

Automatic Colorization of Digital Movies using Decolorization Models and SSIM Index

Andrzej Śluzek, Marcin Dudziński, Tomasz Świsłocki
 0000-0003-4148-2600, 0000-0003-4242-8411, 0000-0002-7679-7533
 Warsaw University of Life Sciences - SGGW,
 Institute of Information Technology,
 Nowoursynowska 159, bld.34,
 02-776 Warsaw, Poland

Email: {andrzej_sluzek, marcin_dudzinski, tomasz_swislocki}@sggw.edu.pl

Abstract—Re-colorization of images or movies is a challenging problem due to the infinite RGB solutions for a monochrome object. In general, the process is assisted by humans, either by providing colorization hints or relevant training data for ML/AI algorithms. Our intention is to develop a mechanism for fully unguided (and with no training data used) colorization of movies. In other words, we aim to create acceptable colored counterparts of movies in domains where only monochrome visualizations physically exist (e.g. IR, UV, MRI, etc. data). Following our past approach to image colorization, the method assumes arbitrary *rgb2gray* models and utilizes a few probabilistic heuristics. Additionally, we maintain the temporal stability of colorization by locally using *structural similarity* (SSIM) between adjacent frames. The paper explains the details of the method, presents exemplary results and compares them to the *state-of-the-art* solutions.

NOTE: All figures are best viewed in color and high resolution.

I. INTRODUCTION AND MOTIVATION

COLORIZATION of monochrome objects is an ill-posed problem due to the infinite number of RGB solutions for given grayscale data. Nonetheless, this topic holds notable practical and commercial significance, particularly in the restoration of historical photos and movies, e.g. [1], [2].

In general, the development of colorization techniques involves incorporating more and more human knowledge and expectations into the algorithms [1]. To that end, earlier methods involved providing reference color images [3], [4] or manually *scribbling* important fragments [5], [6]. Recently, AI-based techniques have dominated, with architectures designed to learn color patterns suitable for specific domains, semantics, and/or contents, such as [7], [8].

Some works consider recognizing/learning the image domain (or specific objects within images) to further improve the results, e.g., [9], [10].

In other works, inter-domain transfer learning is considered [11], or multiple alternative colorizations are proposed by exploiting learned probability densities [12], [13].

Colorization methods for monochrome movies basically follow the same principles. In some works (for example, [2]), there is no distinction between the colorization of still images and movies. That is, each frame of a movie is considered a separate item and is re-colored individually. However,

such an approach may introduce temporal colorization discontinuities. Therefore, in recent works, including [14], [15] or [16], the issue of colorization continuity between adjacent frames is addressed, mainly by combining spatial and temporal consistencies of colors assigned to pixels with similar intensity levels.

Nevertheless, in all typical papers on re-colorization of monochrome objects, the algorithms are assisted by human knowledge/experience/expectations, even if the assistance is disguised as *training datasets* of relevant images/videos.

Therefore, the solution discussed in this paper may seem audacious. We propose a mechanism for fully automatic video colorization without additional metadata, manual assistance, learning processes, or domain identification. In other words, we aim to create plausible colored representations of *gray worlds* using only monochrome videos as the data source. By "plausible," we mean results that are visually attractive, statistically repeatable, and deliver convincingly rich sensations of colors.

It should be noted that we exclude simple pseudo-coloring techniques that use a limited number of colors corresponding to the number of intensity levels, resulting in a limited richness and diversity of coloristic effects.

The problem addressed appears to be relevant and practical because there are numerous domains, such as infrared, ultraviolet, ultrasound, MRI, X-ray, and others, where only single-channel visual data physically exist. Then, realistic-looking automatic colorizations can be synthesized for various reasons, even if it is just for aesthetic purposes (see examples in Fig. 1).

In our recent papers [17], [18], we overviewed the results obtained for unguided colorization of monochrome images, and the method for colorizing videos is a natural extension of those results. Therefore, Section II revisits a number of assumptions and techniques adopted in monochrome image colorization and repeated in video colorization.

Section III presents steps that are specific to the colorization of videos. A summary of the experimental results and corresponding comments is provided in Section IV. Finally, the concluding Section V discusses some supplementary details (mainly related to limiting the number of solutions) and outlines prospective directions for future research.

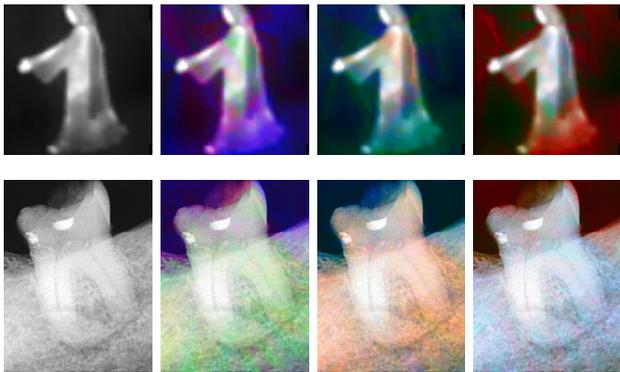


Fig. 1. Examples of monochrome images from non-visual domains (IR and X-ray), and their color-rich and visually plausible colorizations.

II. PRINCIPLES OF AUTOMATIC COLORIZATION

A. Significance of Decolorization Models

Generally, colorization methods assume that the intensity of grayscale objects defines the luminance channel of the colored outputs, so only two channels of chrominance need to be reconstructed. It seems like a matter of arbitrary choice which chrominance models are used (e.g., CIELab in [8], [9] or YUV in [2], [5]). Sometimes, the model is not even specified, and only the final outcomes in RGB format are discussed.

However, almost no papers on re-colorization address the opposite question: **How was the original (real or hypothetical) RGB object decolorized to obtain a grayscale object?**

Typically applied decolorization (*rgb2gray*) models, i.e. YUV or YIQ, assume linear functions of primary colors:

$$I = k_R R + k_G G + k_B B \quad (1)$$

where $k_R = 0.299$, $k_G = 0.587$ and $k_B = 0.114$ (or, based on [19], $k_R = 0.2126$, $k_G = 0.7152$ and $k_B = 0.0722$).

Nevertheless, given that we assume that colored objects only exist hypothetically, any combinations of non-negative coefficients k_R , k_G , and k_B can be formally used (subject to the straightforward condition $k_R + k_G + k_B = 1$).

Therefore, as shown in Fig. 2, a colored object can be converted into a variety of its monochrome counterparts, which may significantly differ depending on the adopted *rgb2gray* model.



Fig. 2. A colored image and its three monochrome variants obtained using three different *rgb2gray* models, namely $[0.299, 0.587, 0.114]$, $[0.44, 0.14, 0.42]$ and $[0.14, 0.11, 0.75]$.

Correspondingly, by assuming a certain *rgb2gray* model, we can restrict the re-colorization results in a specific way.

Any pixel with intensity I can only be assigned colors that satisfy Eq. 1 with the adopted $[k_R, k_G, k_B]$ values. Fig. 3 shows examples of the same image re-colored using the same algorithm, but with different *rgb2gray* models adopted.



Fig. 3. The same monochrome image colored (by the method discussed in [17], [18]) with different *rgb2gray* models adopted.

B. Colorization of Individual Pixels

In coloring monochrome objects depicting some non-existent *color worlds*, we are not restricted by the properties of YUV or YIQ models, which provide the best consistency between the brightness of the color and its monochrome counterpart, as perceived by human observers. Therefore, in the developed colorization scheme, we assume that:

Monochrome visual objects are derived from (only hypothetically existing) color counterparts by an *rgb2gray* model with arbitrarily selected k_R , k_G , and k_B coefficients.

If we assume that the coefficients are uniformly sampled into n values each (for example, $n = 101$ for a 0.01 stepsize), it can be easily obtained that the total number of available *rgb2gray* models is:

$$N = \frac{n(n+1)}{2} \text{ or } N = \frac{(n-3)(n-2)}{2} \quad (2)$$

The first variant includes zero-coefficient models (e.g., $[0.4, 0, 0.6]$ or $[0, 1, 0]$). These models are excluded in the second variant, where each primary color must have a non-zero contribution to the intensity levels.

In the experiments, we use 0.05, 0.01 or 0.002 stepsizes, so that the total numbers of *rgb2gray* models are 231(171), 5151(4851) or 125,751(124,251). However, in the end, we only consider a limited number of 20 – 40 models (see Sections IV and V for more details), resulting in the same number of alternative colorization outputs.

In rendering digital images, we adopt finite numbers of intensities and colors. Thus, given a monochrome intensity level I_p from the discrete range 0 to 255, it can only be assigned colors from a pool of colors that satisfy (subject to color discretization) Eq. 1. The size of those pools varies, depending on the intensity level. As an example, the numbers of colors assigned to intensity levels in two arbitrarily selected *rgb2gray* models are shown in Fig. 4.

The figure shows that the largest pools of colors are for mid-range intensities, with the number of colors gradually decreasing to a single choice for extremely dark or light intensities. (Monochrome white/black should remain white/black in color.)

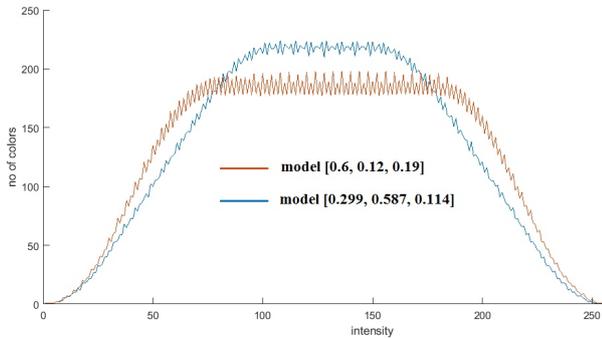


Fig. 4. Numbers of RGB colors assigned to intensities in two exemplary *rgb2gray* models.

As an example, Fig. 5 displays the pool of colors for intensity level 208 under two *rgb2gray* models. Note that the first model is actually YUV, and all colors are perceived as having almost the same brightness. In contrast, the perceived brightness of colors varies significantly for the second model.

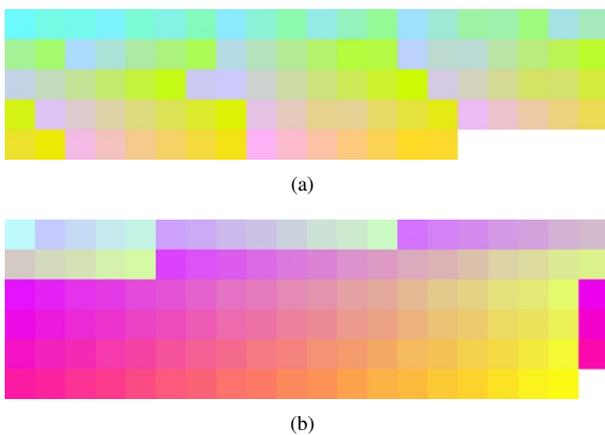


Fig. 5. Colors assigned to 208 intensity in (a) [0.299, 0.587, 0.114] and (b) [0.69, 0.12, 0.19] *rgb2gray* models.

With no prior information provided, all colors available to the I_p level can be assigned to a pixel of that intensity with the same probability. However, if the pixel has an adjacent pixel with an intensity I_1 and its already assigned color C_1 , the probabilities of colors that could be assigned to I_p should be influenced by the intensity and color of the neighbor.

Therefore, we propose a simple but (as shown later) surprisingly effective heuristic rule:

The greater the difference in brightness between adjacent pixels, the higher the likelihood that their assigned colors will also differ significantly.

Under this rule, we prioritize colors from the pool available to the I_p level, which are at distances from the C_1 color proportional to the difference in intensity levels $\|I_p - I_1\|$.

Let's assume a pixel with I_p intensity, which has an already colored neighbor with I_1 value and C_1 color. Let $\mathbf{C} = \{C_{p_1}, \dots, C_{p_N}\}$ be the list of colors assigned to I_p in the adopted *rgb2gray* model.

The neighbor (with I_1 value and C_1 color) contributes a color from the above \mathbf{C} list. First, the list is ordered by the distances of its colors from C_1 , i.e. $\mathbf{Cmod} = \{C_{p_{i_1}}, \dots, C_{p_{i_N}}\}$, where

$$\|C_{p_{i_n}} - C_1\| \leq \|C_{p_{i_{(n+1)}}} - C_1\| \quad (3)$$

It should be noted that inter-color distances are measured in the HSV space, as differences in this space are sufficiently close to represent perceived color similarities, e.g. [20].

Then, a uniform distribution is used to randomly select a color from a specific sub-range of the \mathbf{Cmod} list. The location of this sub-range depends on the difference $\|I_p - I_1\|$. In general, for smaller differences, the sub-range is narrower and shifted to the left of \mathbf{Cmod} , while for larger differences, it is wider and shifted towards the end of \mathbf{Cmod} . Detailed description of this step can be found in [18].

In particular, if the neighboring pixel has the same intensity, i.e. $I_p = I_1$, the I_p intensity pixel would be (preliminarily) assigned the same color C_1 .

Actually, images are colored incrementally (details in the following subsection), and it may happen that an uncolored pixel has several already-colored neighbors. Then, the color selection can be performed several times for that pixel, and the final choice is the mean of the colors obtained from all colored neighbors.

$$C_p = \frac{1}{K} \sum_{j=1}^K C_j \quad (4)$$

where $K = 1, 2, 3$ or 4 .

This step allows us to generate a larger variety of colors than from the (possibly limited) pools of colors that are assigned to individual intensity levels.

C. Incremental Colorization

Colorization of monochrome images is performed incrementally, starting from the darkest and brightest pixels, for which the unique color choice exists (see Fig. 4). They are considered the initial list of already colored pixels for the algorithm, which is actually a randomized variant of a popular *flood-fill* method.

Pseudo-code of the colorization algorithm is provided below.

Step 4 (which does not exist in the standard flood-fill method) is introduced to randomize the expansion of colorized patches, i.e. to avoid unnecessary regularities in the colorization process.

Algorithm 1 Monochrome image colorization**Require:** Initial list L of colored pixels**Ensure:** Colorized images

```

1: remove from  $L$  pixels which do not have uncolored
   neighbors;
2: copy  $L$  to  $Q$  queue
3: while  $Q$  is not empty do
4:   move a random pixel from  $Q$  to the front of  $Q$ ;
5:   get the front pixel  $p$  of  $Q$ ;
6:   if  $ps$  (south neighbor of  $p$ ) exists and is uncolored then
7:     colorize  $ps$  (Eq. 4);
8:     append  $ps$  to the end of  $Q$ ;
9:   end if
10:  repeat Steps 6-9 for  $pn$ ,  $pe$ ,  $pw$  (north, east, west
     neighbors of  $p$ );
11:  remove  $p$  from  $Q$ ;
12: end while

```

D. Other Remarks

Despite the heavy presence of randomizing factors, the results produced by the outlined method are surprisingly repeatable, depending only (as expected) on the adopted *rgb2gray* model. Actually, all images shown in Figs 1 and 3 are generated by the method, and many more examples can be found in [17], [18].

The visual plausibility of the results can be further improved by projecting the colors of each pixel p onto the corresponding YUV plane $I_p = 0.299R + 0.587G + 0.114B$ (or $I_p = 0.2126R + 0.7152G + 0.0722B$). This is because colors assigned to the same intensity level can vary significantly in terms of their perceived brightness, see Fig. 5b. Projecting the colors on the YUV planes unifies the perceived brightness of colors with intensities of the original monochrome image, although it may alter the colors somewhat, as shown in Fig. 6.

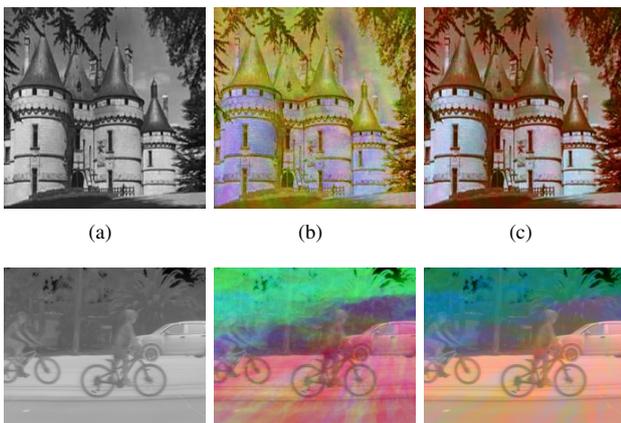


Fig. 6. Original monochrome images (a) and their colorizations before (b) and after (c) projections on the YUV planes.

III. FROM IMAGES TO MOVIES

Formally, the colorization of monochrome movies does not differ from the colorization of images, and both operations are often considered to be almost equivalent, e.g., [21]. Each frame can be independently colorized to provide the corresponding frame of the color movie. Such a simplified approach may be used if image colorization performs nearly flawlessly, which can only be achieved in specific and well-defined domains. Nevertheless, this method has been successful in a number of works, including the commercial system described in [2].

In our application, we cannot use this approach due to randomizing factors in the image colorization scheme. In such cases, even if the same *rgb2gray* model is used, adjacent frames with nearly identical content may have noticeably different colorization. An example is given in Fig. 7. Thus, the temporal continuity/stabilization of rendered colors over sequences of similar frames is particularly important for our problem.

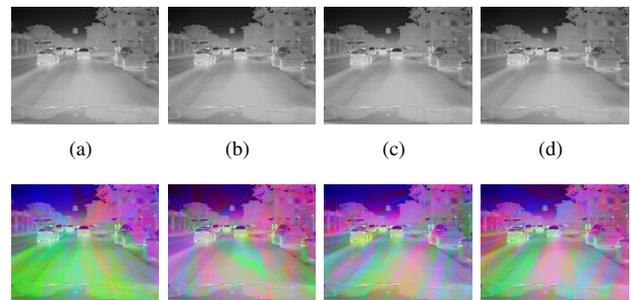


Fig. 7. Four subsequent monochrome frames individually colorized (using the same *rgb2gray* model) by the method outlined in Section II.

Recently, more attention has been paid to the temporal continuity of colors in the colorized movies. Several papers, e.g., [14], [15], [16], consider the regularization or stabilization of colors in adjacent video frames. In [22], a more general problem of stabilizing any visual properties in processed video files is considered.

Nonetheless, in all these papers, the results are obtained by training dedicated neural networks on sufficiently representative ground-truth data. Therefore, these approaches are not applicable to the considered problem of colorizing monochrome videos for which the color counterparts never existed.

In the proposed solution, we assume that the colorization continuity should reflect not only intensity similarities between adjacent frames but also content similarities, particularly the local ones.

After evaluating several popular metrics of image quality and similarity, e.g. [23], [24], the *structural similarity index measure* (SSIM) has been identified as the top candidate, [25]. SSIM is particularly suitable for monochrome images and is applicable to images of any content.

In general, the SSIM is defined by a weighted combination of three measures that broadly represent statistical similarities between the intensity, contrast, and structure of two image

samples, X and Y :

$$SSIM(X, Y) = i(X, Y)^\alpha \times c(X, Y)^\beta \times s(X, Y)^\gamma \quad (5)$$

where $i(X, Y) = \frac{2\mu_X\mu_Y+c_1}{\mu_X^2+\mu_Y^2+c_1}$, $c(X, Y) = \frac{2\sigma_X\sigma_Y+c_2}{\sigma_X^2+\sigma_Y^2+c_2}$ and $s(X, Y) = \frac{\sigma_{XY}+c_3}{\sigma_X\sigma_Y+c_3}$.

Depending on the area over which means and standard deviations are computed, SSIM can indicate image similarities either globally (i.e., between whole images) or locally (i.e., between small neighborhoods of the same coordinates (m, n) in both images).

When coloring the current frame in the context of the previous one, we actually utilize both aspects of SSIM.

First, we calculate the global SSIM measure between the current monochrome frame (to be colored) and its monochrome predecessor (already colored). If the value is too low (the recommended threshold is 0.5), we assume no perceptual similarity between the frames, and the current frame is colored independently.

In practice, such situations occur infrequently, mainly when assembling longer movies from shorter unrelated fragments, and high values of global SSIM similarity between adjacent frames can typically be expected. For example, the SSIM values between the neighboring pairs of monochrome frames in Fig. 7 are as follows: 0.9511 (frames a, b), 0.9756 (frames b, c), and 0.9508 (frames c, d).

Therefore, we normally define colors of the colorized frame I_k as weighted combinations of the independent colorization of I_k and colors of the previous frame I_{k-1} , i.e., for given pixel coordinates (m, n) we use the *local* values of SSIM:

$$Col_k(m, n) = sim \times Col_{k-1}(m, n) + (1 - sim) \times Col_k(m, n) \quad (6)$$

where $sim = SSIM(I_k(m, n), I_{k-1}(m, n))$.

Additionally, the colors computed by Eq. 6 are projected onto the corresponding YUV planes (defined by pixel intensities in frame I_k) as explained in Section II-D.

Fig. 8 provides exemplary effects of the proposed color regularization over neighboring frames. First, we display the local SSIM indexes between the monochrome frames from Fig. 7 in Figs 8(a-c). Then, the lower row of Fig. 8 shows the colored frames after the regularization.

IV. EXPERIMENTS

The proposed method of monochrome movie colorization involves heuristic assumptions, arbitrary model selection, and probabilistic computational schemes. Furthermore, the obtained results cannot be objectively assessed since reference or ground-truth results are assumed to be non-existent.

Therefore, the performance of the method and quality of its outputs can only be evaluated through extensive experimentation. In particular, the final results are typically assessed using subjective criteria such as *visual plausibility*, *coloristic attractiveness*, *aesthetic value*, etc.

One of the main challenges in conducting such experiments is the large number of potential *rgb2gray* models, as discussed

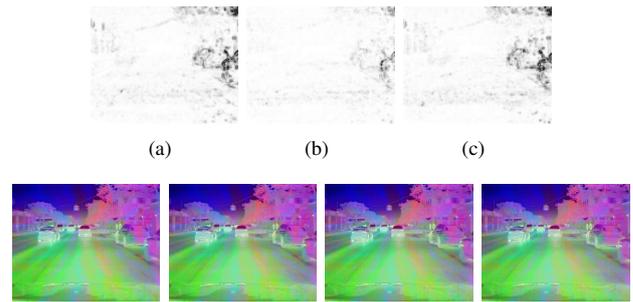


Fig. 8. SSIM maps (intensities proportional to the numerical values) for Fig. 7 pairs of monochrome frames: (a) for frames a, b , (b) for frames b, c and (c) for frames c, d . The bottom row displays the results after the color regularization.

in Subsection II-B. The sheer quantity of alternative colorizations may be overwhelming for human evaluators. Therefore, it is necessary to reduce the number of effectively considered *rgb2gray* models.

For moderate numbers of adopted *rgb2gray* models, e.g., 231(171) models with 0.05 stepsize, we found that the most plausible solutions are normally obtained from the 10 – 15% of results with the lowest value of *colorfulness* (details of this metric are provided in [1], [26]). Specifically, when the monochrome objects depict scenes from the real world, this subset typically includes solutions that vaguely align with human coloristic expectations (more information in [17]).

With a large number of models, e.g., 5151(4851) or 125,751(124,251), the models are preliminarily clustered into a recommended number of classes (20 – 40), and only the cluster medoids are used. Details of the clustering algorithm are outlined in Subsection V-A.

In any case, the users are presented with a limited number of suggested colorizations, from which they can select the most satisfactory option. Subject to the quality constraints of the original monochrome movies, the results always appear attractive and convincing, resembling scenes from ‘fairy tale lands’. Therefore, the preferred colorized version becomes a matter of personal choice.

In the experiments, the non-visual (IR) movies mainly come from FLIR ADAS¹ and CAMEL [27] datasets. The visual-frequency movies, which are used to better highlight the differences between our approach and the ‘traditional’ re-colorization expectations, primarily come from personal collections.

This section includes short representative frame sequences as illustrations. For example, Fig. 9 once again confirms that colorization supported by SSIM-based regularization provides a natural-looking continuity of colored frames, free of flickering and artifacts.

Fig. 10 showcases a rather unusual (but occasionally possible) result of re-colorization of a real-world movie. It can be observed that the selected colorization option appears even more natural than the original color movie!

¹<https://www.flir.eu/oem/adas/adas-dataset-form/>

Similar effects can be observed in Fig. 11, where the re-colored frames of an underwater movie appear more authentic than the original shots.

Nevertheless, in typical cases, even if the monochrome movie has ground-truth colors, there is no correspondence between the original colors and their re-colored versions. In other words, there is no distinction between colorizing monochrome movies with or without existing color originals. The results appear visually plausible, but they may or may not meet the ground-truth coloristic expectations, with the latter being more typical (see examples in Fig. 12).

Finally, Fig. 13 provides exemplary colorization results (arbitrarily selected from a number of alternative results) for a sequence of frames extracted from an outdoor IR movie.

Overall, the experimental results confirm that plausible video colorization can be achieved fully automatically, without the need for learning from relevant training data, human assistance, or supplementary metadata. In other words, it appears possible to synthesize realistically-looking colorful immersion into the 'gray worlds' of monochrome visual data. With the mechanisms provided for pre-selecting the 20 – 40 most promising *rgb2gray* models, users can choose their preferred colorization version from a limited yet sufficiently diverse number of alternative solutions.

However, considering that only subjective assessment criteria are currently utilized, presenting the actual video clips would be a more suitable approach to report the experimental results.

A. Comparing to SOTA

It is generally assumed that AI-based methods deliver *state-of-the-art* (SOTA) results for the colorization of monochrome objects. In particular, some works on video colorization, including [14], [16], use individual frame colorization by SOTA AI methods as benchmarks for the proposed algorithms.

Following the same approach, we colorized sequences of frames from the tested videos using the publicly available (at <https://deepai.org/machine-learning-model/colorizer>) tool which applies one of the most advanced re-colorization methods (outlined in [2]).

The results shown in Fig. 14 are utterly disappointing. While *visual plausibility* is basically unchanged compared to the monochrome images, the other subjective criteria, such as *coloristic attractiveness* or *aesthetic value* fall far below expectations. The color outputs are almost direct replicas of the grayscale values from the monochrome images. Apparently, when facing unfamiliar contents, the algorithm decides to keep the original monochrome colorization. In other words, satisfactory and visually attractive AI-based colorization is not possible for monochrome objects for which no coloristic knowledge or experiences are available. Therefore, the practicality of the proposed approach is somewhat boosted.

TABLE I
COEFFICIENTS $[k_R, k_G, k_B]$ OF 32 ADOPTED *rgb2gray* MODELS.

[0.04, 0.77, 0.19]	[0.05, 0.18, 0.77]
[0.05, 0.05, 0.90]	[0.25, 0.13, 0.62]
[0.83, 0.05, 0.12]	[0.59, 0.11, 0.30]
[0.12, 0.71, 0.17]	[0.75, 0.21, 0.04]
[0.50, 0.20, 0.30]	[0.25, 0.70, 0.05]
[0.42, 0.44, 0.14]	[0.04, 0.91, 0.05]
[0.14, 0.39, 0.47]	[0.58, 0.38, 0.04]
[0.40, 0.33, 0.27]	[0.70, 0.04, 0.26]
[0.14, 0.55, 0.31]	[0.04, 0.37, 0.59]
[0.52, 0.04, 0.44]	[0.71, 0.15, 0.14]
[0.34, 0.04, 0.62]	[0.41, 0.55, 0.04]
[0.26, 0.44, 0.30]	[0.13, 0.24, 0.63]
[0.28, 0.27, 0.45]	[0.04, 0.58, 0.38]
[0.40, 0.14, 0.46]	[0.12, 0.83, 0.05]
[0.57, 0.28, 0.15]	[0.26, 0.58, 0.16]
[0.16, 0.06, 0.78]	[0.91, 0.06, 0.03]

V. CONCLUDING REMARKS

A. Limiting the number of alternative solutions

The main practical obstacle in prospective applications of the proposed approach is (as highlighted in Subsection II-B and Section IV) the large number of available *rgb2gray* models. Human observers are unable to assess all possible colorization outputs, so limiting the number of models to a limited (but sufficiently diversified in terms of the produced results) is an important issue. This is the outline of the proposed remedy.

As an alternative to the representations given in Figs 4 and 5, the colors assigned to the selected intensity I can be visualized as the polygonal intersection of the $k_R R + k_G G + k_B B = I$ plane with the RGB color space cube. An example is provided in Fig. 15 (note locations of the centers of gravity of the depicted polygons).

Thus, the 256×3 matrix of gravity center coordinates of such polygons for $I = 0, \dots, 255$ can be considered a compact representation of the adopted *rgb2gray* model. The matrix can be nicely visualized by a (discrete) curve winding from *black* to *white* in the RGB cube (see examples in Fig. 16). Those curves will be referred to as *mean-color curves*.

Then, the *rgb2gray* models can be clustered by clustering their *mean-color curves* (i.e. 256-dimensional arrays of 3D coordinates). Eventually, *medoids* of the obtained clusters are identified, and only the models corresponding to those medoids are used for colorization.

Therefore, we adopt a limited number of models (for example, only 32 clusters are built regardless the total number of models) which are as diversified as possible in terms of their statistical coloristic properties.

For the total number of 4851 *rgb2gray* models (i.e., 0.01 stepsize), the list of adopted model is given in Table I. It can be noted that one of the models, namely [0.26, 0.58, 0.16], is quite similar to the standard YUV model with [0.299, 0.587, 0.114] coefficients.

B. Summary

In this paper, we propose a method for addressing the ill-posed problem of colorizing monochrome movies without

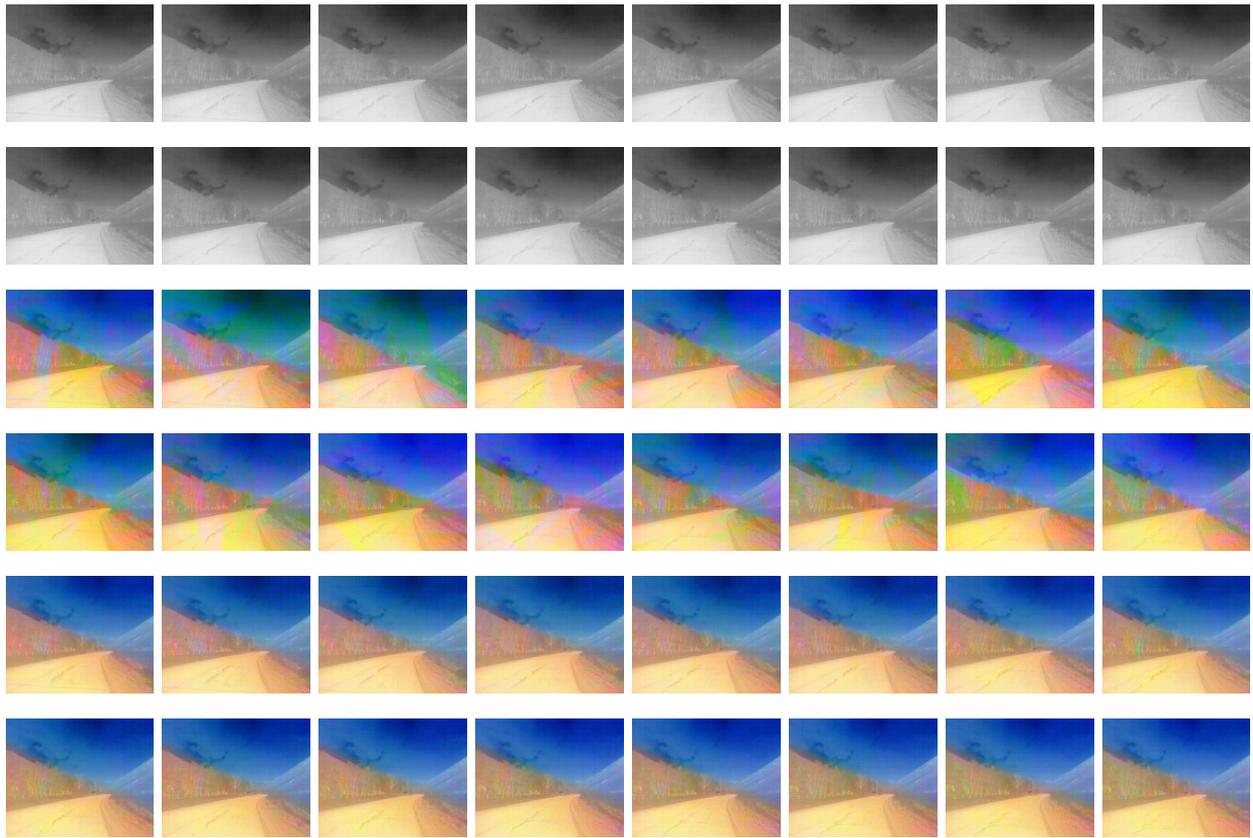


Fig. 9. Monochrome frames of an IR movie (two top rows), frames colorized individually (two middle rows), and frames colorized with the color regularization (two bottom rows).

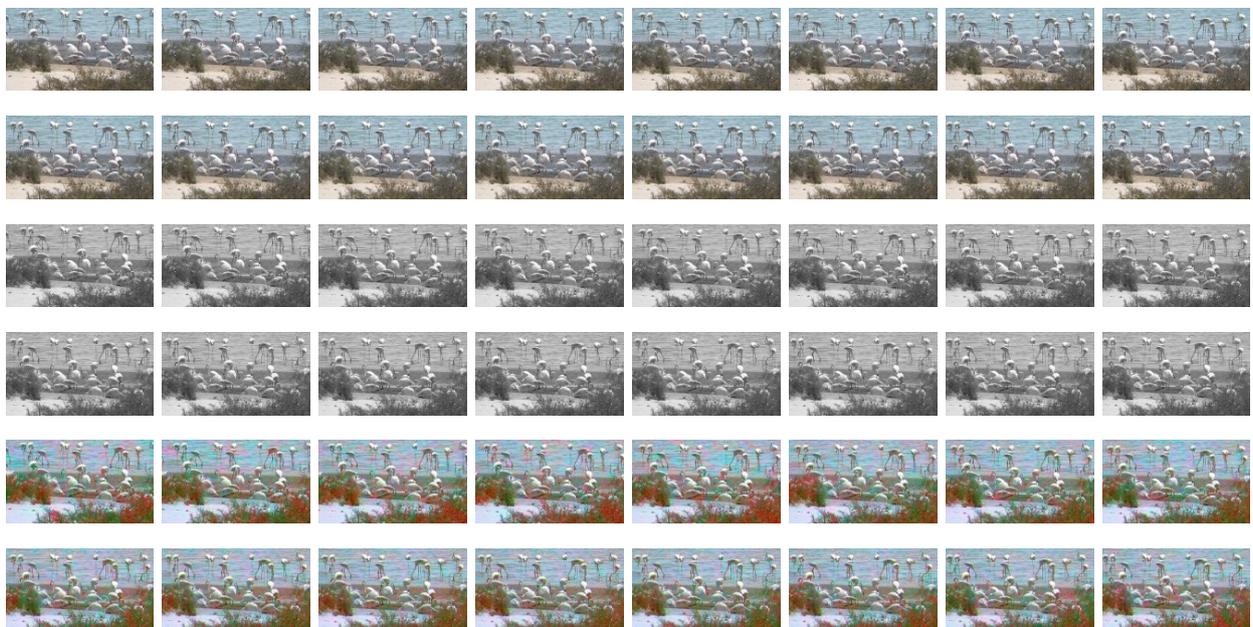


Fig. 10. The original color frames of a movie (two top rows), their decolorized variants (two middle rows), and one of the achieved re-colorization options (two bottom rows).

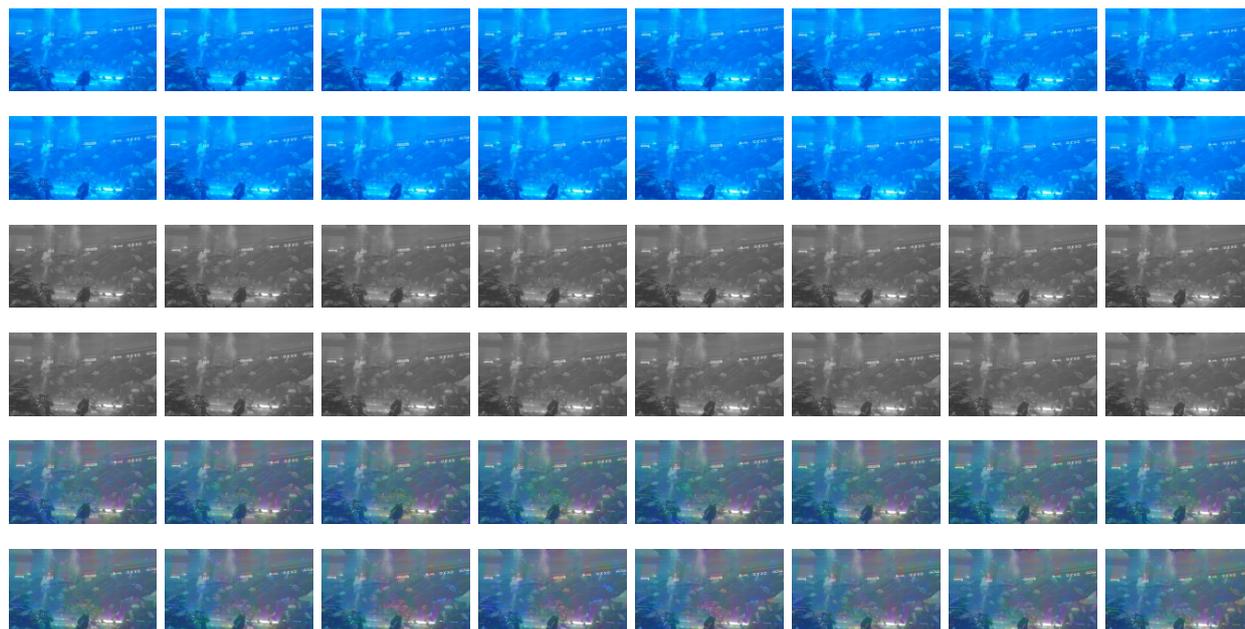


Fig. 11. The original color frames of an underwater movie (two top rows), their decolorized variants (two middle rows), and one of the achieved re-colorization options (two bottom rows).



Fig. 12. The original color frames (two top rows), and their exemplary re-colorization options (pairs of lower rows).

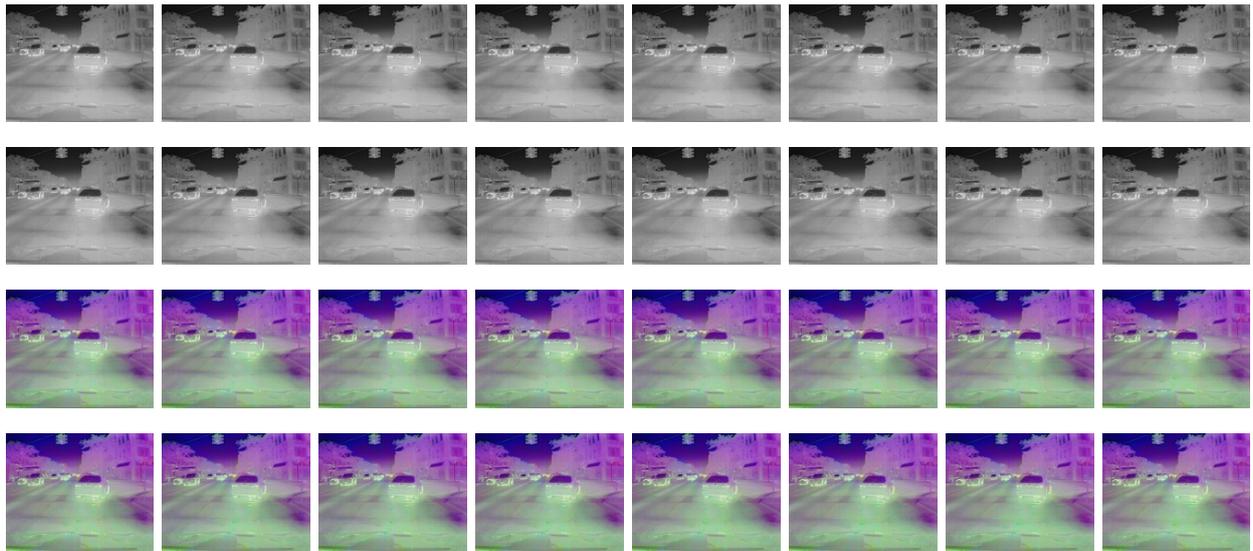


Fig. 13. An exemplary sequence of monochrome frames from an IR movie (top two rows) and its arbitrarily selected colorization (bottom two rows).

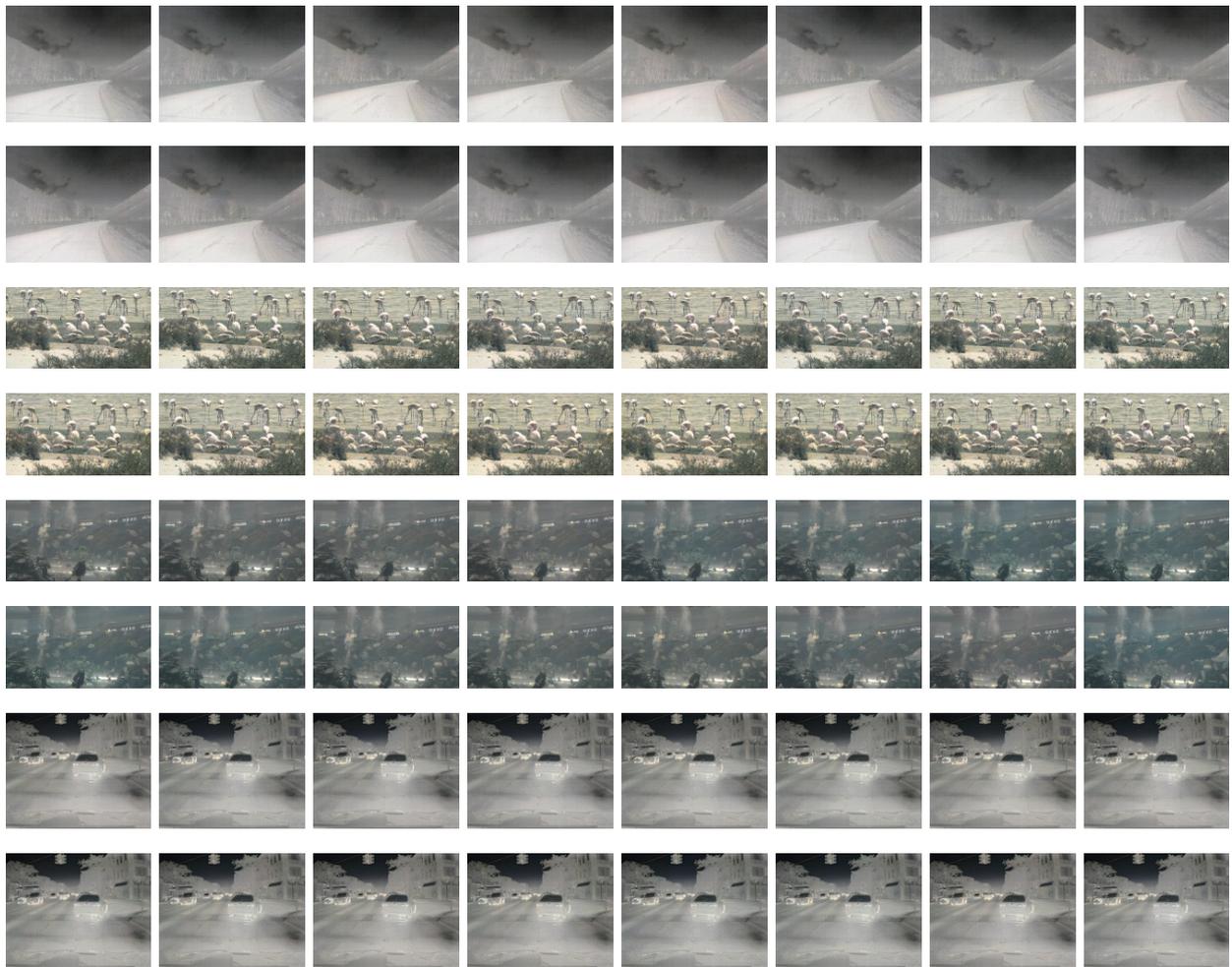


Fig. 14. Sequences of monochrome frames from (top to bottom) Figs 9, 10, 11 and 13 colorized by an AI-based SOTA algorithm.

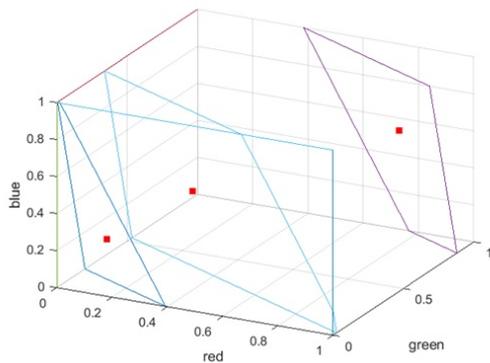


Fig. 15. Distribution of colors assigned to three exemplary intensities 30, 80 and 208 under the $I = 0.299R + 0.587G + 0.114B$ *rgb2gray* model. Red dots indicate the centers of gravity of the intersection polygons.

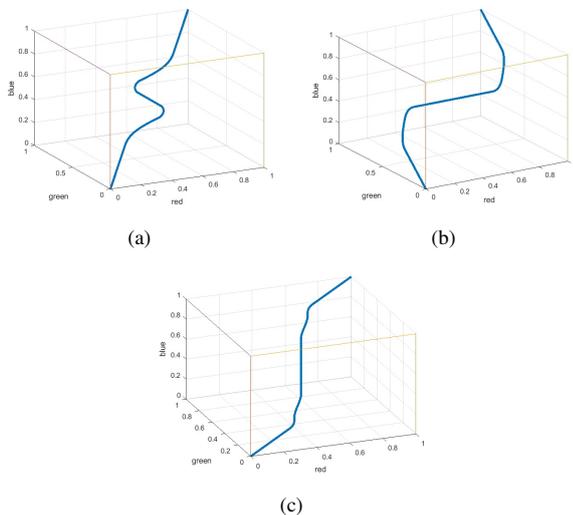


Fig. 16. Mean-color curves representing locations of polygon centers for intensity levels ranging from 0 to 255 for three *rgb2gray* models: (a) [0.299, 0.587, 0.114], (b) [0.69, 0.12, 0.19] and (c) [0.13, 0.175, 0.695].

any direct or indirect human assistance. Our method builds upon our recent results in colorization of monochrome images, where we assume the use of arbitrary decolorization (*rgb2gray*) models.

The movie colorization process involves two operations: image colorization and temporal stabilization of rendered colors. First, individual frames are colored using simple probabilistic heuristics and a randomized *flood-fill* technique, starting from the initial queue of darkest/brightest pixels with deterministic color choices. In the second operation, we utilize the SSIM similarity index to determine whether and to what extent color continuity should be maintained between adjacent frames.

While a large number of *rgb2gray* models can hypothetically be used, we can pre-select a limited number of sufficiently diverse variants. Users can then choose their preferred colorization from these options, typically based on personal

preference.

The method is primarily designed for colorizing monochrome movies in domains where no actual color data exists, such as IR, UV, MRI, etc. In other words, our goal is to transform the monochrome data into convincingly realistic color versions of these *gray worlds*. This may be necessary for various reasons, including aesthetic considerations.

In future work, our intention is to focus on the following problems that have not yet been adequately addressed:

- Analysis of the mathematical properties of the method, which includes exploring alternative probability distributions used in the adopted heuristics, investigating the local (individual frames) and global (movies) convergence of colorization results, etc.
- Developing metrics for the objective evaluation of colorization results, including the selection of appropriate assessment criteria.
- Optimizing the code, including parallelization techniques, optimizing data structures, and other strategies. The ultimate objective may involve achieving real-time performance.
- Integrating the method with selected AI techniques to enhance its capabilities and explore potential synergies.

REFERENCES

- [1] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, "Grayscale image colorization methods: Overview and evaluation," *IEEE Access*, vol. 9, pp. 113 326–113 346, 2021. doi: 10.1109/ACCESS.2021.3104515
- [2] A. Salmona, L. Bouza, and J. Delon, "Deoldify: A review and implementation of an automatic colorization method," *Image Processing On Line*, vol. 12, pp. 347–368, 2022. doi: 10.52011/ipol.2022.403
- [3] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Eurographics Symposium on Rendering (2005)*. The Eurographics Association, 2005. doi: 10.2312/EGWR/EGSR05/201-210. ISBN 3-905673-23-1. ISSN 1727-3463
- [4] R. Gupta, A. Chia, D. Rajan, E. Ng, and Z. Huang, "Image colorization using similar images," in *20th ACM Int. Conf. on Multimedia (MM'12)*, 2012, pp. 369–378.
- [5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics*, vol. 23, pp. 689–694, 06 2004. doi: 10.1145/1015706.1015780
- [6] A. Popowicz and B. Smolka, "Fast image colourisation using the isolines concept," *Multimedia Tools and Applications*, vol. 75, pp. 15 987–16 009, 2017. doi: 10.1007/s11042-016-3892-2
- [7] R. Zhang, P. Isola, and A. Efros, "Colorful image colorization," in *Computer Vision – ECCV 2016*. Springer, 2016. doi: 10.1007/978-3-319-46487-9_40 pp. 649–666.
- [8] E. Farella, S. Malek, and F. Remondino, "Colorizing the past: Deep learning for the automatic colorization of historical aerial images," *Journal of Imaging*, vol. 8, p. 269, 10 2022. doi: 10.3390/jimaging8100269
- [9] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics*, vol. 35, pp. 1–11, 07 2016. doi: 10.1145/2897824.2925974
- [10] J. Su, H. Chu, and J. Huang, "Instance-aware image colorization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2020. doi: 10.1109/CVPR42600.2020.00799 pp. 7965–7974.
- [11] H. Lee, D. Kim, D. Lee, J. Kim, and J. Lee, "Bridging the domain gap towards generalization in automatic colorization," in *Computer Vision – ECCV 2022*. Springer, 2022. doi: 10.1007/978-3-031-19790-1_32 pp. 527–543.
- [12] A. Deshpande, J. Rock, and D. Forsyth, "Learning large-scale automatic image colorization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. doi: 10.1109/ICCV.2015.72 pp. 567–575.

- [13] A. Royer, A. Kolesnikov, and C. Lampert, "Probabilistic image colorization," in *Proc. British Machine Vision Conference (BMVC)*. BMVA Press, September 2017. doi: 10.5244/C.31.85 pp. 85.1–85.12.
- [14] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.00387 pp. 3748–3756.
- [15] M. Hofinger, E. Kobler, A. Effland, and T. Pock, "Learned variational video color propagation," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022. ISBN 978-3-031-20050-2 pp. 512–530.
- [16] M. G. Blanch, N. O'Connor, and M. Mrak, "Scene-adaptive temporal stabilisation for video colourisation using deep video priors," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023. ISBN 978-3-031-25069-9 pp. 644–659.
- [17] A. Śluzek, "Do we always need ai for image colorization?" *Proceedings of the 4rd Polish Conference on Artificial Intelligence, Lodz, Poland*, April 2023, in print.
- [18] A. Śluzek, "On unguided automatic colorization of monochrome images," in *Proc. 31 Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2023, Plzen, Czechia.*, May 2023, in print.
- [19] ITU-R, "Parameter values for the hdtv standards for production and international programme exchange," Geneva, Recommendation BT.709-6, 2015.
- [20] E. Jessee and E. Wiebe, "Visual perception and the hsv color system: Exploring color in the communications technology classroom," *Technology Teacher*, vol. 68, no. 1, pp. 7–11, 2008.
- [21] S.-Y. Chen, J.-Q. Zhang, Y.-Y. Zhao, P. L. Rosin, Y.-K. Lai, and L. Gao, "A review of image and video colorization: From analogies to deep learning," *Visual Informatics*, vol. 6, no. 3, pp. 51–68, 2022. doi: 10.1016/j.visinf.2022.05.003
- [22] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchiescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018. ISBN 978-3-030-01267-0 pp. 179–195.
- [23] K.-H. Thung and P. Raveendran, "A survey of image quality measures," in *2009 International Conference for Technical Postgraduates (TECH-POS)*, 2009. doi: 10.1109/TECHPOS.2009.5412098 pp. 1–4.
- [24] P. Jagalingam and A. V. Hegde, "A review of quality metrics for fused image," *Aquatic Procedia*, vol. 4, pp. 133–142, 2015. doi: <https://doi.org/10.1016/j.aqpro.2015.02.019>
- [25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. doi: 10.1109/TIP.2003.819861
- [26] D. Hasler and S. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, vol. 5007. SPIE, 2003. doi: 10.1117/12.477378 pp. 87–95.
- [27] E. Gebhardt and M. Wolf, "Camel dataset for visual and thermal infrared multiple object detection and tracking," in *2018 15th IEEE Int. Conf. AVSS*, 2018. doi: 10.1109/AVSS.2018.8639094 pp. 1–6.