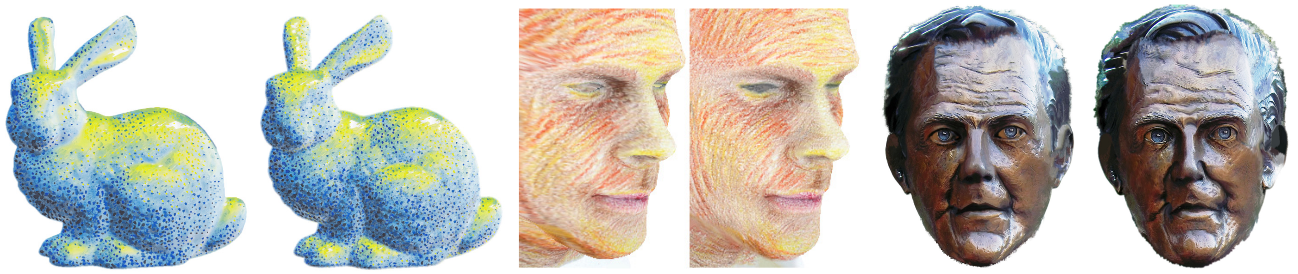
EUROGRAPHICS 2019 / P. Alliez and F. Pellacini
(Guest Editors)

Volume 38 (2019), Number 2

StyleBlit: Fast Example-Based Stylization with Local Guidance

D. Sýkora[1], O. Jamriška[1], O. Texler[1], J. Fišer[2], M. Lukáč[2], J. Lu[2], E. Shechtman[2]

[1]Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic
[2]Adobe Research, USA



(a) StyLit [**56 secs**] (b) our approach [**0.05 sec**] (c) texture mapping (d) our approach (e) FaceStyle [**83 secs**] (f) our approach [**0.1 sec**]

Figure 1: *StyleBlit in applications: (a) style transfer from an exemplar in Fig. 6 to a 3D model using StyLit [FJL* 16]; (b) our approach delivers similar visual quality but is several orders of magnitude faster; (c) regular texture mapping using texture presented in Fig. 10 vs. (d) our approach that better preserves visual characteristics of the used artistic media; (e) style transfer to a portrait image using FaceStyle [FJS* 17] with an exemplar in their supplementary material; (f) our approach produces similar visual quality and is notably faster.*

Abstract

We present StyleBlit—an efficient example-based style transfer algorithm that can deliver high-quality stylized renderings in real-time on a single-core CPU. Our technique is especially suitable for style transfer applications that use local guidance - descriptive guiding channels containing large spatial variations. Local guidance encourages transfer of content from the source exemplar to the target image in a semantically meaningful way. Typical local guidance includes, e.g., normal values, texture coordinates or a displacement field. Contrary to previous style transfer techniques, our approach does not involve any computationally expensive optimization. We demonstrate that when local guidance is used, optimization-based techniques converge to solutions that can be well approximated by simple pixel-level operations. Inspired by this observation, we designed an algorithm that produces results visually similar to, if not better than, the state-of-the-art, and is several orders of magnitude faster. Our approach is suitable for scenarios with low computational budget such as games and mobile applications.

CCS Concepts
• ***Computing methodologies*** $\rightarrow$ ***Non-photorealistic rendering;*** *Image processing;*

1. Introduction

Example-based artistic style transfer recently became popular thanks to advances made by neural-based approaches [GEB16, SED16], patch-based texture synthesis techniques [FJL*16, FJS*17] and their combinations [LW16, LYY*17]. These methods can produce impressive style transfer results with a common limitation of high computational overhead. Although interactive framerate can be achieved when compromising visual quality [JAFF16] or utilizing the GPU [FJL*16], high-quality style transfer remains out of reach for scenarios such as interactive games or mobile applications where the available computational budget is low.

A key concept that distinguishes style transfer from regular texture synthesis [EL99] is the use of guiding channels [HJO*01]. Those encourage the transfer of a specific area in the source exemplar to a corresponding area in the target image. The design of guiding channels is extremely important for achieving semantically meaningful transfer. The guidance can be relatively *fuzzy* with respect to a certain spatial location (e.g., segmentation or blurred

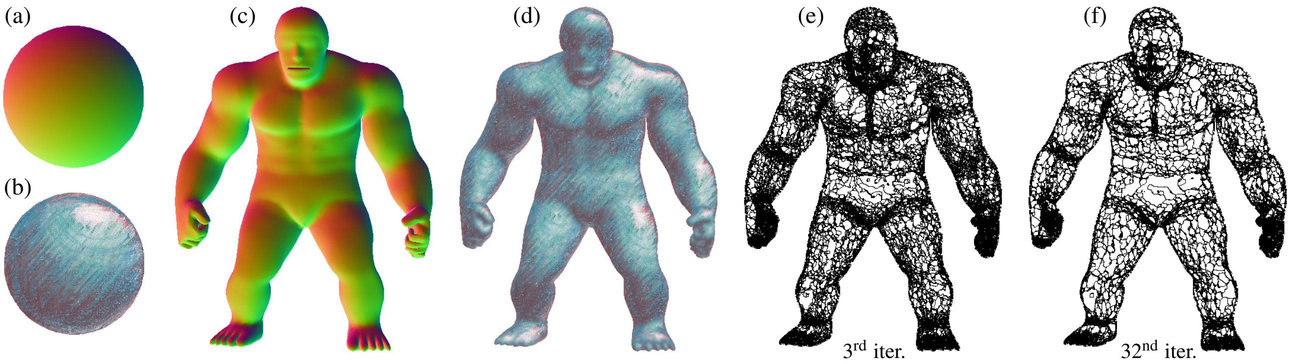


Figure 2: *The motivation for our approach: state-of-the-art guided patch-based synthesis [FJL*16] is used to transfer artistic style from a hand-drawn sphere (b) onto a more complex 3D object (c). Normal maps are used as guidance (a, c). The result (d) preserves well the textural coherence of the original artistic style exemplar since the optimization-based approach converges to a state where large coherent chunks of the source texture (colored white) are copied into the target image forming a mosaic (e). As the optimization progresses, the size of coherent regions increases (f). Style exemplar:* © Pavla Sýkorová

gray-scale gradients used by Hertzmann et al.) or well-localized and descriptive (e.g., a displacement field [SED16, FJS*17], texture coordinates [RRFT14, MNZ*15] or normal values [SMGG01, DBP*15]). We call the latter *local* guidance.

The goal of current state-of-the-art patch-based style-transfer techniques [FJL*16, ZSL*17] is to optimize for a solution that satisfies the prescribed guidance and consists of large coherent chunks of the style exemplar in semantically meaningful regions. This solution represents the most visually-pleasing configuration that maximizes sharpness and fidelity of the synthesized texture since large areas of the exemplar are copied as is (see Fig. 2). To achieve this, however, *textural coherence* [KEBK05, WSI07] needs to be taken into account which results in a computationally demanding energy minimization problem.

In this paper, we demonstrate that when guidance provide good localization and when style exemplar contains stochastic texture, textural coherence becomes less important as the local characteristics of the guide implicitly encourage coherent solutions and the stochastic nature enables visual masking that suppresses visible seams. In this setting, we demonstrate that expensive optimization can be replaced by a set of simple and fast pixel-level operations that gain significant performance speed-up. On a single core modern CPU we can stylize a one-megapixel image at 10 frames per second while on a common GPU we can achieve more than 100 frames per second at a 4K UHD resolution. Despite its simplicity, our new method produces high-quality transfer results for a wide range of styles. Applications include stylization of 3D renderings [FJL*16] (see Fig. 1, left), image-based texture mapping that better preserves the characteristics of natural artistic media [MNZ*15] (Fig. 1, middle), or fast style transfer to faces with comparable results to the method of Fišer et al. [FJS*17] (Fig. 1, right). Our technique can also be used in a more generic MatCap scenario [SMGG01] where instead of using explicit shading models a hand-drawn, captured or synthetically prepared photorealistic material is transferred to a more complex 3D object using normal-based guidance (see Fig. 9). A key advantage of our approach is that compared to the original solution based on environment mapping [SMGG01] our method transfers larger chunks of the source image, which preserves high-frequency features of the texture.

2. Related Work

Over the last two decades, non-photorealistic rendering [KCWI13] evolved considerably. The state-of-the-art techniques can synthesize images resembling real artwork. A popular branch of techniques achieves this goal by mixing a set of predefined strokes or patterns that are selected and positioned according to guiding information provided in 2D [Her98] or 3D [SSGS11] environments. In addition to painterly styles, this line of approaches can also simulate other artistic styles such as pen-and-ink illustration [SWHS97] or hatching [BSM*07]. Nevertheless, these approaches are confined by the limited expressive power of these predefined sets of strokes or patterns.

To alleviate this drawback, an example-based approach called *Image Analogies* was introduced by Hertzmann et al. [HJO*01]. This method allows an artist to prepare an arbitrary stylized version of a target image given an input style example. A one-to-one mapping between the input image and its stylized version is used to guide the transfer by establishing correspondences between the source and target (based, e.g., on color correspondence). The target image can then be stylized according to this analogy. This seminal concept was later extended to animations [BCK*13] and improved by others [BZ17] using better synthesis algorithms [KNL*15, FJL*16] as well as different types of guidance [ZSL*17, FJS*17]. In parallel, an approach similar to *Image Analogies* was introduced by Sloan et al. [SMGG01] and later extended by others [BTM06, TAY13]. Their technique called *The Lit Sphere* (a.k.a. *MatCap*) uses a one-to-one correspondence between normal values to transfer style from a hand-drawn exemplar of a simple object (a sphere) to a more complex 3D model. In this scenario, a simple environment mapping can be used [BN76] to perform the transfer. Recently, Magnenat et al. [MNZ*15] proposed a similar technique where instead of normals, UV coordi-

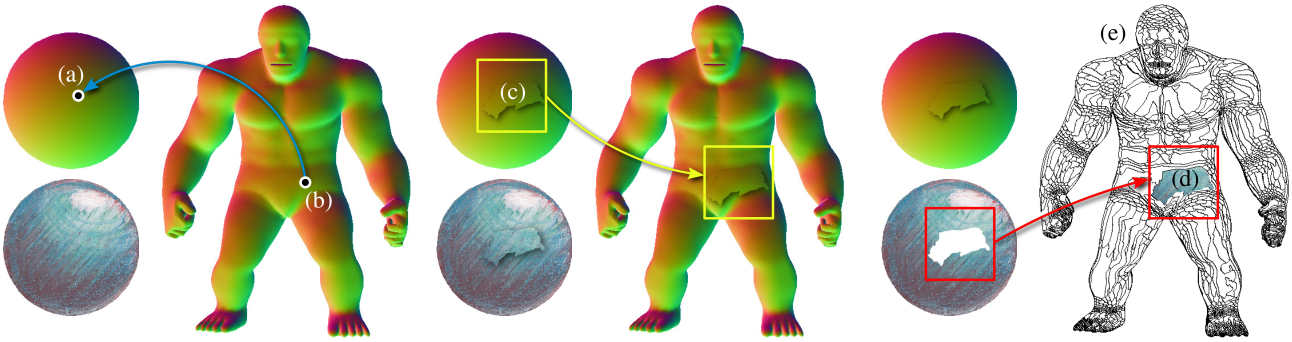


Figure 3: *The core idea behind our method: for each randomly selected seed in the target image (b), we perform a table lookup using its guidance value (in this case a normal) to retrieve the corresponding location in the source exemplar (a). Then we compare the guidance values of source and target pixels in spatially-aligned regions around the seed. Pixels with a guidance value difference below a user-defined threshold belong to the same chunk (c). Finally, we transfer the chunk of example pixels to the target (d). We can produce the final mosaic by repeating this process (e). Style exemplar:* © Pavla Sýkorová

nates are used as guidance so that the artist can draw a stylized version on a 2D projection of a 3D model and then the style is transferred using texture mapping. This approach is similar to image-based texture mapping used in 3D reconstruction [DTM96]. Style transfer can be performed in real-time thanks to its simplicity, but it only works well when the style does not contain distinct high-frequency details. Texture mapping often distorts high-frequency details failing to retain the fidelity of the used artistic medium. Later patch-based synthesis methods [FJL*16, BKR17] have obtained much higher quality results by taking into account not only local guidance but also textural coherence. These improvements, however, came at the cost of notably higher computational overhead.

Recently, Gatys et al. [GEB16] introduced an alternative approach to style transfer based on parametric texture synthesis [PS00] where instead of a steerable pyramid, an alternative parametric representation is used based on a deep neural network trained for object recognition [SZ14]. Their technique inspired a lot of follow-up work [SID17] and became very popular thanks to numerous publicly available implementations. Although it produces impressive results for some style exemplars, it was shown to suffer from certain high-frequency artifacts caused by the parametric nature of the synthesis algorithm [FJL*16, FJS*17]. To prevent texture distortion, researchers have proposed techniques to combine the advantages of patch-based synthesis and the deep features learned by neural network [LW16, LYY*17]. These approaches, however, have significant computational overhead and are not suitable for real-time applications.

Our approach to style transfer bears resemblance to early texture synthesis approaches [PFH00, LLX*01, EF01, KSE*03] that can achieve results similar to patch-based synthesis [KEBK05, WSI07] by transferring larger irregularly-shaped chunks of the source exemplar and composing them seamlessly in the target image. In particular *Lapped Textures* [PFH00] can tile the target surface with a set of source patches, however, there is no specific guidance for the patch placement, the patches need to be prepared in advance to have minimal features on boundaries (to avoid seams), and the approach requires an additional growing operation to fill in gaps. In appearance-space texture synthesis [LH06], small appearance vectors are used instead of color patches to compress neighborhood information, but an iterative optimization [LH05] is still necessary to obtain the final result.

In another related work [PKVP09], a graph labeling problem is solved to find the optimal shift of every pixel in the output image from its source in an input image. Nevertheless, additional smoothness term is needed to avoid discontinuities, and so computationally demanding optimization is required.

In this paper, we demonstrate that for style exemplars which contain mostly stochastic textures the interplay between local guidance and textural masking effect described by Ashikhmin [Ash01] makes seams between the individual chunks barely visible and thus simple blending operation can be used to suppress them without the need to take into account texture coherence explicitly.

3. Our Approach

In this section, we describe the core idea behind our approach and discuss implementation details. As a motivation, we first describe a simple experiment that inspired us to develop our method.

To understand the properties of optimization-based approaches, we applied the StyLit algorithm [FJL*16] to transfer the style from a hand-drawn image of a sphere to a more complex 3D model using normals as guidance (see Fig. 2). The texture coherence term in the original energy formulation, and the mechanism for preventing excessive utilization of source patches, help the optimization converge to a state where large chunks of the original source texture (Fig. 2b) are copied to the target image resulting in a high-fidelity transfer (Fig. 2d).

Inside each coherent chunks, the errors of texture coherence term are equal to zero. Errors of the guidance term can be bounded by a small upper bound, i.e., we can find a chunk of the normal field on the exemplar sphere to roughly approximate the corresponding chunk of normals on the target 3D model within a certain error threshold. The black lines in Fig. 2e, f show the boundaries between chunks within which all pixels have guidance errors below some predefined error bound. The lines get sparser and the regions grow larger as the bound increases.

This fact inspired us to seek large coherent chunks of style regions directly using simple pixel-level operations foregoing expensive patch-based optimization.

3.1. Basic Algorithm

To build such a mosaic of coherent chunks, we need to estimate the shape and spatial location of each individual chunk. This is done by going in the scan-line order or by picking a random pixel (seed) in the target image and finding its corresponding location in the source exemplar (see Fig. 3a, b). Usually, the local guidance at each target pixel consists of two values that indirectly specify the corresponding pixel coordinates in the source exemplar. This fact enables us to use a simple look-up table to retrieve, for each target pixel, the corresponding location in the source exemplar. In a more complex scenario where additional guiding channels are used, we can accelerate the retrieval using search trees [AMN*98]. Once we know the corresponding source pixel, we calculate the difference between the guidance values in local spatially-aligned regions. The target pixels having guidance difference smaller than a user-defined threshold belong to the current chunk (Fig. 3c). We copy those corresponding pixels and paste them in the target image (Fig. 3d). By repeating the searching and copying steps, we eventually cover all pixels in the target image (Fig. 3e and Fig. 7, left).

Our approach does not explicitly enforce textural coherence. One might expect that seams between individual chunks will be visible. Surprisingly, for a relatively large variety of exemplars, seams are either not apparent or can be effectively suppressed using linear blending applied around the boundaries of individual chunks. The reasons are twofold: (1) local guidance is often smooth and continuous and thus two neighboring chunks are usually roughly aligned; (2) hand-drawn exemplars are typically highly stochastic which intrigues the human visual system and makes the structural inconsistencies less noticeable [Ash01].

3.2. Implementation Details

The basic algorithm can be implemented in a brute-force manner (see supplementary material for pseudocode). Though simple, it is highly inefficient due to the redundant visiting of target pixels and the inherent sequential nature that prohibits parallel implementation.

To overcome the mentioned drawbacks, we use a more efficient approach that is fully parallel and guarantees that every target pixel will be visited only once (see Algorithm 1). The key idea here is to define an implicit hierarchy of target seeds $\mathbf{q}$ (see Fig. 4) with different granularity. On the top level, seeds are distributed randomly far apart. On the lower levels, the distance between them is gradually decreased by a factor of 2. Algorithmically we build this hierarchy by placing dots at regular grid points whose positions are randomly perturbed. Then for every target pixel $\mathbf{p}$, we start at the top level of our seed hierarchy and find the spatially nearest target seed $\mathbf{q}_l$ within the same level l.

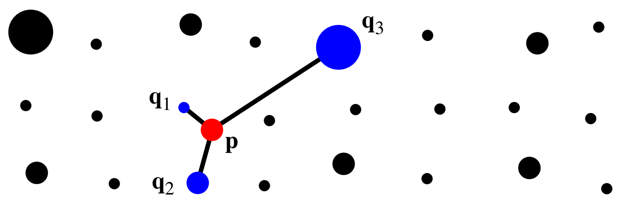


Figure 4: *An example hierarchy of spatially distributed seeds $\mathbf{q}_l$ (black and blue dots). The hierarchy level l corresponds to the size of the dots: the dots in the top level are the largest. For every target pixel $\mathbf{p}$ (red dot), we proceed from the top level to the bottom $l = \{3,2,1\}$. At the top level, we retrieve the spatially nearest seed q_3, and check whether the guidance value between p and q_3 falls below a specified threshold. If not, we proceed to the nearest seed in the next lower level q_2 and then q_1.*

Algorithm 1: ParallelStyleBlit

Inputs : target pixel $\mathbf{p}$, target guides G_T, source guides G_S, source style exemplar C_S, threshold t, number of levels L.
Output: stylized target pixel color $C_T[\mathbf{p}]$.

SeedPoint(pixel $\mathbf{p}$, seed spacing h)**:**
  $\mathbf{b} = \lfloor \mathbf{p}/h \rfloor$; $\mathbf{j} =$ `RandomJitterTable`$[\mathbf{b}]$
  return $\lfloor h \cdot (\mathbf{b} + \mathbf{j}) \rfloor$

NearestSeed(pixel $\mathbf{p}$, seed spacing h)**:**
  $d^\star = \infty$
  for $x \in \{-1, 0, +1\}$ **do**
  for $y \in \{-1, 0, +1\}$ **do**
    $\mathbf{s} =$ SeedPoint$(\mathbf{p} + h \cdot (x, y),\ h)$
    $d = ||\mathbf{s} - \mathbf{p}||$
    if $d < d^\star$ **then**
      $\mathbf{s}^\star = \mathbf{s}$; $d^\star = d$
  return $\mathbf{s}^\star$

ParallelStyleBlit(pixel $\mathbf{p}$)**:**
  for *each level* $l \in (L, \dots, 1)$ **do**
    $\mathbf{q}_l =$ NearestSeed$(\mathbf{p}, 2^l)$
    $\mathbf{u}^\star = argmin_{\mathbf{u}}\, ||G_T[\mathbf{q}_l] - G_S[\mathbf{u}]||$ ← *found via lookup, or a tree search.*
    $e = ||G_T[\mathbf{p}] - G_S[\mathbf{u}^\star + (\mathbf{p} - \mathbf{q}_l)]||$
    if $e < t$ **then**
      $C_T[\mathbf{p}] = C_S[\mathbf{u}^\star + (\mathbf{p} - \mathbf{q}_l)]$
      break

If the nearest seed yields guidance error below a specific threshold, we transfer the corresponding style color to the target pixel and stop the traversal, otherwise we enter the next lower level of the hierarchy and continue until we reach the bottom level.

When seams become apparent, we can optionally perform blending on the boundaries of individual chunks. This can be simply implemented by replacing the transfer of pixel colors with the transfer of pixel coordinates, i.e., every target pixel will be assigned its cor-

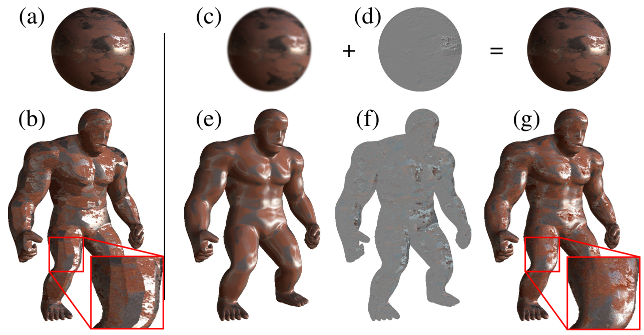


Figure 5: *Multi-layer approach: style exemplar with smooth gradients and high-frequency details (a) may introduce visible seams (b). By decomposing the exemplar into base (c) and detail (d) layer one can employ The Lit Sphere algorithm [SMGG01] for the base (e), then apply our algorithm on the detail (f), and finally, make the composition which preserves both smoothness as well as high-frequency details (g). Style exemplar:* © Free PBR

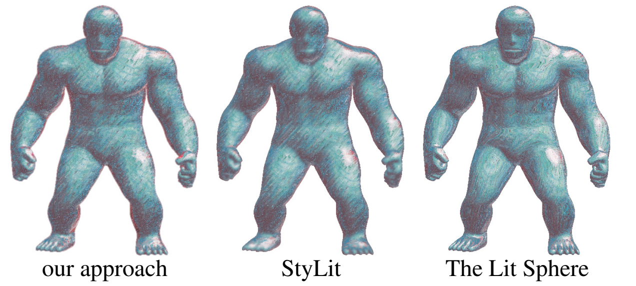


Figure 7: *Stylized results produced by our method (left), StyLit [FJL*16] (middle) and The Lit Sphere [SMGG01] (right). Compared to StyLit, our approach is orders of magnitude faster and produces similar result quality without explicitly enforcing textural coherence. Compared to The Lit Sphere, our algorithm is equally fast, but retains the high-level structure of the used artistic media; i.e., large directional brush strokes are better preserved.*

responding source pixel coordinates. This structure is equivalent to the nearest neighbor field used in patch-based synthesis. Then, the final colors are obtained using a voting step [KEBK05, WSI07] where the color of every target pixel is computed as the average color of co-located pixels from a set of source patches that intersect the currently processed target pixel. This operation is simple to implement and is, in fact, equivalent to performing blending only at chunk boundaries.

3.3. Extensions

Our method is suitable both for hand-drawn style exemplars as well as realistic materials that have stochastic nature. Those, however, may contain smooth gradients together with high-frequency features (see Fig. 5a). In this case, finding a threshold that would preserve both smoothness and high-frequency details could be difficult (Fig. 5b). We resolve this problem by employing a multi-layer approach [BA83, HRRG08, GSDC17]. We first separate the input style exemplar into a smooth base layer (Fig. 5c) and a high-frequency detail layer (Fig. 5d). To obtain the base layer, we first filter the original style image with Gaussian filter and then we subtract the filtered image from the original to get the detail layer. Style transfer is then performed in each layer separately. In the base layer, we employ The Lit Sphere algorithm [SMGG01] which works well for low-frequency content (Fig. 5e). For the detail layer, we apply our algorithm which preserves high-frequency content (Fig. 5f) and finally, we make the seamless composition by summing synthesized base and detail layers (Fig. 5g).

Our approach can also be extended to animations. The local guidance implicitly encourages temporal coherence in the synthesized content while the randomization of seed points slightly perturbs the structure of the resulting mosaic. This creates a slight temporal flickering effect which gives the observer an illusion of a hand-colored animation where every frame is drawn independently by hand [FLJ*14]. Moreover, the amount of flickering can be controlled by changing the guidance threshold. Higher threshold gives rise to larger chunks and more visible visual changes between consecutive frames, and thus the amount of flickering is increased.

4. Results

We implemented our approach on the CPU using C++ and on the GPU using OpenGL with GLSL (for desktop) as well as WebGL (for mobile devices). As a default threshold value, we use $t = 24$ and the number of seed levels is set to $L = 7$. Table `RandomJitterTable` contains random values between $(0,1)$. On a single core CPU (Core i7, 2.8 GHz), we stylize a one-megapixel image at 10 frames per second while on the GPU (GeForce GTX 970) we can achieve more than 100 frames per second at 4K resolution. This represents three orders of magnitude speedup as compared to the original StyLit algorithm [FJL*16] which requires computationally demanding iterative optimization. Such improvement enables us to perform real-time style transfer even on devices with a lower computational budget including mid-range mobile phones (using WebGL 1.0 we can achieve, e.g., 15 frames per second full screen on the Samsung Galaxy A3).

We tested our approach in three different style-transfer scenarios where local guidance is used: normals (see Fig. 6 and 9), texture coordinates (Fig. 7 and 10), and a displacement field (Fig. 11). For additional results see also Fig. 1 and the supplementary material.

For normal-based guidance, we compared our approach with the StyLit algorithm [FJL*16] to confirm that we produce comparable results that preserve visually important characteristics of artistic media (see Fig. 1, 7, 6, and the supplementary material that includes results of a perceptual study). In addition, our approach also better preserves geometric details (cf., e.g., head result in Fig. 6) since it compares guidance channels per pixel and does not involve any patch-based averaging used in the StyLit algorithm. Such averaging acts as a low-pass filter applied on the guidance channel. In the supplementary video, we present a recording of an interactive session (on the GPU as well as on a smartphone) where the user manipulates and animates a 3D model on which a selected artistic style is transferred in real-time. We also demonstrate controllable temporal flickering effect following the concept of Fišer

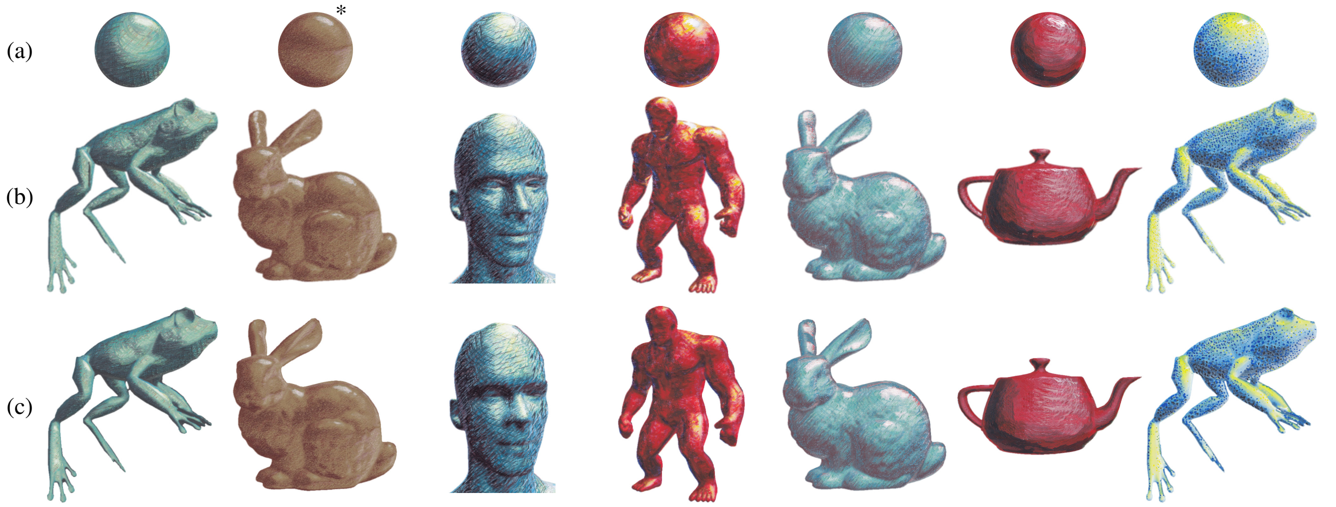


Figure 6: *Comparison with StyLit [FJL*16]: original style exemplar (a), the result of our approach (b), and the result of StyLit (c). Style exemplars:* © Pavla Sýkorová *and* Daichi Ito*

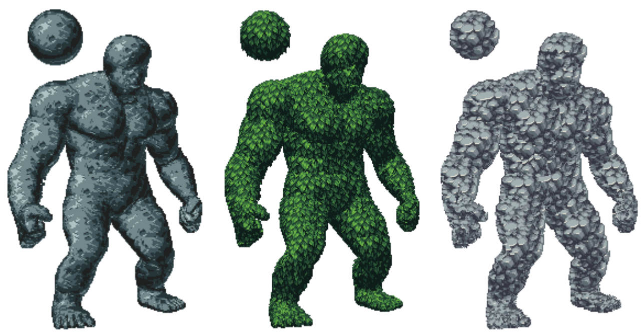


Figure 8: *Examples of stylization where normal-based guidance is used to transfer delicate pixel art styles. In this scenario, copy-and-paste nature of our approach is crucial as it allows to retain essential details on the pixel level which are important to preserve the fidelity of images that has been created manually pixel by pixel. Style exemplars:* © *Lachlan Cartland*

et al. [FLJ*14]. Our approach is suitable also for transferring delicate pixel art styles where even small blurring artifacts may become apparent (see Fig. 8).

We also compare our technique with The Lit Sphere algorithm [SMGG01], i.e., MatCap scenario which is based on environment mapping. It directly maps colors between corresponding pixels according to a one-to-one mapping specified by the normal values. Due to pixel-level processing, high-level structures visible in the style exemplar become distorted, and thus only low-frequency exemplars can be used. In contrast, our approach copies larger chunks and thus better preserves high-level structures which are important to retain fidelity of the original style exemplar (see Fig. 7, Fig. 9 and the supplementary material). This improvement is visible also in the case where texture coordinates are derived directly from a planar parametrization (unwrap) of the target 3D mesh (see Fig. 1, 10, and the supplementary material). Here the style exemplar can be painted on a specific 2D projection of the 3D mesh [MNZ*15] or directly on the planar unwrap. In both cases, our approach transfers larger chunks of the original texture which effectively removes artifacts caused by texture mapping and better preserves the fidelity of the style exemplar. To do that, however, a larger threshold is required which can break the structure of high-level geometric features. To avoid this artifact, we use additional segmentation guide which prevents chunks from crossing boundaries of semantically important regions (see supplementary material for examples of these additional guiding channels).

Finally, we tested our approach in a scenario where a dense displacement field is used as a local guide. An example of such setting is artistic style transfer to human portraits [FJS*17]. Here the displacement field is defined by a set of corresponding facial landmarks detected in the source exemplar and in the target subject. Moving least squares deformation [SMW06] is used to compute dense correspondences, i.e., the resulting displacement field. Besides the local guide, two additional guidance channels are used for patch-based synthesis: a segmentation map containing semantically important facial parts (head, hair, eyes, eyebrows, nose, and mouth) and an appearance guide that helps to preserve subject's identity (see the supplementary material for examples of all guiding channels). The resulting visual quality is comparable or a bit inferior to the previous work, but sufficient for applications with limited computational resources (see Fig. 1, 11, and the supplementary material). To demonstrate such an application a recording of a live session with real-time facial style transfer to a video stream is presented in the supplementary material. To highlight the benefit of our method, the result of our algorithm is compared side-by-side with a simple texture mapping scheme. Note how our approach better preserves the fidelity of the original artistic media.

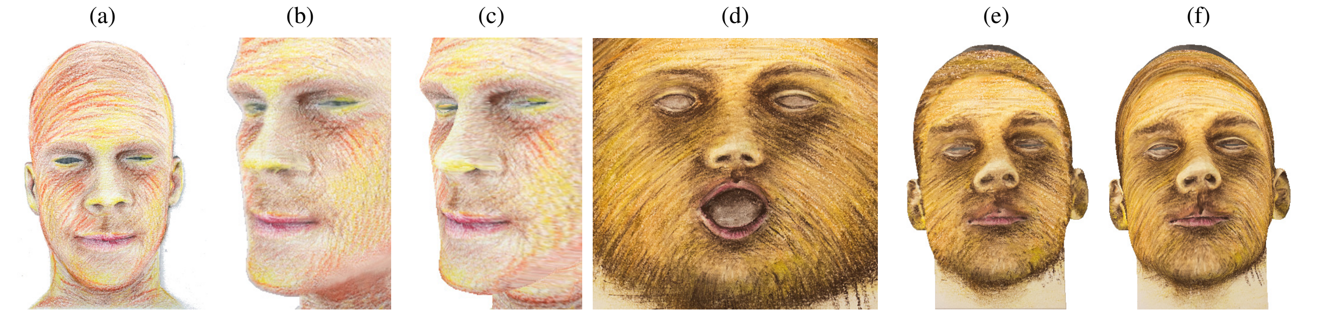


Figure 10: *Comparison with texture mapping: original artwork (a, d), new viewpoint generated using our approach (b, e) and using texture mapping (c, f). Style exemplars:* © Pavla Sýkorová

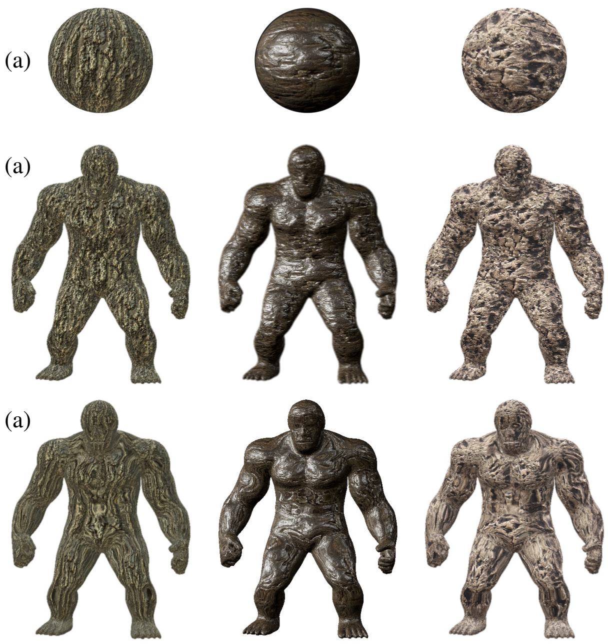


Figure 9: *Comparison with The Lit Sphere [SMGG01]: style exemplar (a), our approach (normal-based guidance) (b), and The Lit Sphere result (c). Note how our approach enables MatCap scenario also for materials that contain distinct high-level features while the computational overhead is still comparable to the original Lit Sphere method which is not applicable in this context. Style exemplars:* © Free PBR

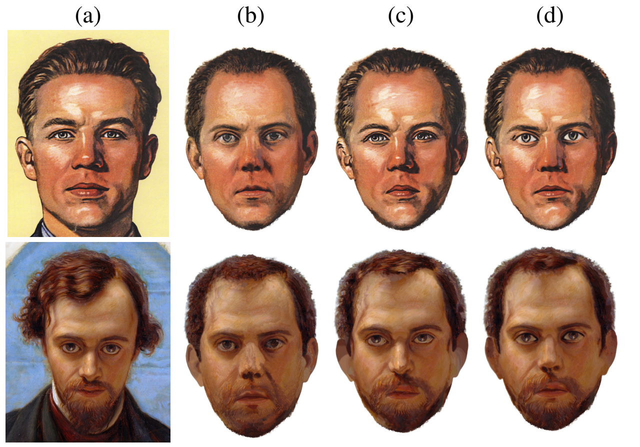


Figure 11: *Comparison with FaceStyle [FJS*17]: original style exemplar (a), the result of our method using strong (b) and weak (c) appearance guide, and the result of FaceStyle (d).*

5. Limitations and Future Work

Although our method produces visually pleasing results for a variety of different style exemplars and different types of guidance, there are some limitations that need to be taken into account.

For non-stochastic (semi-)regular textures like a brick wall, our approach may introduce visible misalignment of regular structures (see Fig. 12a). To suppress this artifact one may employ post-transfer alignment of individual chunks using the method of Lucas and Kanade [LK81]. This operation can be performed relatively quickly as it requires only inexpensive accumulation of image gradients and pixel differences over chunk boundaries and since the misalignment is usually small, only a few iterations are necessary to get a better alignment (see Fig. 12b). Nevertheless, the quality is still inferior as compared to full-fledged synthesis (see Fig. 12c).

Visible misalignment of individual chunks can also be apparent in cases when a set of guidance channels used for the style transfer does not contain local guide or when the influence of local guide is low as compared to other channels. Example of such scenario can be the usage of light path expressions in [FJL*16] (see Fig. 13a). In this case, we envision a more sophisticated post-transfer alignment mechanism would also handle larger discrepancies.

Our approach shares limitations with techniques that use guided patch-based synthesis [KNL*15, FJL*16]. They may produce excessive repetition in cases when the scale of the target object is fairly different as compared to the object in the style exemplar, e.g., during zoom-in operations or when there is not enough variability in the guidance, e.g., when stylizing flat surface using spherical exemplar (see Fig. 13b). This drawback can be alleviated by adjusting

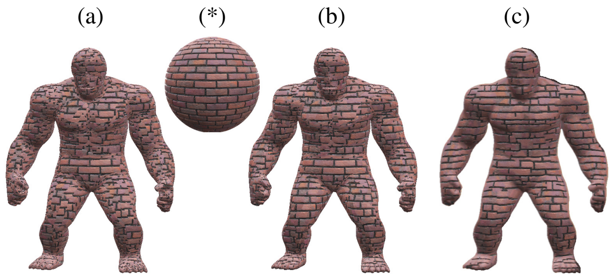


Figure 12: *Limitation: when a (semi-)regular texture (*) is used as a style exemplar, our method may introduce visible misalignment of regular features (a). To suppress this artifact, post-transfer alignment of individual chunks can be performed (b) to get a result which is closer to the output of StyLit algorithm [FJL*16] (c). Style exemplar:* © Free PBR

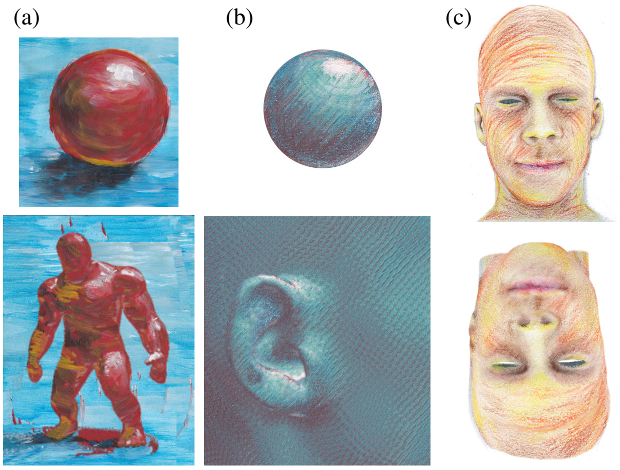


Figure 13: *Limitations: when a set of guiding channels does not contain local guide, for instance when light path expressions are used [FJL*16], our approach may introduce visible seams (a); when the target contains large areas of pixels having constant guidance values, our method produces a visible texture repetition (b); when the orientation of local guide changes considerably (vertically flipped), translation cannot accommodate this change, and our technique starts to produce smaller chunks (c). Style exemplars:* © Pavla Sýkorová

the global scale or by preparing a different style exemplar that contains similar structures as the target objects.

Another limitation is related to the rotation in the image plane when texture coordinates or displacement field are used for guidance. In this situation corresponding counterparts of target seeds can be found easily, however, as their neighborhoods have notably different content caused by rotation, the error threshold limits the size of the target chunks, and the method will introduce blur into the result (see Fig. 13c). To alleviate this issue, one can pre-rotate the source guidance to match with the dominant orientation in the target channel as in [FJS*17].

6. Conclusion

We have presented a new approach for example-based style transfer suitable for applications where strong local guidance is used. We demonstrated that in this scenario computationally demanding patch-based synthesis converges to a solution that can be easily mimicked using a relatively simple algorithm with notably lower computational overhead. We also showed that considering textural coherence is not crucial for successful style transfer as local guidance in conjunction with the visual masking effectively suppresses visible seams for a variety of hand-drawn as well as photorealistic style exemplars. Since our method is several orders of magnitude faster as compared to the current state-of-the-art, it enables real-time style transfer even in applications with limited computational resources available.

Acknowledgements

We would like to thank Brett Ineson for providing the animated face model, Michal Kučera for performing the perceptual study, and all anonymous reviewers for their fruitful comments and suggestions. This research was supported by the Fulbright Commission in the Czech Republic, the Technology Agency of the Czech Republic under research program TE01020415 (V3C – Visual Computing Competence Center), by the Grant Agency of the Czech Technical University in Prague, grant No. SGS19/179/OHK3/3T/13 (Research of Modern Computer Graphics Methods), by Research Center for Informatics No. CZ.02.1.01/0.0/0.0/16_019/0000765, and by Adobe.

References

[AMN*98] Arya S., Mount D. M., Netanyahu N. S., Silverman R., Wu A. Y.: An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM 45*, 6 (1998), 891–923. 4

[Ash01] Ashikhmin M.: Synthesizing natural textures. In *Proceedings of Symposium on Interactive 3D Graphics* (2001), pp. 217–226. 3, 4

[BA83] Burt J. R., Adelson E. H.: A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics 2*, 4 (1983), 217–236. 5

[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. 2

[BKR17] Bi S., Kalantari N. K., Ramamoorthi R.: Patch-based optimization for image-based texture mapping. *ACM Transactions on Graphics 36*, 4 (2017), 106. 3

[BN76] Blinn J. F., Newell M. E.: Texture and reflection in computer generated images. *Communications of the ACM 19*, 10 (1976), 542–547. 2

[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. 2

[BTM06] Barla P., Thollot J., Markosian L.: X-toon: An extended toon shader. In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2006), pp. 127–132. 2

[BZ17] Barnes C., Zhang F.-L.: A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media 3*, 1 (2017), 3–20. 2

[DBP*15] DIAMANTI O., BARNES C., PARIS S., SHECHTMAN E., SORKINE-HORNUNG O.: Synthesis of complex image appearance from limited exemplars. *ACM Transactions on Graphics 34*, 2 (2015), 22. 2

[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. 3

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *SIGGRAPH Conference Proceedings* (2001), pp. 341–346. 3

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *Proceedings of IEEE International Conference on Computer Vision* (1999), pp. 1033–1038. 1

[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. 1, 2, 3, 5, 6, 7, 8

[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. 1, 2, 3, 6, 7, 8

[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. 5, 6

[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. 1, 3

[GSDC17] GUINGO G., SAUVAGE B., DISCHLER J.-M., CANI M.-P.: Bi-layer textures: A model for synthesis and deformation of composite textures. *Computer Graphics Forum 36*, 4 (2017), 111–122. 5

[Her98] HERTZMANN A.: Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH Conference Proceedings* (1998), pp. 453–460. 2

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *SIGGRAPH Conference Proceedings* (2001), pp. 327–340. 1, 2

[HRRG08] HAN C., RISSER E., RAMAMOORTHI R., GRINSPUN E.: Multiscale texture synthesis. *ACM Transactions on Graphics 27*, 3 (2008), 51. 5

[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. 1

[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. 2

[KEBK05] KWATRA V., ESSA I. A., BOBICK A. F., KWATRA N.: Texture optimization for example-based synthesis. *ACM Transactions on Graphics 24*, 3 (2005), 795–802. 2, 3, 5

[KNL*15] KASPAR A., NEUBERT B., LISCHINSKI D., PAULY M., KOPF J.: Self tuning texture optimization. *Computer Graphics Forum 34*, 2 (2015), 349–360. 2, 7

[KSE*03] KWATRA V., SCHÖDL A., ESSA I. A., TURK G., BOBICK A. F.: Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics 22*, 3 (2003), 277–286. 3

[LH05] LEFEBVRE S., HOPPE H.: Parallel controllable texture synthesis. *ACM Transactions on Graphics 24*, 3 (2005), 777–786. 3

[LH06] LEFEBVRE S., HOPPE H.: Appearance-space texture synthesis. *ACM Transactions on Graphics 25*, 3 (2006), 541–548. 3

[LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence* (1981), pp. 674–679. 7

[LLX*01] LIANG L., LIU C., XU Y.-Q., GUO B., SHUM H.-Y.: Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics 20*, 3 (2001), 127–150. 3

[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. 1, 3

[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. 1, 3

[MNZ*15] MAGNENAT S., NGO D. T., ZÃIJND F., RYFFEL M., NORIS G., ROETHLIN G., MARRA A., NITTI M., FUA P., GROSS M. H., SUMNER R. W.: Live texturing of augmented reality characters from colored drawings. *IEEE Transactions on Visualization and Computer Graphics 21*, 11 (2015), 1201–1210. 2, 6

[PFH00] PRAUN E., FINKELSTEIN A., HOPPE H.: Lapped textures. In *SIGGRAPH Conference Proceedings* (2000), pp. 465–470. 3

[PKVP09] PRITCH Y., KAV-VENAKI E., PELEG S.: Shift-map image editing. In *Proceedings of IEEE International Conference on Computer Vision* (2009), pp. 151–158. 3

[PS00] PORTILLA J., SIMONCELLI E. P.: A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision 40*, 1 (2000), 49–70. 3

[RRFT14] REMATAS K., RITSCHEL T., FRITZ M., TUYTELAARS T.: Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 3898–3905. 2

[SED16] SELIM A., ELGHARIB M., DOYLE L.: Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics 35*, 4 (2016), 129. 1, 2

[SID17] SEMMO A., ISENBERG T., DÖLLNER J.: Neural style transfer: A paradigm shift for image-based artistic rendering? In *Proceedings of International Symposium on Non-Photorealistic Animation and Rendering* (2017), p. 5. 3

[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. 2, 5, 6, 7

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3 (2006), 533–540. 6

[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. 2

[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. 2

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014). 3

[TAY13] TODO H., ANJYO K., YOKOYAMA S.: Lit-sphere extension for artistic rendering. *The Visual Computer 29*, 6–8 (2013), 473–480. 2

[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. 2, 3, 5

[ZSL*17] ZHOU Y., SHI H., LISCHINSKI D., GONG M., KOPF J., HUANG H.: Analysis and controlled synthesis of inhomogeneous textures. *Computer Graphics Forum 36*, 2 (2017), 199–212. 2