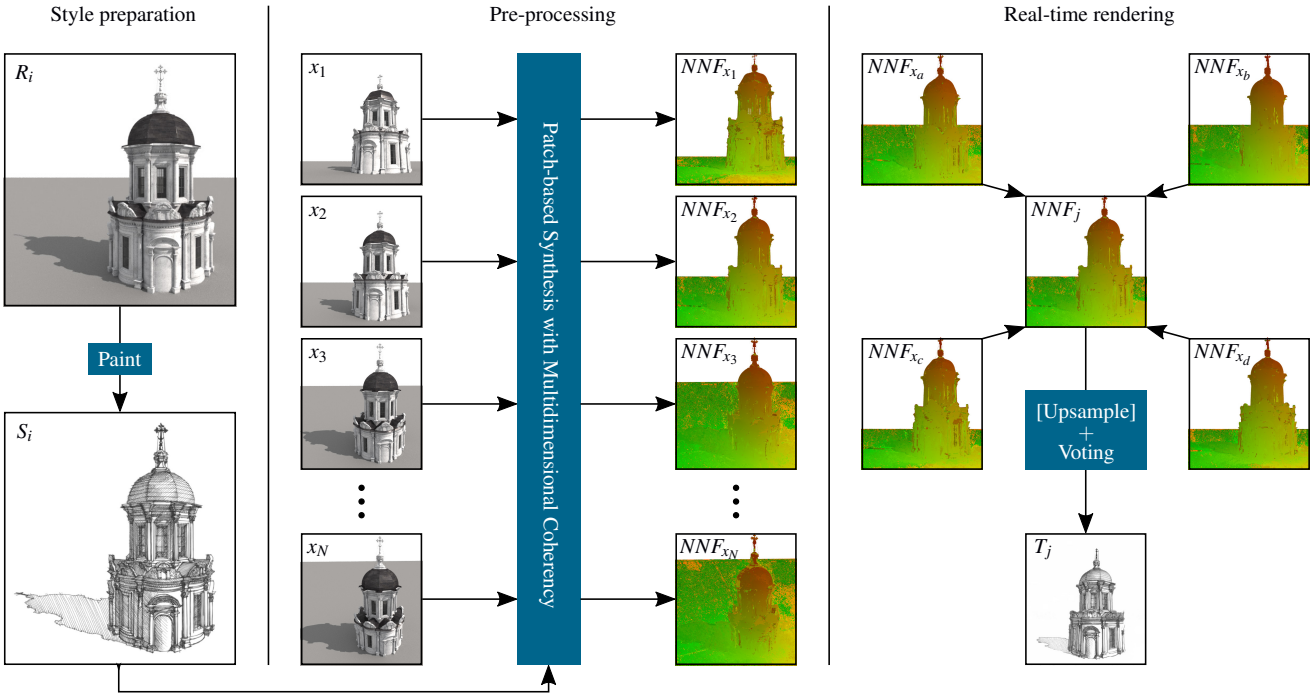

Figure 3: *An overview of our method: First, a model M is rendered in preselected interaction state i to produce a stencil R_i over which an artist paints the style exemplar S_i. Also, a set of source guiding channels G_S is rendered at the state i. Then in the pre-processing phase, available interaction space $\mathcal{I}$ is sampled to a set of states $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathcal{I}$ and for each such state, the full render, as well as other guiding channels in G_T, are computed. Those serve as an input to our patch-based synthesis algorithm that maintains coherence in multiple dimensions, i.e.; it takes into account consistency between nearby interaction states in X. This algorithm's output is a set of nearest neighbor fields (NNF) at each interaction state x. Those provide a latent representation from which the corresponding stylized image T can be reconstructed (not shown in this figure). Finally, in the real-time rendering phase, the user browses to an arbitrary interaction state $j \in \mathcal{I}$, and at that location, pre-computed NNFs of nearby states $x_a, \ldots, x_d$ are combined to produce NNF_j from which the final target image T_j is reconstructed using voting operation. Alternatively, the NNF upsampling technique of Texler et al. [TFF*20] can be used to increase the resolution of the output image. See the text for further details. Style exemplar S_i courtesy of © Jan Pokorný.*

The task for our method is to render T_j (see Fig. 3)—a stylized counterpart of the target model M seen from a different location $j \in \mathcal{I}$. We would like T_j to be still perceived as a painting/drawing on a canvas/paper comparable to S_i, i.e., we need to preserve planar structures typical for the used artistic media such as brush strokes or canvas pattern. In addition, we would like to retain the artist's intention, i.e., stylize a particular feature in T_j in a similar way as it was stylized in the original style exemplar S_i. Finally, our aim is to render T_j in real-time on a mobile device while maintaining temporal consistency during the interactions in the available interaction space $\mathcal{I}$.

We approach this task in two steps. First, we generate N samples of the interaction space $\mathcal{I}$, i.e., $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathcal{I}$. Then for each sample $x \in \mathcal{X}$, we synthesize T_x using a patch-based synthesis algorithm that respects planar structures of the original style exemplar S_i. Moreover, we also ensure that when a user moves from a sample x_k to a nearby sample x_l, the transition will be visually consistent. Finally, we store a latent representation of T_x denoted as NNF_x, and during the interactive exploration when the user browses through $\mathcal{I}$ into a location j, we retrieve NNFs of all nearby samples around j: $\mathcal{N}_j \subset \mathcal{X}$ and use them to quickly reconstruct the stylized image T_j. A key advantage of combining latent representations instead of blending images is that the final stylized image will look comparable to the original patch-based synthesis algorithm's output.

The task described above has a substantial difference compared to previous patch-based synthesis techniques [BCK*13, FJS*17, JvST*19] where the coherence is maintained only in one dimension—in time. In our scenario, we need to achieve consistency in all possible dimensions of $\mathcal{I}$. To do that, we extend the patch-based synthesis algorithm of Fišer et al. [FJL*16] (StyLit) to support multidimensional coherence. We provide a brief overview of the original StyLit algorithm, and then we propose its extension.

3.1. StyLit algorithm overview

In its original form, StyLit algorithm aims to minimize the following error over all patches in the target synthesized image T:

$$\mathcal{E}(S,T,G_S,G_T) = \sum_{q \in Q_T} \min_{p \in Q_S} \left(E_t(S,T,p,q) + E_g(G_S,G_T,p,q)\right). \quad (1)$$

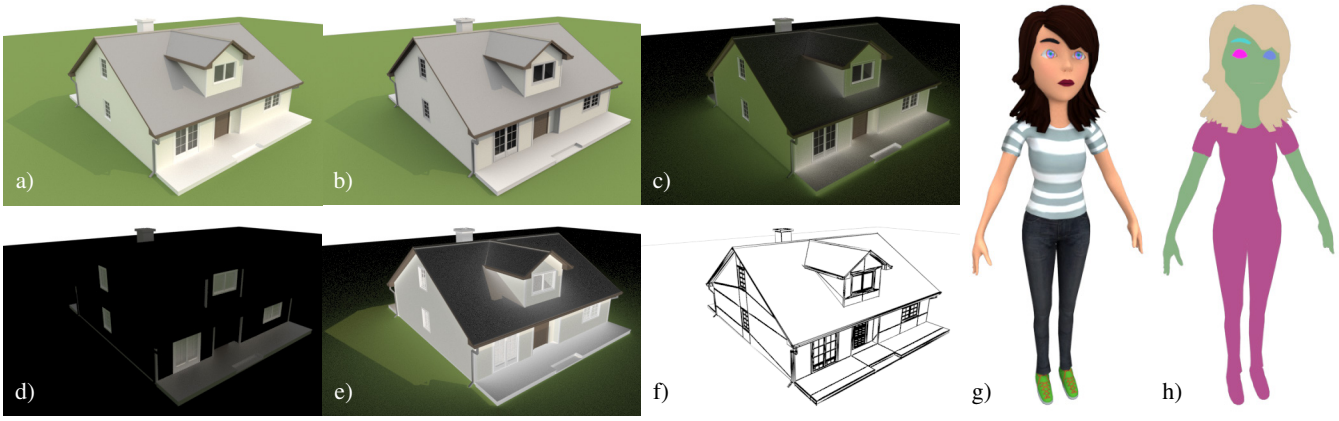

Figure 4: *For buildings we used the following set of guiding channels: full global illumination (a), direct (b), indirect (c), specular (d) components, shadow guide (e), and edge guide (f). For characters, to distinguish between different body parts, we used material ID (h) together with full global illumination (g).*

Here Q_S & Q_T are sets of patches in the source style exemplar S and the target synthesized image T, E_t is the texture coherence error:

$$E_t(S,T,p,q) = \|S(p) - T(q)\|^2 \tag{2}$$

and E_g is the guidance error:

$$E_g(G_S,G_T,p,q) = \|G_S(p) - G_T(q)\|^2 \tag{3}$$

where G_S & G_T are source and target multichannel guides, which are computed using a 3D renderer. Besides a full global illumination channel, its direct/indirect/specular components, and shadow channel (used in the original StyLit algorithm), we added an edge channel (c.f. Fig. 4a–f):

$$G_{\{S,T\}} = \{full, direct, indirect, specular, shadow, edge\}. \tag{4}$$

For 3D characters we use the full global illumination channel and the material ID channel (c.f. Fig. 4g–h):

$$G'_{\{S,T\}} = \{full, id\}. \tag{5}$$

To compute $\mathcal{E}$, the nearest neighbor field (*NNF*) is constructed between the sets of source and target patches Q_S & Q_T. *NNF* is a look-up table in which each target patch $q \in Q_T$ has stored coordinates of its corresponding source patch $p \in Q_S$. The p corresponds to q if it has the lowest sum of style and guide errors E_t & E_g among all patches in Q_S. Also, during the retrieval of p, an allowable error budget is taken into account to prevent some source patches from being assigned too often as the closest ones (please refer to Fišer et al. [FJL*16] for detailed description).

To obtain the final stylized image T, the StyLit algorithm uses an iterative EM-like algorithm initially proposed by Wexler et al. [WSI07]. It alternates two steps: First, in *search step*, *NNF* is constructed between the source patches Q_S and target patches Q_T. Then in *voting step*, an updated version of T is reconstructed using *NNF* by computing a weighted average of all co-located pixels from corresponding source patches.

To maintain temporal coherence in Fišer et al. [FJS*17] and later in Jamriška et al. [JvST*19], the error (1) was extended by an additional temporal coherence term:

$$E_c(S,T',p,q) = \|S(p) - T'(q)\|^2 \tag{6}$$

where T' is a previously synthesized frame that was shifted to match with the position of the current frame T. Therefore, the extended error $\mathcal{E}$ which is minimized looks as follows:

$$\mathcal{E}(S,T,T',G_S,G_T) = \sum_{q\in Q_T} \min_{p\in Q_S} \big(E_t(S,T,p,q) + E_g(G_S,G_T,p,q) + E_c(S,T',p,q)\big). \tag{7}$$

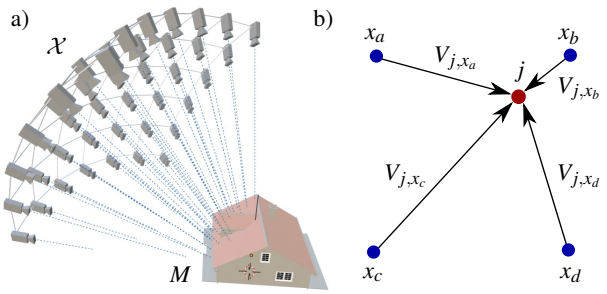

Figure 5: *An illustration of a discrete subset $\mathcal{X}$ of an interaction space $\mathcal{I}$ where the pre-calculation of stylized images T_x of the target model M is performed (a). In order to reconstruct a target image T_j in arbitrary location j within the interaction space $\mathcal{I}$, a latent representation NNF_x of nearby images T_x at locations $\{x_a,\ldots,x_d\}$ are shifted towards j using motion vectors $\{V_{j,x_a},\ldots,V_{j,x_d}\}$ and combined to produce NNF_j from which the target image T_j is subsequently reconstructed (b). See the text for detailed description.*

3.2. Multidimensional coherence

The error $\mathcal{E}$ requires the motion-compensated version of the previous frame T' to perform the evaluation. However, in our sce-

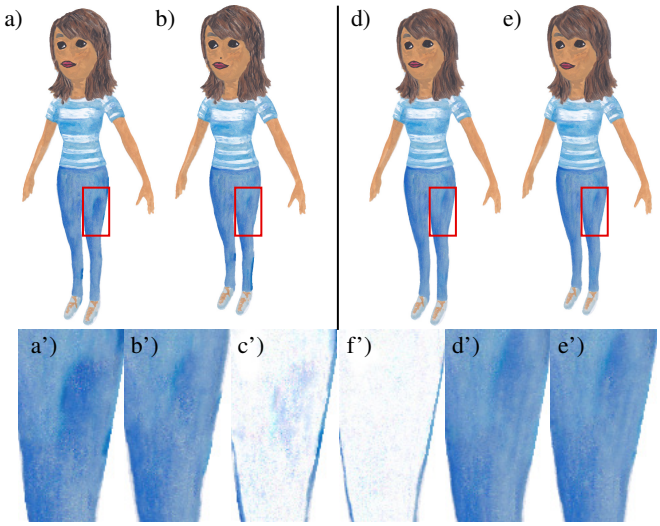

Figure 6: *Demonstration of coherence enforcement: (a, b) two consecutive states stylized without handling the coherence, (d, e) the same states with coherence enforced. The area in a red rectangle is enlarged below; notice the difference in (a', b'), while (d', e') appear identical; (c', f) visualize an inverted subtraction of (a') from (b') and (d') from (e'), respectively. Note that (f') is almost white (almost zero difference). Although these differences might not seem prominent on still images, they can be distracting in motion (c.f. our supplementary video).*

nario, we do not have a sequence of frames, but a multidimensional space of all possible interaction states $\mathcal{I}$ (see Fig. 5a). To address such a multidimensional coherence problem, we sample $\mathcal{I}$ to $\mathcal{X} = \{x_1, \ldots, x_N\} \subset \mathcal{I}$ and start the computation of all T_x in parallel. During each search-vote iteration, we get an intermediate stylized result of T_x and warp it to all neighboring interaction states $\mathcal{N}_x \subset \mathcal{X}$. To do that, we leverage the existence of the underlying 3D model to generate accurate motion fields $V_{x,n}$ that capture movement of individual pixels between nearby interaction states. Such a shifted result T'_n is then used as a new coherence guide, i.e., our goal is to minimize a joint error computed over all sampled interaction states $\mathcal{X}$:

$$\sum_{x \in \mathcal{X}} \sum_{n \in \mathcal{N}_x} \mathcal{E}(S, T_x, T'_n, G_S, G_T). \tag{8}$$

The importance of this coherence enforcement is demonstrated in Fig. 6a–b where two consecutive stylized frames are synthesized without coherence enforcement while in Fig. 6d–e coherence is enforced. In the zoom-in patches, significant changes between Fig. 6a' and Fig. 6b' are visible, but Fig. 6d' and Fig. 6e' appear almost identical. Even though the changes might not seem very prominent on still images, they could be distracting while in movement (c.f. our supplementary video).

3.3. Real-time rendering

The algorithm described in the previous section outputs a coherently stylized model for each sample x of sparsely sampled interaction space $\mathcal{X}$. However, for our target interactive application we need to reconstruct a stylized image T_j at an arbitrary location $j \in \mathcal{I}$. Moreover, we need to retain the visual quality of the original synthesis algorithm, i.e., T_j needs to be a mosaic of larger bitmap chunks taken from the original style exemplar S_i. Therefore, instead of doing some kind of blending operation on the stylized images T_x, we leverage the existence of NNF_x that were used to generate T_x.

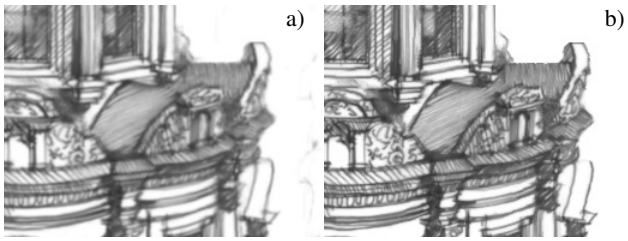

Figure 7: *Improving spatial coherency of a combined NNF_j^*: (a) an image T^* produced from NNF_j^* that was combined from nearby pre-computed states of interaction space $\mathcal{N}_j^*$, note blurriness caused by spatial incoherency of NNF_j^*, (b) a sharper image T produced from a refined NNF_j that has better spatial coherency (see the text for a detailed description).*

We again use the underlying 3D geometry to generate motion vectors $V_{j,x}$ that we use to shift nearby pre-computed NNF_x to a position of the current interaction state $j \in \mathcal{I}$ (see Fig. 5b). To generate the combined NNF_j^* at every target pixel t, we set $NNF_j^*(t) = NNF_{\hat{x}}(t - V_{j,k}(t))$ where $\hat{x}$ is the most suitable sample from the set of nearby states $\mathcal{N}_j$. To select $\hat{x}$, we first exclude states that have different object IDs, i.e., that lie outside the object located at the pixel t. Those will form a set of feasible samples $\mathcal{N}_j^*$. Then we generate a small random displacement vector r that is unique for each target pixel t and does not change during the interaction. Finally, we pick $\hat{x}$ such that:

$$\hat{x} = \arg \min_{x \in \mathcal{N}_j^*} \|x - j + r\| \tag{9}$$

Such a perturbated closest sample selection does not introduce additional flicker and helps to avoid larger abrupt swaps when the sample x changes.

Since the combined NNF_j^* mixes pixel coordinates from different NNFs of nearby interaction samples it may suffer from lower spatial coherency, i.e., contain smaller coherent chunks when compared to the original NNFs. This may lead to blurring artifacts when NNF_j^* is applied directly in the subsequent voting step (see Fig. 7). To improve spatial coherency of the resulting NNF_j, we first apply voting step to obtain an intermediate blurred version of T_j denoted as T_j^*. Then for each patch $q \in Q_{T^*}$ at a pixel t we compute the error $E_c(S, T^*, p, q)$ for every $p \in Q_S$ of with coordinates given by the shifted $NNF_x(t - V_{j,x}(t))$. The coordinates of a patch p with the lowest error E_c are then stored to the refined NNF_j that is used for final voting step to produce T_j. The improvement caused by this refinement can be seen in Fig. 7.

Figure 8: *A complex architectonic model of a chapel stylized using markers (a, c) and watercolor (b, d) style exemplars (left) from various viewpoints using our method. The camera is rotating and zooming in/out (right). Note how important planar structures (such as individual pen/brush strokes or a paper grain) typical for the corresponding artistic media are preserved in each result. For the stylization in motion, please, refer to our supplementary video. Style exemplar (a,c) courtesy of © Jan Pokorný and (b,d) © Štěpánka Sýkorová.*

4. Results

We implemented our patch-based synthesis algorithm in C++ and CUDA. To generate guiding channels, we use GPU implementation of [Kaj86]. The overall computation (guiding channels and synthesis) takes around 10 seconds for one interaction sample with resolution of 1280x720 on Nvidia RTX 2080 GPU. For $\mathcal{I}$ where camera is rotating around the model (see Fig. 5) in a range of 180 degrees for horizontal direction and 50 degrees for vertical direction with sampling rate 10 degrees we get 90 samples of $\mathcal{X}$ that can be generated in less than 15 minutes. The second part of our approach—*NNF* merging and rendering is implemented in Unity framework using HLSL shaders that can fully utilize GPU. Thanks to this integration, the renderer can easily be deployed on desktop machines as well as on a wide range of mobile devices (c.f. Fig. 1i–j).

Resulting *NNF*s are stored as 2D arrays where each entry contains two coordinates (short integers). After the LZMA compres-

Figure 9: *A model of family house painted using color pencils (a). Results of our method (b, c) faithfully represent the original style exemplar and respect the content of an underlying 3D model. The stylized model can be viewed on computer or mobile phone in real-time (see our supplementary video). Style exemplars (a) courtesy of © Barbora Kociánová.*

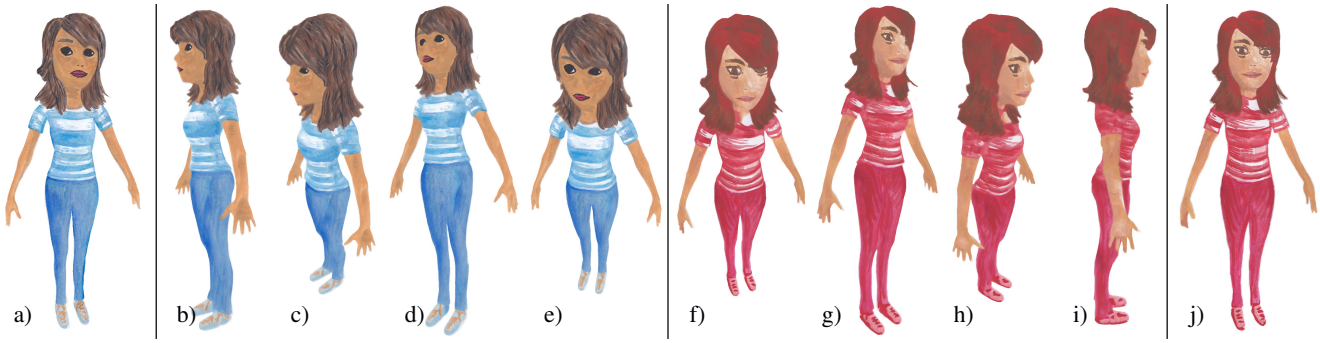

Figure 10: *A model of a girl stylized using two different watercolor styles (a, j). The results (b–e) were produced using style (a) and results (f–i) using style (j). Even in extreme poses, stylized images (b) and (i) retain the content of an underlying 3D model well. The stylized model can be viewed on desktop computer as well as on a mobile phone in real-time, see our supplementary video. Style exemplars (a, j) courtesy of © Štěpánka Sýkorová.*

sion (in Unity), such a lossless latent representation is smaller than the final stylized image stored in PNG format or roughly the same size as a medium–high quality JPEG image of the same resolution. The compressed bundle of 90 samples takes around 19MB of space.

Both the computation time for guiding channels and patch-based synthesis and the memory footprint can further be reduced by using *NNF upscaling* method of Texler et al. [TFF*20]. When *NNF* is upscaled two times, the resulting quality is still acceptable while the computational overhead is reduced to roughly 4 minutes and the size of 90 *NNF* samples is only 5MB. Thus, the entire interaction space for a new style exemplar can be sampled, stylized, transferred, and viewed on a mobile device relatively quickly.

To evaluate our method, we choose two characters and two architectonic models for which we let artists to prepare different style exemplars using watercolor, markers, color pencils, and chalk. We sampled two different interaction spaces: (1) camera moving around the object and (2) camera moving in a horizontal direction and zooming in/out. In Fig. 8 we show results on a architectonic model of chapel stylized in four different artistic styles. Another architectonic model is shown in Fig. 9. In Figures 1, 10, and 11 we present results on two different character models. To demonstrate the potential of our approach to be executed in real-time on a mobile device, in Fig. 1i–j we show our method running on Samsung Galaxy Note 8 at 20 frames per second. For full recordings of real-time interaction sessions please refer to our supplementary video.

An important parameter of our method is the sampling rate of the available interaction space (e.g., angular difference between nearby camera viewpoints). In Fig. 12 we compare results created using four different sampling rates, their memory requirements, and the time of pre-calculation. The visual quality difference between the sampling rate of 2 and 5 degrees is almost negligible. For sampling rate of 10 degrees some blurring artifacts start to show up, however, those are not visible on small screens, e.g., mobile phones, and thus 10 degrees can serve as a good compromise between visual quality, storage space and computational overhead. The results with sampling rate of 20 degrees and more already show considerable artifacts.

In Fig. 13 and in our supplementary video we compared our approach with the current state-of-the-art in patch-based synthesis as well as with some neural-based techniques.

The seminal example-based method of Bénard et al. [BCK*13] (Fig. 13b) as well as the current improvement of Jamriška et al. [JvST*19] (Fig. 13c) were designed for image sequences, therefore, they suffer from discontinuities when brows-

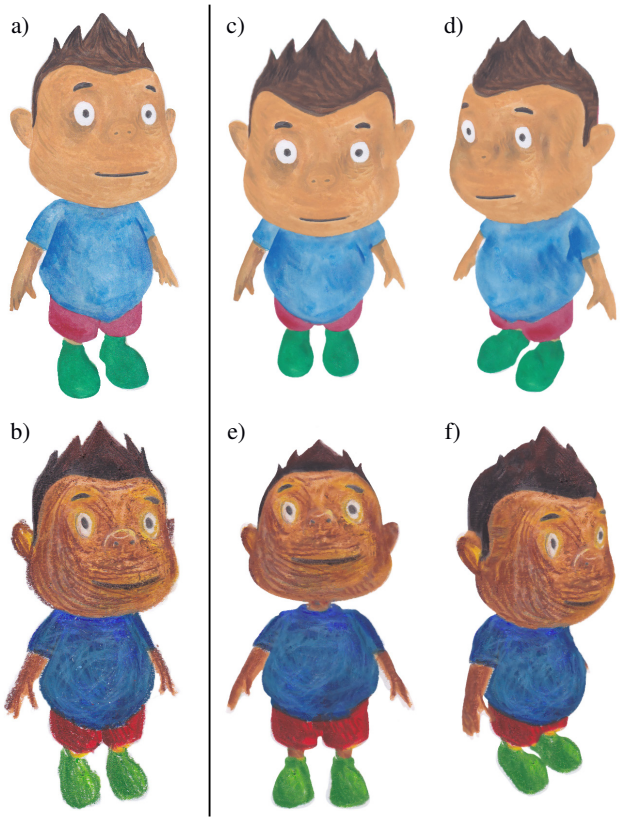

Figure 11: *A model of a boy stylized using watercolor (top row) and chalk (bottom row). Our results (c–f) faithfully mimic the original style exemplars (a, b), preserving the notion of a painting/drawing created by hand on the paper (c.f. our supplementary video for the model in motion). Style exemplars (a, b) courtesy of © Štěpánka Sýkorová.*

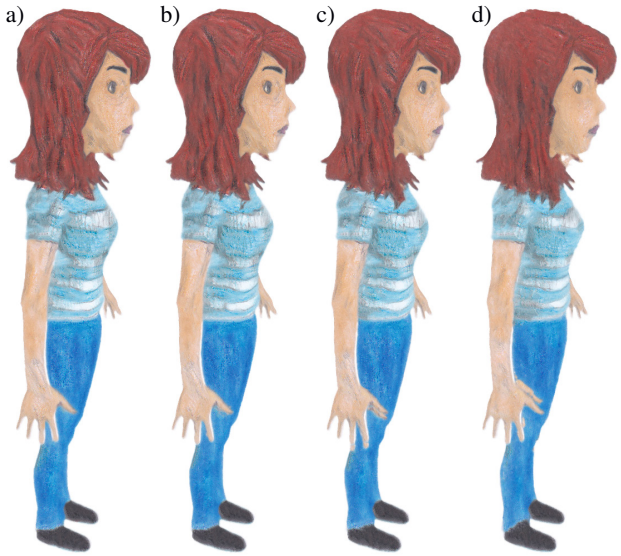

Figure 12: *Comparison of four different sampling rates. From left to right, dense sampling to sparse sampling; (a) sampled every 2 angular degrees, (b) 5 degrees, (c) 10 degrees, and (d) 20 degrees. Sampling rate defines trade-off between quality and performance, i.e., with dense sampling the quality is high, however, time required to run the patch-based synthesis and size of the package might be intractable. Compare the visual quality of (a) and (d) and their respective memory and computational time requirements. We found that sampling rate of 5 or 10 degrees is a good compromise.*

ing in multidimensional interaction space. We used those previous techniques to illustrate this limitation and precalculate a few linear trajectories over the entire interaction space. During the viewing session, we then let the user navigate freely in the interaction space, pick the closest pre-computed path, and replay its frames as long as its direction remains similar to the user's intent. In the case when the user starts navigate differently, we pick another trajectory that is closer to a new path. Due to the one-dimensional coherence, such a hard jump leads to abrupt changes in the appearance, as is visible in our supplementary video. Performance-wise the method of Bénard et al. took several minutes per frame to compute and thus is not applicable in our interactive scenario. The method of Jamriška et al. [JvST*19] running on the GPU is notably faster (few seconds per frame), however, still not fast enough for interactive use.

The method of Debevec et al. [DTM96] (Fig. 13e) and recent approach of Sýkora et al. [SJT*19] (Fig. 13d) can run at interactive rates, however, since they use texture mapping coordinates to perform the re-projection / style transfer, they fail to handle parts of the model which are not properly stylized in the original exemplar S_i.

Although the StyleBlit algorithm can use additional guides (such as object IDs) that are less restrictive and allow for better generalization, one local guide still needs to remain in the set of guiding channels to satisfy the StyleBlit requirements. To perform a meaningful comparison using our guiding channels (Fig. 4g–h), which are not local, we had to add a local guide, i.e., texture mapping coordinates. Due to this reason, the results shown in Fig. 13d suffer from similar artifacts as the method of Debevec et al. [DTM96]. An essential advantage of our approach is that it could potentially work with any guiding channels as the original StyLit algorithm [FJL*16].

Neural-based techniques are slow to compute (tens of seconds per frame) and in general have difficulties to preserve important high-frequency details of the original artistic media as is visible in the output of Li et al. [LFY*17] (Fig. 13f) and Gu et al. [GCLY18] (Fig. 13h). While deep image analogies [LYY*17] (Fig. 13g) performs better with respect to high-frequency details, they cannot properly handle temporal coherence.

5. Limitations and Future Work

Although our approach enables interactive exploration of a stylized 3D model on a mobile device while faithfully reproducing unique

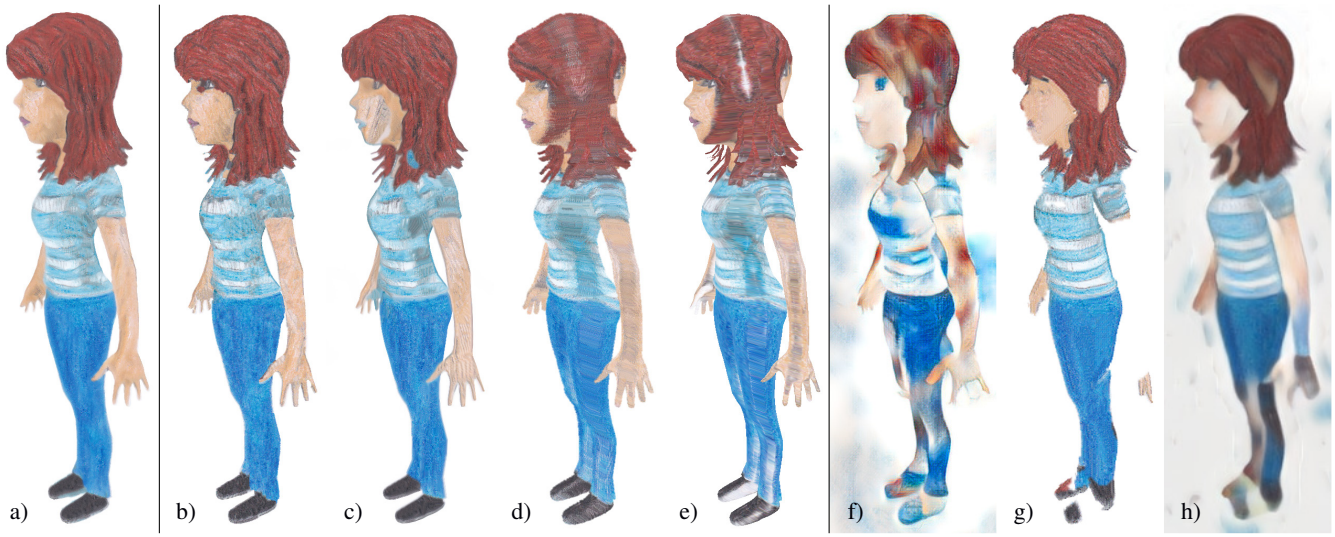

Figure 13: *Comparison of our approach (a) with other example-based stylization methods. Method of Bénard et al. [BCK*13] (b) and Jamriška et al. [JvST*19] (c) show artifacts due to their inability to maintain temporal coherence in multiple dimensions; Sýkora et al. [SJT*19] (d) and Debevec et al. [DTM96] (e) fail to stylize parts of the model which are not well covered by texture in the original style exemplar; Li et al. [LFY*17] (f) fail to reproduce appearance of the style exemplar; Liao et al. [LYY*17] (g) preserve texture properties faithfully, however, do not maintain global consistency; Gu et al. [GCLY18] (h) yield poor texture as well as content quality. Please, refer to our supplementary video to see this comparison in motion.*

visual characteristics of the used artistic media and preserving temporal coherency, there are still some limitations that could motivate future work.

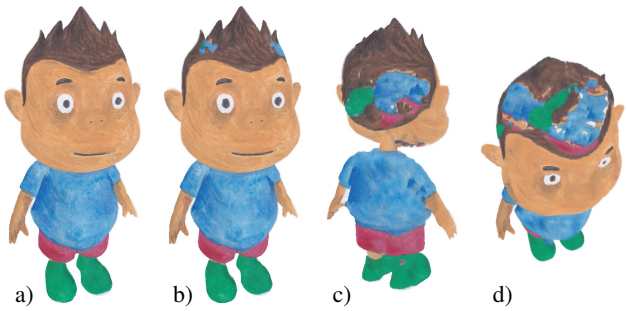

Figure 14: *Comparing results computed using large and smaller interaction space: (a) result generated from a small interaction space where all samples are stylized without visible artifacts, (b) result of the same viewpoint as (a), but taken from a synthesis running on a larger interaction space. Note the artifacts on the hair region propagated from samples far from the style exemplar (c, d) that are not stylized correctly due to significant content difference.*

As our technique uses guided patch-based synthesis [FJL*16] it also shares its drawbacks. The style exemplar needs to be aligned relatively well with the original render, i.e., notable discrepancies (e.g., shape caricature) may lead to a structural mismatch. Tiny details such as nostrils may occasionally disappear due to relatively small spatial support. For those parts, adding a specific guide would be beneficial. Although the original algorithm [FJL*16] handles brush strokes crossing the object boundaries, in our real-time *NNF* combination phase, the object ID masking mechanism may lead to visible discontinuities. Special handling would be necessary to preserve the appearance of structured boundaries.

Despite the fact that our approach explicitly handles multidimensional coherence, it may not always achieve fully coherent results. Since the result of patch-based synthesis is a seamless mosaic of small, translated chunks of the original style exemplar, occasional popping is inevitable. This effect was also apparent in previous patch-based methods (see, e.g., [JvST*19]) where it can bring the notion of hand-colored sequence [FLJ*14] but it may also introduce unwanted distraction. In future work, we plan to control it by combining patch-based and neural techniques.

Our method can suffer from significant artifacts when executed on a larger interaction space where, e.g., the camera viewpoint differs significantly from the one used for creation of style exemplar *S*. The synthesis then fails to find appropriate exemplar patches for the unseen content and it starts to use patches from inappropriate areas. Since the coherence is enforced in all dimensions, the synthesis may propagate those errors across the entire interaction space. Due to this reason artifacts from distant interaction states (w.r.t. *S*) diffuse to nearby states which would otherwise be stylized properly if a smaller interaction space is used. This limitation is illustrated in Fig. 14 (and in our supplementary video) where two samples from the same viewpoint are displayed side-by-side: one is generated from a larger interaction space that goes beyond the limits of our method and the other uses a smaller space.

6. Conclusion

We introduced an interactive approach to the stylization of 3D models that faithfully reproduces a given hand-drawn exemplar while preserving coherence during its exploration. To allow this, (1) we designed a novel variant of a patch-based synthesis algorithm that can produce a sparse set of samples from the available interaction space. Those are produced in a way that all nearby states are stylized coherently. Then, during the real-time rendering phase (2), we demonstrate how to swiftly combine those pre-calculated samples to produce the final stylized image at an arbitrary location. Thanks to this two-stage approach, a real-time 3D model exploration is feasible even on a mobile device. We verified our method on various 3D models and hand-drawn styles and compared them with the current state-of-the-art.

Acknowledgements

This paper is dedicated to Alexis Jaroslav Křivánek and Angeliki Michalopoulou in memory of their great father and husband Jaroslav. We thank the anonymous reviewers for their valuable feedback and comments. We are also grateful to Štěpánka Sýkorová, Barbora Kociánová, and Jan Pokorný, for providing their artistic skills. This research was supported by Chaos Czech, the Grant Agency of the Czech Technical University in Prague, grant No. SGS19/179/OHK3/3T/13 (Research of Modern Computer Graphics Methods), the Research Center for Informatics, grant No. CZ.02.1.01/0.0/0.0/16_019/0000765, and by the Czech Science Foundation under project GA19-07626S.

References

[BCK*13] BÉNARD P., COLE F., KASS M., MORDATCH I., HEGARTY J., SENN M. S., FLEISCHER K., PESARE D., BREEDEN K.: Stylizing animation by example. *ACM Transactions on Graphics 32*, 4 (2013), 119.

[BKTS06] BOUSSEAU A., KAPLAN M., THOLLOT J., SILLION F. X.: Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2006), pp. 141–149.

[BLV*10] BÉNARD P., LAGAE A., VANGORP P., LEFEBVRE S., DRETTAKIS G., THOLLOT J.: A dynamic noise primitive for coherent stylization. *Computer Graphics Forum 29*, 4 (2010), 1497–1506.

[BN76] BLINN J. F., NEWELL M. E.: Texture and reflection in computer generated images. *Communications of the ACM 19*, 10 (1976), 542–547.

[BNTS07] BOUSSEAU A., NEYRET F., THOLLOT J., SALESIN D.: Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics 26*, 3 (2007), 104.

[BSM*07] BRESLAV S., SZERSZEN K., MARKOSIAN L., BARLA P., THOLLOT J.: Dynamic 2D patterns for shading 3D scenes. *ACM Transactions on Graphics 26*, 3 (2007), 20.

[CAS*97] CURTIS C. J., ANDERSON S. E., SEIMS J. E., FLEISCHER K. W., SALESIN D. H.: Computer-generated watercolor. In *SIGGRAPH Conference Proceedings* (1997), pp. 421–430.

[CLY*17] CHEN D., LIAO J., YUAN L., YU N., HUA G.: Coherent online video style transfer. In *Proceedings of IEEE International Conference on Computer Vision* (2017), pp. 1114–1123.

[DLKS18] DVOROŽŇÁK M., LI W., KIM V. G., SÝKORA D.: ToonSynth: Example-based synthesis of hand-colored cartoon animations. *ACM Transactions on Graphics 37*, 4 (2018), 167.

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH Conference Proceedings* (1996), pp. 11–20.

[FCC*19] FUTSCHIK D., CHAI M., CAO C., MA C., STOLIAR A., KOROLEV S., TULYAKOV S., KUČERA M., SÝKORA D.: Real-time patch-based stylization of portraits using generative adversarial network. In *Proceedings of the ACM/EG Expressive Symposium* (2019), pp. 33–42.

[FJL*16] FIŠER J., JAMRIŠKA O., LUKÁČ M., SHECHTMAN E., ASENTE P., LU J., SÝKORA D.: StyLit: Illumination-guided example-based stylization of 3D renderings. *ACM Transactions on Graphics 35*, 4 (2016), 92.

[FJS*17] FIŠER J., JAMRIŠKA O., SIMONS D., SHECHTMAN E., LU J., ASENTE P., LUKÁČ M., SÝKORA D.: Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics 36*, 4 (2017), 155.

[FLJ*14] FIŠER J., LUKÁČ M., JAMRIŠKA O., ČADÍK M., GINGOLD Y., ASENTE P., SÝKORA D.: Color Me Noisy: Example-based rendering of hand-colored animations with temporal noise control. *Computer Graphics Forum 33*, 4 (2014), 1–10.

[FSDH19] FRIGO O., SABATER N., DELON J., HELLIER P.: Video style transfer by consistent adaptive patch sampling. *The Visual Computer 35*, 3 (2019), 429–443.

[GCLY18] GU S., CHEN C., LIAO J., YUAN L.: Arbitrary style transfer with deep feature reshuffle. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8222–8231.

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2414–2423.

[GJAFF17] GUPTA A., JOHNSON J., ALAHI A., FEI-FEI L.: Characterizing and improving stability in neural style transfer. In *Proceedings of IEEE International Conference on Computer Vision* (2017), pp. 4087–4096.

[GPAM*14] GOODFELLOW I. J., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A. C., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems* (2014), pp. 2672–2680.

[HB17] HUANG X., BELONGIE S. J.: Arbitrary style transfer in real-time with adaptive instance normalization. *Proceedings of IEEE International Conference on Computer Vision* (2017), 1510–1519.

[HE04] HAYS J., ESSA I. A.: Image and video based painterly animation. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2004), pp. 113–120.

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *SIGGRAPH Conference Proceedings* (2001), pp. 327–340.

[HLFR07] HAEVRE W. V., LAERHOVEN T. V., FIORE F. D., REETH F. V.: From Dust Till Drawn: A real-time bidirectional pastel simulation. *The Visual Computer 23*, 9–11 (2007), 925–934.

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. pp. 5967–5976.

[JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision* (2016), pp. 694–711.

[JFA*15] JAMRIŠKA O., FIŠER J., ASENTE P., LU J., SHECHTMAN E., SÝKORA D.: LazyFluids: Appearance transfer for fluid animations. *ACM Transactions on Graphics 34*, 4 (2015), 92.

[JvST*19] JAMRIŠKA O., ŠÁRKA SOCHOROVÁ, TEXLER O., LUKÁČ M., FIŠER J., LU J., SHECHTMAN E., SÝKORA D.: Stylizing video by example. *ACM Transactions on Graphics 38*, 4 (2019), 107.

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Computer Graphics 20*, 4 (1986), 143–150.

[KCWI13] KYPRIANIDIS J. E., COLLOMOSSE J., WANG T., ISENBERG T.: State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics 19*, 5 (2013), 866–885.

[KNL*15] KASPAR A., NEUBERT B., LISCHINSKI D., PAULY M., KOPF J.: Self tuning texture optimization. *Computer Graphics Forum 34*, 2 (2015), 349–360.

[KSLO19] KOTOVENKO D., SANAKOYEU A., LANG S., OMMER B.: Content and style disentanglement for artistic style transfer. In *Proceedings of IEEE International Conference on Computer Vision* (2019), pp. 4421–4430.

[KSM*19] KOTOVENKO D., SANAKOYEU A., MA P., LANG S., OMMER B.: A content transformation block for image style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10032–10041.

[KSS19] KOLKIN N. I., SALAVON J., SHAKHNAROVICH G.: Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 10051–10060.

[LFY*17] LI Y., FANG C., YANG J., WANG Z., LU X., YANG M.-H.: Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems* (2017), pp. 385–395.

[LHW*18] LAI W., HUANG J., WANG O., SHECHTMAN E., YUMER E., YANG M.: Learning blind video temporal consistency. In *Proceedings of European Conference on Computer Vision* (2018), pp. 179–195.

[Lit97] LITWINOWICZ P.: Processing images and video for an impressionist effect. In *SIGGRAPH* (1997), pp. 407–414.

[LW16] LI C., WAND M.: Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2479–2486.

[LXJ12] LU C., XU L., JIA J.: Combining sketch and tone for pencil drawing production. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2012), pp. 65–73.

[LYY*17] LIAO J., YAO Y., YUAN L., HUA G., KANG S. B.: Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics 36*, 4 (2017), 120.

[LZY*17] LU M., ZHAO H., YAO A., XU F., CHEN Y., LIN X.: Decoder network over lightweight reconstructed feature for fast semantic style transfer. *Proceedings of IEEE International Conference on Computer Vision* (2017), 2488–2496.

[MSC*19] MILDENHALL B., SRINIVASAN P. P., CAYON R. O., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics 38*, 4 (2019), 29.

[MSS*18] MONTESDEOCA S. E., SEAH H. S., SEMMO A., BÉNARD P., VERGNE R., THOLLOT J., BENVENUTI D.: 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* (2018), p. 11.

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-time hatching. In *SIGGRAPH* (2001), pp. 581–586.

[RDB18] RUDER M., DOSOVITSKIY A., BROX T.: Artistic style transfer for videos and spherical images. *International Journal of Computer Vision 126*, 11 (2018), 1199–1219.

[SJT*19] SÝKORA D., JAMRIŠKA O., TEXLER O., FIŠER J., LUKÁČ M., LU J., SHECHTMAN E.: StyleBlit: Fast example-based stylization with local guidance. *Computer Graphics Forum 38*, 2 (2019), 83–91.

[SKLO18] SANAKOYEU A., KOTOVENKO D., LANG S., OMMER B.: A style-aware content loss for real-time hd style transfer. In *Proceedings of European Conference on Computer Vision* (2018), pp. 715–731.

[SMGG01] SLOAN P.-P. J., MARTIN W., GOOCH A., GOOCH B.: The Lit Sphere: A model for capturing NPR shading from art. In *Proceedings of Graphics Interface* (2001), pp. 143–150.

[SSGS11] SCHMID J., SENN M. S., GROSS M., SUMNER R. W.: Overcoat: an implicit canvas for 3D painting. *ACM Transactions on Graphics 30*, 4 (2011), 28.

[STB*19] SRINIVASAN P. P., TUCKER R., BARRON J. T., RAMAMOORTHI R., NG R., SNAVELY N.: Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 175–184.

[SWHS97] SALISBURY M. P., WONG M. T., HUGHES J. F., SALESIN D. H.: Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH Conference Proceedings* (1997), pp. 401–406.

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014).

[SZKC06] SNAVELY N., ZITNICK C. L., KANG S. B., COHEN M. F.: Stylizing 2.5-D video. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2006), pp. 63–69.

[TFF*20] TEXLER O., FUTSCHIK D., FIŠER J., LUKÁČ M., LU J., SHECHTMAN E., SÝKORA D.: Arbitrary style transfer using neurally-guided patch-based synthesis. *Computers & Graphics 87* (2020), 62–71.

[TLYK18] TULYAKOV S., LIU M.-Y., YANG X., KAUTZ J.: Mocogan: Decomposing motion and content for video generation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1526–1535.

[ULVL16] ULYANOV D., LEBEDEV V., VEDALDI A., LEMPITSKY V. S.: Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML* (2016), vol. 48, pp. 1349–1357.

[UVL16] ULYANOV D., VEDALDI A., LEMPITSKY V. S.: Instance normalization: The missing ingredient for fast stylization. *CoRR abs/1607.08022* (2016).

[UVL17] ULYANOV D., VEDALDI A., LEMPITSKY V. S.: 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* (2017), pp. 4105–4113.

[WLZ*18] WANG T.-C., LIU M.-Y., ZHU J.-Y., TAO A., KAUTZ J., CATANZARO B.: High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8798–8807.

[WOZW17] WANG X., OXHOLM G., ZHANG D., WANG Y.-F.: Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 7178–7186.

[WRB17] WILMOT P., RISSER E., BARNES C.: Stable and controllable neural texture synthesis and style transfer using histogram losses. *CoRR abs/1701.08893* (2017).

[WSI07] WEXLER Y., SHECHTMAN E., IRANI M.: Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 3 (2007), 463–476.

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision* (2017), pp. 2242–2251.

[ZZ11] ZHAO M., ZHU S.-C.: Portrait painting using active templates. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2011), pp. 117–124.

[ZZP*17] ZHU J.-Y., ZHANG R., PATHAK D., DARRELL T., EFROS A. A., WANG O., SHECHTMAN E.: Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems* (2017), pp. 465–476.