



Learned Variational Video Color Propagation

Markus Hofinger¹(✉) , Erich Kobler² , Alexander Effland³ ,
and Thomas Pock¹

¹ Institute of Computer Graphics and Vision, Graz University of Technology,
Graz, Austria

{markus.hofinger,pock}@icg.tugraz.at

² Institute of Computer Graphics, University of Linz, Linz, Austria

³ Institute for Applied Mathematics, University of Bonn, Bonn, Germany

Abstract. In this paper, we propose a novel method for color propagation that is used to recolor gray-scale videos (e.g. historic movies). Our energy-based model combines deep learning with a variational formulation. At its core, the method optimizes over a set of plausible color proposals that are extracted from motion and semantic feature matches, together with a learned regularizer that resolves color ambiguities by enforcing spatial color smoothness. Our approach allows interpreting intermediate results and to incorporate extensions like using multiple reference frames even after training. We achieve state-of-the-art results on a number of standard benchmark datasets with multiple metrics and also provide convincing results on real historical videos – even though such types of video are not present during training. Moreover, a user evaluation shows that our method propagates initial colors more faithfully and temporally consistent.

Keywords: Video color propagation · Learned variational refinement

1 Introduction

Interestingly, adding color to monochromatic images is as old as photography itself [18]. Lately, even entire movies have been meticulously colorized [20], e.g. *They Shall Not Grow Old* – Peter Jackson, to make historic movies more accessible to audiences used to the high-quality imagery of today’s cinema. An enormous manual effort has been spent to get the color of, for instance, gear or

Markus Hofinger and Erich Kobler are shared co-first authors. Source code can be found on <https://github.com/VLOGroup/LVVCP>.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20050-2_30.

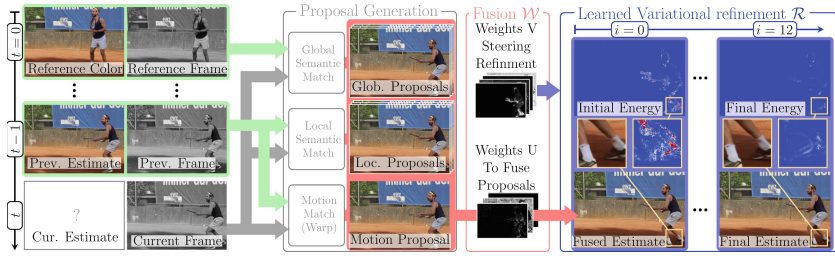


Fig. 1. Method overview: Video color propagation using color proposals from motion and semantic feature matching (to reference and prev. frame) are fused and then refined using a learned variational refinement. For details see page 516. Best viewed on screen.

uniforms correct from a historical perspective [31], which goes far beyond rough color guesses from pure fully automatic colorization.

One way of still keeping the colorization effort low is to avoid manual colorization of each frame but instead propagating the color from a single high quality manually colorized reference frame to subsequent frames. However, this also comes along with its own challenges, as can be seen at the top of Fig. 2. Here, a tree occludes parts of a car and simply propagating the color, as traditionally done using optical flow, leads to color artifacts at non-matched regions. Therefore, refinement is needed to keep results faithful over multiple frames.

Classical methods propose to solve this problem by setting up energy-based optimization problems [32] which assume that similar gray regions exhibit the same color, while at the same time enforcing color smoothness by using *hand-crafted* edge-aware regularizers [43]. Iteratively solving these optimization problems leads to smooth results with inpainted occlusions. Recent works demonstrate that a *deep learning* inspired total deep variation (TDV) regularizer \mathcal{R} can outperform *hand-crafted* regularizers on various image restoration tasks [26, 44]. Indeed, a TDV regularizer can be taught to inpaint color in an edge-aware fashion, as shown in a proof of concept in Fig. 2 (second row). Recently, deep learning-based approaches were proposed that transfer colors without optimization but rather use CNNs to regress colors based on color proposals from deep feature matches to the reference [38, 60], or use CNNs for temporal smoothing [30].

In this work, we therefore propose a novel method that combines the benefits of deep learning and optimization based methods. We use colors warped by motion together with alternative plausible color proposals. These are generated via deep feature matches to the global reference frame and to the previous colorized frame, which is more local in time and therefore more similar. All these estimates are subsequently fused in a data-driven manner and further refined in an unrolled optimization scheme using a learned modified TDV regularizer. Moreover, the whole optimization process is steered by image-dependent data-driven weights, which are independently estimated for each image by a CNN termed WeightNet. Thus, our energy-based model learns to refine the color estimates and to resolve color ambiguities among the different color proposals. In

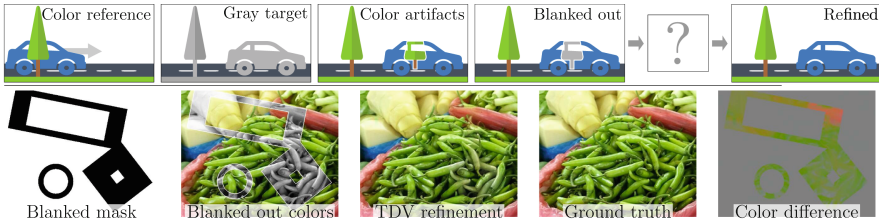


Fig. 2. Top: Propagating color from a color reference to a gray image according to the object motion (optical flow) leads to artifacts. These need to be detected and filled. Bottom: Proof of concept. A learned TDV is able to restore missing colors edge-aware. (Color figure online)

particular, the mathematical structure leads to interpretable and user overrideable intermediate results. Overall, the main contributions are as follows:

- Generation of multiple plausible color proposals of different types (global/local) by a sophisticated feature-matching process as well as motion.
- Learned color proposal fusion and guiding of the subsequently learned variational refinement to resolve ambiguities among numerous color proposals.
- Variational structure improves mathematical controllability and interpretable intermediate results allowing extensions to the method even after training.
- State-of-the-art results on several video color propagation datasets, metrics and promising qualitative results also validated by a user evaluation.

2 Related Work

Video color propagation is closely related to image colorization. While classic image colorization propagates color information spatially within one image, video color propagation additionally has to incorporate multiple frames. In literature, diverse colorization approaches exist that can be roughly classified into interactive [32, 59, 63], reference or exemplar-based [17, 23, 25, 38, 40, 53, 57], and fully automatic methods [11, 22, 48, 61]. Our method is reference-based.

Interactive methods rely on some kind of user input, e.g. scribbles, defining the color for selected image pixels. Classically, the color of the remaining pixels is determined by diffusing color over the image or between frames using a locally adaptive distance, based on handcrafted similarity measures such as luminance [4, 32], geodesic distance [59], or texture features [1, 35, 46]. Later, the amount of required user interaction was reduced by learning image-specific similarity measures using, for instance, local linear embeddings [10], iterative feature discrimination [58], or CNNs [14]. Motivated by the success of deep learning [29] and the availability of large-scale image datasets [34, 49], Zhang et al. [63] learned a deep CNN that colorizes an image given either scribbles or a color histogram.

In contrast to the aforementioned approaches, *reference-based* methods utilize a reference image to transfer its colors to similar regions within a destination

frame. Initial approaches transferred the color from the reference image solely based on luminance [47] and texture similarity [57], which often lead to spatially varying colors within an image region. Thus, optimization-based spatial regularization techniques were introduced to refine a coarse colorization based on correspondences [5, 9, 23, 40, 43]. Charpiat et al. [9] rephrased the colorization problem as a discrete labeling problem and resolved local ambiguities by minimizing a Markov random field (MRF) energy, which resulted in a spatially consistent image with discrete colors with a final variational refinement. Pierre et al. [40, 43] advocated a variational method utilizing a hand-crafted regularizer favoring spatially consistent colors, and a dataterm for plausible pixel color proposals. They further extended this explicitly to video [41] by integrating motion-based color proposals from PatchMatch [3] and TV- ℓ_1 optical flow [7]. Soon after, VPN [25] used a completely different deep learning-based approach using learnable bilateral filters to propagate color in videos. Other deep learning approaches tailored to videos followed [21, 38, 60]. These methods computed correspondences based on deep features [16, 51] of the gray-scale images. Interestingly, Vondrick et al. [56] showed that networks learned tracking when trained on color propagation. DeepRemaster [21] utilized a temporal attention mechanism to colorize historic videos for a given example image. Deep image analogy [33] developed a PatchMatch for deep features, bridging larger domain gaps. He et al. [17] extended this approach for exemplar-based image colorization, filling non-matched regions using a CNN trained on a large database. The video extension DEB [60] focused on automatic exemplar-based video colorization by using not necessarily related exemplar images as references, allowing color deviations from the reference. In contrast, DVCP [38] paid close attention to staying close to the reference by combining motion estimation and feature-based matching to the global reference frame to avoid color drifts. Our method also focuses on staying faithful to the colors from a provided high quality reference.

Fully automatic colorization approaches predict reasonable colors without user input by learning on large-scale image datasets [34, 49]. While classical methods are based on, e.g., conditional Gaussian random fields [13], more recent approaches [28, 48, 52, 61] proposed different CNN architectures to address the multi-modality of colorization and introduced semantics from different perspectives. These colorization techniques relied on, e.g. semantic features [22], the prediction of a color histogram for each pixel of a gray-scale image [28, 61], variational refinement [39], autoregressive neural networks [48], conditional variational autoencoders [12], conditional GANs [6, 24], a color diversity loss [30], or conditional autoregressive transformers [27]. In contrast to automatic colorization, we focus on faithful propagation of high-quality references.

For further details on colorization and color propagation, we recommend the surveys [2, 15, 42], for (learnable) variational refinement [8, 26].

3 Method

Overview. As shown in detail in Fig 1, our method propagates color from a given color reference frame sequentially to the following gray-scale frames. To avoid the aforementioned motion artifacts, our model fuses and refines diverse color proposals of various sources to a final color estimate. In detail, we extract multiple color proposals with confidences, for each pixel of a frame based on semantic matches to the Global reference frame and the already colorized previous (Local) frame. Further, we also use the Motion-compensated previous frame as color proposal. All these different color proposals are fused into an initial color estimate via a learned WeightNet \mathcal{W} . Then, this estimate is refined by unrolling an energy-based optimization algorithm that facilitates the learned edge-aware total deep variation (TDV [26]) regularizer \mathcal{R} . The optimization is further guided by pixel-wise weights, provided by \mathcal{W} for each frame. Further details on proposal generation, fusion, and refinement are given in Sects. 3.1, 3.2 and 3.3, respectively.

Setup and Notation. We primarily operate in the CIE-Lab color space Ω^{lab} as it mimicks the human color perception. *Color images* like the ground truth y , the color estimate x , or the color proposals c always refer to ab channels, if no other channel subscript is given like in y_g (original gray-scale), y_l (CIE-Lab luminance) or y_{lab} . The *numbers of* pixels is denoted by N_P , pyramid levels by N_J , color proposals by $N_M/N_G/N_L$ and training frame augmentations by N_A and we frequently use corresponding subscripts (p, j, \dots) as indices. Further, to index the *color proposal types* (Motion, Global, Local) we use $\gamma \in \{M, G, L\}$, i.e. c_M denotes a motion color proposal, while c_γ is a generic placeholder for any color proposal type. In addition, we indicate the pixel-wise product using broadcasting over color channels by \odot . To ease notation, we frequently omit the frame superscript t if clear from the context.

3.1 Color Proposal Generation and Matching

In this section, we describe, how we generate our three color proposal *types* (see Fig. 1). In a nutshell, we bilinearly interpolate colors from matched positions in either the global reference $t=0$ (G), or the previously colorized frame $t-1$ (L,M). On the previous (Local) frame we extract proposals via feature matching C_L and via Motion C_M . We also use feature matching to the Global reference C_G . Each feature matching yields multiple proposals per pixel along with a confidence. The following paragraphs elaborate on the details.

For our color proposals based on *motion* C_M , we use RAFT [54] to estimate motion m_M between the current frame y_g^t and its previous frame y_g^{t-1} . We further compute the forward-backward motion difference δ_M for occlusion reasoning (following [19, suppl. Equation 10]), and also use it as a confidence, as motion provides wrong colors for occluded areas. To provide plausible colors for such areas, we use semantic feature matching, described next.

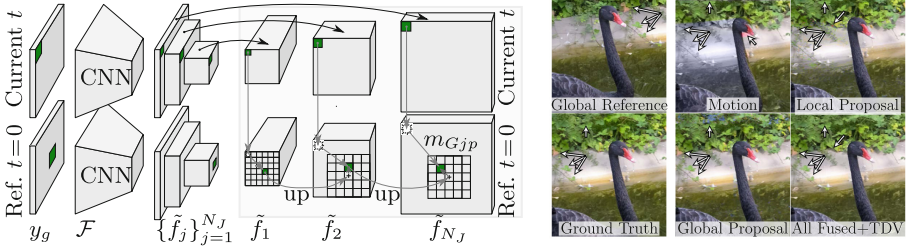


Fig. 3. Left: Global matching procedure for gray-scale images y_g^t and y_g^0 using a feature pyramid $\{\tilde{f}_j^t\}_{j=1}^{N_J}$ with N_J -levels of features \tilde{f}_j^t . Right: Results of colorizing multiple frames with a single color proposal type (noisy) vs. our learned fusion \mathcal{W} and TDV refinement; Mind the errors on the fast moving background. (Color figure online)

Our *global semantic matching* to the reference y_g^0 , finds and refines N_G best matches using a CNN feature encoder \mathcal{F} . In detail, we convert each gray frame to a pyramid of semantic features $\{f_j^t\}_{j=1}^{N_J} = \mathcal{F}(y_g^t)$ with different spatial resolutions on N_J levels, as seen on the left in Fig. 3. Ablation experiments (see suppl.) revealed that VGG16 with batchnorm pre-trained for classification works best. We use instance normalization [55] yielding features \tilde{f} of similar magnitude. For each pixel location p within \tilde{f}_j^t , we search in locations q around a neighborhood $\mathcal{N}(p)$ in the corresponding features \tilde{f}_j^0 of the reference image y_g^0 , i.e.

$$\hat{k}_{Gjp}^t = \max_{q \in \mathcal{N}(p)} \left\{ \text{ReLU} \left(\left\langle \frac{\tilde{f}_{jp}^t}{\|\tilde{f}_{jp}^t\|_2}, \frac{\tilde{f}_{jq}^0}{\|\tilde{f}_{jq}^0\|_2} \right\rangle \right) \right\}. \quad (1)$$

Here, \hat{k}_{Gjp} is the best global confidence for each pixel p of the current level j , using a truncated normalized cross correlation. We define m_{Gjp} as the according match (2D offset: $q - p$) that maximized \hat{k}_{Gjp} . The search process is repeated on the next finer level, centered around the position indicated by the upsampled m_{Gjp} . We use nearest neighbor upsampling and rescale to compensate the larger pixel spacing. Repeating this for all pixels and levels leads to a field of dense matches m_G on the final level. While we use the whole image as the search neighborhood $\mathcal{N}(p)$ on the coarsest level, we restrict $\mathcal{N}(p)$ to ± 2 pixels for refinement on the finer levels (see Fig 3). On the finest level we compute a final confidence by multiplying all confidences, using nearest neighbor upsampling (\uparrow), i.e.

$$k_G = \prod_{j=1}^{N_J} \uparrow (\hat{k}_{Gj}). \quad (2)$$

The corresponding Global color proposal c_G for each match and pixel is computed by sampling from the color reference y^0 using the matched positions m_G . The global matching and refinement process is repeated N_G times, using the N_G most confident matches on the coarsest level. This yields a set of $C_G = \{c_G^n\}_{n=1}^{N_G}$ color proposal images, with a corresponding set of final confidences K_G .

The *Local matching* process closely follows the global matching. In contrast, we match against the *previous* gray frame y_g^{t-1} and sample from the last color estimate x^{t-1} to get C_L . Moreover, we do not search over the whole image on the coarsest level, as motions are smaller, but around a small neighborhood $\mathcal{N}(p) = \pm 8$ pixels around the positions indicated by the motion estimate m_M .

To summarize, our color proposal generation process yields a set of diverse proposal types $C = \{C_M, C_G, C_L\}$, containing all one or more color proposals. The *best proposal* per type \hat{c}_γ , is either the single proposal for motion \hat{c}_M , or the best via the pixel-wise confidence, yielding \hat{c}_G or \hat{c}_L . To compare the differences of the proposal types, we propagate colors using each type’s best proposal separately over many frames, see Fig. 3. While motion color proposals bleed into occluded areas, local proposals have less accumulated errors. Global proposals allow fixing objects that were occluded for multiple frames, like the leaves on the right, but at the cost of higher base noise. Fusing the color proposals in each step together with our learned refinement yields the best result as described next.

3.2 Initial Fusion with Weight Network

Since the color proposal types have very different properties (see Fig. 3), we use a CNN based UNet termed WeightNet \mathcal{W} (details in suppl.) to fuse the initial best color proposals $(\hat{c}_M, \hat{c}_L, \hat{c}_G)$ using pixel-wise weights $U = \{u_M, u_G, u_L, u_0\}$. Moreover, we use \mathcal{W} to predict an additional set of pixel-wise weights $V = \{v_M, v_G, v_L, v_0, v_R\}$ to locally guide the subsequent variational refinement. To enable a propagation of weights across multiple frames, we also feed the (motion compensated) U^{t-1} and V^{t-1} from the previous frame into \mathcal{W} , i.e.

$$(U^t, V^t) = \mathcal{W}_\theta(y_l^t, \widehat{Z}^t, U^{t-1}, V^{t-1}). \quad (3)$$

Here, y_l^t is the current frame’s luminance, \widehat{Z}^t concatenates the best color proposal per pixel for each proposal type, together with its associated confidence or motion delta δ_M and absolute luminance difference. All weights U^t and V^t have a pixel value in $[0, 1]$. Using the fusion weights U^t , the initial color estimate reads as

$$x^{t,0} = \sum_{\gamma \in \{M,G,L\}} u_\gamma^t \odot \hat{c}_\gamma^t, \quad (4)$$

which essentially implements a pixel-wise soft-selection of the best type of color proposals \hat{c}_γ^t or no proposal at all, since we enforce for each pixel that

$$\sum_{\gamma \in \{M,G,L\}} u_{\gamma,p}^t \leq 1, \quad \sum_{\gamma \in \{M,G,L\}} v_{\gamma,p}^t \leq 1. \quad (5)$$

This is implemented via a pixelwise softmax using u_0 and v_0 allowing inequality.

Hence, \mathcal{W} can blend colors or fade them out if all matches seem implausible. Recall that the best color proposal of each type \hat{c}_γ^t is defined as the one whose associated pixel-wise confidence is maximal. Since only the best color proposal of each type \hat{c}_γ^t is fed into the WeightNet, we can adapt the number of proposal per pixel individually for each type without any retraining. This also holds true for the subsequent learned variational refinement, which even uses the full set of proposals C , allowing it to undo initial wrong choices, as we will elaborate next.

3.3 Learned Variational Refinement

This section describes the details of our learned variational refinement, which we perform independently for each frame. In a nutshell, our model uses an unrolled optimization algorithm, to decrease an energy \mathcal{E} consisting of a dataterm energy \mathcal{D} that models coherence to the color proposals, and a total deep variation regularizer (TDV [26]) \mathcal{R} that learns to model spatial color smoothness. The whole process is steered by pixel-wise weights provided by WeightNet \mathcal{W} , and automatically picks the best reference, as we will explain in the following.

In detail, we combine and extend the approaches [26, 43] and let

$$\mathcal{E}(x) = \mathcal{D}(x, \mathbf{C}) + \mathcal{R}_\theta(x, y_l) \quad (6)$$

be our learnable energy, which is a function of the current color estimate x . Here, y_l is the current frames luminance, and $\mathbf{C} = \{C_M, C_G, C_L\}$ the set of all color proposal types. For example, $C_G = \{c_G^n\}_{n=1}^{N_G}$ denotes the global color proposal, which already provides N_G different proposals per pixel. The regularizer

$$\mathcal{R}_\theta(x, y_l) = \sum_{p=1}^{N_P} v_{\mathcal{R},p} \cdot r_\theta(x, y_l)_p. \quad (7)$$

is weighted per pixel p with a scalar weight $v_{\mathcal{R},p} \in [0, 1]$ generated by WeightNet \mathcal{W} . Therefore, \mathcal{W} can allow to preserve high frequency textures in regions of high confidence and rely on the regularizer in uncertain regions. The regularizer itself, is a special twice differentiable UNet, detailed in TDV [26], with learnable parameters θ . Similar to the regularizer, the dataterm

$$\mathcal{D}(x, \mathbf{C}) = \sum_{\gamma \in \{M, L, G\}} \lambda_\gamma \sum_{p=1}^{N_P} v_{\gamma,p} \cdot d(x_p, C_{\gamma,p}) \quad (8)$$

also consists of a weighted combination of the pixel-wise dataterms d of each proposal type (M,L,G). The learned scalars $\lambda_\gamma \in \mathbb{R}^+$ balance the different dataterm types based on dataset statistics, and the scalar fields v_γ are again predicted by WeightNet \mathcal{W} . This allows \mathcal{W} to shift attention between the dataterms of the proposal types, focusing on the most trusted type for each pixel.

While a standard ℓ_2 dataterm uses a single fixed reference, we use a multi-well dataterm which automatically chooses the best reference from the N_γ proposals per pixel. This means for each pixel p we use the dataterm

$$d(x_p, \{c_{\gamma,p}^n\}_{n=1}^{N_\gamma}) = \min_{\tilde{c}_p \in \{c_{\gamma,p}^n\}_{n=1}^{N_\gamma}} \|x_p - \tilde{c}_p\|_2^2. \quad (9)$$

Hence, although the color proposal fusion only used the *best* proposal based on confidences, this multi-well dataterm can still choose from *all* proposals per pixel and type. Therefore, the optimization scheme does not only refine x but also cleans the dataterm reference from initial color noise as illustrated in Fig. 4.

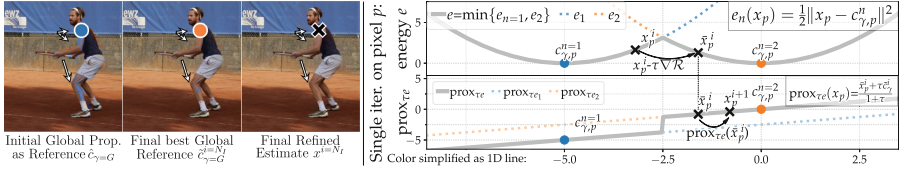


Fig. 4. Auto-selection of closest dataterm reference: Left shows how initial noisy ref. updates to a better option per pixel, easing TDV refinement of x . Right, details on the marked pixel’s iterative update: Multi-well dataterm energy e is a simplification of (9) for single pixel x_p . A step on $\nabla\mathcal{R}$ improves colors of estimate $x_p^i \rightarrow \bar{x}_p^i$, away from wrong blue ref. $c_{\gamma,p}^{n=1}$. The dataterm energy e_2 of proposal $c_{\gamma,p}^{n=2}$ is now lower than e_1 , leading to the reference update. Hence, $\text{prox}_{\tau e}$ now uses the better orange ref. $c_{\gamma,p}^{n=2}$ for \bar{x}_p^i .

Given the overall energy (6), our model refines the initial fused x^0 using N_I unrolled iterations of a proximal gradient scheme [8]. A step on $\nabla\mathcal{R}$ for spatial color smoothness, is followed by a proximal dataterm step, updating x^i to

$$x^{i+1} = \text{prox}_{\tau\mathcal{D}}(\bar{x}^i) = \text{prox}_{\tau\mathcal{D}}(x^i - \tau\nabla_1\mathcal{R}_\theta(x^i, y_l)) \quad (10)$$

in each iteration i . We use the proximal map

$$\text{prox}_{\tau\mathcal{D}}(\bar{x}^i) = \frac{\bar{x}^i + \tau \sum_{\gamma \in \{M,G,L\}} \lambda_\gamma v_\gamma \odot \tilde{c}_\gamma^i}{1 + \tau \sum_{\gamma \in \{M,G,L\}} \lambda_\gamma v_\gamma}, \quad (11)$$

where $\text{prox}_{\tau\mathcal{D}}$ is basically a convex combination of the intermediate estimate \bar{x}^i and its currently closest color proposal per proposal type \tilde{c}_γ^i , with \odot being the pixel-wise product with broadcasting along color channels (derivation in suppl.).

To summarize, our iterative approach refines the initial fused most confident color proposals, and enables an automatic adaption of the dataterm references in each iteration. Hence, if a regularizer update changes noisy pixels to favor color smoothness, the dataterms can change their pixel-wise color references $\hat{c}_{\gamma,p}$ from an initial most confident but noisy value to the best in the set of all proposals per type $C_{\gamma,p}$ for each pixel p . Finally, the interplay of all parts is shown in Fig. 5, where \tilde{M}_O^t indicates occluded regions as explained in the supplementary.

3.4 Training

For training, we use an MSE loss in the ab space, in combination with online-hard-example-mining (OHEM [50]) to focus on the 25% most difficult pixels per image, as most regions soon work very well. We train on a batch of frame pairs. In addition, we use the estimated result as augmented input (gradient-stopped) for the next frame and repeat this N_A times to simulate realistic artifact accumulation. Although this teaches our model real artifacts, extreme cases can occur in the initial training phase, which would require a heuristic increase of the number of propagated frames with training duration. To avoid this, we rescale

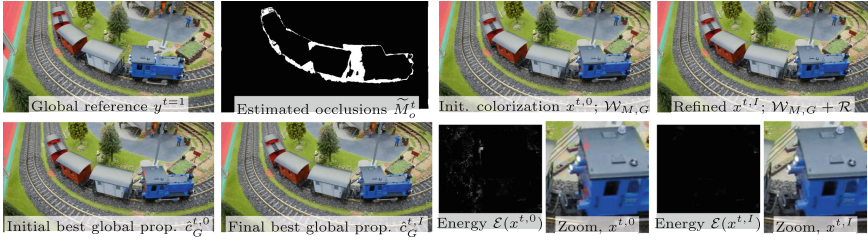


Fig. 5. Color refinement example; The backside of the train is initially occluded (\tilde{M}_O^t); \mathcal{W} fuses the initially most confident color proposals, which can contain noise; An unrolled optimization ($I=12$) with learned TDV further refines results for smooth colors.

the loss of each frame pair based on an oracle estimating the best currently possible initial proposal $\hat{c}_o^{t,0}$ from all proposals via

$$\hat{c}_{p,o}^{t,0} = \underset{c \in \{c_M^t, c_{G1}^t, \dots, c_{GN_k}^t, c_{L1}^t, \dots, c_{LN_k}^t\}}{\operatorname{argmin}} \|c_p - y_p^t\|_2, \quad (12)$$

$$\mathcal{L}_o(x, y) = \sum_{t=1}^{N_A} \frac{\mathcal{L}(x^t, y^t)}{\varepsilon_o + \mathcal{L}(\hat{c}_o^{t,0}, y^t)}, \quad (13)$$

where we set ε_o to roughly 1% of the loss the model generates without loss rescaling and use $N_A = 5$ as default. To speed-up training, we pre-compute the gray-scale matches and train only \mathcal{W} and \mathcal{R} . However, our method allows for full end-to-end training. Further details on training can be found in the supplementary material.

Multimodel Training. Using more proposal *types* for \mathcal{W} typically provides better initial estimates. However, this also means that fewer errors remain for the TDV regularizer to train on. Hence, we propose to train a *shared regularizer* \mathcal{R} with \mathcal{W} using different color proposal types e.g. $\mathcal{W}_{M,G}$ and $\mathcal{W}_{M,G,L}$ at the same time. This allows to train \mathcal{R} with a much wider variety of hard and easy cases.

4 Experiments

In this section, we show various quantitative and qualitative experimental results. Further ablation results, interactive experiments, and a discussion of limitations can be found in the supplementary material. The source code is on [github](#).

Baselines and datasets. As baselines we use three color propagation methods VPN [25], DeepRemaster [21] and DVCP [38], as well as the exemplar-based colorization method DEB [60]. As DeepRemaster and DEB require image sizes to be a multiple of 32, we zero-pad the inputs and crop the results for them.

We report results on multiple datasets. For training and evaluation, we use the splits of DAVIS 2017 [45] as defined in the VPN [25] codebase. It consists of 35 training and 15 evaluation sequences of 25 consecutive frames each. For testing we use the 27 sequences from DAVIS-2017-test that are at least 45 frames long. We resample the original high-resolution sources to remove JPG artifacts and get highest quality ground-truth. Furthermore, we report results on NDVCP, the non-DAVIS subset from the test-set of [38] (55 videos of 45+ frames; Fig. 7) to avoid overlaps. We received the DVCP results and data from the authors¹, and re-ran all open source methods. For datasets where we did not receive DVCP results (e.g. DAVIS-2017-test), we picked the next best open source method.

Metrics. In literature, metrics are computed quite differently, e.g. PSNR with different color spaces and normalizations [25, 38], or showing averages over the first t frames [38] vs. reporting each step t [60], which prevents direct comparisons. Hence, to ensure fairness, we identically compute all metrics on the results of all methods. We compute PSNR over the CIE-Lab ab color channels (PSNR $_{ab}$), as the luminance is fixed (details in suppl.). Furthermore, we report CIDE2000 [36] in the supplementary. Finally, we compute the open source LPIPS metric [62], which corresponds well to human perception on patch level.

Comparison on DAVIS. Fig. 6 shows our results and ablations on DAVIS-2017-val [25]. Using our fused proposal generation ($N_G = N_L = 8$) alone already outperforms some baselines such as VPN or Levin. From the color proposals, $\hat{c}_L^{t,0}$ works best up to roughly 11 frames, when global color proposals perform better as they do not accumulate errors. Fusing motion and global color proposals ($\mathcal{W}_{M,G}$) already outperforms all baselines, and adding regularization ($\mathcal{W}_{M,G} + \mathcal{R}$) yields further improvement. Adding local color proposals further enhances results on initial frames. Training with frame propagation augmentation over 9 frames ($\mathcal{W}_{M,G,L} + \mathcal{R} + N_A = 9$) improves the long-range quality, without adding additional inference time. Using $N_G = N_L = 3$ ('fast') reduces inference time at similar performance. Performance improves further with multimodel training 'mm'. Hence, we use these two models for all further comparisons.

Comparison on NDVCP and DAVIS-2017-test Dataset.

Using the previously best methods, we computed results also on larger and longer datasets. Figure 7 shows results for PSNR $_{ab}$ (higher is better) of all pixels as well as on occluded areas for both datasets separately. The occlusions are estimated using the heuristic from [37] (see supplementary for details). Even though we train only on the 35 DAVIS sequences, we greatly outperform the baselines on both datasets also in occlusions. In addition, we compute the perceptual metric LPIPS (v0.1 VGG) [62], and also there our method outperforms the baselines by a clear margin (Fig. 8).

¹ We thank DVCP authors for the data. As their results exclude the DAVIS-2017-val video *mallard-water*, we also omit it for fair comparison resulting in 14 sequences.

method \ frame:	1.0	5.0	10.0	15.0	20.0	24.0
Baselines						
Gray	25.30	25.40	25.49	25.46	25.59	25.60
VPN	29.74	28.65	27.84	27.69	27.51	26.99
Levin	39.35	33.76	30.94	29.27	28.40	27.78
DeepRemaster	32.30	31.88	31.52	31.22	31.03	30.71
DEB	34.67	33.57	32.80	31.93	31.86	31.58
DVCP	39.40	35.88	34.57	33.43	33.09	32.89
Our Methods						
$c_{M,G}^{1.0}$ Motion Cand.	40.52	35.42	32.78	31.20	30.27	29.46
$c_{M,G}^{5.0}$ Local Cand.	42.06	36.87	34.29	32.59	31.69	30.90
$c_{M,G}^{1.0}$ Global Cand.	41.00	35.70	33.88	32.83	32.15	31.52
Our Methods + Refinement						
$W_{M,G}$	43.46	39.10	36.81	35.35	34.83	34.20
$W_{M,G} + \mathcal{R}$	44.05	39.90	37.60	36.21	35.69	35.01
$W_{M,G,L} + \mathcal{R}$	44.27	40.19	37.91	36.38	35.76	35.05
$W_{M,G,L} + \mathcal{R} + N_A=9$	44.25	40.18	37.94	36.44	35.86	35.16
$W_{M,G,L} + \mathcal{R} + N_A=9, \text{ fast}$	44.19	40.15	37.92	36.37	35.87	35.20
$W_{M,G,L} + \mathcal{R} + N_A=9, \text{ fast, mm}$	44.17	40.20	37.97	36.42	35.91	35.23

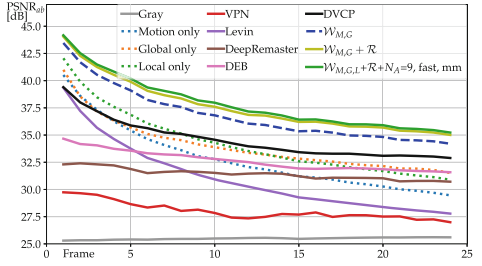


Fig. 6. PSNR_{ab} on DAVIS-2017-val Dataset; Already our weakest model W_{MG} , outperforms all baselines. Adding refinement ($W_{MG} + \mathcal{R}$), adds further improvements.

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
NDVCP Dataset						
Gray	26.90	26.90	26.96	26.92	26.98	27.07
DeepRemaster [21]	34.38	34.22	33.61	33.16	32.97	32.49
DEB [60]	35.39	35.08	34.30	33.83	33.48	33.18
DVCP [38]	43.67	40.37	37.78	36.21	35.45	34.60
LVVCP (Ours fast)	47.85	42.91	39.74	37.99	36.97	36.13
LVVCP (Ours mm fast)	47.87	43.05	39.78	38.03	36.99	36.15
NDVCP - occld						
Gray	27.32	27.53	27.48	27.19	27.00	27.16
DeepRemaster [21]	33.21	33.03	32.61	32.16	32.05	31.73
DEB [60]	34.17	34.00	33.16	32.74	32.46	32.33
DVCP [38]	39.83	37.32	35.20	33.77	33.45	32.85
LVVCP (Ours fast)	40.52	38.94	36.47	34.98	34.40	34.03
LVVCP (Ours mm fast)	40.46	39.09	36.55	34.99	34.36	33.98

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
DAVIS-2017-test						
Gray	24.92	24.94	24.89	24.91	24.91	24.95
DeepRemaster [21]	32.08	31.61	30.74	30.29	29.98	29.71
DEB [60]	33.82	32.91	31.74	31.29	30.75	30.45
LVVCP (Ours, fast)	42.91	39.03	35.61	34.39	33.44	32.64
LVVCP (Ours, mm, fast)	42.91	39.05	35.60	34.46	33.57	32.78
DAVIS-2017-test - occld						
Gray	26.11	25.54	25.27	25.11	25.13	25.13
DeepRemaster [21]	31.03	30.61	30.01	29.59	29.42	29.27
DEB [60]	33.32	32.29	31.09	30.55	30.07	29.83
LVVCP (Ours, fast)	37.56	35.51	33.32	32.48	31.83	31.32
LVVCP (Ours, mm, fast)	37.58	35.53	33.26	32.53	31.95	31.47

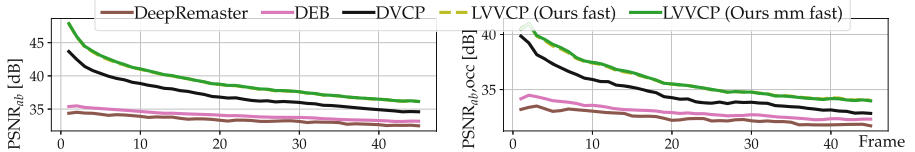


Fig. 7. PSNR_{ab}(↑) on NDVCP and DAVIS-2017-test datasets. Top of each table = all pixels; Bottom = occluded regions only; Graph shows results on NDVCP subset. Performance in occluded regions is lower for all methods; We still outperform all baselines.

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
NDVCP						
Gray	0.1216	0.1220	0.1219	0.1227	0.1230	0.1216
DeepRemaster [21]	0.0723	0.0734	0.0773	0.0795	0.0820	0.0839
DEB [60]	0.0568	0.0585	0.0628	0.0656	0.0687	0.0710
DVCP [38]	0.0127	0.0246	0.0361	0.0445	0.0505	0.0559
↓ LVVCP (Ours fast)	0.0067	0.0162	0.0274	0.0351	0.0418	0.0465
↓ LVVCP (Ours mm fast)	0.0065	0.0155	0.0267	0.0344	0.0410	0.0455

method \ frame:	1.0	5.0	15.0	25.0	35.0	45.0
DAVIS-2017-test						
Gray	0.1569	0.1569	0.1575	0.1551	0.1553	0.1534
DeepRemaster	0.0872	0.0908	0.0979	0.1001	0.1019	0.1036
DEB	0.0635	0.0705	0.0775	0.0803	0.0814	0.0854
↓ LVVCP (Ours, fast)	0.0161	0.0310	0.0470	0.0544	0.0612	0.0677
↓ LVVCP (Ours, mm, fast)	0.0161	0.0305	0.0463	0.0525	0.0585	0.0649

Fig. 8. LPIPS [62] (↓ lower is better) on NDVCP (Left) and DAVIS-2017-test (Right); Both our models with and without multimodal training show lowest perceptual errors.

User Evaluation. To better assess the quality of the models for our task of faithful video color propagation, we asked 30 users to rate the models on the DAVIS-2017-val and DAVIS-2017-test sets Fig. 9. Each video was converted to grayscale and recolored by different methods given the ground-truth colored first frame. In particular, we asked the users which method propagates the colors from the

still image reference most faithfully and consistent over time. The users then had to rank the methods from best to worst for each video sequence independently. Figure 9 shows how often each method ranked from best (green) to worst (red). As can be seen the users clearly prefer our method with a consistent large gap.

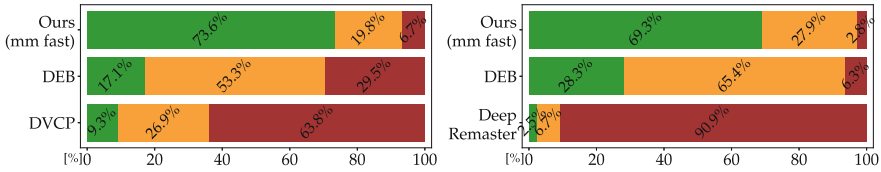


Fig. 9. User evaluation; Our method ranks best more than twice as the baselines; Left = DAVIS 2017 val, Right = DAVIS 2017 test; green = best, orange = 2nd, red = worst. (Color figure online)

Qualitative Comparison. Figure 10 shows a qualitative comparisons to the best performing baselines DEB [60] and DVCP [38] on a complex scene to reveal error patterns. While DVCP lost most of the color of the soapbox and the drivers

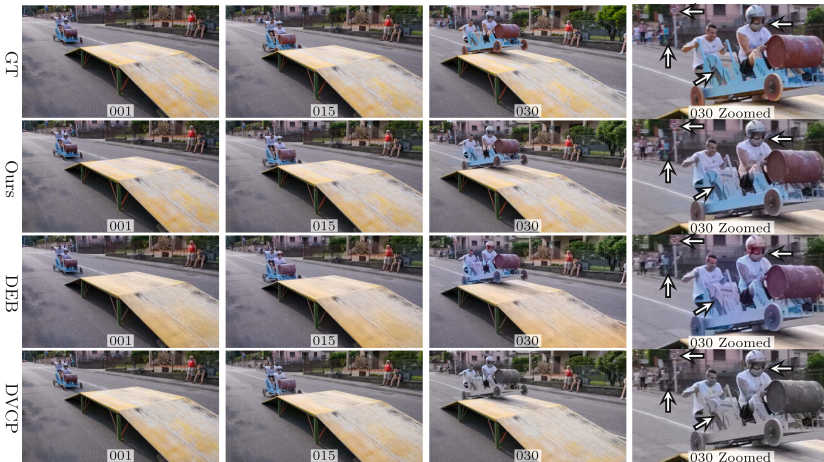


Fig. 10. Qualitative results on DAVIS soapbox shows we keep realistic colors details for longer compared to DEB [60] and DVCP [38]. (Best viewed in color on screen) (Color figure online)

after 30 frames, DEB shows color drifts [60, Fig. 16 arXiv] and oversmoothing even on the background clearly visible in the reference, and over-saturates the heads to red. In contrast, our model manages to keep the details – even after

occlusion (e.g. crowd in frame 30, was occluded in frame 15), while at the same time keeping the soapbox driver colorized, with some minimal color bleeding on the shirt, despite the drastic appearance and size changes. More qualitative results including a discussion on limiting cases can be found in the supplementary.

Historic Western - 2 References. Figure 11 demonstrates the extensibility of our method without re-training, on a historic scene from 1925. In frame 15, large



Fig. 11. Colorization of a historic sequence with FilmGrain - out of training domain; Without re-training our model can be extended to use multiple keyframe references.

portions of the arm and the jacket are occluded and are later visible with very different and new appearance. Therefore the matching to the global reference frame can be reduced and local feature matches take over. As a result, close-by similar textures dominate leading to wrong colorization. However, our flexible framework allows to add a slightly reworked frame as a second global reference. With the initial confidences and the multiwell dataterm, our method automatically selects best color proposal from *both* reference images. Using both also improves intermediate results, even though our method was never trained to work with two global references. An example on how to users can override the fusion of the color proposals can be found in the supplementary.

Historic Theater. Figure 12 compares our method to DEB [60] and DeepRemaster [21] – the best competitors with available source code – on a historic video of a theater play in 1902, with a manually colorized reference frame. While DeepRemaster loses colors, DEB over saturates them like on the floor, and shows a color drift on the wall. Both fail to keep details intact like the yellow sash.



Fig. 12. Colorized historic sequence from 1902; Mind details like the yellow sash, face, or color drift. Our method keeps colors more faithful to the manually colored reference. (Color figure online)

5 Conclusion

In this work, we proposed a method that successfully combines classical energy-based methods with deep learning to propagate colors in videos. Our method advanced the state-of-the-art – both quantitatively and qualitatively on multiple datasets and metrics as well as user ranking – even with much less training data. Further, our flexible mathematical structure allows for extensions like integrating additional references without retraining. Future work includes extension of user input capabilities and elaboration of loss functions such as adversarial losses.

Acknowledgement. This work was supported by the FFG-Program BRIDGE with short title RE:Color (No. 877161). Alexander Effland was also supported by the German Research Foundation under Germany’s Excellence Strategy EXC-2047/1-390685813 and EXC2151-390873048.

References

1. An, X., Pellacini, F.: Approp: all-pairs appearance-space edit propagation. In: ACM SIGGRAPH, pp. 1–9 (2008)
2. Anwar, S., Tahir, M., Li, C., Mian, A., Khan, F.S., Muzaffar, A.W.: Image colorization: a survey and dataset. [arXiv:2008.10774](https://arxiv.org/abs/2008.10774) (2020)
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **28**(3), 24 (2009)
4. Barron, J.T., Poole, B.: The fast bilateral solver. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9907, pp. 617–632. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_38

5. Bugeau, A., Ta, V.T., Papadakis, N.: Variational exemplar-based image colorization. *IEEE Trans. Image Process.* **23**(1), 298–307 (2013)
6. Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 151–166 (2017)
7. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**, 120–145 (2011)
8. Chambolle, A., Pock, T.: An introduction to continuous optimization for imaging. *Acta Numer* **25**, 161–319 (2016). <https://doi.org/10.1017/S096249291600009X>
9. Charpiat, G., Hofmann, M., Schölkopf, B.: Automatic image colorization via multimodal predictions. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008*. LNCS, vol. 5304, pp. 126–139. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88690-7_10
10. Chen, X., Zou, D., Zhao, Q., Tan, P.: Manifold preserving edit propagation. *ACM Trans. Graph.* **31**(6), 1–7 (2012)
11. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: *International Conference on Computer Vision*, pp. 415–423 (2015)
12. Deshpande, A., Lu, J., Yeh, M.C., Jin Chong, M., Forsyth, D.: Learning diverse image colorization. In: *Conference on Computer Vision and Pattern Recognition* (2017)
13. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: *International Conference on Computer Vision*, pp. 567–575 (2015)
14. Endo, Y., Iizuka, S., Kanamori, Y., Mitani, J.: Deepprop: extracting deep features from a single image for edit propagation. In: *Computer Graphics Forum*, pp. 189–201 (2016)
15. Faridul, H.S., Pouli, T., Chamaret, C., Stauder, J., Trémeau, A., Reinhard, E., et al.: A survey of color mapping and its applications. In: *Eurographics (State of the Art Reports)*, vol. 3 (2014)
16. 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)
17. He, M., Chen, D., Liao, J., Sander, P.V., Yuan, L.: Deep exemplar-based colorization. *ACM Trans. Graph.* **37**(4), 47 (2018)
18. Hensch, H.K., Hensch, B.A.: *The Painted photograph, 1839–1914: Origins, Techniques, Aspirations*. Pennsylvania State University Press University Park (1996)
19. Hofinger, M., Bulò, S.R., Porzi, L., Knapitsch, A., Pock, T., Kotschieder, P.: Improving optical flow on a pyramid level. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12373, pp. 770–786. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_46
20. Hurwitz, M.: Real war: how Peter Jackson’s they shall not grow old breathed life into 100-year-old archival footage (2019). <https://www.studiodaily.com/2019/05/real-war-peter-jacksons-shall-not-grow-old-breathed-life-100-year-old-archival-footage/>
21. Iizuka, S., Simo-Serra, E.: Deepre-master: temporal source-reference attention networks for comprehensive video enhancement. *ACM Trans. Graph.* **38**(6), 1–13 (2019)
22. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.* **35**(4), 1–11 (2016)
23. Irony, R., Cohen-Or, D., Lischinski, D.: Colorization by example. In: *Eurographics Symposium on Rendering*, pp. 201–210 (2005)

24. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
25. Jampani, V., Gadde, R., Gehler, P.V.: Video propagation networks. In: Conference on Computer Vision and Pattern Recognition, pp. 451–461 (2017)
26. Kobler, E., Effland, A., Kunisch, K., Pock, T.: Total deep variation for linear inverse problems. In: IEEE Conference on Computer Vision and Pattern Recognition (2020)
27. Kumar, M., Weissenborn, D., Kalchbrenner, N.: Colorization transformer. In: International Conference on Learning Representations (2021)
28. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 577–593. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_35
29. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
30. Lei, C., Chen, Q.: Fully automatic video colorization with self-regularization and diversity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3753–3761 (2019)
31. Leitner, D.: The documentary masterpiece that is Peter Jackson’s they shall not grow old (2018). <https://filmmakermagazine.com/106589-the-documentary-masterpiece-that-is-peter-jacksons-they-shall-not-grow-old>
32. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, pp. 689–694 (2004)
33. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint [arXiv:1705.01088](https://arxiv.org/abs/1705.01088) (2017)
34. Lin, T.-S.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
35. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.Q., Shum, H.Y.: Natural image colorization. In: Eurographics Conference on Rendering Techniques, pp. 309–320 (2007)
36. Luo, M., Cui, G., Rigg, B.: The development of the CIE 2000 colour-difference formula: Ciede 2000. *Color Res. Appl.* **26**, 340–350 (2001). <https://doi.org/10.1002/col.1049>
37. Meister, S., Hur, J., Roth, S.: Unflow: unsupervised learning of optical flow with a bidirectional census loss. In: AAAI (2018)
38. Meyer, S., Cornillère, V., Djelouah, A., Schroers, C., Gross, M.: Deep video color propagation. In: British Machine Vision Conference (2018)
39. Mouzon, T., Pierre, F., Berger, M.O.: Joint CNN and variational model for fully-automatic image colorization. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 535–546 (2019)
40. Pierre, F., Aujol, J.F., Bugeau, A., Papadakis, N., Ta, V.T.: Luminance-chrominance model for image colorization. *SIAM J. Imaging Sci.* **8**(1), 536–563 (2015)
41. Pierre, F., Aujol, J.F., Bugeau, A., Ta, V.T.: Interactive video colorization within a variational framework. *SIAM J. Imaging Sci.* **10**(4), 2293–2325 (2017)
42. Pierre, F., Aujol, J.F.: Recent Approaches for Image Colorization (2020). <https://hal.archives-ouvertes.fr/hal-02965137>

43. Pierre, F., Aujol, J.-F., Bugeau, A., Ta, V.-T.: A unified model for image colorization. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) ECCV 2014. LNCS, vol. 8927, pp. 297–308. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16199-0_21
44. Pinetz, T., Kobler, E., Pock, T., Effland, A.: Shared prior learning of energy-based models for image reconstruction. [arXiv:2011.06539](https://arxiv.org/abs/2011.06539) (2020)
45. Pont-Tuset, J., et al.: The 2017 davis challenge on video object segmentation. [arXiv:1704.00675](https://arxiv.org/abs/1704.00675) (2017)
46. Qu, Y., Wong, T.T., Heng, P.A.: Manga colorization. *ACM Trans. Graph.* **25**(3), 1214–1220 (2006)
47. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Comput. Graphics Appl.* **21**(5), 34–41 (2001)
48. Royer, A., Kolesnikov, A., Lampert, C.H.: Probabilistic image colorization. In: British Machine Vision Conference (2018)
49. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **115**(3), 211–252 (2015)
50. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 761–769 (2016)
51. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
52. Su, J.W., Chu, H.K., Huang, J.B.: Instance-aware image colorization. In: Conference on Computer Vision and Pattern Recognition, pp. 7968–7977 (2020)
53. Šykora, D., Buriánek, J., Žára, J.: Unsupervised colorization of black-and-white cartoons. In: International Symposium on Non-photorealistic Animation and Rendering, pp. 121–127 (2004)
54. Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 402–419. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_24
55. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: the missing ingredient for fast stylization. [arXiv:1607.08022](https://arxiv.org/abs/1607.08022) (2016)
56. Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., Murphy, K.: Tracking emerges by colorizing videos. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 402–419. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_24
57. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. In: Conference on Computer Graphics and Interactive Techniques, pp. 277–280 (2002)
58. Xu, L., Yan, Q., Jia, J.: A sparse control model for image and video editing. *ACM Trans. Graph.* **32**(6), 1–10 (2013)
59. Yatziv, L., Sapiro, G.: Fast image and video colorization using chrominance blending. *IEEE Trans. Image Process.* **15**(5), 1120–1129 (2006)
60. Zhang, B., et al.: Deep exemplar-based video colorization. In: Conference on Computer Vision and Pattern Recognition, pp. 8052–8061 (2019)
61. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_40

62. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, pp. 586–595 (2018)
63. Zhang, R., et al.: Real-time user-guided image colorization with learned deep priors. *ACM Trans. Graph.* **36**(4), 1–11 (2017)