


RELATIONSHIPS BETWEEN COLORIZATION AND PSEUDO-COLORIZATION OF MONOCHROME IMAGES

Andrzej Śluzek 

Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Warsaw, Poland

Abstract This paper investigates the relationship between colorization and pseudo-colorization techniques for converting grayscale images to color. Colorization strives to create visually believable color versions of monochrome images, either replicating the original colors or generating realistic, alternative color schemes. In contrast, pseudo-colorization maps grayscale intensities to pre-defined color palettes to improve visual appeal, enhance content understanding, or aid visual analysis. While colorization is an ill-posed problem with infinitely many RGB solutions, pseudo-colorization relies on mapping functions to deterministically assign colors. This work bridges these techniques by exploring the two following operations: first – deriving pseudo-color from colorized images – this allows for creating stylized or abstract representations from existing colorizations, and second – enriching color diversity in pseudo-colored images – this enhances visual appeal and attractiveness of pseudo-colored images. The paper emphasizes the centrality of decolorization (*rgb-to-gray*) models in both processes. It focuses on the theoretical underpinnings of these problems but complements them with illustrative examples for clarity.

Keywords: colorization, pseudo-colorization, decolorization, rgb-to-gray models, color maps, randomized flood-fill.

1. Introduction and motivation

While both (re)colorization and pseudo-colorization aim to create color versions of monochrome images, their approaches, complexities, and applications differ significantly.

Colorization is the process of converting monochrome images into visually convincing color-rich counterparts. In the case of recolorization, the goal is to replicate the original colors of the image, even though they are unknown to the algorithm, e.g. [3, 6, 16, 18, 22, 23], etc. Alternatively, the objective is to create realistic-looking hypothetical color versions of grayscale images, representing worlds that might not have originally been in color, e.g. [10, 25].

This field holds significant practical and commercial value, particularly in restoring historical photos and movies. Over the past two decades, researchers have proposed numerous and diverse (re)colorization algorithms. However, a fundamental challenge of colorization is its inherent ambiguity. There are infinitely many valid RGB combinations that, when converted back to grayscale, would appear identical.

Thus, colorization techniques typically leverage human knowledge and expectations to guide the algorithms, [22]. Early methods relied on providing reference color images [5, 7] or manual coloring (*scribbling*) of specific image regions [11, 13]. Recently, deep



Fig. 1. Two grayscale images (from a popular SUN dataset [21] and an IR image) and their exemplary colorizations. The results are produced by a method outlined in [24,25] (i.e., without the use of training data, AI tools, semantic analysis, human assistance, etc.).

learning approaches have become dominant, with neural networks designed to learn color patterns suitable for specific domains, semantics, or content, [3,23]. Some techniques additionally incorporate recognition or learning of image domains or objects to further enhance colorization, e.g., [6,18].

While some papers claim fully automatic (re)colorization, e.g. [10,19], they often still rely on implicit human color knowledge through training datasets and semantic identification mechanisms. In contrast, our recent works [24,25] propose a mechanism for truly automatic image colorization, entirely without human intervention, training data, learning processes, or semantic analysis. This approach specifically targets images from domains where original color versions likely never existed.

Fig. 1 summarizes this concept by showcasing grayscale images with various colorized options for each. For the first image (from the real world), one result might resemble the actual colors, although the *ground-truth* is unknown.

Unlike colorization, pseudo-colorization is a simpler process. It assigns a pre-defined color map to image intensities. The key challenge lies in selecting or designing this mapping function to achieve specific goals, such as enhancing visual appeal, improving content perception, or facilitating the visual analysis of objects or processes within the image, e.g. [9,12,15,20].

There are a number of standard color maps, commonly referred to as *hot*, *warm*, *rainbow*, *springtime*, *jet*, *sine*, etc. (see [1,15,17]). Examples are given in Figs. 2 and 3. They are often used in typical applications of pseudo-coloring, including thermography, roentgenography or geographical visualizations.

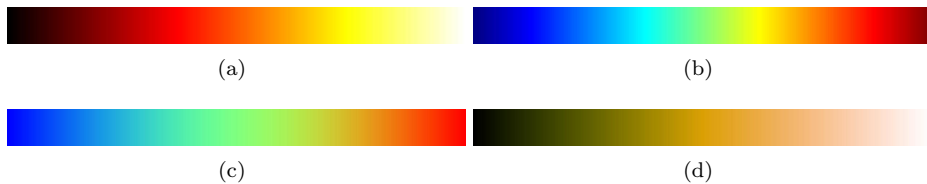


Fig. 2. Some popular color maps used for pseudo-colorization: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*.

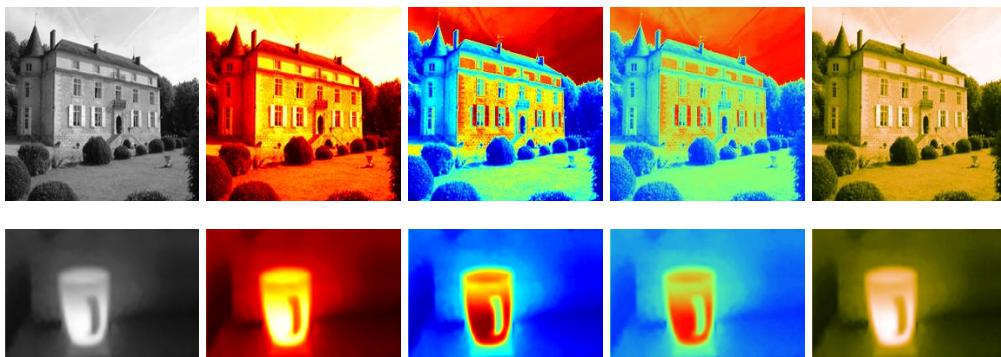


Fig. 3. Two grayscale images and their exemplary pseudo-colorizations obtained by using the corresponding color maps from Fig. 2 (compare to Fig. 1).

Finally, it should be noted that in this paper, we do not consider the so-called *pseudo-coloring problem* (PsCP), which, despite its similar name, is a significantly different task. PsCP involves segmenting a monochrome image into regions and the objective is to assign a set of colors to these regions, with the colors being as visually distinct as possible (e.g., [2, 14]).

Colorization can yield infinitely many valid results for a single grayscale image, while pseudo-colorization produces unambiguous colors based on a pre-defined mapping function. This paper bridges the gap between these two seemingly distinct color-rendering techniques. We aim to explore how to transform colorized images into pseudo-colored ones and vice versa, while preserving the overall color perception of the image. Notably, these methods should be generalizable, functioning across various image content and characteristics.

To our knowledge, no prior research has investigated these connections between colorization and pseudo-colorization. We believe this exploration holds significant potential, driven not only by practical applications but also by sheer scientific curiosity.

To that end, Section 2 presents formal models of image decolorization and colorization

(both regular colorization and pseudo-colorization). In particular, for regular colorization, we revisit models proposed in [24, 25].

In Section 3, the easier problem of converting colorized images into the most similar pseudo-color equivalents is discussed. This operation can be used, for example, to create a more abstract or stylized representation of an image.

The more complex task of transforming pseudo-colored images into visually similar (though, obviously, with much richer colorization effects) fully-colored images is presented in Section 4. Such operations may be required if we intend to make pseudo-colored images more aesthetically pleasing and visually engaging.

In the final Section 5, we summarize the paper and link its outcomes to the related problems of image processing and computer graphics.

Finally, it should be highlighted that this paper presents improved and updated materials originally presented at the 9th Conference on Symbiosis of Technology and IT (SIT) in Kiry, June 2023. It is important to note that these materials have not been previously published.

2. Formal models

2.1. Models of image decolorization

Given a monochrome image, we may want to know what decolorization (*rgb-to-gray*) models describe the conversion of the color original (possibly not even existing) into the grayscale results.

For rendering digital images, the RGB/sRGB color models are universally adopted, and the intensities I of decolorized images are usually defined by linear functions of primary colors, mainly by the Y channel of the YUV (or YIQ) models:

$$I = 0.299 R + 0.587 G + 0.114 B \quad (1)$$

Alternatively, another set of coefficients can be used almost equivalently, as recommended in [8]:

$$I = 0.2126 R + 0.7152 G + 0.0722 B \quad (2)$$

The advantage of Eqs. (1) and (2) is that they take human perception into account, and the perceived brightness of resulting grayscale images looks very similar to the brightness of color originals (see examples in Figs. 4b, c).

However, if we assume that original color images do not exist (i.e., the *ground-truth* colors are unspecified), we can use any set of *rgb-to-gray* coefficients (satisfying $k_R + k_G + k_B = 1$) to ‘decolorize’ the hypothetical color sources:

$$I = k_R R + k_G G + k_B B \quad (3)$$



Fig. 4. A color image and its monochrome counterparts obtained by Eq.1 (b), Eq.2 (c), and by Eq.3 with two random sets of coefficients, $[0.14, 0.11, 0.75]$ in (d) and $[0.44, 0.14, 0.42]$ in (e).

In case of actual color images, their monochrome counterparts obtained by using (3) may sometimes look strange when compared to the original images (see Figs. 4d, e) but, nevertheless, they are visually convincing.

Thus, applying (3) with different coefficients $[k_R, k_G, k_B]$ allows us to generate an infinite spectrum of monochrome versions of a color image. This might seem counter-intuitive, but it aligns with the concept of having infinite RGB combinations that can create a single grayscale image.

Actually, decolorization models are the backbone of other methods discussed in this paper. Their equations indirectly define the colorization results (Subsection 2.3), they are instrumental in converting color images into pseudo-colored ones (Section 3), and in the operation of rich colorization of pseudo-colored images (Section 4).

2.2. Models of pseudo-colorization

Pseudo-colorization models are defined by vector-functions (*color maps*) of the grayscale intensities I :

$$R = R(I), \quad G = G(I), \quad B = B(I) \quad (4)$$

where the $[0 : 1]$ range is generally assumed for the intensities and (subsequently) for the color values.

For the exemplary *sine* model the equations are as follows:

$$R(I) = -0.5 \cos(\pi I) + 0.5, \quad G(I) = \sin(\pi I), \quad B(I) = 0.5 \cos(\pi I) + 0.5, \quad (5)$$

with more examples of color map equations available in [1, 15].

It can be observed that the *color map equations* define parametric curves within the RGB unit cube. The curves start at $[R(0), G(0), B(0)]$ and terminate at $[R(1), G(1), B(1)]$. Normally, the colors assigned to intensities change incrementally, i.e., the curves are continuous. Shapes of such curves for exemplary color maps are given in Fig. 5.

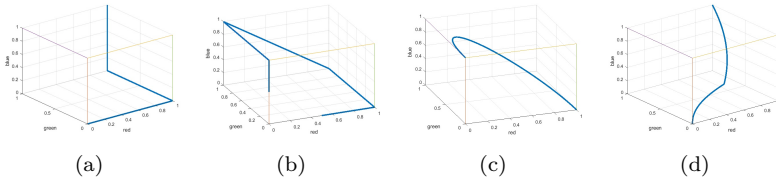


Fig. 5. Curves defined within the *RGB* cube by exemplary color mapping functions: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*.

In some tasks, we may want pseudo-colorization results that preserve the perceived brightness of the original monochrome images. To that end, a color $C_u = [R_u, G_u, B_u]$ assigned to the intensity I_u can be replaced by another color $C_{pr} = [R_{pr}, G_{pr}, B_{pr}]$ which satisfies

$$0.299R_{pr} + 0.587G_{pr} + 0.114B_{pr} = I_u \quad (6)$$

and minimizes $\|C_u - C_{pr}\|$ (alternatively, in the above formula, Eq. (2) can be used instead of (1)).

Usually, the solution of (6) is obtained by a simple orthogonal projection of the C_u color onto the $0.299R + 0.587G + 0.114B = I_u$ plane, unless the result lies outside the *RGB* cube. In such cases, additional steps are needed to find the closest color on the boundary of the *RGB* cube.

Such color transformations will be referred to as *YUV* projections. Exemplary results of such projections are given in Fig. 6.

2.3. Models of colorization

Colorization techniques typically assume that the intensity of monochrome images defines the luminance of colored outputs, and only two chrominance channels should be reconstructed. Usually, there are no other formal models used in the popular (re)colorization methods, which are currently dominated by AI.

In this work, however, we propose to use decolorization (*rgb-to-gray*) models, as specified in Subsection 2.1, to define the models of colorization. Thus, given a monochrome image, we assume that it was obtained from a hypothetical color image by applying Eq. 3 with specific values $[k_R, k_G, k_B]$ coefficients.

The number of such models can be very large, depending on the quantization step applied to the coefficients (which can range from 0 to 1, subject to $k_R + k_G + k_B = 1$). For example, there are 5151 decolorization models with the 0.01 quantization increment and 125,751 models with the 0.002 stepsize. Formally, any number of decolorization models can be used. However, in practice, it is best to use only a limited subset, typically between 20 and 40, of the most representative models. This results in a number of

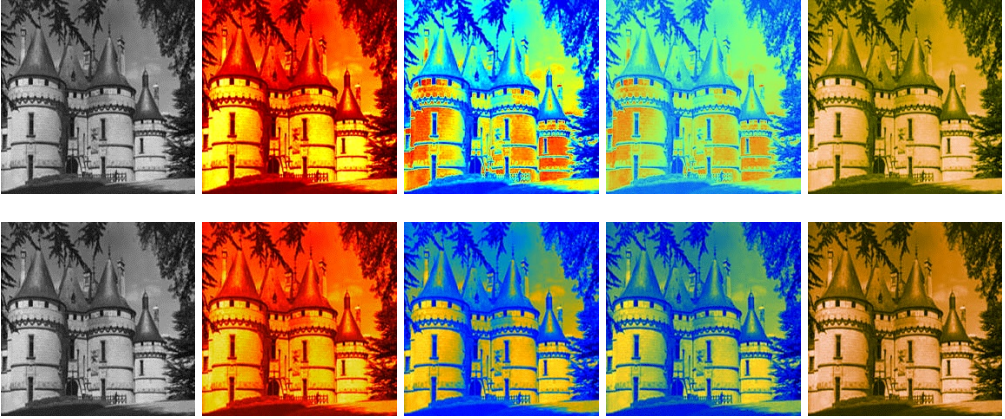


Fig. 6. *Hot*, *jet*, *sine* and *warm* pseudo-colorizations (of an exemplary monochrome image from SUN dataset) before (top row) and after (bottom row) the YUV projections (by using Eq.6).

colorization variants from which the user can choose the most plausible one. If too many *rgb-to-gray* models are used simultaneously, the sheer number of alternative colorizations may be overwhelming for human evaluators. (Details are explained in [26].)

Given, in digital images, the finite number of intensity levels (e.g., from 0 to 255) and, correspondingly, finite numbers of RGB colors, a pixel with intensity I can only be assigned colors that satisfy Eq. (3) with the adopted $[k_R, k_G, k_B]$ values. All such colors should be considered viable choices for coloring an I -valued monochrome pixel. It can be noted that the pool of available colors actually diminishes for darker/lighter intensities (as shown in Fig. 7), and for the extreme values the choice is deterministic (i.e., grayscale *white/black* should remain *white/black* in color).

As an illustration, Fig. 8 shows colors assigned to 208 intensity in two exemplary *rgb-to-gray* models. Note that the first model is actually YUV (and all colors are perceived as having almost the same brightness) while for the second model the perceived brightness of colors varies significantly.

Alternatively, all colors assigned to the selected intensity I_u can be visualized as the polygonal intersection of the $k_R R + k_G G + k_B B = I_u$ plane with the RGB color space cube. An example is given in Fig. 9.

With no prior information provided, all colors available to the I_u intensity can be assigned to a pixel of that value 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_u should be influenced by the intensity and color of the neighbor.

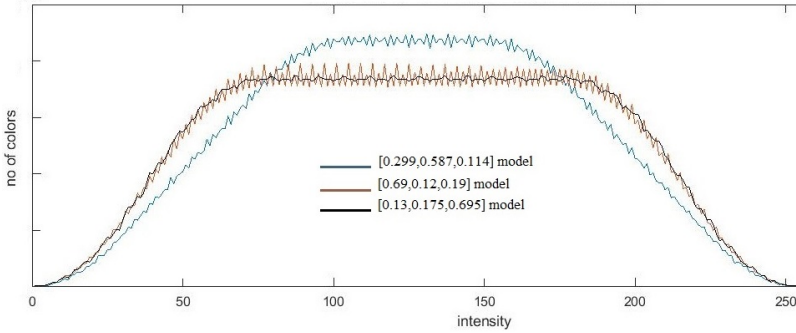


Fig. 7. Changes in the numbers of colors assigned to various intensities for $[0.299, 0.587, 0.114]$, $[0.69, 0.12, 0.19]$ and $[0.13, 0.175, 0.695]$ *rgb-to-gray* models.

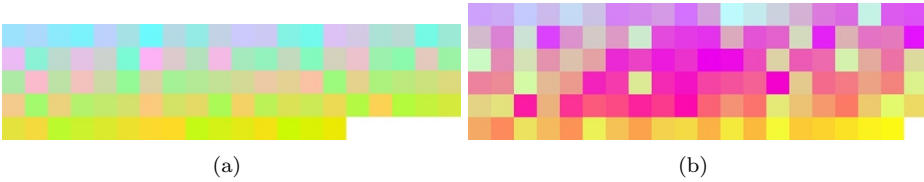


Fig. 8. Exemplary colors assigned to 208 intensity in (a) $[0.299, 0.587, 0.114]$ and (b) $[0.69, 0.12, 0.19]$ models.

Therefore, we use a simple but surprisingly effective (as shown in [24,25,26]) 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 level I_u , which are at distances from the color C_1 proportional to the difference in intensity levels $\|I_u - I_1\|$.

Then, pixels can be colored using the following color assignment method:

1. Let's assume a pixel with I_u intensity, which has an already colored neighbor with I_1 intensity and C_1 color. Let $\mathbf{C} = \{C_{u_1}, \dots, C_{u_N}\}$ be the list of colors available to I_u in the adopted *rgb-to-gray* model (see examples in Fig. 8).
2. 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.,

$$\mathbf{C}_{\text{mod}} = \{C_{u_{i_1}}, \dots, C_{u_{i_N}}\}, \text{ where } \|C_{u_{i_n}} - C_1\| \leq \|C_{u_{i_{(n+1)}}} - C_1\| \quad (7)$$

In fact, the displayed lists of colors in Fig. 8 are already ordered (for $I_1 = 40$ and $C_1 = [20, 42, 137]$).

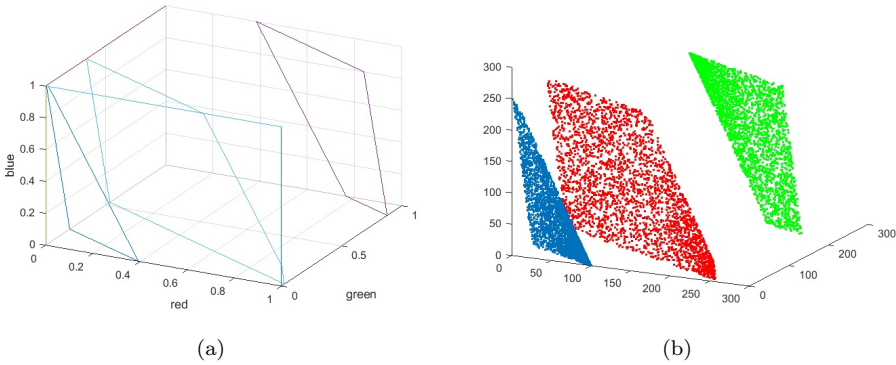


Fig. 9. Distribution of colors assigned to three exemplary intensities in (a) RGB cube and (b) sRGB cube. The assumed *rgb-to-gray* model is $I = 0.299R + 0.587G + 0.114B$. The intensities are 30, 80 and 208 (in sRGB), i.e., 0.118, 0.314 and 0.816 (in RGB).

3. A uniform distribution is used to randomly select a color from a specific sub-range of the **Cmod** list. The location of this sub-range depends on the difference $\|I_u - I_1\|$. In general, smaller differences favor colors from a narrower range near the top of the **Cmod** list, while larger differences favor a wider range of colors towards the end of **Cmod**. Refer to [25] for a detailed explanation of this step.

To achieve a more natural look and avoid unnecessarily regular patterns, the colorization process employs a randomized variant of the queue-based *flood-fill* algorithm, e.g., [4]. This means the active pixel for coloring is randomly chosen from the current queue.

The initial queue consists of the brightest and darkest pixels, for which the color selection is usually deterministic (see Fig. 7). Pixels are then colored using the previously described steps until the queue is empty. However, during colorization, an uncolored pixel might have several already colored neighbors. In these cases, the color selection process (described in Point 3) is repeated multiple times. The final color assigned to the pixel is then the average of the colors obtained from all its colored neighbors.

$$C_u = \frac{1}{K} \sum_{k=1}^K C_{u_k}, \text{ where } K = 1, 2, 3 \text{ or } 4. \quad (8)$$

Despite all these randomizing factors, the results produced by the outlined method are surprisingly repeatable, depending mainly (as expected) on the adopted *rgb-to-gray* model. In fact, the images shown in Fig. 1 are generated by this method, and many more examples can be found in [24, 25, 26].

The visual plausibility of the results can be further improved by applying the YUV

projection outlined in Subsection 2.2. Through visual inspection, it could be identified that several images in Fig. 1 are actually rectified using this approach.

2.4. Remarks on alternative *rgb-to-gray* models

In the decolorization and colorization operations discussed in Subsections 2.1 and 2.3, we assume *rgb-to-gray* models with the fixed values of adopted $[k_R, k_G, k_B]$ coefficients. Nevertheless, the process of colorization can be generalized by assuming that $[k_R, k_G, k_B]$ coefficients are individually assigned to each intensity level. In this case, the following equation should be used instead of Eq. 3 (without affecting the presented methodology of colorization):

$$I = k_R(I)R + k_G(I)G + k_B(I)B \quad (9)$$

However, such a generalization has two weaknesses. First, the number of alternative colorization variants would become unimaginably large. For example, with a step size of 0.002 for the model coefficients, the number of variants would approach $125\,751^{256}$. Second, the *rgb-to-gray* models by Eq. (9) are useless in the decolorization operation because the intensity values are computed from the coefficients, which depend on the unknown intensities themselves. Nevertheless, we will further explore these alternative models in Section 4.

3. Pseudo-colorization of colorized images

The conversion of color (or colorized) images into pseudo-colored ones seems straightforward at first. We simply need to reduce the number of colors to 256, for example by clustering all colors present in an image into 256 classes and then performing the corresponding substitutions.

However, there are certain formal complications. As discussed in Subsection 2.2, pseudo-colorization models are defined by their *color maps*. This means that pseudo-colorization can only be performed on a monochrome image. Therefore, for any color (or colorized) image, we first need to have its monochrome version, which can then be pseudo-colored.

In Subsection 2.1, we assumed that grayscale images are obtained from color images by applying decolorization (*rgb-to-gray*) models defined by Eq. 3, where the $[k_R, k_G, k_B]$ coefficients have specific values. Once these values are assumed or identified, pseudo-colorization can be easily performed as explained in the following description.

First, as highlighted in Subsection 2.3, for the adopted *rgb-to-gray* model, any intensity defines the polygonal intersection with the RGB cube (see Fig. 9). The centers of mass of such polygons, i.e., the average colors assigned to intensities ranging from 0 to 1 (0 to 255), actually form a curve winding (from *black* to *white*) within the cube.

Such curves are equivalent to the curves defining the *color maps* (see Fig. 5) and can be employed as such.

Formally, for the adopted $[k_R, k_G, k_B]$ model, the pseudo-color assigned to I intensity is defined by

$$[R(I), G(I), B(I)] = \text{center of mass}[(\text{RGB cube}) \cap (I = k_R R + k_G G + k_B B \text{ plane})] \quad (10)$$

Fig. 10 shows the curves, i.e. the *color maps*, defined by three exemplary *rgb-to-gray* models. Correspondingly, Figs. 11 and 12 compare outcomes of colorization and pseudo-colorization of selected monochrome images using the same *rgb-to-gray* models.

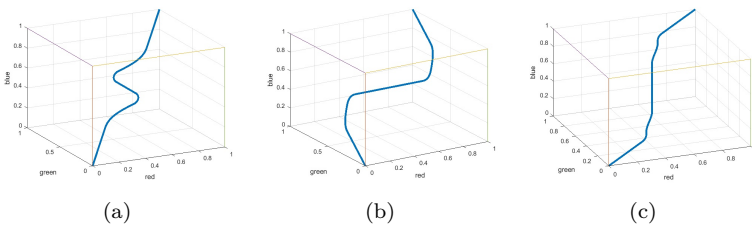


Fig. 10. Curves representing the *color maps* obtained from three *rgb-to-gray* models: (a) $[0.299, 0.587, 0.114]$, (b) $[0.69, 0.12, 0.19]$ and (c) $[0.13, 0.175, 0.695]$.

The presented examples indicate that, despite a reduced number of colors, the pseudo-colored images approximately preserve the coloristic perception of the (fully-)colorized images, although the general impression is simplified and more abstract.

4. Colorization of pseudo-colored images

Colorization of pseudo-colored images is once again based on *rgb-to-gray* models. Given a color map CM (and its equivalent curve, see Figure 5), we attempt to identify an *rgb-to-gray* model $\text{MOD}(\text{CM})$ that would define (as explained in Section 3) a color map as close as possible to CM. Then, a monochrome image can be colorized (using the method outlined in Subsection 2.3) with the identified $\text{MOD}(\text{CM})$ model adopted.

Unfortunately, the color map curves obtained from *rgb-to-gray* models by averaging colors assigned to particular intensities always start at *black* and terminate at *white* (see Fig. 10), while curves of arbitrary color maps can start and terminate at any colors within the RGB cube (e.g., Fig. 5b, c).

Therefore, as the preliminary step, the color map CM is transformed by the YUV

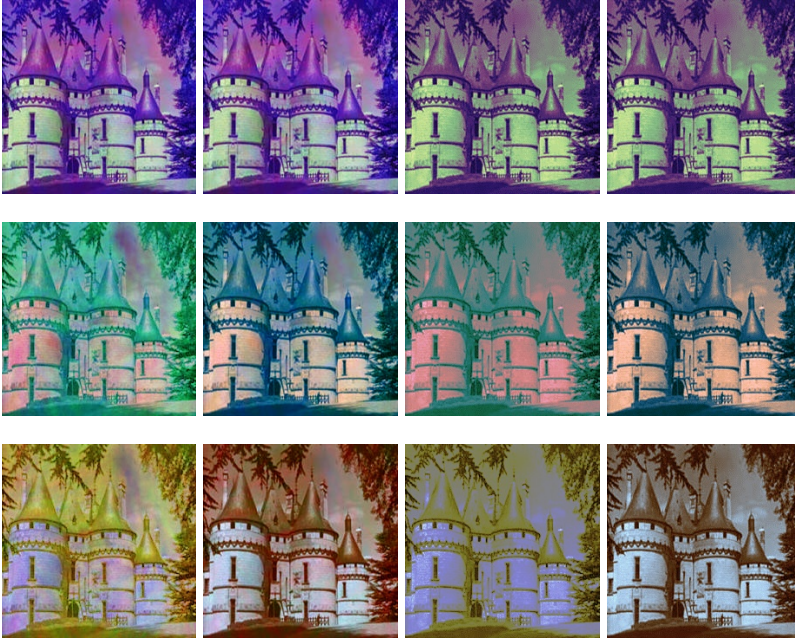


Fig. 11. Results for the grayscale image (from Fig. 6), i.e., its colorized variants (1st column), colorized variants after the YUV projection (2nd column), pseudo-colored variants (3rd column) and pseudo-colored variants after the YUV projection (4th column). The results are obtained using the following *rgb-to-gray* models: [0.299, 0.587, 0.114] (top), [0.69, 0.12, 0.19] (center) and [0.13, 0.175, 0.695] (bottom). Note that the first model is actually YUV, so the results are the same before and after the projection.

projection, and for each intensity I the shift from its original pseudo-color to the YUV-projected pseudo-color is represented by the T_{yuv} vector:

$$T_{yuv}(I) = [R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)] - [R(I), G(I), B(I)], \quad (11)$$

where the original $[R(I), G(I), B(I)]$ pseudo-color is transformed by the YUV projection into $[R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)]$.

Thus, the curves of YUV-projected color maps always start at *black* and terminate at *white* (see examples in Fig. 13).

Next, for each intensity I , we individually identify the decolorization model (e.g., from 125 751 available *rgb-to-gray* models) which minimizes

$$[k_R(I), k_G(I), k_B(I)] = \min \|k_R R_{YUV}(I) + k_G G_{YUV}(I) + k_B B_{YUV}(I) - I\|, \quad (12)$$

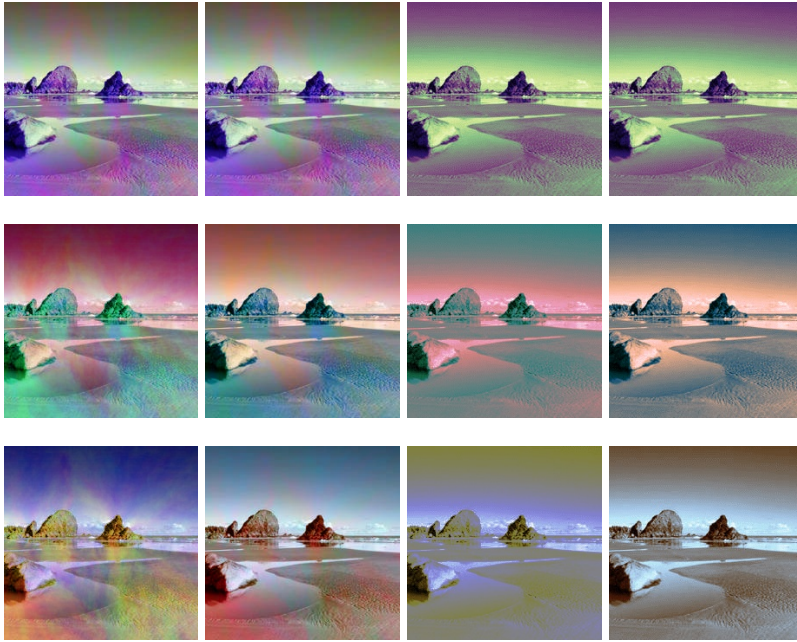


Fig. 12. The same results as in Fig. 11 for another grayscale image (from SUN dataset).

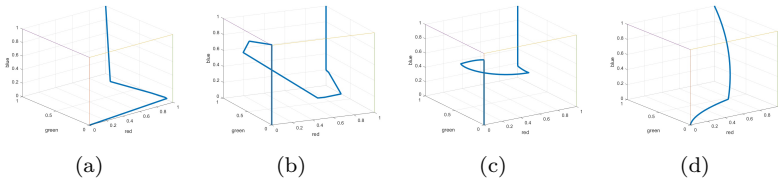


Fig. 13. The YUV-projected curves of exemplary *color maps*: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*. Compare the shapes to Fig. 5

i.e., the model which most accurately converts the modified pseudo-color into the corresponding intensity.

The coefficients obtained in (12) define the optimum *rgb-to-gray* model $MOD_{opt}(CM)$, which will be adopted for colorizing monochrome images which are pseudo-colored with the original CM color map.

Crucially, Eq. 12 defines separate $[k_R(I), k_G(I), k_B(I)]$ coefficients for each intensity

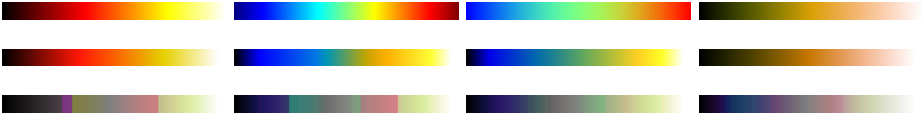


Fig. 14. Exemplary color maps (from left to right: *hot*, *jet*, *sine* and *warm*) with their original pseudo-colors (top row), pseudo-colors after YUV projections (middle row) and colors of the maps converted from the $MOD_{opt}(CM)$ models (bottom row).

level. This implies that Eq. (9) (see Subsection 2.4) truly captures the essence of the $MOD_{opt}(CM)$ model, not Eq. (3). Nevertheless, such a modification does not affect the model of colorization presented in Section 2.3.

The obtained model $MOD_{opt}(CM)$ can be back-converted into the color map using the approach discussed in Section 3. In this approach, for each intensity I its pseudo-color is specified by the center of mass of the polygon defined by the $[k_R(I), k_G(I), k_B(I)]$ coefficients.

Obviously, the color map obtained in this way differs not only from the original CM map but also from the YUV-projected map. This is because the centers of polygons defined by $[k_R(I), k_G(I), k_B(I)]$ coefficients (these centers will be referred to as $[R_{CoM}(I), G_{CoM}(I), B_{CoM}(I)]$) are almost always far from the $[R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)]$ colors (i.e., YUV-projected pseudo-colors of the original CM map) which can be anywhere within the corresponding polygons.

Fig. 14 shows exemplary original color maps, the maps after applying the YUV projections, and eventually compares them to the maps obtained from the corresponding $MOD_{opt}(CM)$ models.

Therefore, the images colorized by the $MOD_{opt}(CM)$ models significantly differ in terms of coloristic perception from the images pseudo-colored by the original CM maps. However, we propose a simple correction that reverses the changes introduced in the color maps so that the corrected images are perceptually consistent with the original pseudo-color images, albeit with much richer diversity of colors.

Given a pixel of (x, y) coordinates and with the original I intensity, its color is corrected as follows:

$$RGB_{new}(x, y) = RGB(x, y) - T_{off} \{I(x, y)\}, \quad (13)$$

where $T_{off}(I) = [R_{CoM}(I), G_{CoM}(I), B_{CoM}(I)] - [R(I), G(I), B(I)]$, i.e. it represents the offset between the original pseudo-color and the corresponding pseudo-color of the map reconstructed from the $MOD_{opt}(CM)$ model.

If the vector resulting from Eq. (13) extends outside the RGB cube, we select the color at the intersection of this vector with the cube boundary.

Figs. 15 and 16 illustratively summarize the results of converting pseudo-colored

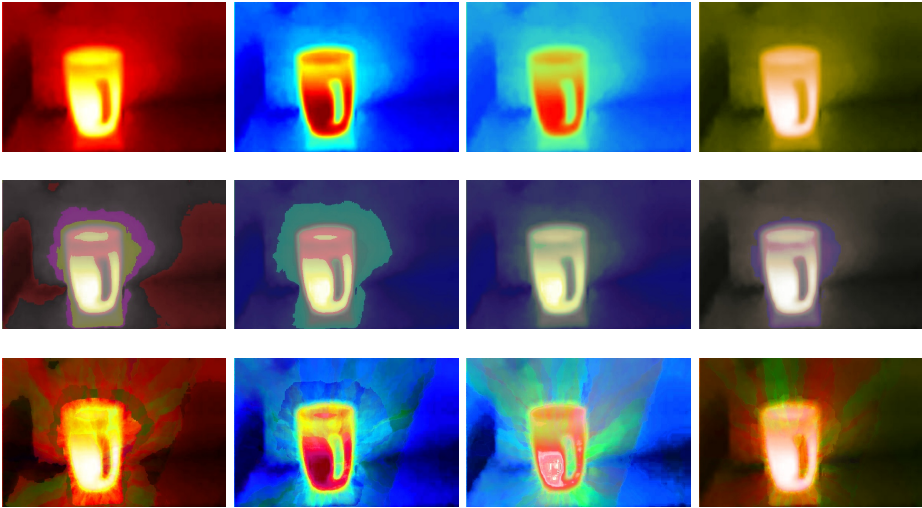


Fig. 15. First row: an image pseudo-colored by (from left to right) *hot*, *jet*, *sine* and *warm* color maps. Second row: the corresponding colored images before the color correction by Eq. 13. Third row: the colored images after the color correction by Eq. 13.

versions of two previously discussed exemplary images into their fully-colored variants, using four different color maps.

The colored images correspond well to the pseudo-colored originals in terms of overall coloristic perception, but the added variability of colors enriches the attractiveness and visual appeal of the results.

5. Conclusions

This paper proposes a novel and general approach that bridges the gap between image colorization and pseudo-colorization, which are fundamentally distinct concepts. The key unifying element is the adopted model for image decolorization (*rgb-to-gray*) in which we hypothesize that grayscale image intensities are linear combinations (represented by $[k_R, k_G, k_B]$ coefficients) of primary colors from a potentially non-existent color image.

In image colorization, these coefficients are freely chosen, allowing for the creation of diverse colored versions of a grayscale image. These variations may not necessarily reflect the “true” colors of the scene, but rather offer aesthetically pleasing visualizations of imagined worlds. For each colored image, a corresponding pseudo-colored variant can be derived by leveraging the chosen $[k_R, k_G, k_B]$ coefficients.

Colorization of pseudo-colored images follows a different approach. Here, the color

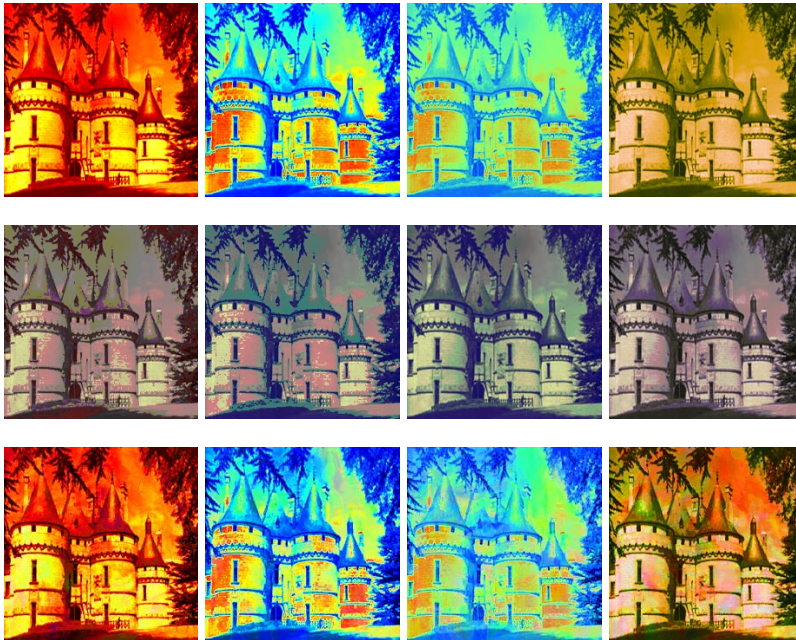


Fig. 16. The same results as in Fig. 15 for another image pseudo-colored by four color maps.

maps themselves define the *rgb-to-gray* models, with unique $[k_R, k_G, k_B]$ coefficients for each intensity level.

Notably, both the conversion from colorized to pseudo-colored images and vice versa result in images with perceptually similar color palettes, containing either a reduced or enriched range of colors.

The proposed methodologies, verified through experimentation, offer a solution to an intriguing challenge, though their immediate practicality might be limited. Nonetheless, this solution is directly applicable for scenarios where preserving the overall color perception of an image is crucial, but the goal is to create either an abstract/stylized representation of a full-color image or a more visually engaging and expressive variant of a pseudo-colored image.

Beyond the scope of this paper, preliminary investigations suggest the potential of this approach for a challenging task: recovering visual artifacts lost during the conversion of color originals to monochrome or pseudo-colored images. This promising avenue merits further exploration in future work.

Furthermore, converting pseudo-colored images into fully colorized ones could serve

as a stepping stone for creating or enriching training datasets for AI-based colorization. This is particularly relevant for domains beyond human color perception, where we only have access to grayscale information, such as infrared, ultraviolet, ultrasound, MRI, or X-ray modalities.

References

- [1] B. Abidi, Y. Zheng, A. Gribok, and M. Abidi. Screener evaluation of pseudo-colored single energy X-ray luggage images. In: *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) – Workshops*, pp. 35–35, 07 2005. doi:[10.1109/CVPR.2005.521](https://doi.org/10.1109/CVPR.2005.521).
- [2] R. C. Contreras, O. Morandin Junior, and M. S. Viana. A new local search adaptive genetic algorithm for the pseudo-coloring problem. In: Y. Tan, Y. Shi, and M. Tuba, eds., *Advances in Swarm Intelligence*, pp. 349–361. Springer International Publishing, 2020. doi:[10.1007/978-3-030-53956-6_31](https://doi.org/10.1007/978-3-030-53956-6_31).
- [3] E. Farella, S. Malek, and F. Remondino. Colorizing the past: Deep learning for the automatic colorization of historical aerial images. *Journal of Imaging*, 8:269, 10 2022. doi:[10.3390/jimaging8100269](https://doi.org/10.3390/jimaging8100269).
- [4] K. Fishkin and B. Barsky. An analysis and algorithm for filling propagation. In: *Computer-Generated Images. The State of the Art Proceedings of Graphics Interface '85*, pp. 56–76. Springer, 1985. doi:[10.1007/978-4-431-68033-8_6](https://doi.org/10.1007/978-4-431-68033-8_6).
- [5] R. Gupta, A. Chia, D. Rajan, E. Ng, and Z. Huang. Image colorization using similar images. In: *Proc. 20th ACM Int. Conf. on Multimedia (MM'12)*, pp. 369–378, 2012. doi:[10.1145/2393347.2393402](https://doi.org/10.1145/2393347.2393402).
- [6] 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*, 35:1–11, 07 2016. doi:[10.1145/2897824.2925974](https://doi.org/10.1145/2897824.2925974).
- [7] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In: *Proc. Eurographics Symposium on Rendering (2005)*. The Eurographics Association, 2005. doi:[10.2312/EGWR/EGSR05/201-210](https://doi.org/10.2312/EGWR/EGSR05/201-210).
- [8] ITU-R. Parameter values for the HDTV standards for telecommunication and international programme exchange. Recommendation BT.709-6, International Telecommunication Union, Geneva, 2015. <https://www.itu.int/rec/R-REC-BT.709-6-201506-I/>.
- [9] M. Khan, Y. Gotoh, and N. Nida. Medical image colorization for better visualization and segmentation. In: M. Valdés Hernández and V. González-Castro, eds., *Medical Image Understanding and Analysis*, pp. 571–580. Springer International Publishing, 2017. doi:[10.1007/978-3-319-60964-5_50](https://doi.org/10.1007/978-3-319-60964-5_50).
- [10] C. Lei and Q. Chen. Fully automatic video colorization with self-regularization and diversity. In: *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3748–3756, 2019. doi:[10.1109/CVPR.2019.00387](https://doi.org/10.1109/CVPR.2019.00387).
- [11] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23:689–694, 06 2004. doi:[10.1145/1015706.1015780](https://doi.org/10.1145/1015706.1015780).
- [12] K. Moreland. Why we use bad color maps and what you can do about it. In: *Proc. IS&T Int. Symp. Electronic Imaging: Human Vision and Electronic Imaging*, vol. 28, 2016. doi:[10.2352/ISSN.2470-1173.2016.16.HVEI-133](https://doi.org/10.2352/ISSN.2470-1173.2016.16.HVEI-133).
- [13] A. Popowicz and B. Smolka. Fast image colourisation using the isolines concept. *Multimedia Tools and Applications*, 75:15987–16009, 2017. doi:[10.1007/s11042-016-3892-2](https://doi.org/10.1007/s11042-016-3892-2).

- [14] K. Radlak and B. Smolka. Visualization enhancement of segmented images using genetic algorithm. In: *Proc. 2014 Int. Conf. Multimedia Computing and Systems (ICMCS)*, pp. 391–396, 2014. doi:10.1109/ICMCS.2014.6911269.
- [15] A. Rahimian, M. Etehadtavakol, M. Moslehi, and E. Ng. Comparing different algorithms for the pseudo-coloring of myocardial perfusion single-photon emission computed tomography images. *Journal of Imaging*, 8(12):331, 12 2022. doi:10.3390/jimaging8120331.
- [16] A. Salmona, L. Bouza, and J. Delon. Deoldify: A review and implementation of an automatic colorization method. *Image Processing On Line*, 12:347–368, 2022. doi:10.5201/ipol.2022.403.
- [17] X.-Q. Shi, P. Sällström, and U. Welander. A color coding method for radiographic images. *Image and Vision Computing*, 20:761–767, 2002. doi:10.1016/S0262-8856(02)00045-8.
- [18] J. Su, H. Chu, and J. Huang. Instance-aware image colorization. In: *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7965–7974, jun 2020. doi:10.1109/CVPR42600.2020.00799.
- [19] D. Varga and T. Sziranyi. Fully automatic image colorization based on convolutional neural network. In: *Proc. 23rd Int. Conf. Pattern Recognition (ICPR)*, pp. 3691–3696, 2016. doi:10.1109/ICPR.2016.7900208.
- [20] A. Visvanathan, S. Reichenbach, and Q. Tao. Gradient-based value mapping for pseudocolor images. *Journal of Electronic Imaging*, 16(3):033004, 2007. doi:10.1117/1.2778426.
- [21] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abby to zoo. In: *Proc. 2010 IEEE Conference CVPR*, pp. 3485–3492, 2010. doi:10.1109/CVPR.2010.5539970.
- [22] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul. Grayscale image colorization methods: Overview and evaluation. *IEEE Access*, 9:113326–113346, 2021. doi:10.1109/ACCESS.2021.3104515.
- [23] R. Zhang, P. Isola, and A. Efros. Colorful image colorization. In: *Proc. European Cong. Computer Vision – ECCV 2016*, pp. 649–666. Springer, 2016. doi:10.1007/978-3-319-46487-9_40.
- [24] A. Śluzek. Do we always need ai for image colorization? *Proc. of 4th Polish Conference on Artificial Intelligence*, pp. 31–36, 2023. doi:10.34658/9788366741928.3.
- [25] A. Śluzek. On unguided automatic colorization of monochrome images. In: *WSCG 2023 Proceedings*, vol. 3301 of *Computer Science Research Notes*, pp. 379–384, May 2023. doi:10.24132/CSRN.3301.38.
- [26] A. Śluzek, M. Dudziński, and T. Świsłocki. Automatic colorization of digital movies using de-colorization models and SSIM index. In: *Preproc. 18th Conf. Computer Science and Intelligence Systems (FedCSIS 2023)*, pp. 837–847, September 2023. doi:10.15439/2023F3017.



Andrzej Śluzek received his M.Sc., Ph.D., and D.Sc. (*habilitation*) degrees from Warsaw University of Technology. He is currently a full professor in the Institute of Information Technology (Department of Artificial Intelligence) at Warsaw University of Life Sciences – SGGW.

From 1992 to 2011, he worked at Nanyang Technological University (School of Computer Science and Engineering) in Singapore. From 1994 to 2010, he was the deputy director of the Robotic Research Centre at the same university. From 2011 to 2020, he was an associate professor at Khalifa University (Department of Electrical Engineering and Computer Science) in Abu Dhabi, United Arab Emirates. In 2015, he was granted the title of professor by the President of Poland.