Learned Variational Video Color Propagation
Supplementary material – ECCV 2022

Markus Hofinger, Erich Kobler
Alexander Effland, and Thomas Pock

A Implementation Details

Inference Time. The run-time of our method depends on the number of used color proposals. Fig. 14 shows that already 3 proposals yield almost maximal PSNR scores (ours fast). Our unoptimzed PyTorch code takes approximately 1.75 seconds per frame (DAVIS, 854x480) on a NVIDIA Titan RTX. Approximately 1.2 seconds are spent in color proposal generation (fast = 3x local + 3x global proposals + 12 RAFT iterations) and roughly 0.55 seconds are spent in our unrolled refinement optimization (12 iterations) with our learned regularizer. Thus, our model takes roughly 4.3 s/MPixel, which is slightly faster than the 5.4 s/MPixel reported by DVCP [11]. With an additional CUDA sampling operator (not in main paper but on github), the inference can be improved from 1.75 to approximately 1 seconds with similar qualitative results. Since the feature matching and motion estimation can be precomputed once for a whole sequence, the 0.55 seconds become the relevant part if our model is used by a human operator in an interactive way. Page 5 provides another example of how user input can be easily integrated into our model after training.

$PSNR_{ab}$. In literature, evaluation metrics are computed quite differently, e.g. different color spaces are considered or not mentioned at all [7,11], or some report averages over the first t time steps [11] vs. reporting at time step t [15], and there are also non-negligible implementation differences, which unfortunately prevent direct comparisons of values reported across papers. We therefore identically re-evaluated all models from their public sources on the all datasets described in the main paper. For each time step t we compute PSNR_{ab}– the PSNR of the chrominance ab channels of the CIE-Lab space, as luminance is kept fixed. Since the ab-space Ω^{ab}contains negative values, one needs to take these into account when computing the maximum signal range. This leads to the PSNR_{ab} formula

$$\text{PSNR}_{ab}(I_1, I_2) = 10 \log_{10}\left(\text{diam}(\Omega^{ab})^2\right) - 10 \log_{10}\left(\tfrac{1}{2N_p}\|I_1^{ab} - I_2^{ab}\|_2^2\right), \quad (14)$$

where we compute $\text{diam}(\Omega^{ab}) \coloneqq \max\{\max_{x,y\in\Omega^a}\|x-y\|, \max_{x,y\in\Omega^b}\|x-y\|\} \approx 202.3354$ as the maximal diameter of the Ω^{ab}-colorspace for converted 3x8Bit-PNG images. Furthermore, we want to raise awareness, that using uint8 for the non-linear Lab conversion leads to a reduction of the unique ∼16.7 Mio RGB colors to ∼2.1 Mio Lab colors, which affects PSNR computation. Thus, floating point datatypes are required for PSNR in CIE-Lab space.

Occlusion estimation for PSNR$_{ab}$occl. We estimate occlusions masks M_o between frames using the heuristic of UnFlow [10, Eq. 1]

$$M_o = \left| m_M^f + \text{warp}\left(m_M^b, m_M^f\right)\right|^2 < 0.01\left(\left|m_M^f\right|^2 + \left|\text{warp}\left(m_M^b, m_M^f\right)\right|^2\right) + 0.5 \tag{15}$$

based on the optical flow (using RAFT [14] with 20 iterations) between the groundtruth color frames in forward m_M^f and backward m_M^b direction. Since the occluded regions accumulate over time, we also accumulate the motion compensated (warped) occlusion masks with the current occlusion masks, performing a pixelwise logic or operation $\oplus$, i.e.

$$\widetilde{M}_o^t = M_o^t \oplus \text{warp}\left(\widetilde{M}_o^{t-1}, m_M^b\right). \tag{16}$$

As can be seen in Fig. 13, the accumulated occlusion masks $\widetilde{M}_o^t$ represent image regions that have been occluded over time. Then, the PSNR$_{ab}$ for occluded pixels is computed by considering the areas defined by the accumulated occlusion mask $\widetilde{M}_o^t$. Note that we exclude frames with 0 occluded pixels, where the PSNR$_{ab}$ is undefined.

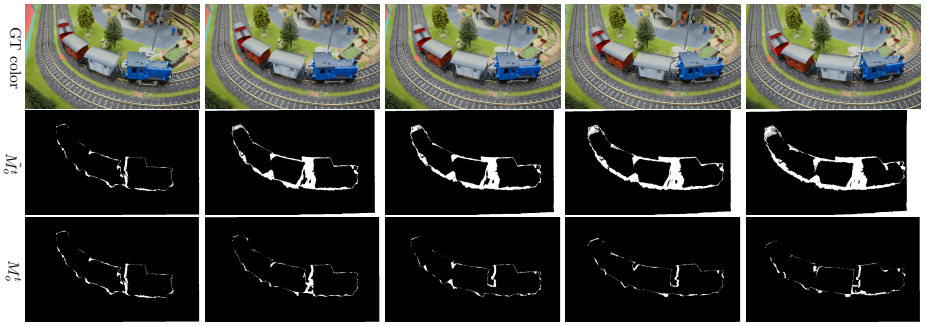

Fig. 13. Example of the occlusion estimation process used for the computation of PSNR$_{ab}$ occl. Here, M_o^t shows occlusions between frames and $\widetilde{M}_o^t$ their accumulation.

TDV and $\mathcal{W}$- structure and details For the TDV, we use one macroblock, 3 scales, 32 feature channels, and tanh non-linearities as we also use for $\mathcal{W}$. However, for the TDV we additionally use the standard zero-mean constraints and student-t as final activation function for the energy.

To ensure the pixel-wise constraints for the fusion weights $\{\mathrm{u}_G, \mathrm{u}_L, \mathrm{u}_M, \mathrm{u}_0\}$, as well as the weights steering the refinement $\{\mathrm{v}_G, \mathrm{v}_L, \mathrm{v}_M, \mathrm{v}_0\}$, we use a pixelwise softmax for both groups, where v_0 and u_0 are additional pixelwise weights, allowing the sum of the other masks (G,L,M) to be ≤ 1. For $\mathrm{v}_{\mathcal{R}}$ we use a sigmoid

function. Hence, WeightNet $\mathcal{W}$ can independently adjust the initial fusion, as well as the dataterms, and the regularization strength form $0 - 100\%$, and also shift the balance between each type (G,L,M) based on the image data.

Training For training, we follow [12] using the ADAM optimizer [8] with $\beta_1 = 0.5, \beta_2 = 0.9$ for 400 epochs and an initial learning rate of 10^{-4}, halving it after 100, 150, 200, 250 and 300 epochs. To account for the different distribution of gradient norms between the convolution kernel parameters, biases, and the scalar parameters, i.e. step size τ or balancing weight λ), we used a 500 times larger learning rate for the scalar parameters.

Background on the historic scene data The historic Western scene is from the movie 'Go West'[2] by Buster Keaton from 1925, and hence already in public domain, as also stated on the official homepage[1]. The theater sequence from ARRI [1] is dated even earlier (1902) and also in the public domain. We hereby express our gratitude to the data providers.

B Further ablations

Ablation of Backbones. Prior to the development of the full method we ran ablations on various pre-trained CNN backbones using an early variant of our global matching algorithm, as can be seen in Table 1.

Inst.Norm	Levels	Task	Backbone	frame:	1.0	5.0	10.0	15.0	20.0	24.0
			Gray baseline		25.24	25.30	25.40	25.36	25.54	25.53
✗	4	Class.	ResNet101 [4]		32.32	28.92	28.16	27.77	27.64	27.49
✗	4	Flow	ResNet (from RAFT [14])		38.14	33.26	31.59	30.39	29.70	29.20
✗	4	Segm.	ResNet101 (from DeeplabV3 [3])		37.92	32.86	31.50	30.42	29.93	29.67
✗	4	Class.	VGG16 [13]		39.47	34.23	32.96	31.70	31.17	30.79
✗	4	Class.	VGG16BN [13]		40.48	35.31	33.62	32.65	31.97	31.33
✓	4	Class.	ResNet101 [4]		32.17	28.82	28.00	27.68	27.53	27.42
✓	4	Flow	ResNet (from RAFT [14])		38.42	33.53	31.80	30.59	29.93	29.39
✓	4	Segm.	ResNet101 (from DeeplabV3 [3])		38.56	33.48	32.10	31.12	30.66	30.28
✓	2	Class.	VGG16 (2Levels) [13]		40.06	34.55	33.30	31.99	31.44	31.01
✓	4	Class.	VGG16 [13]		40.77	35.37	33.81	32.82	32.24	31.56
✓	4	Class.	VGG16BN [13]		**41.18**	**35.80**	**34.03**	**33.12**	**32.42**	**31.72**

Table 1. Performance of various pre-trained backbones for global matching, measured in PSNR_{ab} over the DAVIS evaluation sequences.

For this evaluation, we computed a global matching with 8 proposals per pixel and reduced them to a single best matching position $\hat{m}_G^t$ per pixel according to the extracted confidences. We then extracted the *ab* colors from the best

[1] https://www.busterkeaton.org/gowest

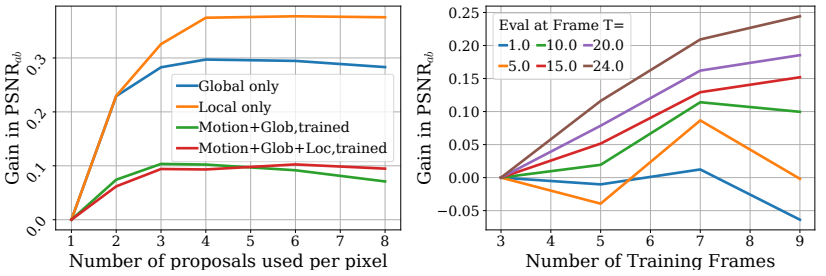

Fig. 14. Left: PSNR$_{ab}$ gain vs. proposal count; Right: PSNR$_{ab}$ gain vs. training frames

match per pixel, yielding a best color estimate $\hat{c}_G^t$, which we compared against the ground truth. Table 1 shows the result of this evaluation, conducted over the validation dataset described in the main paper. Surprisingly, the simple VGG16 architecture trained with batch normalization on ImageNet classification outperforms the more recent ResNet architecture, even when using a ResNet backbone trained for motion estimation RAFT [14].

For VGG16 we extracted the features right after the batchnorm layers, which corresponds to the layers 1, 8, 15 and 25. These are basically the batchnorm layers after the first convolution layers per resolution level. Additionally, we added one experiment, where we did not refine iteratively on each of the 4 levels but just once (2-Levels). For the VGG16 version without batchnorm, we again use the output of the first convolutions per resolution level, but this time directly at the convolution layer since there is no batchnorm layer (layers 0, 5, 10, 17).

For ResNet101-type networks, we also used the outputs of the first batchnorm layers after downsampling. Additionally, we recompute the first conv layer with a stride of 1 to also get high-resolution input features at full resolution.

Ablation on 'color proposal count'. The left side of Fig. 14 shows the average gain in PSNR$_{ab}$ when changing the number of color proposals N_L, N_G, per pixel and for local and global proposal types. The gain is computed over the whole evaluation dataset for 24 frames. On average, the gain is roughly 0.3dB when using untrained best local or global color proposals. But even after fusion, refinement and training an average gain of 0.1dB remains when using at least three proposals per pixel for both (local and global) color proposal types.

Ablation on 'training frame count'. The right side of Fig. 14 shows the gain in PSNR$_{ab}$ vs. the amount of frame propagation augmentation N_A used during training. The PSNR$_{ab}$ gain for later frames increases with higher amount of training frame propagations. This confirms our qualitative findings that results appear improved when more accumulated errors are also present during training.

Ablation on TDV iterations. Fig. 15 shows the improvements on PSNR$_{ab}$ vs. different amounts of refinement iterations, when varying the number of refinement steps after training. The diagram on the left shows, that performance improves

Fig. 15. Gain in PSNR$_{ab}$ v.s. number of TDV iterations on DAVIS-2017-val.

with more iterations, but saturates at around 8-12 iterations. The curves are shown for different frames, revealing more PSNR$_{ab}$ gain for earlier frames. The images on the right provide a qualitative comparison for the same setup.

C Example of interpretable results with user interaction.

Fig. 16 gives an example on how our method allows to integrate user interactions on the example of overriding the initial fusion decisions. The top row shows the

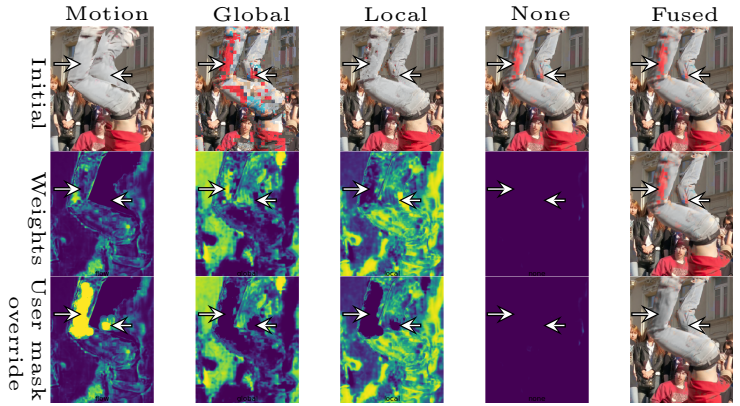

Fig. 16. Example of possible integration of user interaction for a particular hard case. Our method allows to override the weight masks (here shown for the initial fusion masks u) to correct hard cases, even before the final refinement.

best color proposal per type for a very hard case for demonstration, as indicated by the arrows. Due to large motion the appearance of the trousers changes so much that parts of it now better match to the red pullover, which has a similar grayscale appearance. Furthermore, even though the motion's color proposal would provide correct colors for that region, the motion is so large and complex that $\mathcal{W}$ gets confused and trusts more on the colors from the local and global color proposals. While such stain type artifacts also can happen in methods

like DVCP if appearance similarity is misleading, we can show where each color proposal comes from, allowing a user – if desired – to override a bad proposal fusion already before our TDV refinement. Note that the weight masks U, which simply represent the percentage of how much each proposal is used on a per pixel level, is shown underneath the color proposal images.

The *bottom row* of Fig. 16, shows the result after an user interaction with a simple GUI. This result was generated by letting a user draw strokes on the motion mask (overlaid by the according proposal image). To keep the constraints on the masks, we performed a simplex projection after each user input.

Such types of insights, and hence options to override network decisions are also possible for other parts of the network, such as the dataterm and regularization masks. Therefore our method can be used to fully automatic propagate colors as well as integrated into tools with user interaction.

D Derivation of the proximal mapping for the multi-well dataterm.

In what follows, we derive the proximal map for the multi-well dataterm. Let $\mathrm{C}_{\gamma,p} = \left\{c^n_{\gamma,p}\right\}_{n=1}^{N_\gamma}$ be a set of N_γ plausible color proposals per pixel p, of an arbitrary type γ, and let $\mathcal{D}$ be a multi-well dataterm which is given by

$$\mathcal{D}(x, \mathrm{C}_\gamma) = \lambda_\gamma \sum_{p=1}^{N_P} \mathrm{v}_{\gamma,p} \min_{\tilde{c}_{\gamma,p} \in \left\{c^n_{\gamma,p}\right\}_{n=1}^{N_\gamma}} \frac{1}{2}\|x - \tilde{c}_{\gamma,p}\|_2^2, \tag{17}$$

where λ_γ is a non-negative scalar and $\mathrm{v}_{\gamma,p}$ a pixelwise non-negative mask. Then the proximal map for the dataterm is given by

$$\mathrm{prox}_{\tau\mathcal{D}(\,\cdot\,,\mathrm{C}_\gamma))}\left(\bar{x}^i\right) = \frac{\bar{x}^i + \tau\lambda_\gamma \mathrm{v}_\gamma \odot \tilde{c}^i_\gamma}{1 + \tau\lambda_\gamma \mathrm{v}_\gamma}, \tag{18}$$

where $\tilde{c}^i_\gamma$ denotes the pixelwise closest color proposal to $\bar{x}^i$ in every iteration i. To show this, we start by inserting $\tau\mathcal{D}$ into the definition of the proximal mapping

$$\begin{aligned}
&\mathrm{prox}_{\tau\mathcal{D}(\,\cdot\,,\mathrm{C}_\gamma))}\left(\bar{x}^i\right) = \operatorname*{argmin}_x \left\{\tau\mathcal{D}(x, \mathrm{C}_\gamma) + \frac{1}{2}\|x - \bar{x}^i\|_2^2\right\} \\
&= \operatorname*{argmin}_x \left\{\tau\lambda_\gamma \sum_{p=1}^{N_P} \mathrm{v}_{\gamma,p} \min_{c_{\gamma,p} \in \left\{c^n_{\gamma,p}\right\}_{n=1}^{N_\gamma}} \frac{1}{2}\|x_p - c_{\gamma,p}\|_2^2 + \frac{1}{2}\|x_p - \bar{x}^i_p\|_2^2\right\}.
\end{aligned} \tag{19}$$

Since the function on the right-hand side of (19) is non-negative, lower semi-continuous and coercive, the minimum $\hat{x}$ is attained by the direct method of the calculus of variations. Due to the multi-well structure, all N_γ possible $n \in \mathbb{N}^\gamma$ proposals for the minimum need to be checked simultaneously. We therefore

compute all possible minima $\{\hat{x}^n\}_{n=1}^{N_\gamma}$ independently for all pixels and select the one with the lowest dataterm energy. The individual optimization steps are:

$$0 \stackrel{!}{=} \tau\lambda_\gamma \mathrm{v}_\gamma(\hat{x}^n - c_\gamma^n) + (\hat{x}^n - \bar{x}^i) \tag{20}$$

$$= \tau\lambda_\gamma \mathrm{v}_\gamma \hat{x}^n - \tau\lambda_\gamma \mathrm{v}_\gamma c_\gamma^n + \hat{x}_k - \bar{x}^i \tag{21}$$

$$\Rightarrow \hat{x}^n = \frac{\bar{x}^i + \tau\lambda_\gamma \mathrm{v}_\gamma c_\gamma^n}{1 + \tau\lambda_\gamma \mathrm{v}_\gamma} \tag{22}$$

Using $\mathbb{N}^\gamma = \{1, \ldots, N_\gamma\}$, and inserting into the pixelwise version of the dataterm d reveals the index n_p for the lowest energy per pixel p as follows:

$$n_p = \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} d(\hat{x}_p^n, \{c_{\gamma,p}^n\}) \tag{23}$$

$$= \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} \left\{ \frac{\lambda_\gamma \mathrm{v}_{\gamma,p}}{2} \|\hat{x}_p^n - c_{\gamma,p}^n\|_2^2 \right\} \tag{24}$$

$$= \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} \left\{ \frac{\lambda_\gamma \mathrm{v}_{\gamma,p}}{2} \left\| \frac{\bar{x}_p^i + \tau\lambda_\gamma \mathrm{v}_{\gamma,p} c_{\gamma,p}^n}{(1 + \tau\lambda_\gamma \mathrm{v}_{\gamma,p})} - c_{\gamma,p}^n \right\|_2^2 \right\} \tag{25}$$

$$= \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} \left\{ \frac{\lambda_\gamma \mathrm{v}_{\gamma,p}}{2} \left\| \frac{\bar{x}_p^i - c_{\gamma,p}^n}{(1 + \tau\lambda_\gamma \mathrm{v}_{\gamma,p})} \right\|_2^2 \right\} \tag{26}$$

$$= \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} \left\{ \|\bar{x}_p^i - c_{\gamma,p}^n\|_2^2 \right\}. \tag{27}$$

Hence, the index only depends on the initial distance between $\bar{x}^i$ and the different c_γ^n. However, there is still the possibility that multiple color proposals share the same distance to $\bar{x}^i$. In this case, we choose the first proposal – since they are ordered by confidence. Hence, we favor a single proposal over a color blending. Using Eqn.(27), we select the best per pixel reference per iteration i as

$$\tilde{c}_{\gamma,p}^i = c_{\gamma,p}^{i,n_p} \qquad \text{with} \qquad n_p = \operatorname*{argmin}_{n \in \mathbb{N}^\gamma} \left\{ \|\bar{x}_p^i - c_{\gamma,p}^n\|_2^2 \right\}. \tag{28}$$

Combining the above results, the proximal mapping can be written as

$$\mathrm{prox}_{\tau\mathcal{D}(\,\cdot\,, \mathrm{C}_\gamma))}\left(\bar{x}^i\right) = \frac{\bar{x}^i + \tau\lambda_\gamma \mathrm{v}_\gamma \odot \tilde{c}_\gamma^i}{1 + \tau\lambda_\gamma \mathrm{v}_\gamma}, \tag{29}$$

using the Hadamard product $\odot$ with broadcasting along color channels, and the simplification that $\tilde{c}_\gamma^i$ uses the best proposal for each pixel location. Here, we also denote the ^{i} subscript, to clarify that this has to happen in every iteration!

To combine the global and local multi-well datatermes with the standard motion dataterm we use the following approximation

$$\mathrm{prox}_{\tau\mathcal{D}}\left(\bar{x}^i\right) = \frac{\bar{x}^i + \tau \sum_{\gamma \in \{M,G,L\}} \lambda_\gamma \mathrm{v}_\gamma \odot \tilde{c}_\gamma^i}{1 + \tau \sum_{\gamma \in \{M,G,L\}} \lambda_\gamma \mathrm{v}_\gamma}, \tag{30}$$

which generalizes the individual proximal mappings and follows a similar derivation. It is an approximation for efficiency, as it assumes that the different dataterms are independent and neglects jumps among different minimizers. Therefore, each multi-well color proposal's best index per pixel can be efficiently precomputed, only considering the distance to the initial starting point $\bar{x}^i$. For all cases, where this index does not change if the dataterms are used independently or jointly, the approximation is also exact. Note that the best color proposal can change freely between the different iteration of the optimization scheme.

E User Evaluation details

We conducted a user evaluation asking 30 people to rank different color propagation methods. As input we use real color videos from DAVIS, where we use the initial color frame as reference, and convert the remaining frames to gray before passing them to the different color propagation methods to restore the colors. During the rating process the user sees the results from the methods side-by-side simultaneously as synchronized videos, next to the initial real color frame. The spatial ordering of the methods changes randomly for each sequence, and the sequence is shown in a loop until the user finishes the rating and continues to the next sequence. All users were using the same PC setup and similar lightning conditions. The group consisted of only non-color blind people, with 23 males and 7 females, and ages ranging from 24 to 38 years with a mean age of 30.0 years. All users where asked for their consent in written form and provided with information on how the data will be used, and how they can opt out if desired.

The Tables 2 show the percentage of the ranks (Top 1=best, Top 3=worst) and the average rank over all sequences of each dataset and over all users. Since we do not have access to DVCP results for DAVIS-2017-test we use DeepRemaster as the next best alternative. As can be seen, our method is ranked best 73.6% of the time for DAVIS-2017-val and 69.3% for DAVIS-2017-test. This greatly outperforms all competing methods, also on average rank.

rank	DVCP	DEB	LVVCP (Ours)
Top 1	9.3%	17.1%	**73.6%**
Top 2	26.9%	53.3%	19.8%
Top 3	63.8%	29.5%	6.7%
Avg rank	2.5 ± 0.7	2.1 ± 0.7	**1.3 ± 0.6**

rank	DeepRemaster	DEB	LVVCP (Ours)
Top 1	2.5%	28.3%	**69.3%**
Top 2	6.7%	65.4%	27.9%
Top 3	90.9%	6.3%	2.8%
Avg rank	2.9 ± 0.4	1.8 ± 0.5	**1.3 ± 0.5**

Table 2. User Evaluation - Average Ranks on DAVIS-2017-val (Left) and DAVIS-2017-test (Right). On average, our method is the Top 1 choice of the users.

F Limitations, assumptions, corner cases

Color propagation vs. colorization Our method is designed for faithful color propagation from a reference and puts focus on keeping details and minimizing color drift. Colorization of new objects with very different appearance than provided

Fig. 17. Failure Case comparison: Fixed zoom in on the soapbox sequence to reveal failure cases. Semantic similarity matching allows us to propagate color to the initially occluded second knee and the crowd in the background, while refinement reduces bleeding and artifacts.

in any of the references (global or previous frame) is not a main application. However, as our method allows to integrate multiple color proposals, a possible future extension could investigate the inclusion of color proposals for new objects from a pure colorization technique. Likewise, our method could easily be extended to add additional references - similar to the 2 reference example - but only for regions that contain new objects entering the scene.

Large untextured regions with different colors. Like most color propagation and colorization methods, our method relies on the gray image. If the reference image features different colors in regions that are hard to distinguish in the gray image, our method, like most methods can get confused. To remedy such problems, future work could focus on regularizers with larger receptive fields in combination with special losses such as spatial smoothness and adversarial training.

Failure case comparison - with and without appearance changes. As color propagation is a very ill-posed task, quality of all current methods decreases over distance to the reference, as indicated by the PSNR curves. Here, we investigate different failure cases of the best methods on the example of the moving objects in the soapbox sequence. While the first part shows how our method is very well able to handle e.g. disocclusions if reasonable references are present, the second part compares how our method and the baselines start to struggle as the appearance changes become larger.

Fig.17 shows the first half of the sequence. Up to frame 10 all methods propagate well. In frame 13 the people with blue and red shirts of the small crowd in the background are fully occluded and become dis-occluded from frame 15 onward. Our method is the only one that can recover the colors of these

Fig. 18. Failure Case comparison: Fixed zoom in on the soapbox sequence to reveal failure cases - hard cases. We show more temporal stability, while keeping more regions colored; See text for detailed description.

details, although they are visible in the global reference. Also visible from frame 15 onward is a loss of color in DVCP, and a color drift in DEB due a lack of temporal consistecy as reported in [5, arXiv Appendix F. Fig. 16]. The latter is visible, e.g., on the helmet turning red or the arms and knees and shirts turning blue. In contrast, our method manages to keep these details colored faithfully. In addition, we manage to colorize the initially occluded second knee of the man on the back of the soapbox, as it is similar to the reference and the other foot in the previous frame. This highlights the ability of our model to colorize new objects provided that they are similar to other objects in the previously colorized frame or the global reference frame.

Next, we focus on the harder cases shown in Fig. 18. DVCP has already lost most colors and further fades out. DEB still has colors, but red and blue tones are bleeding onto the shirt and the soapbox. Also some regions drift back and forth between red and blue such as the original blue disc on the left with the donkey head, or between gray and red such as the helmet and the head. While we keep such regions that are visible in the previous frame colorized, *there is also an interesting failure case of our method* on the knees of the rider in the back. After being hidden in frame 30, the knees gradually appear in frame 33/35 where our method sees more similarity to the close by blue stick, than to the initial knee. Hence, a wrong color is selected. As the knee gradually becomes larger, the self-similarity to the previous frame leads to a temporal consistent but a consistently wrong coloring of the knee. A similar effect happens to the shirt of the driver, gradually appearing after frame 15. One way to resolve such issues is by loosening temporal consistency such as for DEB, where the left knee starts blue and turns red later. A more promising way could be to also consider future frames when colorizing the current frame, which we pursue in future work.

G More Results and metrics

Qualitative training results vs. overfitting. The following images are from evaluating on our training set, and show an interesting behavior, about how our model resolves newly appearing input. As we built our model structure to stay faithful to the initially provided color reference, we avoid the blind creation of new colors. Hence, newly appearing objects need to be colorized in the style of the reference. Fig. 19 demonstrates this well by the signs in the background.

Fig. 19. Our model continues to propagate colors from the provided reference frame 001 even if it is different from the trained color – see the 'TONE' sign in frame 20, 24.

Fig. 20. Newly appearing hurdles to the right are being colorized according to the reference in frame 001.

For instance, although the 'TONE' sign is red and white during training, our model colorizes it in the style of the reference frame, where the only colored sign is blue and yellow. The same effect can be seen in Fig. 20, where the hurdle appearing on the right is green in the training data, but the style provided by the reference image is red. Even though both sequences have been part of the training set – teaching it a different color via the loss – our model has learned to colorize the new objects according to the provided reference. We expect this to further improve with dedicated losses, which we consider in future work.

Ablation on search ranges. Table 3 shows an ablation on the effect of different search ranges for the local neighborhood search. We achieve best results using a

frame method	1.0	5.0	10.0	15.0	20.0	24.0
Gray	25.30	25.40	25.49	25.46	25.59	25.60
Coarse:±2, Fine: ±6	44.07	40.08	37.82	36.18	35.61	34.96
Coarse:±4, Fine: ±4	**44.20**	40.17	37.94	36.40	35.86	35.16
Coarse:±6, Fine: ±2	**44.20**	**40.19**	**37.97**	**36.41**	**35.90**	35.22
Coarse:±8, Fine: ±2	44.18	**40.19**	**37.97**	**36.41**	**35.90**	**35.23**

Table 3. Ablation on search ranges using a model with CUDA sampling operator.

search window of ±8 pixels on the coarse low resolution, and refining with ±2. This results used a fast C++ CUDA sampling operator, but similar results can be achieved with the default pytorch operator.

CIDE2000 Fig. 21, 22 compare the performance on the CIDE2000 metric [9], which measures realistic appearance of color shifts. DEB and DeepRemaster already start with a significant hue shift, which is also clearly visible in the background of the qualitative results in the main paper (Fig.12, Fig.13). DVCP improves this on the data provided by the authors. Our method shows further improvements and reaches the lowest color shifts over all frames on both datasets.

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
CIEDE2000 ↓ Gray	9.25	9.25	9.18	9.22	9.14	9.06
DeepRemaster [6]	4.12	4.18	4.39	4.60	4.76	4.97
DEB [15]	3.12	3.19	3.46	3.70	3.92	4.19
DVCP [11]	1.16	1.64	2.14	2.53	2.81	3.09
LVVCP (Ours fast)	**0.85**	1.33	1.82	2.20	2.52	2.82
LVVCP (Ours mm fast)	**0.85**	**1.31**	**1.81**	**2.19**	**2.51**	**2.81**

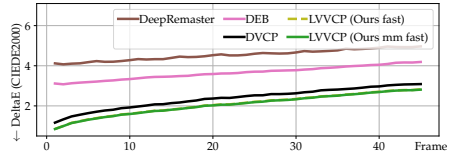

Fig. 21. CIDE2000 (↓ lower is better) on NDVCP Dataset; Our method has less initial color drifts and keeps it lowest.

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
CIEDE2000 ↓ Gray	10.62	10.61	10.59	10.60	10.66	10.59
DeepRemaster [6]	4.83	5.09	5.61	5.94	6.16	6.32
DEB [15]	3.31	3.76	4.28	4.53	4.81	5.11
LVVCP (Ours, fast)	1.35	2.08	2.92	3.24	3.64	4.07
LVVCP (Ours, mm, fast)	**1.35**	**2.07**	**2.93**	**3.22**	**3.58**	**4.00**

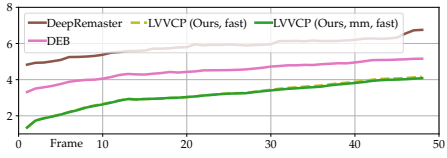

Fig. 22. CIDE2000 (↓ lower is better) on DAVIS-2017-test Dataset; Also here our method achieves lowest color drifts.

More Qualitative Results. Finally, Fig. 23, and 24 show a few qualitative results of our method on untrained sequences from DAVIS 2017 and DAVIS 2019. Even though some artifacts may occur – e.g. the racket of the tennis player in frame 030 – the resulting color propagations look plausible and promising.

Fig. 23. Qualitative Results of our Method on DAVIS-2017-test sequences

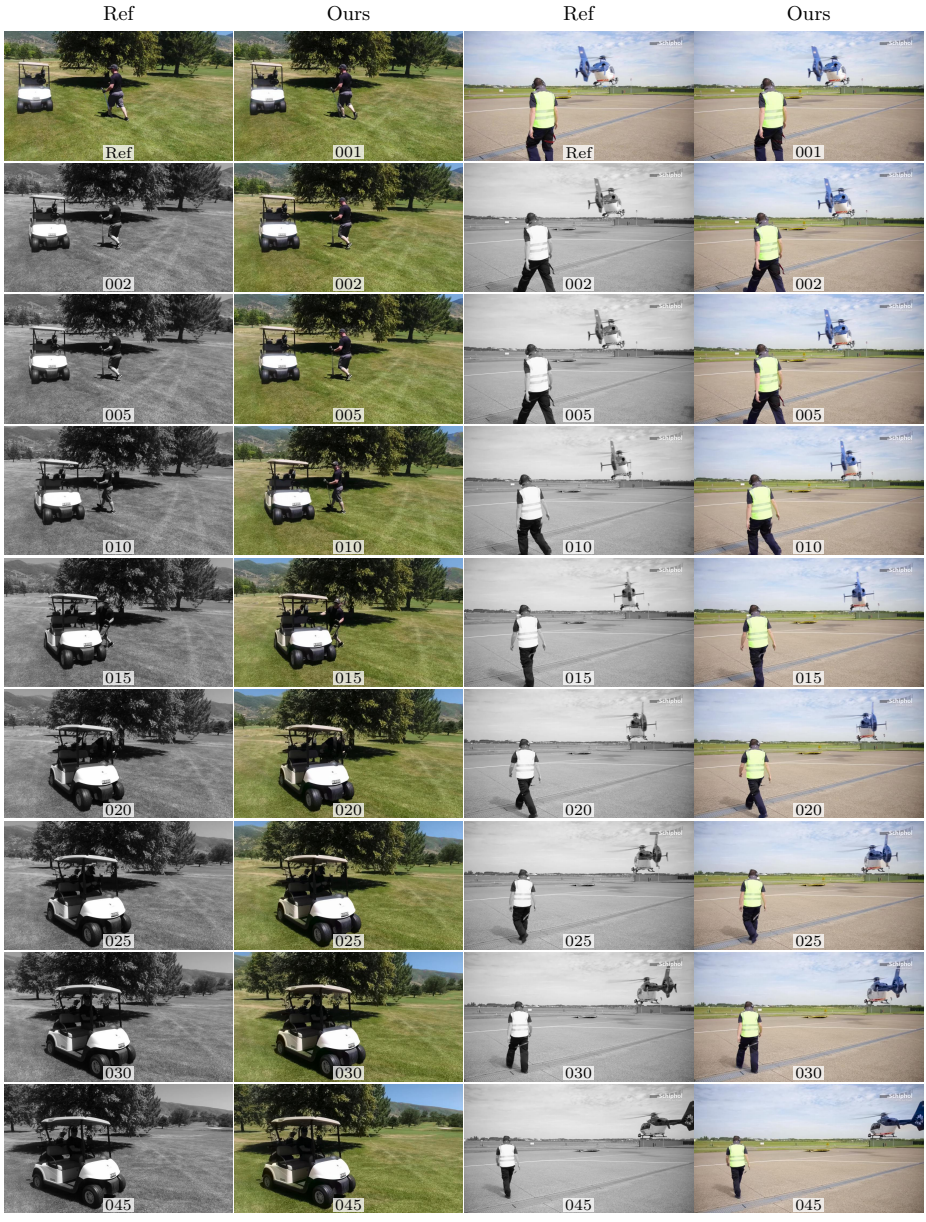

Fig. 24. Qualitative Results of our Method on DAVIS-2017-test sequences

References

1. Arnold & Richter Cine Technik GmbH & Co. Betriebs KG: Historic theater (movie sequence) (1902), www.arri.com 3
2. Buster Keaton: Go west (movie) (1925) 3
3. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer Vision – ECCV 2018. pp. 833–851. Springer International Publishing, Cham (2018) 3
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016) 3
5. He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. ACM Transactions on Graphics **37**(4), 47 (2018) 10
6. Iizuka, S., Simo-Serra, E.: Deepremaster: temporal source-reference attention networks for comprehensive video enhancement. ACM Transactions on Graphics **38**(6), 1–13 (2019) 12
7. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: Conference on Computer Vision and Pattern Recognition. pp. 451–461 (2017) 1
8. Kingma, D.P., Ba, J.L.: ADAM: a method for stochastic optimization. In: International Conference on Learning Representations (2015) 3
9. Luo, M., Cui, G., Rigg, B.: The development of the cie 2000 colour-difference formula: Ciede2000. Color Research & Application **26**, 340 – 350 (10 2001). https://doi.org/10.1002/col.1049 12
10. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: AAAI (2018) 2
11. Meyer, S., Cornillère, V., Djelouah, A., Schroers, C., Gross, M.: Deep video color propagation. In: British Machine Vision Conference (2018) 1, 12
12. Pinetz, T., Kobler, E., Pock, T., Effland, A.: Shared prior learning of energy-based models for image reconstruction. arXiv:2011.06539 (2020) 3
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015) 3
14. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision. pp. 402–419 (2020) 2, 3, 4
15. Zhang, B., He, M., Liao, J., Sander, P.V., Yuan, L., Bermak, A., Chen, D.: Deep exemplar-based video colorization. In: Conference on Computer Vision and Pattern Recognition. pp. 8052–8061 (2019) 1, 12