The Visual Computer manuscript No.
(will be inserted by the editor)

Video Style Transfer by Consistent Adaptive Patch Sampling

Oriel Frigo · Neus Sabater · Julie Delon · Pierre Hellier

Received: xx/xx/xxxx / Accepted: xx/xx/xxxx

Abstract This paper addresses the example-based stylization of videos. Style transfer aims at editing an image so that it matches the style of an example. This topic has recently been investigated massively, both in the industry and academia. The difficulty lies in how to capture the style of an image and correctly transferring it to a video. In this paper, we build on our previous work "Split and Match" for still pictures, based on adaptive patch synthesis. We address the issue of extending that particular technique to video, ensuring that the solution is spatially and temporally consistent. Results show that our video style transfer is visually plausible, while being very competitive regarding computation time and memory when compared to neural network approaches.

Keywords Style Transfer · Texture Synthesis · Non-photorealistic Rendering · Video Processing

1 Introduction

Style transfer consists in transferring the style of an example, typically a painting, to another image or video. This problem has received recently a considerable interest [1,2], especially with the blooming of convolutional neural network approaches [3,4,5]. There are many applications to style transfer, ranging from social networks, virtual reality to movie post-production.

The upcoming film "Loving Vincent" [1] can be seen as a remarkable example of the practical possibilities of style transfer. This endeavor is claimed to be the first fully painted feature film to be made, and includes an average of 12 oil paintings per second of video. Note that more than 100 painters were involved in the production, which is a painstaking work that could be facilitated by style transfer techniques.

In this paper, we build on our recent paper [6] that proposed a patch-based method for style transfer, and extend this approach to video content. Example-based patch methods have been sucessfully used in various cases, ranging from texture synthesis [7], inpainting [8], to super-resolution [9]. These methods have proven their capability to capture knowledge about the image based on the content intra-similarity. In this paper, we extend this patch-based approach to video, where an efficient optical flow method is used to track parts of the image, while inconsistencies of the tracking (captured by a metric on motion reliability) are handled by a spatially and temporally consistent patch synthesis.

2 Related Work

Style transfer can be traced back to two seminal papers, "Image Analogies" [10] and "Image quilting" [7], which presented non-photorealistic rendering based on example images as one possible application of texture synthesis. Closely related is the concept of color transfer [11], where one seeks to transfer the color palette from a target to a source image.

In Image Analogies, style transfer is computed as a pixelwise texture synthesis, inspired by the non-parametric sampling approach of [12]. An analogy is defined as the relationship (transformation) between a pair of aligned example images, typically a non-filtered picture of a scene and a stylized painting of this same scene. Then, the task is to apply an analogous transformation to a given input image, such that it has the same rendering of the example painting. Every pixel to be synthesized

Address(es) of author(s) should be given

[1] http://join.lovingvincent.com/

in the output image is selected from the example painting, by minimizing an energy accounting to the similarity between the input picture and the non-filtered example picture. A seminal work in video style transfer was conducted in [13], as an extension of image analogies to stylization of video animations. Temporal coherence is enforced and neighborhood matching is accelerated by using the randomized patch correspondences of "PatchMatch" [14]. Nevertheless, the method still has high computational complexity as it perform a pixel-wise style synthesis, which results in large search space for neighborhood matching. Furthermore, it is worthy noting that approaches based on image analogies assume the existence of a registered pair of example images from which the style analogy is learned, which constrain the practical use of such methods.

The work of [7], in parallel with [15], have introduced the concept of patch sampling for texture synthesis and texture transfer. These methods perform example-based synthesis by sampling from a pool of candidate patches, which are selected from an example image. Sampling patches instead of pixels have the advantage of lower computational complexity, but a post-processing step may be required to overcome blocking artifacts. In [15], linear interpolation is used to produce seamless patch blending, while in image quilting [7], optimal patch boundary cuts are computed.

Our previous work [6] has shown that the quality of patch-based style transfer can be improved by adapting the patch dimensions to the structure of the original and example images. Inspired by [16] and [17] the "Split and Match" approach considers a Markov Random Field (MRF) probability density modeling and computes an approximate Maximum a Posteriori (MAP) solution through loopy belief propagation [18]. A follow-up patch-based approach [19] for style transfer is also based on adaptive patches for capturing the style from example images. In this paper, we extend our previous still image style transfer [6] to image sequences, as a fast and efficient approach for video style transfer.

The paper of [3] has introduced a technique that builds upon deep Convolutional Neural Networks (CNN) to separate and recombine the content and the style of an original and an example image. Their main idea is to represent style by correlations between features from different layers of a CNN, and to represent content by feature responses in higher layers of a CNN. Their method produces impressive results for style transfer, but has the drawback of high computational complexity. This work received a great attention in vision community, inspiring other papers such as [20, **?**] and resulted in the development of products such as "DeepArt" [2] and an accelerated smartphone application "Prisma" [3]. Furthermore, an adaptation of CNN-based style transfer for post-production of a feature film has been described by [21].

Recently, neural style transfer for videos has been proposed by [22], where temporal coherence is enforced to the stylization by using optical flow guidance. Our approach for temporal coherence is similar in spirit to [22]: propagate style by motion warping where optical flow is reliable, and resynthesizing style where optical flow is not reliable.

It should be noted that image and video style transfer based on deep CNN's differs considerably from our approach, since it assumes a pre-trained neural network architecture. Although results of neural style transfer are mostly excellent, as remarked by [23], the stylization by neural networks tends to be unpredictable in practice. On the other hand, patch based approaches may be advantageous for a more predictable stylization that better preserves the main structures in the original image.

3 Split and Match for Still Images

In this section, we briefly review our "Split and Match" approach, which proposes a patch-based algorithm to transfer the style of an example (or style) image to an input image. A more detailed version of this style transfer method can be found in our recent paper [6]. This approach, illustrated by Figure 2, starts by constructing an adaptive quadtree segmentation of the input image geometry, taking into account the capacity of the patches in the quadtree to be represented by patches of the style image. This adaptive partition is necessary to respect the original image geometry, using small patches (fine brushes) along geometric features, while letting large patches (rougher brushes) to capture the style of the example image everywhere else. Let us denote by $u : \Omega_u \to \mathbb{R}^3$ the input image and $v : \Omega_v \to \mathbb{R}^3$ the example image. For two regions R and R' of similar size of Ω_u and Ω_v, we write $u(R)$ and $v(R')$ the respective restriction of u and v to R and R', and we define the normalized distance between $u(R)$ and $v(R')$ by

$$d[u(R), v(R')] = \frac{||u(R) - v(R')||^2}{|R|}, \text{ with } |R| \text{ the size of } R. \tag{1}$$

The Split and Match decomposition works as follows. Starting from the region $R_1 := \Omega_u$, each rectangle R_i

[2] https://deepart.io/
[3] http://prisma-ai.com/

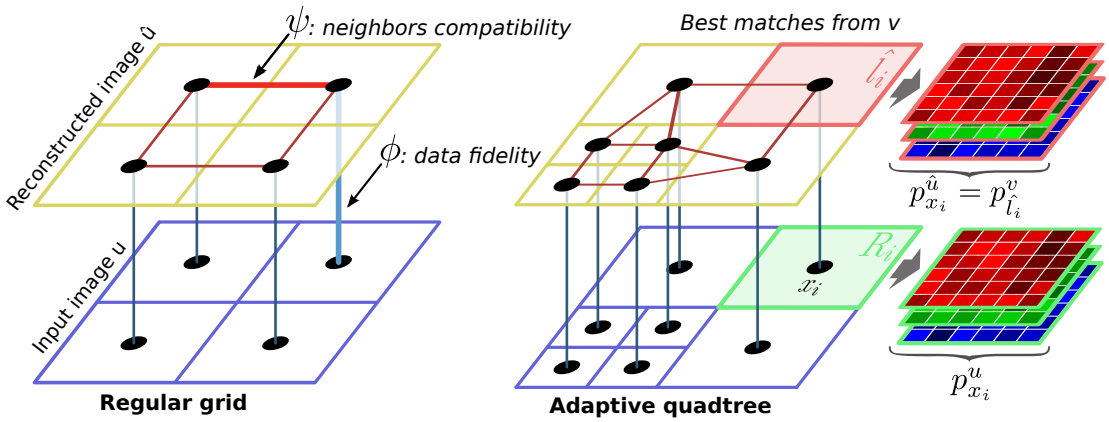

Fig. 1: On the left, MRF over a regular grid and on the right MRF over an adaptive image partition. The bottom layers represent image units from the observed scene, while nodes in the top layer represent hidden image units that we search to estimate through inference. Vertical edges represent data fidelity terms, while horizontal edges represent pairwise compatibility terms.

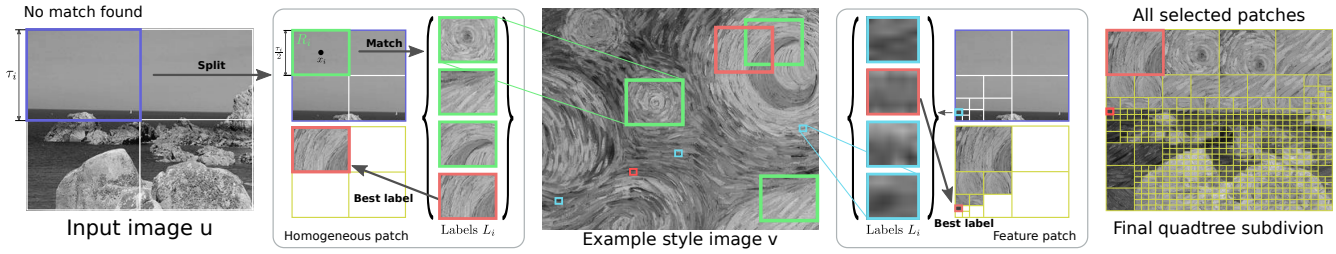

Fig. 2: Illustration of our Split and Match approach [6] for style transfer between still images. From left to right: the input image u is recursively splitted into patches. Each patch is matched with the example image, and if there is a match sufficiently close, or if the patch variance is sufficiently small, the splitting is ended. Each patch over region R_i has a set of label candidate patches L_i, from which a best label is computed by loopy belief propagation. As it can be seen in the figure, the patch labels on the left part account mostly for style (homogeneous patches in the original image), while the label patches on the right part account mostly for smaller feature patches, which account for reconstructing details and geometry in the stylized image.

of the partition is split into four equal rectangles if

$$\left(\sqrt{\mathrm{Var}(u(R_i))} + d[u(R_i), v(R_i^v)] > \omega \ \textbf{and} \ \sqrt{|R_i|} > \Upsilon_0\right) \textbf{or} \left(\sqrt{|R_i|} > \Upsilon_1\right), \quad (2)$$

where R_i^v is the rectangle R of the style image v minimizing the distance $d[u(R_i), v(R)]$, ω is a similarity threshold (fixed to $\omega := 15$), Υ_0 is the minimum patch size and Υ_1 the maximum patch size allowed in the quadtree (respectively fixed to 8^2 and 256^2). At the end of this decomposition, for each region R_i of the final quadtree a set of M candidate labels $L_i = \{l_{i_m}\}_{m=1}^{M}$ is selected by computing the M-nearest neighbors of the patch $u(R_i)$ in v. The label l_{i_m} is the central pixel of the m^{th} nearest region of $u(R_i)$ in v. These patch labels are also required to be sufficiently distant from each other to favor label variety.

In a second step, we rely on a Markov Random Field model over the final partition $\{R_i\}_{i=1}^{n}$ to minimize pairwise distances and patch differences. More precisely, we search for the set of label assignments $\hat{L} = \{\hat{l}_i\}_{i=1}^{n}$ maximizing a probability density

$$P(L) = \frac{1}{Z} \prod_i \phi(l_i) \prod_{(i,j)\in\mathcal{N}} \psi(l_i, l_j), \quad (3)$$

where Z is a normalization constant, ϕ is a data fidelity term and ψ a compatibility term between neighboring regions ($(i,j) \in \mathcal{N}$ means that R_i and R_j are neighbors in the quadtree). The data fidelity term ϕ is defined as

$$\phi(l_i) = \exp(-\lambda_d \, d[u(R_i), v(R_{l_i})]), \quad (4)$$

where R_{l_i} is the region of label l_i in v, and λ_d a positive weight (fixed to 2 in practice). The role of the function ψ

is to ensure that neighboring candidate patches are similar enough at the vicinity of their common frontier. Extending each region R_i of the quadtree in each direction by 50% yields overlapping regions $\widetilde{R_i}$. For $(i,j) \in \mathcal{N}$, $\psi(l_i, l_j)$ is then defined as a mix of a smoothness term and a term penalizing local label repetitions

$$\psi(l_i, l_j) = \exp(-\lambda_s \, d_\cap[v(\widetilde{R_{l_i}}), v(\widetilde{R_{l_j}})] + \lambda_r \, |l_i - l_j|^2), \quad (5)$$

with λ_s and λ_r two positive weights (fixed to 2 and 1 in all experiments), and where $d_\cap[v(\widetilde{R_{l_i}}), v(\widetilde{R_{l_j}})]$ is the normalized quadratic distance between the candidates $v(\widetilde{R_{l_i}})$ and $v(\widetilde{R_{l_j}})$ on the overlap $\widetilde{R_i} \cap \widetilde{R_j}$.

In order to maximize the probability density (3), we adopt the *Loopy Belief Propagation* method [18], [24]. In this approach, neighboring variables update their likelihoods by *message passing* which permits to maximize the density (3) in a few number of iterations.

To avoid stitching artifacts along patch borders (see Figure 3), patches are eventually merged through a linear alpha blending, ensuring smoother transitions between neighbors. The last and optional step of the algorithm is a color transfer [25] used to match consistently the color palettes of the original and example images, combined with a global contrast specification approximated by a power law model. The whole process results in a stylized image $\tilde{u}$.

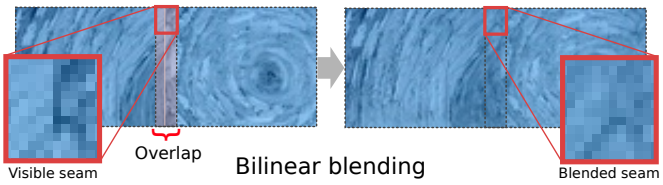

Fig. 3: Illustration of the bilinear blending used to avoid stitching artifacts.

4 Video Style Transfer

We explain in this section how to adapt our Split and Match style transfer to image sequences. Obviously, applying an independent style transfer to each frame of a sequence leads to strong texture flickering and poor visual results, even if neighbor frames share most of their geometry and color (see the left column of Figure 9). In order to impose coherence between frames, we propose a motion-based temporally coherent stylization, illustrated by Figure 4.

In the following, we denote by $U = \{u_t\}_{t=1}^{D}$ the input image sequence and by v the example (style) image. Each discrete image $u_t : \Omega \to \mathbb{R}^3$ is defined over the same discrete domain Ω. A "keyframe rate" r is chosen in such a way that we have one keyframe for each second of video (typically, $r := 25$ for a 25 fps video). We write $U_{k,k+r} = \{u_t\}_{t=k+1}^{k+r-1}$ the set of all frames delimited between the keyframes u_k (also called left keyframe) and u_{k+r} (right keyframe). At the beginning of the algorithm, our Split and Match algorithm is applied to transfer the style of the example image v to the central keyframe of the movie. To ensure a temporally stable stylization, we rely on optical flow to propagate the style to all other keyframes, recomputing the style transfer in regions where motion is unreliable. We call this process Temporal Style Propagation (TSP). Once all keyframes have been stylized, for each set of frames $U_{k,k+r}$, we propagate the style from the stylized left keyframe $\widetilde{u}_k$ in forward direction to all images u_t from $U_{k,k+r}$, then we propagate the style from $\widetilde{u}_{k+r}$ in backward direction and finally we blend the forward and backward stylized frames by linear interpolation. This stylization "by chunks" is repeated until the end of sequence, as summarized in Algorithm 1 and illustrated by Figure 5.

Algorithm 1 Video Style Transfer

Input: Video U, example style v, keyframe rate r, length D
Output: Stylized video $\widetilde{U}$
1: $k_0 \leftarrow \frac{D}{2}$
2: Split and Match Style Transfer (SMT) for keyframe u_{k_0}:
3: $\widetilde{u}_{k_0} \leftarrow SMT(u_{k_0}, v)$
4: $\widetilde{U} \leftarrow \{\widetilde{u}_{k_0}\}$
5: Set of keyframe indexes, starting from center:
6: $K \leftarrow \{k_0, k_0 + r, k_0 + 2r, ..., D\} \cup \{k_0 - r, k_0 - 2r, ..., 1\}$
7: Forward-backward Propagation between keyframes:
8: $\epsilon \leftarrow r$
9: $\widetilde{U} \leftarrow \widetilde{U} \cup$ FBP$(U, \widetilde{U}, \epsilon, K)$
10: Forward-backward Propagation for remaining frames:
11: $\epsilon \leftarrow 1$
12: $\widetilde{U} \leftarrow \widetilde{U} \cup$ FBP$(U, \widetilde{U}, \epsilon, K)$

Algorithm 2 FBP (Forward-backward propagation)

Input: $U, \widetilde{U}, \epsilon, K$
Output: $\{\widetilde{u}_i\}$, $i = \{1, 1 + \epsilon, 1 + 2\epsilon, ..., D\}$
1: **for** $k \in K$ **do**
2: Temporal Style Propagation in forward direction:
3: $\widetilde{U}^f_{k,k+r} \leftarrow$ TSP$(U_{k,k+r}, \widetilde{u}_k, k, \epsilon)$
4: Temporal Style Propagation in backward direction:
5: $\widehat{\epsilon} \leftarrow -\epsilon$
6: $\widetilde{U}^b_{k,k+r} \leftarrow$ TSP$(U_{k,k+r}, \widetilde{u}_{k+r}, k + r, \widehat{\epsilon})$
7: Forward-backward blending
8: $\widetilde{U}_{k,k+r} \leftarrow \alpha_t \widetilde{U}^f_{k,k+r} + (1 - \alpha_t) \, \widetilde{U}^b_{k,k+r}$
9: **end for**

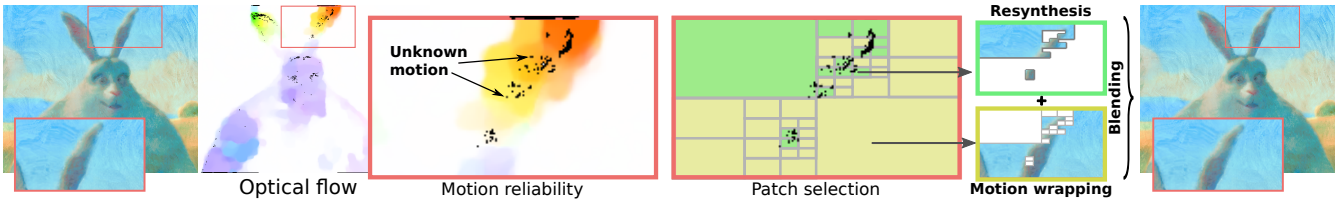

Fig. 4: Overview of our Temporal Style Propagation (TSP). Assuming that the frame $u_{t^\star}$ has been stylized, we show how to propagate the stylization to u_t ($t^\star = t-1$ when we propagate in the forward direction). On the left, the stylized frame $\widetilde{u}_{t^\star}$. **Motion warping** is applied to the stylized keyframe $\widetilde{u}_{t^\star}$ to obtain a first estimate of the stylized frame $\widetilde{u}_t$. Since motion is not reliable everywhere, a motion reliability map is also deduced (unreliable pixels are shown in black). To improve the first estimate of $\widetilde{u}_t$, we re-synthesize all regions containing unreliable pixels by solving an optimal labeling problem. The final stylized image $\widetilde{u}_t$ (on the right) is obtained by **blending** the re-synthesized regions with the motion warped estimate.

4.1 Temporal Style Propagation

We describe in this section the details of our Temporal Style Propagation, used first to ensure coherence between keyframes and then to propagate the style to all frames. In a nutshell, Temporal Style Propagation is a combination of optical flow warping and optimal patch labeling in regions where motion is unreliable. The whole propagation process is illustrated by Figure 4.

For the sake of simplicity, we assume in the following that a keyframe u_k has been stylized (the stylized version is written $\widetilde{u}_k$) and that we wish to propagate the style to all the frames u_t for t in the set $[k+1, k+r-1]$. Writing $t^\star = t-1$, style is propagated in the forward direction from both images $u_{t^\star}$ and u_k to u_t (if $t = k+1$, both images $u_{t^\star}$ and u_k are identical). The propagation in the backward direction follows the same path, using $t^* = t+1$. The propagation between two keyframes in the first part of the algorithm is similar, starting from the central keyframe stylized by Split an Match, and propagating first to subsequent frames and second to previous keyframes.

To estimate motion fields between successive frames, we rely on the recent *DeepFlow* algorithm [26], which combines a variational approach with descriptor matching to compute a dense offset map from a pair of images. The motion field between $u_{t^\star}$ and u_t is written as an offset map $\Delta_{t^\star,t}$ such that $u_{t^*}(x + \Delta_{t^\star,t}(x)) \simeq u_t(x)$ everywhere. We denote the warping of $u_{t^\star}$ by

$$u_t^w := W(u_{t^\star}, \Delta_{t^\star,t}) = u_{t^*}(. + \Delta_{t^\star,t}(.)),$$

where u_{t^*} is interpolated by bicubic interpolation on non integer coordinates. Optical flow between non successive frames (for instance two keyframes) is obtained by composition of successive pairwise optical flows.

A first estimate of the stylized frame $\widetilde{u}_t$ is given by $\widetilde{u}_t^w := W(\widetilde{u}_{t^\star}, \Delta_{t^\star,t})$. Obviously, warping the previously stylized frame $\widetilde{u}_{t^\star}$ by optical flow encourages temporal coherence between $\widetilde{u}_t$ and $\widetilde{u}_{t^\star}$, but optical flow estimation is not always reliable, and motion vectors remain unknown in some areas (occlusions, domain boundaries). To address these limitations, we compute a reliability map, defined as a combination of an occlusion map and a motion accuracy map. The motion accuracy map is defined as the set of pixels such that the absolute error between the original frame and the motion warped frame is larger than a threshold τ_e (set in practice to 25)

$$A_t(\boldsymbol{x}) = \begin{cases} 1, & \text{if } |u_t(\boldsymbol{x}) - u_t^w(\boldsymbol{x})| < \tau_e, \\ 0, & \text{otherwise.} \end{cases}$$

According to [27], a simple and effective approach to compute occlusions is to take the residual between the forward and the backward optical flow:

$$O_t(x,y) = \begin{cases} 1, & \text{if } |\Delta_{t^\star,t}(\boldsymbol{x}) + \Delta_{t,t^\star}(\boldsymbol{x})|^2 < \tau_m, \\ 0, & \text{otherwise,} \end{cases}$$

where $\tau_m := 0.01(|\Delta_{t^\star,t}|^2 + |\Delta_{t+1,t}|^2) + 0.05$. Thus, the set of coordinates where optical flow is not reliable and where style transfer needs to be resynthesized is given by

$$\chi_t = \left\{ (x,y) \in \Omega \;\middle|\; A_t(x,y)O_t(x,y) = 0 \right\}. \tag{6}$$

In order to re-synthesize style on χ_t, we start by computing a quadtree partition $R = \{R_i\}_{i=1}^n$ for u_t, with a variance only stopping criteria (threshold set to 15 in practice) for the quadtree splitting. Next, we divide this partition in two sets $R = R^{'} \cup R^{''}$, where

$$R^{'} = \{R_i \in R';\ \chi_t \cap R_i \neq \emptyset\}$$

is the set of patches to be relabeled and $R^{''}$ the set of patches considered as already stylized. Temporal style

propagation on R' can be posed as a supervised style transfer, in the spirit of [10]. Indeed, we know both the keyframe u_k and its stylized version $\widetilde{u}_k$, and we can take advantage of this knowledge to compute $\widetilde{u}_t$. The core idea is that if two patches $u_t(R_i)$ and $u_k(R)$ are similar in structure, then the patch $\widetilde{u}_t(R_i)$ should be similar in style to $\widetilde{u}_k(R)$. Since u_t and u_k are expected to have considerable overlapping content, it also turns out to be much easier to find patch correspondences between them than between u_t and the exemple image v. Consequently, for each region R_i in R', a set of M candidate labels $L_i = \{l_{i_m}\}_{m=1}^{M}$ is selected by computing the M-nearest neighbors of the patch $u_t(R_i)$ in u_k.

Next, we search for the optimal set of label assignments $\widehat{L} = \{\widehat{l}_i\}_{i=1}^{\#R'}$ (with $\#R'$ the size of R') minimizing the energy

$$E(L) = \lambda_d E_d(L) + \lambda_s E_s(L) + \lambda_t E_t(L), \tag{7}$$

where E_d is a data fidelity term, E_s is a spatial smoothness term, E_t a temporal coherence term and $\lambda_d, \lambda_s, \lambda_t$ the respective weights of each term. These three terms are defined as follows:

1. The **data fidelity** term is given by

$$E_d(L) = \sum_{i=1}^{\#R'} d[u_k(R_{l_i}), u_t(R'_i)]. \tag{8}$$

   It can be noted that this data term differs from the one used for single image style transfer, since it compares patches of the non-stylized images u_t and u_k.
2. The **spatial smoothness** term is defined as

$$E_s(L) = \sum_{(i,j)\in\mathcal{N}} d_{\cap}[\widetilde{u}_k(\widetilde{R_{l_i}}), \widetilde{u}_k(\widetilde{R_{l_j}})]. \tag{9}$$

   to encourage smooth transitions between stylized patches (we remind that the notation $\widetilde{R}$ denotes extended regions, see Section 3).
3. The **temporal coherence** term is given by

$$E_t(L) = \sum_{i=1}^{\#R'} d[u_t^c(R_i), \widetilde{u}_k(R_{l_i})], \tag{10}$$

   where u_t^c is a combination of $\widetilde{u}_t^w$ (stylized frame $\widetilde{u}_{t^\star}$ warped by optical flow) and u_t, defined as

$$u_t^c := [(1 - \mathbf{1}_{\chi_t}) \odot \widetilde{u}_{t^\star}^w] + [\mathbf{1}_{\chi_t} \odot u_t]. \tag{11}$$

   This temporal coherence cost favors intensities of a stylized patch at time t to remain similar to the warped intensities from $t^\star$ for all coordinates where optical flow is reliable. This temporal coherence is already guaranteed for patches contained in R''.

Once the optimal set $\widehat{L}$ has been estimated by Loopy Belief Propagation, we use the bilinear patch blending described in Section 3 to obtain an image $\widetilde{u}_t^l$. The final reconstructed image is

$$\widetilde{u}_t := G_t \odot \widetilde{u}_t^w + (1 - G_t) \odot \widetilde{u}_t^l, \tag{12}$$

where $G_t := G^\sigma * (1 - \mathbf{1}_{\chi_t})$ is the optical flow reliability map convolved by a gaussian kernel G^σ with standard deviation σ, for spatial blending between the motion warped image $\widetilde{u}_t^w$ and the labeled image $\widetilde{u}_t^l$. Temporal style propagation is summarized in Algorithm 3.

Algorithm 3 TSP (Temporal Style Propagation)

Input: $(U_{k,k+r}, \widetilde{u}_k, k, \epsilon)$
Output: $\widetilde{U}_{k,k+r}$

$t \leftarrow k + \epsilon$
while $t \neq k + r$ **do**
  $t^\star \leftarrow t - \epsilon$
  $\Delta_{t^\star,t} \leftarrow$ optical flow between $u_{t^\star}$ and u_t

  Warp $u_{t^\star}$ and $\widetilde{u}_{t^\star}$:
  $u_t^w \leftarrow W(u_{t^\star}, \Delta_{t^\star,t})$
  $\widetilde{u}_t^w \leftarrow W(\widetilde{u}_{t^\star}, \Delta_{t^\star,t})$

  Compute the motion accuracy map
  $A_t(\boldsymbol{x}) \leftarrow |u_t(\boldsymbol{x}) - u_t^w(\boldsymbol{x})| < \tau_e$
  Compute the occlusion map
  $O_t(\boldsymbol{x}) \leftarrow |\Delta_{t^\star,t}(\boldsymbol{x}) + \Delta_{t,t^\star}(\boldsymbol{x})|^2 < \tau_m$
  Compute the set of coordinates to be resynthesized
  $\chi_t \leftarrow \left\{ (\boldsymbol{x}) \in \Omega \,\middle|\, O_t(\boldsymbol{x}) A_t(\boldsymbol{x}) = 0 \right\}$

  Select the set of patches R' to be labelled by style transfer:
  $R' \leftarrow \{\}$
  for every region $R_i \in R$ **do**
    if $R_i \cap \chi_t \neq \emptyset$ **then**
      $R' \leftarrow R' \cup \{R_i\}$
    end if
  end for
  Compute M-nearest neighbors:
  $L_i \leftarrow L_i = \{l_{i_m}\}_{m=1}^{M}$ with $|l_{i_m} - l_{i_{m+1}}| > \chi$
  Compute optimal labels $\widehat{L}$ for patches in R':
  $\widehat{L} \leftarrow \min E(L)$
  Compute $\widetilde{u}_t^l$
  $\widetilde{u}_t \leftarrow \widetilde{u}_t^l$ blended with $\widetilde{u}_t^w$

  $t \leftarrow t + \epsilon$
end while

4.2 Forward-backward blending

Temporal style propagation is applied for all sets of frames $\{u_{k+1}, u_{k+2}, ..., u_{k+r-1}\}$ delimited by a "left keyframe" u_k and "right keyframe" u_{k+r}, both in a forward and in a backward pass. We denote the stylized sequence

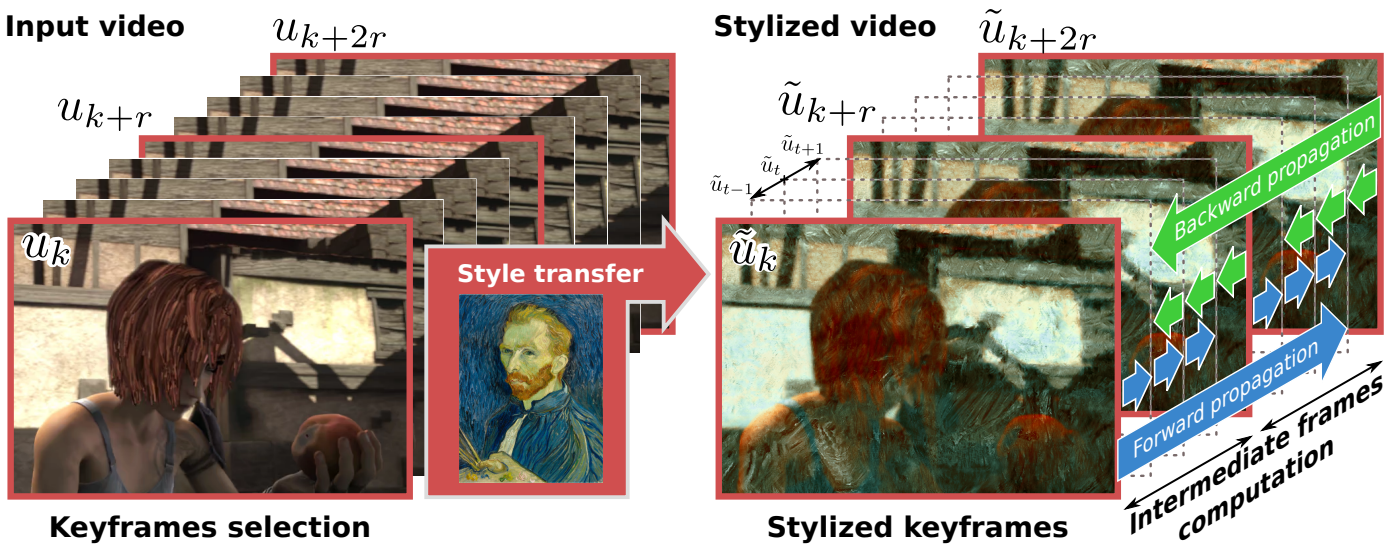

Fig. 5: **Temporal Style Propagation**. On the left, original frames, and on the right, stylized frames. Once all keyframes have been stylized, for each "chunk" of frames between two keyframes u_k and u_{k+r}, we propagate the style from the stylized left keyframe $\widetilde{u}_k$ in forward direction to all images u_t, then we propagate the style from $\widetilde{u}_{k+r}$ in backward direction and finally we blend the forward and backward stylized frames by linear interpolation. This stylization "by chunks" is repeated until the end of sequence.

resulting from forward style propagation by $\widetilde{U}^f_{k,k+r} = \{\widetilde{u}^f_t\}_{t=k+1}^{k+r-1}$ and the sequence resulting from backward propagation as $\widetilde{U}^b_{k,k+r} = \{\widetilde{u}^b_t\}_{t=k+1}^{k+r-1}$.

We can finally compute $\widetilde{u}_t$ as a blend of the forward and backward pass

$$\widetilde{u}_t = \alpha_t \widetilde{u}^f_t + (1-\alpha_t)\widetilde{u}^b_t, \tag{13}$$

where α_t is the linear temporal weighting factor given by

$$\alpha_t = \frac{k+r-t}{k+r-k}. \tag{14}$$

5 Experiments

In this section, we provide some results obtained with the style transfer method described in this paper. In Fig. 6 we present an experiment with our style transfer method applied to two different still images with the styles of Signac and Seurat. Remaining experiments concern our method applied to videos. Video results can be found at our project website: `http://oriel.github.io/video_style_transfer.html`.

The results in Fig. 7 and Fig. 8 were obtained with sequences taken from the Sintel dataset [28], which includes ground truth optical flow and occlusions. Thus, for these results, we use the optical flow and occlusion maps provided with the dataset.

In Fig. 9 we illustrate the interest of a temporally coherent video style transfer in comparison to the application of an image style transfer method in a frame-by-frame manner. It can be noted in Fig. 9, (in particular on the highlighted red and blue rectangles) that our method guarantees temporal coherency of style, while a stylization performed independently for every frame unsurprisingly results in severe texture variation.

Finally, in Fig. 11, we compare our method to the recent work [22] on a real sequence. In Fig 10, we present the data that were used to compare our method to [22]. The top row shows frames of the original sequence, while the second row shows the various style images that were used. Results are then shown in Fig. 11.

For [22], we used a GPU GTX980 with $4G$ memory. To fit in memory, the video had to be downsampled to a resolution 440×400, and the computation took several hours. Our method runs on CPU and the computation took 45 minutes.

It must first be stated that quantitative evaluation is impossible since there is no ground truth, and different artistic choices can be rated differently by users. The neural method consists in matching statistics over the layers of a deep pre-trained convolutional network. It has been known that some of these layers corresponds to Gabor filters. For this reason, results of [22] exhibit a better stylization of contours. However, we think that our results preserve better the color tones

Fig. 6: Results of our still image style transfer method with Signac's and Seurat's paintings as examples. Top row: example images, Left column: original images.

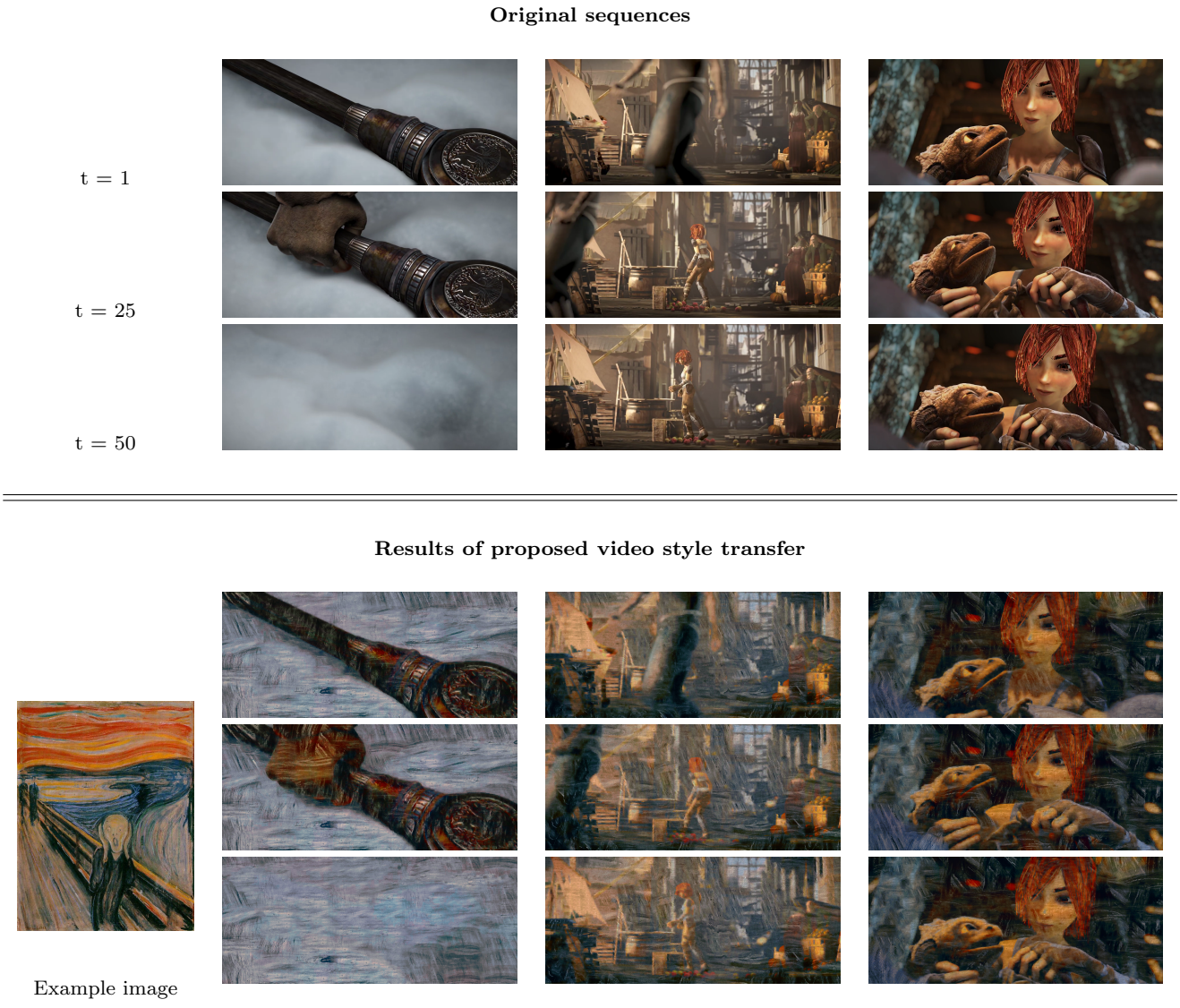

Fig. 7: Video style transfer results for three different sequences from Sintel [28] with the painting "Cry" by Edvard Munch as example image. It can be noted that the stylized frames resulting from our video style transfer method have temporally coherent style.

of the style image, while being a possible and different artistic choice.

6 Conclusion

This paper has proposed a new approach for video style transfer, in which temporal style propagation is used to obtain temporal coherence between stylized frames. We have seen that the extension of style transfer from images to videos is not straightforward, as a stylization in a frame-by-frame basis is very likely to result in flickering. Thus, we proposed a technique that guarantees a temporally coherent stylization by propagating the style from keyframes. For that, we relied in a combination of optical flow warping and style resynthesis. Our results show that such a technique is well adapted for stylization of videos. However, it should be noted that the quality of video stylization is strongly dependent on the accuracy of optical flow estimation. Hence, our video style transfer would clearly benefit from advances in the field of optical flow estimation.

Furthermore, we have shown that decomposing input images into an adaptive patch partition is an efficient approach not only for image style transfer, but also for the extension of style transfer to videos. In this sense, our results suggest, together with [19], that adaptive patches are useful to obtain a completely unsupervised style transfer in which no previous learning

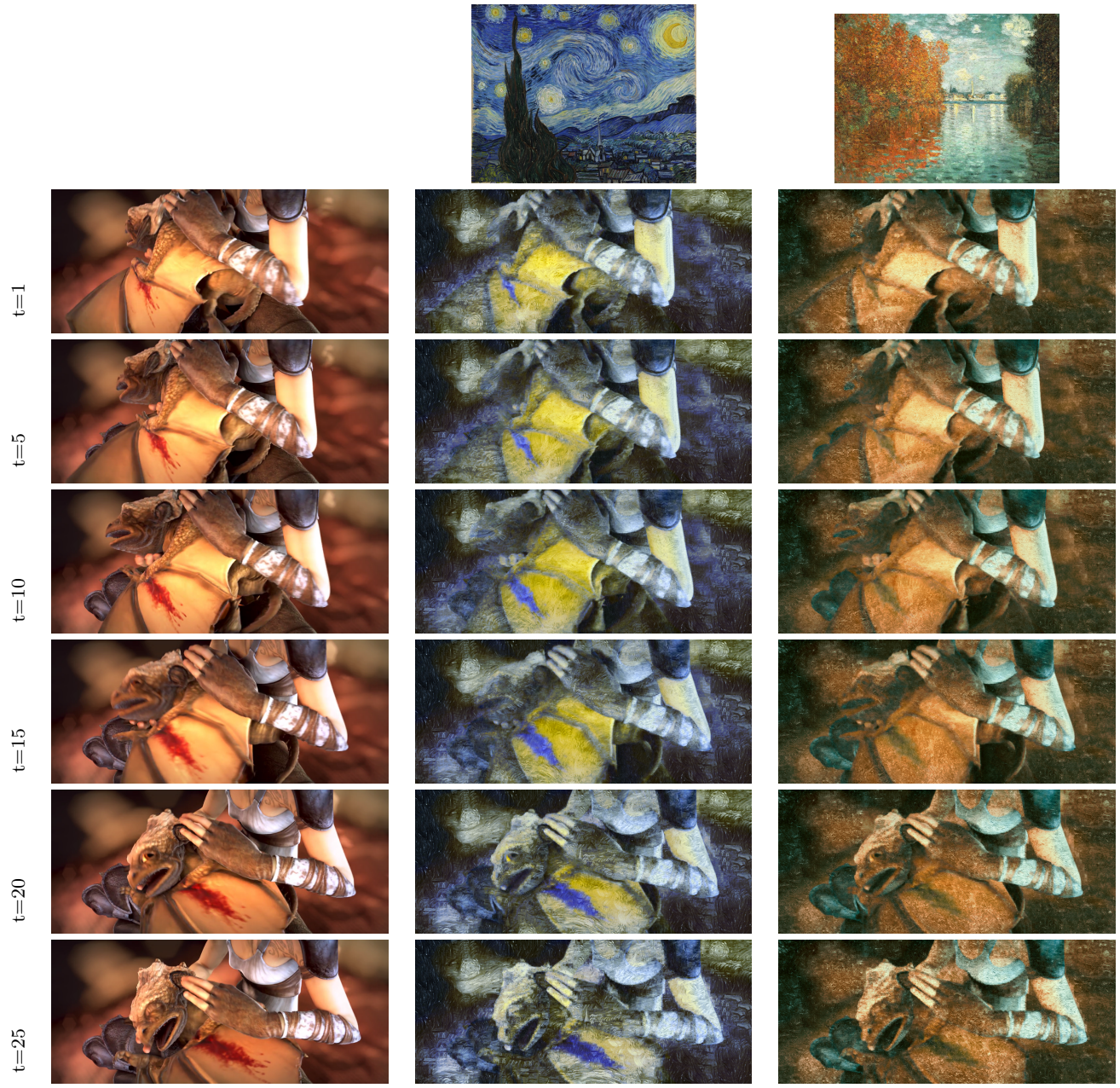

Fig. 8: Video style transfer results for Sintel synthetic sequence [28] and two different example styles. Top row: example images, Left column: original images. It can be noted that the stylized frames resulting from our video style transfer method have temporally coherent style.

is required, differently to the neural network approach [22]. Another clear advantage of our approach is the reduced computational complexity: while our method needed less than a hour to stylize the video of Fig. 11 using the CPU, the neural network approach needed several hours of processing in a GPU.

References

1. F. Durand, "An invitation to discuss computer depiction," in *NPAR*, New York, NY, USA, 2002, pp. 111–124. [Online]. Available: http://doi.acm.org/10.1145/508530.508550
2. J. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the art: A taxonomy of artistic stylization techniques for images and video," *IEEE TVCG*, vol. 19, no. 5, pp. 866–885, May 2013.
3. L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *CoRR*, vol. abs/1508.06576,

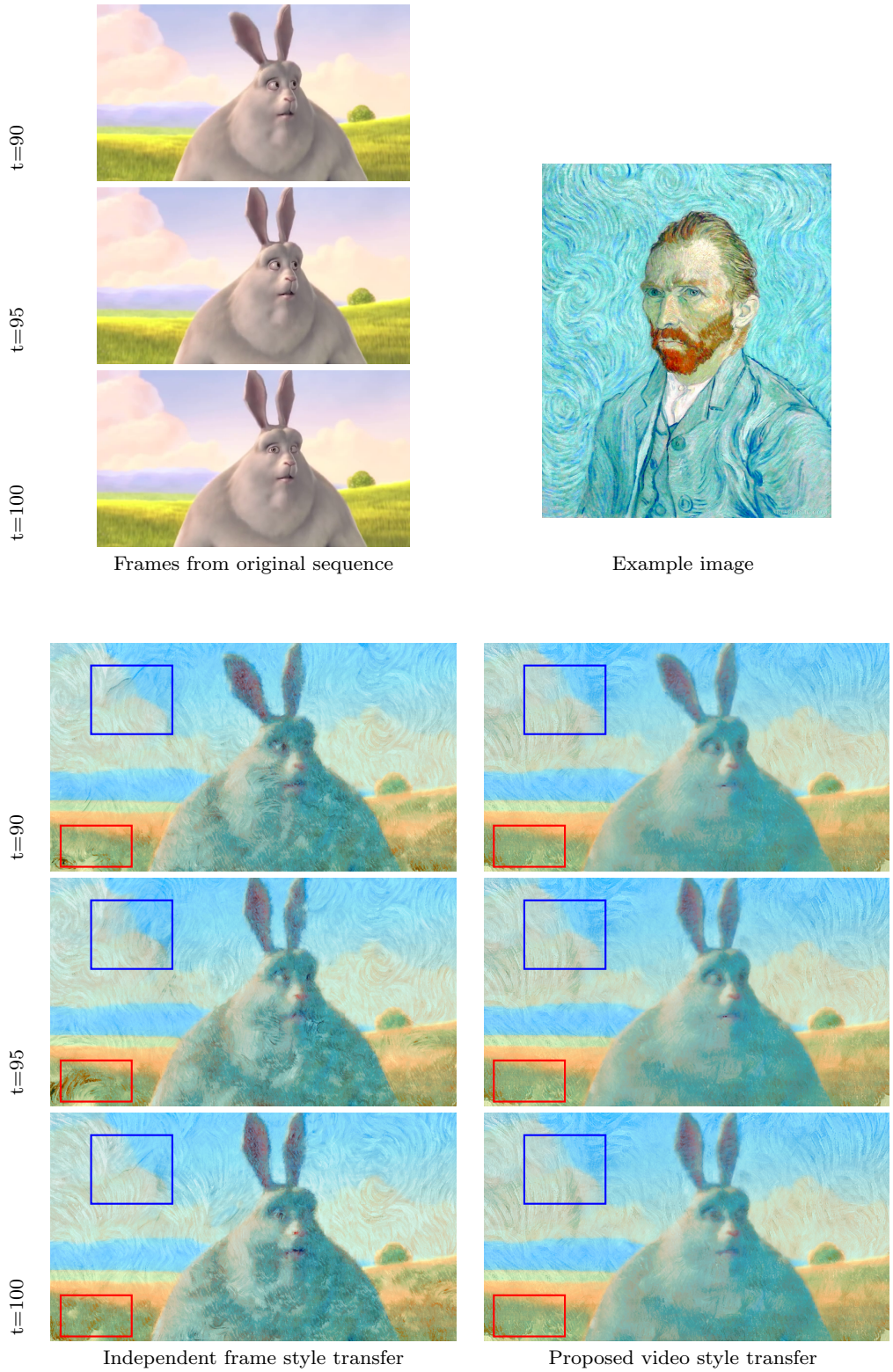

Fig. 9: Illustration of our temporally coherent video style transfer. The top row shows the middle frame of the original sequence, as well as the example style image (Van Gogh). Next rows show the results of the style transfer, using the frame-independent technique of [6] (left column) and the presented algorithm (right column). It can be noted that the stylized frames generated with frame-by-frame stylization, suffer from texture flicker. The brush strokes highlighted by the red and blue rectangles change abruptly from a frame to the next. On the right column, we show the stylized frames resulting from our video style transfer method, which has temporally coherent style.

Frames from the original sequence

Canaletto sketch style

Van gogh style

Examples

Fig. 10: Data used for the real sequence comparison. The top row shows the frames of the original sequence, while the second ow shows the style images.

2015. [Online]. Available: http://arxiv.org/abs/1508.06576

4. C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 702–716.
5. V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *ICLR*, 2017.
6. O. Frigo, N. Sabater, J. Delon, and P. Hellier, "Split and match: Example-based adaptive patch sampling for unsupervised style transfer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
7. A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH*, New York, NY, USA, 2001, pp. 341–346. [Online]. Available: http://doi.acm.org/10.1145/383259.383296
8. A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE T-IP*, vol. 13, no. 9, pp. 1200–1212, Sept 2004.
9. W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar 2002.
10. A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *SIGGRAPH*, New York, NY, USA, 2001, pp. 327–340. [Online]. Available: http://doi.acm.org/10.1145/383259.383295
11. E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 34–41, Sep. 2001. [Online]. Available: http://dx.doi.org/10.1109/38.946629
12. A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *ICCV*, Washington, DC, USA, 1999, pp. 1033–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851569
13. P. Bénard, F. Cole, M. Kass, I. Mordatch, J. Hegarty, M. S. Senn, K. Fleischer, D. Pesare, and K. Breeden, "Stylizing animation by example," *ACM TOG*, vol. 32, no. 4, pp. 119:1–119:12, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2461929
14. C. Barnes, E. Shechtman, D. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," in *Computer Vision ECCV 2010*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6313, pp. 29–43. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15558-1_3
15. L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, "Real-time texture synthesis by patch-based sampling," *ACM Trans. Graph.*, vol. 20, no. 3, pp. 127–150, Jul. 2001. [Online]. Available: http://doi.acm.org/10.1145/501786.501787
16. W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *IJCV*, vol. 40, no. 1, pp. 25–47, 2000.
17. X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE T-PAMI*, vol. 31, no. 11, pp. 1955–1967, 2009. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.222
18. Y. Weiss, "Belief propagation and revision in networks with loops," Cambridge, MA, USA, Tech. Rep., 1997.
19. M. Elad and P. Milanfar, "Style-transfer via texture-synthesis," *CoRR*, vol. abs/1609.03057, 2016. [Online]. Available: http://arxiv.org/abs/1609.03057
20. D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: http://arxiv.org/abs/1607.08022
21. B. J. Joshi, K. Stewart, and D. Shapiro, "Bringing impressionism to life with neural style transfer in

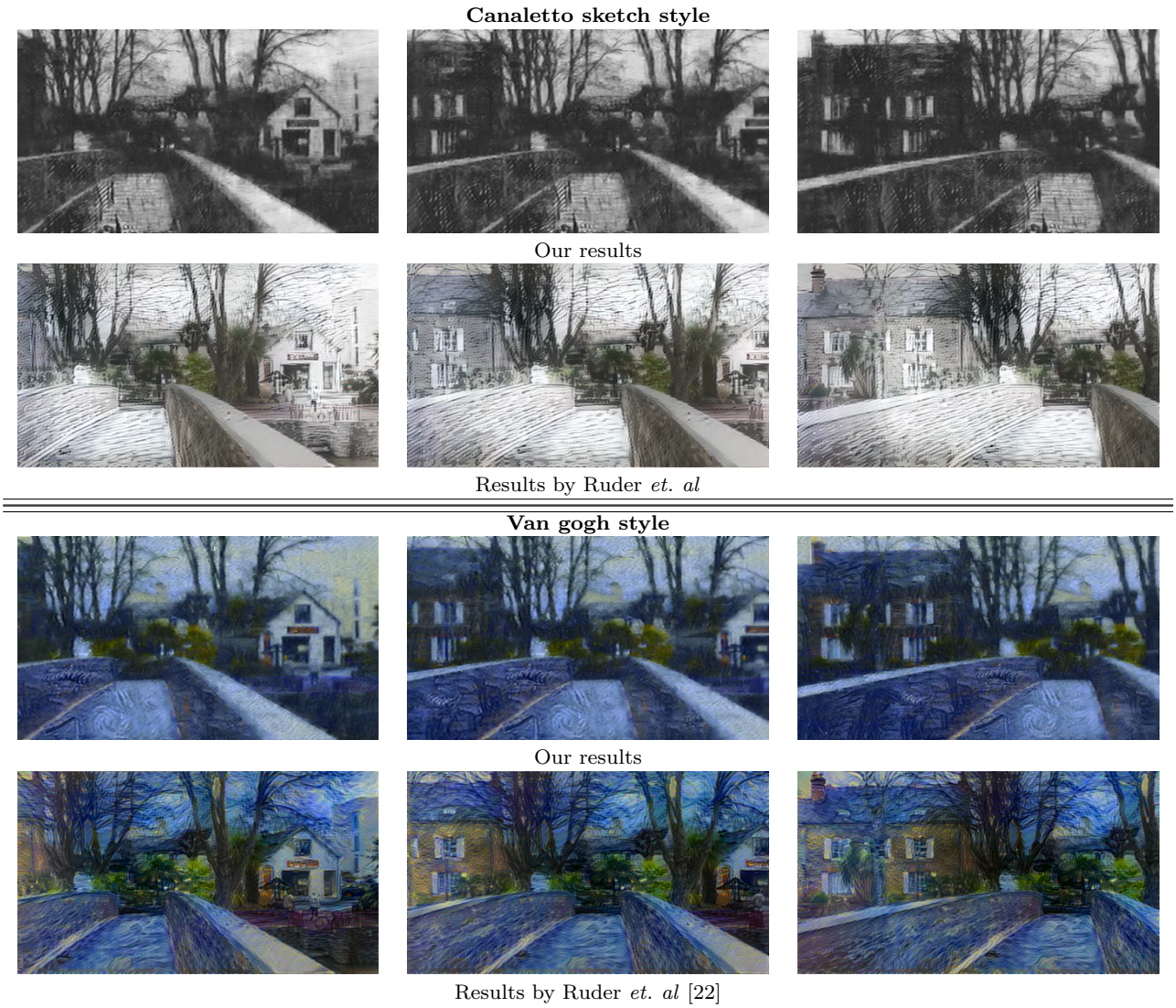

Fig. 11: Real sequence results, where we compare out method to the neural artistic style transfer for videos [22]. For each style image, the first row corresponds to our method and the second row is [22]. Result comparison may be subjective and therefore challenging. Nevertheless, for the sketch style, it can be noted that our method stylize consistently the two walls of the bridge, while in the result of [22], the left wall is brighter and the right wall is darker. As remarked by [23], stylization by neural networks may be unpredictable at times.

come swim," *CoRR*, vol. abs/1701.04928, 2017. [Online]. Available: http://arxiv.org/abs/1701.04928

22. M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," *CoRR*, vol. abs/1604.08610, 2016. [Online]. Available: http://arxiv.org/abs/1604.08610
23. J. Fišer, O. Jamriška, M. Lukáč, E. Shechtman, P. Asente, J. Lu, and D. Sýkora, "Stylit: Illumination-guided example-based stylization of 3d renderings," *ACM Transactions on Graphics*, vol. 35, no. 4, 2016.
24. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.
25. O. Frigo, N. Sabater, V. Demoulin, and P. Hellier, "Optimal transportation for example-guided color transfer," in *ACCV*, 2014, pp. 655–670.
26. P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013. [Online]. Available: http://hal.inria.fr/hal-00873592
27. N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science. Springer, Sept. 2010. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2010/Bro10e
28. D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.