Bildeinfärbung alter Fotografien auf der Basis erweiterter Abstandstransformation

Old photographs colorization based on extended distance transformation

Przemyslaw Lagodzinski [1)], Bogdan Smloka [2)]
Silesian University of Technology, Institute of Computer Science [1)], Institute of Automatic Control [2)]
Akademicka 16, 44-100 Gliwice, Poland
E-mail: Przemyslaw.Lagodzinski@polsl.pl, Bogdan.Smolka@polsl.pl

Zusammenfassung:

Der Begriff Bildeinfärbung wird verwendet, um einen automatisierten Prozess für das Hinzufügen von Farbe zu schwarzweißen Abbildungen, Filmen oder Fernsehprogrammen zu beschreiben. Dieser Prozess ersetzt die Intensität oder Helligkeit der Pixel durch einen Vektor in einem dreidimensionalen Farbraum mit Farbton, Sättigung und Helligkeit oder einfach RGB. Der Nutzen der Bildeinfärbung wird manchmal angezweifelt, jedoch erhöhen Farben die Bildwirkung z. B. einer alten Schwarzweiß-Fotografie oder lassen einen alten Film attraktiver erscheinen. Da die Zuordnung von Intensität zu Farbe kein inhärent "korrektes" Ergebnis besitzt, spielen menschliche Beeinflussung und/oder externe Informationen üblicher Weise eine große Rolle. In diesem Beitrag stellen wir eine neue Methode der Bildeinfärbung vor, welche die morphologische Abstands-transformation und die Bildstruktur ausnutzt, um vom Benutzer festgelegte Farben automatisch in einem Grauwertbild fortzupflanzen. Die Leistungsfähigkeit des Algorithmus erlaubt dem Benutzer, interaktiv zu arbeiten und das Resultat unverzüglich nach der Festlegung der Farbe zu sehen. Im Vortrag wird gezeigt, dass für Bilder die vorgeschlagene Methode ohne präzise Segmentierung Einfärbungsergebnisse von hoher Qualität liefern kann.

Abstract:

Colorization is a term used to describe a computerized process for adding color to back and white pictures, movies or TV programs. This process involves replacing a scalar value that represents pixels' intensity or luminance by a vector in a three dimensional color space with hue, saturation and luminance or simply RGB.The value of colorization, is sometimes controversial, nevertheless colors increase the visual appeal of an image such as an old black and white photograph or make an old movie look more attractive.

Since the mapping between intensity and color has no inherently "correct" solution, human interaction and/or external information usually play a large role. In this paper we present a novel colorization method that takes advantage of the morphological distance transform and image structures to automatically propagate the color scribbled by a user within the gray scale image. The effectiveness of the algorithm allows the user to work interactively and obtain the desired results promptly after providing the color. In the paper we show that the proposed method allows for high quality colorization results for still images without precise segmentation.

1. INTRODUCTION

The very first approaches in image colorization are dated for 1842 and possibly earlier, when the movies ware colorized manually by painting the celluloid film frame by frame. By around 1905, Pathé brothers introduced Pathécome, a stencil process that involved cutting glass stencils for each frame with a pantograph. These earliest form of colorization introduced limited color into a black and white film using dyes to create the desired visual effect. Movies colorized with these techniques have softer contrast and fairly pale, flat and washed out color.

With the rapid development of computer technology, adding color to the gray scale images and movies in a way that looks natural to most human observers became a problem that challenged the motion picture industry within the computer vision community. In the last few years, several advanced and effective techniques have been proposed. These techniques are based on: *luminance keying, color transfer* [1], *image analogies* [2], *motion estimation* [3], *segmentation* [4], *color prediction* [5], *probabilistic relaxation* [6], *chrominance blending* [7], among many other techniques.

In this paper we show that using our novel colorization method based on the generalized distance transformation, it is possible to obtain satisfactory colorization results in a very short time and with small amount of work.

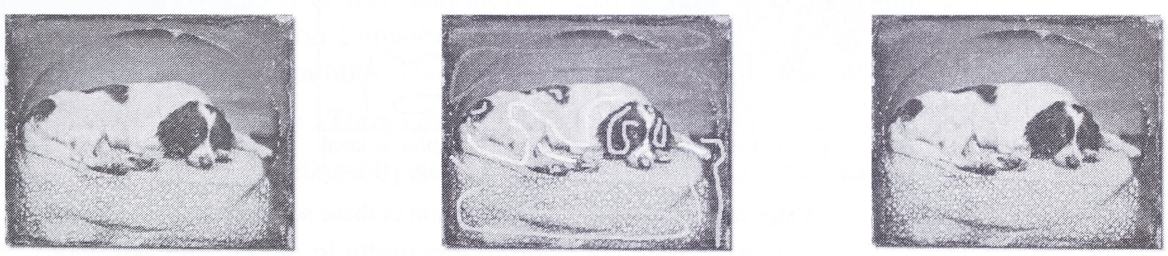
This paper is organized as follows. The next section presents the proposed colorization algorithm. Subsection 2.1 reviews a standard fast algorithm for computing an approximation of the Euclidean distance from a pixel to a set of pixels. In Subsection 2.2 we extend the distance transform to take into account image structures, and in Subsection 2.3 we describe the color blending process. Finally, in Section 3 we present our colorization results and in Section 4 we formulate the final conclusions.

2. THE PROPOSED COLORIZATION ALGORITHM

In this section we present our novel algorithm for image colorization that exploits the distance transformation (DT) and luminance changes within the source grayscale image. Our goal is to create a fast and effective colorization algorithm which does not require precise segmentation as well as any other color images as a reference, (see Fig. 1).

2.1 Distance transform

Since, the color is provided simply by scribbling the image, the first step of our algorithm, after the user scribbles the image, is to isolate the color scribbles and compute the distance from each pixel of the source gray scale image to these scribbles. For this purpose we have implemented three different kinds of distances, [8]: 4-connectivity city-block distance (d_4), 8-connectivity chess-board distance (d_8) and chamfer 5-7-11 distance (d_{ch}). As a result of this transformation, we obtain a gray scale image whose intensities show the distance to the object (scribble) from each image pixel.



(a) Gray scale image. (b) Scribbled image. (c) Colorized image.

Fig.1: Illustration of the proposed colorization process.

Thus, let P be a binary image defined on an image domain grid G in which:

$$\langle \mathrm{P} \rangle = \{ p : p \in \mathrm{G} \wedge \mathrm{P}(p) = 1 \}, \quad \langle \overline{\mathrm{P}} \rangle = \{ p : p \in \mathrm{G} \wedge \mathrm{P}(p) = 0 \}, \tag{1}$$

are proper subsets of G. For any grid metric, the d_α distance transform of P associates with every pixel p of $\langle \mathrm{P} \rangle$ the d_α distance from p to $\langle \overline{\mathrm{P}} \rangle$.

The d_4, d_8 or d_{ch} distance transform of P are computed by scanning G twice with a suitable structuring element and performing a series of local operations. The distance transforms influence the accuracy of the Euclidean distance approximation. The structuring element related to the distance transform, consists of a pattern specified as the coordinates of a number of discrete points relative to some origin, (see Fig. 2). In our algorithm the best results are obtained with chamfer 5-7-11 structuring element (Fig. 2c), since d_{ch} provides the best approximation of Euclidean distance.

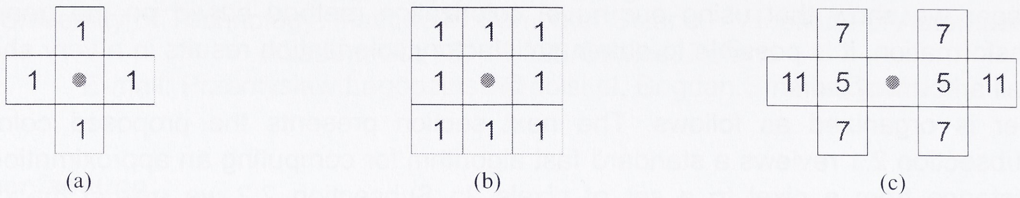


Fig.2: Structuring elements for: d_4 (a), d_8 (b) and d_{ch} (c) distance. The dot represents its origin.

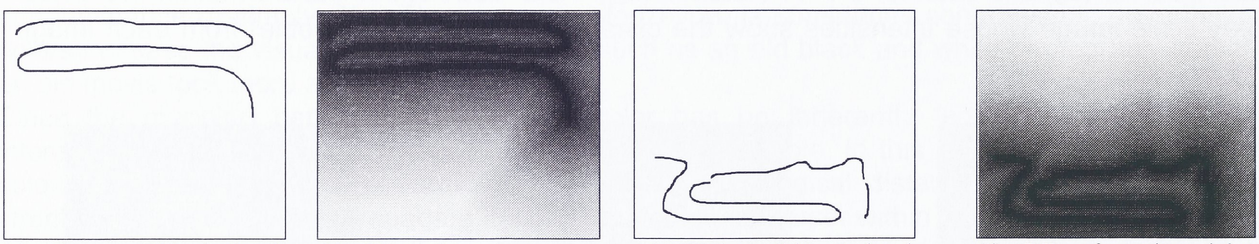
For any $p \in \mathrm{G}$ let $\mathrm{B}(p)$ (before scan) be the set of pixels adjacent to *p* that precedes *q* when G is scanned, and let $\mathrm{A}(p)$ (after scan) be the remaining neighbors of *p*. Then, during the first scan (in top-left to bottom-right direction) we compute:

$$f_1(p) = \begin{cases} 0 & if \quad p \in \langle \mathrm{P} \rangle, \\ \min\{f_1(q)+1 : q \in \mathrm{B}(p)\} & if \quad p \in \langle \bar{\mathrm{P}} \rangle. \end{cases} \tag{2}$$

After the first scan, we approach to the second scan in reverse direction i.e. bottom-right to top-left, and compute the following:

$$f_2(p) = \min\{f_1(p), f_2(q)+1 : q \in \mathrm{A}(p)\}. \tag{3}$$

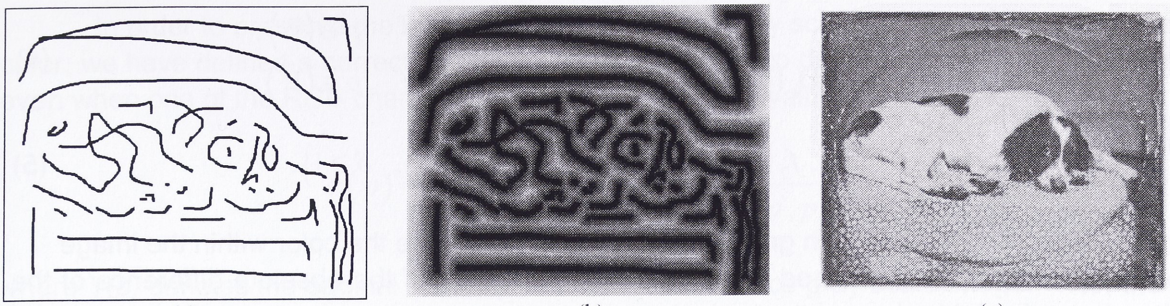
Thus, after the second scan we obtain the distance values that can be expressed as intensities of points within the grayscale image (Fig. 3b).



(a) Green scribbles isolated from the image (1b)
(b) DT performed on green scribbles.
(c) Violet scribbles isolated from the image (1b)
(d) DT performed on violet scribbles.

Fig.3: Scribbles isolated from the image and standard distance transform of these scribbles using the d_{ch} metric.

Colorization using standard distance transformation methods produces promising results. However, images obtained with this scheme show, that the method based on standard DT does not have the ability to detect boundaries between objects, preserve original image structures and requires a large amount of work while scribbling the image, (see Fig. 4).

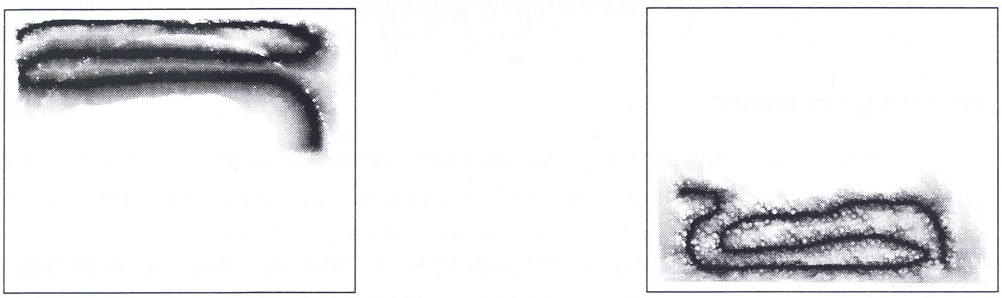


(a) (b) (c)

Fig.4: Example of a colorization process using standard d_{ch} transform.

2.2 Extended distance transform

Another aspect, that is considered in our algorithm are intensity changes within the source gray scale image. We are investigating these changes in order to detect boundaries between objects and preserve the original image structures, (see Fig. 5).



(a) Extended DT performed on green scribbles. (b) Extended DT performed on violet scribbles.

Fig.5: Extended distance transform of green and violet scribbles from Figs. 3a, c, using d_{ch} metric.

They are calculated by taking the absolute value of the difference between the intensity values of two neighboring image points *p* and *q*, and are defined as: $\mathrm{D}(p,q)=\left|\mathrm{Y}(p)-\mathrm{Y}(q)\right|$, where $\mathrm{Y}(p)$ denotes intensity value at a point *p*.

The intensity values of the neighboring points within the image are usually very close to each other and the transition between objects is very smooth, which causes that the boundary between them is hardly noticeable. Therefore, in order to amplify the intensity changes, we decided to rise D to the power of γ. Thus we have: $\mathrm{D}_e(p,q)=\left|\mathrm{Y}(p)-\mathrm{Y}(q)\right|^{\gamma}$, where $\gamma>1$ denotes the exponent whose value is defined by a user. Usually, satisfactory results are obtained for the $\gamma=2$ (see Fig. 6).

Since the calculation of intensity changes D_e, as well as computation of distances (Eqs. 2, 3), requires a sequence of local operations on neighboring pixels within the gray scale image, we have decided to merge these steps together. Taking this opportunity, we have also decided to introduce one more parameter δ, which will allow us to investigate the influence of topological distance on the quality of resulting colorized images.

Finally, we obtain the following equations that define the distance transform, which is an extension of the DT proposed in [9]:

$$f_1(p)=\begin{cases}0 & if \quad p\in\langle \mathrm{P}\rangle,\\ \min\{f_1(q)+\mathrm{D}_e(p,q)+\delta : q\in(p),\ \delta\geq 1\} & if \quad p\in\langle \bar{\mathrm{P}}\rangle.\end{cases} \tag{4}$$

$$f_2(p)=\min\{f_1(p), f_2(q)+\mathrm{D}_e(p,q)+\delta : q\in \mathrm{A}(p),\ \delta\geq 1\}. \tag{5}$$

that are next used in conjunction with gradient functions to propagate the color within the image

Good results were also obtained using a following function of the absolute difference of the gray scale values of neighboring points:

$$f_1(p)=\begin{cases}0 & if \quad p\in\langle \mathrm{P}\rangle,\\ \min\left\{f_1(p)+\left(1+\sum\exp\left(-\dfrac{\mathrm{D}(p,q)}{\beta}\right)\right)^{-1}+\delta : q\in \mathrm{B}(p),\ \delta\geq 1\right\} & if \quad p\in\langle \bar{\mathrm{P}}\rangle\end{cases}. \tag{6}$$

$$f_2(p)=\min\left\{f_1(p), f_2(q)+\left(1+\sum\exp-\left(\frac{\mathrm{D}(p,q)}{\beta}\right)\right)^{-1}+\delta : q\in \mathrm{A}(p),\ \delta\geq 1\right\}, \tag{7}$$

where β is a smoothing parameter.

2.3 Gradient functions

The intensities within the gray scale image can change in various ways i.e. creating a smooth transition between two shades or sharp-edge boundary between objects. That suggest us the way the color, we are adding to the gray scale image, should change. To make use of these indications, we have decided to use kernel functions that are able to reproduce the structures of the source image. We have implemented two kernel functions in out algorithm; linear $f_l(d)$ and a Gaussian like $f_g(d)$:

$$f_l(d)=\left|\frac{d-1}{h}-1\right|, \qquad f_g(d)=\exp\left(-\left(\frac{d}{h}\right)^2\right), \tag{8}$$

where parameter *h,* set by the user, determines the smoothness of the functions. In both cases *d* denotes the value of the extended DT we obtain from Eqs. 6 and 7 for each point of the gray scale image.

These functions are used next as weights to determine the color *C* of a given point *p* during the additive color mixing process:

$$C(p)=\frac{C_1\cdot f_1(d,p)+C_2\cdot f_2(d,p)}{f_1(d,p)+f_2(d,p)}, \tag{9}$$

where C_1 and C_2 are colors scribbled by the user and $f_1(d,p)$, $f_2(d,p)$ are weights obtained for a given point *p* using one of the presented kernel gradient functions. Since the proposed algorithm is iterative and colors are added one by one, indexes 1 and 2 in Eqs. 9-11 correspond respectively to the current and previous colorization step.

In order to preserve the intensity from the source gray scale image in the newly propagated color, we have defined a correction factor r which allows us to determine the desired pixel intensity even when one of the RGB channels reaches the maximum value:

$$r(p)=\frac{Y(p)}{\max\left(C_1\cdot f_1(d,p)+C_2\cdot f_2(d,p)\right)}, \tag{10}$$

where *Y(p)* denotes the original intensity value from the source gray scale image. Thus, the final color can be achieved from:

$$C(p)=\frac{C_1\cdot f_1(d,p)+C_2\cdot f_2(d,p)}{f_1(d,p)+f_2(d,p)}\cdot r(p). \tag{11}$$

In this way, we are able to determine the color for each point of the image taking the colors scribbled by a user as a basis, and propagate them, preserving the original intensity values within the color image. Our method uses the max function within the RGB color space to determine the luminance value. However, the proposed method can exploit various fuzzy norms, other color spaces like HSV or lαβ and various methods of luminance calculation within the color image.

3. RESULTS

The results shown here were all obtained using the presented method that works on the basis of the extended distance transformation. The proposed solution is iterative and adds color one by one. This method was implemented using Microsoft Visual C# .NET 2.0. Although the code is not fully optimized, this implementation of algorithm works fast enough to allow the user for interactive work without noticeable delays and achieving real-time preview. In [10], the authors present the comparison of the processing time between their and the algorithm presented in [4]. Taking the same set of test images our algorithm produces following results:

Test image	Image size	Time [s], [4]	Time [s], [10]	Proposed [s]
Cats	319x267	15.20	0.71	1.21
Girl	318x238	10.12	0.42	0.92
Building	399x299	15.26	0.91	1.40

Tab.1: Processing time comparison of colorization methods.

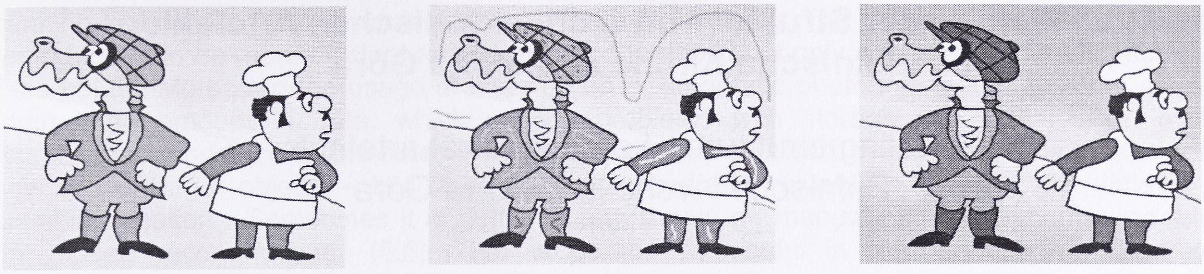
4. CONCLUSIONS

In the future work, we will experiment with new fast distance transformation algorithms to make our algorithm more efficient. The optimization of the code of our application will allow us to shorten the time of colorization process in the case of very large images. We are very optimistic that these improvements will allow us to produce even better colorization results in the near future.

Fig.6: Work of art colorization examples. In columns: original color images, gray scale images scribbled with color and our colorization results, (re-creation of lost painting in Artus Court in Gdańsk, source: http://www.izdebski.pl).

(a) (b) (c)

Fig.7: Movie frame colorization examples: (a) gray scale image, (b) scribbled image and (c) our colorization result.



(a) (b) (c)

Fig.8: Cartoon colorization example: (a) grayscale image, (b) scribbled image and (c) our colorization result.*

REFERENCES

[1] REINHARD E., ASHIKHMIN M., GOOCH B, and SHIRLEY P., Color transfer between images. IEEE Computer Graphics and Applications, Vol. 21, No. 5, pp. 34-41, May 2001.

[2] WELSH T., ASHIKHMIN M., and MUELLER K., Transferring color to grayscale images. Proc. ACM SIGGRAPH Conf., 2002, pp. 277-280.

[3] PAN Z., DONG Z., and ZANG M., A new algorithm for adding color to video and animation clips. Proc. WSCG, International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2004, pp. 515-519.

[4] LEVIN A., LISCHINSKI D., and WEISS Y., Colorization using optimization. Proc. ACM SIGGRAPH Conf., 2004, pp. 689-694.

[5] MADEIRA J. S., STORK A., and GROB M. H., An approach to computer supported cartooning. The Visual Computer, Vol. 12, 1996, pp. 1-17.

[6] HORIUCHI T., Colorization algorithm for gray-level image by probabilistic relaxation. Proc. IEEE International Conference on Pattern Recognition, 2003, pp. 867-870.

[7] YATZIV L., and SAPIRO G, Fast image video colorization using chrominance blending. IEEE Transactions on Image Processing, Vol. 15, May 2006, pp. 1120-1129.

[8] KLETTE R., and ROSENFELD A., Digital geometry – Geometric methods for digital picture analysis. Morgan Kauffman, 2004.

[9] TAIVONEN P. J., New geodesic distance transform for gray-scale images. Elsevier – Pattern Recognition Letters, Vol. 17, 1995, pp. 437-450.

[10] LI. Y., LIZHUANG M., and DI W., Fast colorization using edge and gradient constraints. Proc. of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2007, pp. 309-316.

* The color images are available at: http://www.rybweb.eu/colorization