colorization, distance transformation, colour blending

Przemysław ŁAGODZIŃSKI*, Bogdan SMOŁKA**

MEDICAL IMAGE COLORIZATION

Colorization is a term used to describe a computerized process for adding colour to black 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 colour space with luminance, saturation and hue or simply RGB. The colorization process is also used to convert the grey scale to colour on medical images. Colour increases the visual appeal of an image and it also makes a medical visualization more attractive. Changes in colour are more easily perceived then changes in shades of grey and therefore this procedure makes the interpretation and understanding of the image easier. Since the mapping between intensity and colour has no inherently "correct" solution, human interaction and/or external information usually plays a large role. In this paper we present a novel colorization method that takes advantage of the morphological distance transformation and image structures to automatically propagate the colour scribbled by the user within the grey scale image. The effectiveness of the algorithm allows the user to work interactively and obtain the desired results promptly after providing the colour. In the paper we show that the proposed method allows for high quality colorization results for still images without precise segmentation.

1. INTRODUCTION

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

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 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

* Silesian University of Technology, 44-100 Gliwice, Akademicka 16, Institute of Computer Science, Przemyslaw.Lagodzinski@polsl.pl,
** Silesian University of Technology, 44-100 Gliwice, Akademicka 16, Institute of Automatic Control, Bogdan.Smolka@polsl.pl

distance transform to take into account image structures, and in subsection 2.3 we describe the colour 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

2.1. DISTANCE TRANSFORM

Since, the colour is provided simply by scribbling the image, the first step of our algorithm, after the user scribbles the image, is to isolate the colour scribbles and compute the distance from each pixel of the source grey scale image to these scribbles. For this purpose we have implemented three different kinds of distances, [8]: 4-connectivity city-block distance (d4), 8-connectivity chess-board distance (d8) and chamfer 5-7-11 distance (dch). As a result of this transformation, we obtain a grey scale image whose intensities show the distance to the object (scribble) from each image pixel.

Grey scale image. Scribbled image. Colorized image.

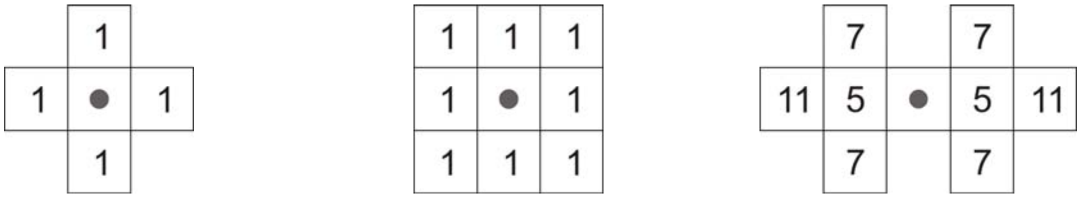
Fig.1. Illustration of the proposed colorization process.

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

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

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 $\langle \bar{\mathrm{P}} \rangle$.

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.



(a) (b) (c)

Fig.2. Structuring elements for: d_4 (a), d_8 (b) and d_{ch} (c) distance. The dot in the middle of each structuring element represents its origin.

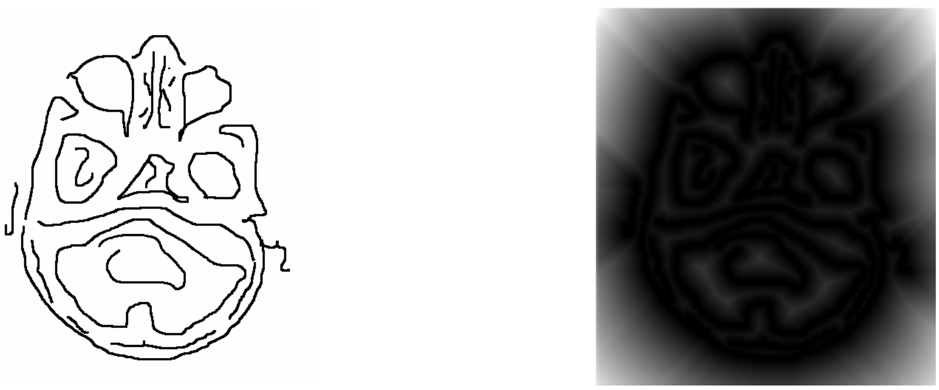
For any $p \in \mathrm{G}$ let $\mathrm{B}(p)$ (before scan) be the set of pixels adjacent to *p* that precedes *q* when G is scanned, and let $\mathrm{A}(p)$ (after scan) be the remaining neighbours of *p*. Then, during the first scan (in top-left to bottom-right direction) we compute:

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

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 greyscale image (Fig. 3b).



Scribbles isolated from the image (1b) DT performed on this 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 grey scale image. We are investigating these changes in order to detect boundaries between objects and preserve the original image structures (see Fig. 5).

Fig.5. Extended distance transform of scribbles from Fig. 3a using the d_{ch} metric.

They are calculated by taking the absolute value of the difference between the intensity values of two neighbouring image points p and q, and are defined as: $D(p,q) = |Y(p) - Y(q)|$,

where $Y(p)$ denotes intensity value at a point p.

The intensity values of the neighbouring 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: $\mathrm{D}_e(p,q) = |\mathrm{Y}(p) - \mathrm{Y}(q)|^{\gamma}$,

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).

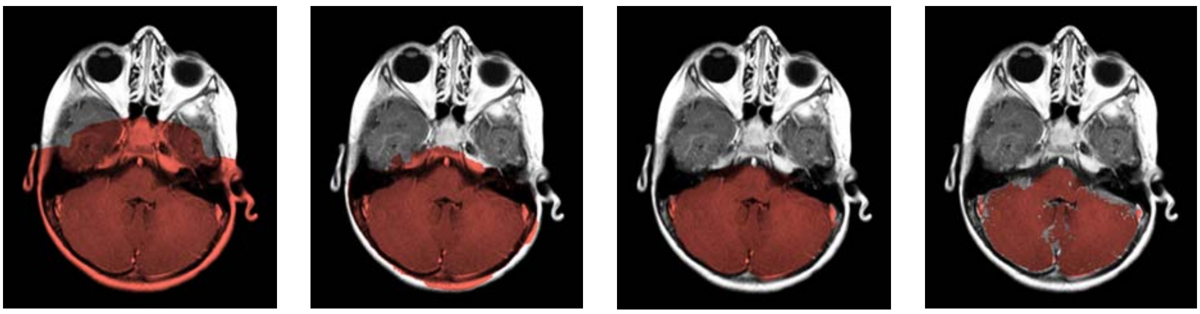
Since the calculation of intensity of changes D_e, as well as computation of distances (Eqs. 2, 3), requires a sequence of local operations on neighbouring pixels within the grey scale image, we have decided to merge these steps together. Taking this opportunity, we have also decided to introduce one 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]:

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

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

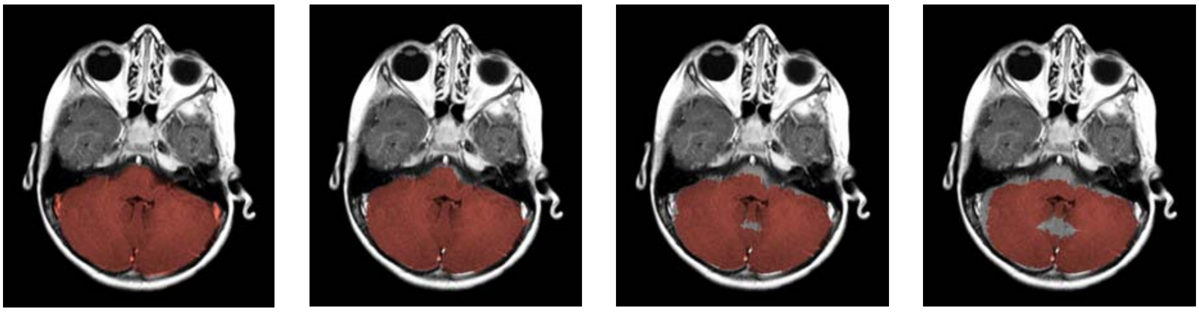
that are next used in conjunction with gradient functions to propagate the colour within the image.



γ = 0 γ = 1 γ = 2 γ = 0

Fig.6. Illustration of the influence of γ on the quality of resulting colorized image.

While the γ parameter makes the method more sensible to intensity changes and boundaries between object as depicted in Fig. 6, the δ makes the proposed method sensitive to topological distances on the image domain (Fig. 7).



δ = 1 δ = 5 δ = 10 δ = 15

Fig.7. Illustration of the influence of δ on the colorization results.

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

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

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

where β is a smoothing parameter.

2.3. GRADIENT FUNCTIONS

The intensities within the grey 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 colour, we are adding to the grey 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 $f_g(d)$:

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

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 grey scale image.

These functions are used next as weights to determine the colour C of a given point p during the additive colour 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 colours 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 (see Fig. 8).

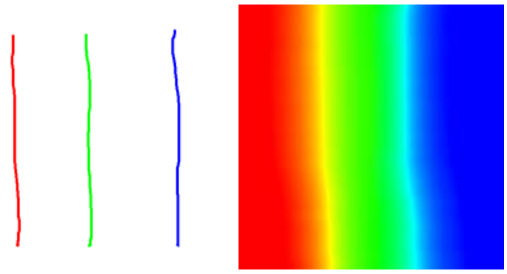
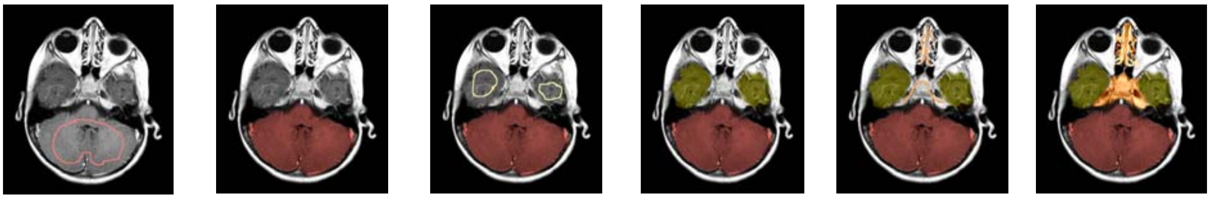


Fig.8. Illustration of additive colour blending using weights.

Since the proposed algorithm is iterative and colours are added one by one, indexes 1 and 2 in Eq. 9 correspond respectively to the current and previous colorization step as shown in Fig. 9.



(a) (b) (c) (d) (e) (f)

Fig.9. Illustration of the iterative colour application.

In order to preserve the intensity from the source grey scale image in the newly propagated colour, we have introduced three different approaches depending on the specifics of a chosen colour space: *RGB*, *HSV* or *YUV*.

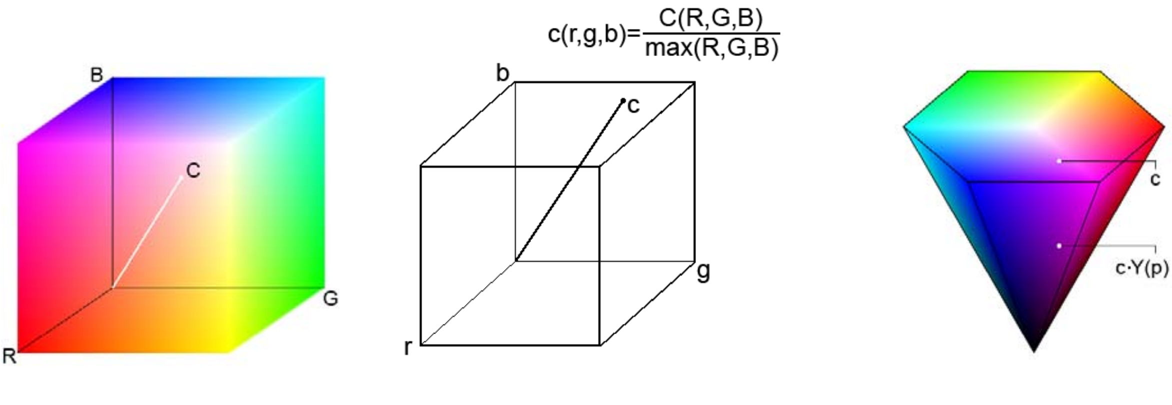
In case of the *RGB* colour space, in the first step we compute the ratio between the channels using the following:

$$r=\frac{R}{\mathrm{I}},\quad g=\frac{G}{\mathrm{I}},\quad b=\frac{B}{\mathrm{I}}, \tag{10}$$

where $\mathrm{I}=R+G+B$. The second step consists of the multiplication r, g, b ratio factors by the original intensity value $\mathrm{Y}(p)$ taken from the source grey scale image. In this way, we have obtained the colour with the desired intensity within the *RGB* colour space.

Considering the intensity computation within the *HSV* colour space, we have noticed, that using the *max* function it is possible to pass smoothly from *RGB* to *HSV* colour space. Like in the case of *RGB* colour space in the first step we have to compute ratio between *RGB* channels (Eq. 10) by dividing each channel with the intensity taken as: $\mathrm{I} = \max(R,G,B)$.

As a result we obtain a vector placed inside a unit cube. Multiplying this vector with the original intensity value $\mathrm{Y}(p)$, we get the final colour with desired intensity value (Fig. 10).



(a) (b) (c)

Fig.10. Illustration of intensity calculation process in the *HSV* colour space.

To obtain the representation of the colour *C* within the *YUV* colour space, we determine the *U* and *V* from:

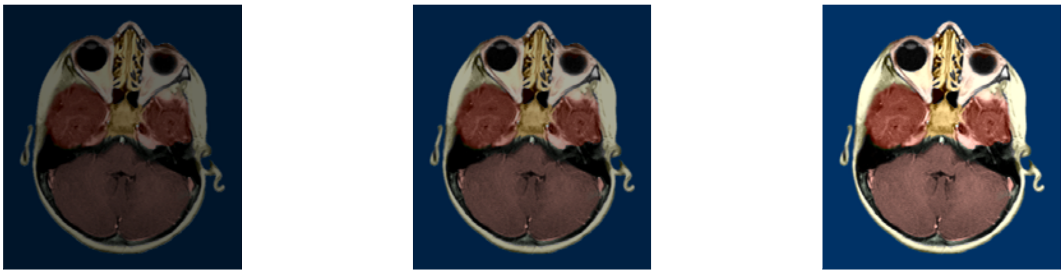
$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{11}$$

and then replace *Y* channel representing the luminance with the proper intensity value $\mathrm{Y}(p)$. At the end we convert the colour back to the *RGB* colour space:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{Y} \\ U \\ V \end{bmatrix} \tag{12}$$

to obtain the final colour.

As you can observe in Fig. 11, images obtained using the same colorization algorithm but with colour intensity calculated in three different ways differ from each other. The image presented in Fig. 11c, whose intensity was calculated using *YUV* colour model, seems to look more realistic then two others.



(a) (b) (c)

Fig.11. Illustration of differences between images where the colour intensity was calculated in three different colour spaces: a) *RGB*, b) *HSV* and c) *YUV*.

3. COLORIZATION 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 colour 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 Levin's algorithm [4]. Taking the same set of test images our algorithm produces following results:

Table 1. Processing time comparison of colorization methods.

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

The following figures present colorization examples of medical images acquired with Computer Tomography (CT) (Fig. 12), Magnetic Resonance Imaging (MRI) (Fig. 13) and X-Ray (Fig. 14).

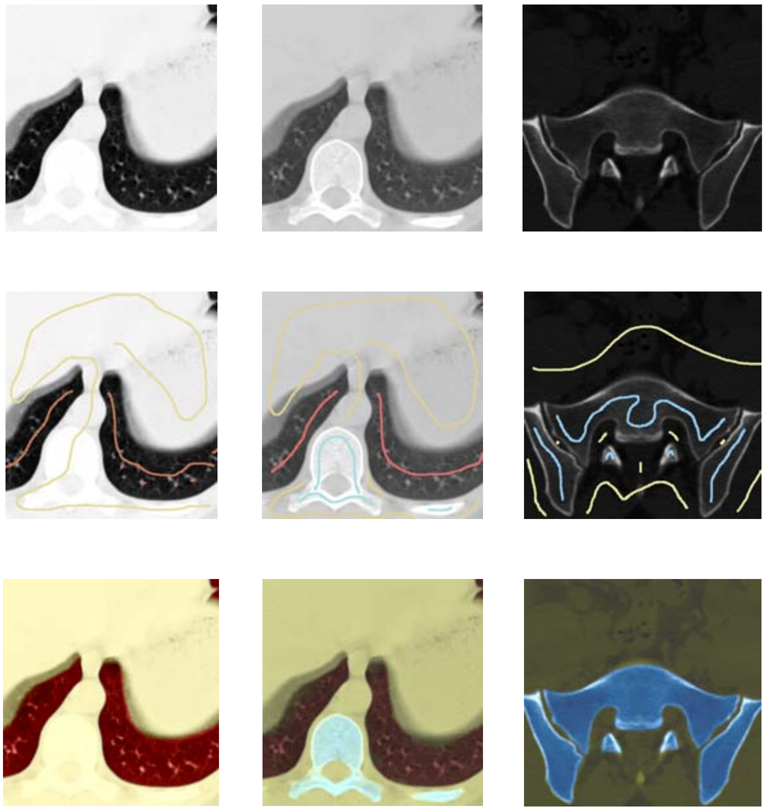


Fig.12. Colorization examples of Computer Tomography images. In rows: original grey scale images, scribbles images and our colorization result.

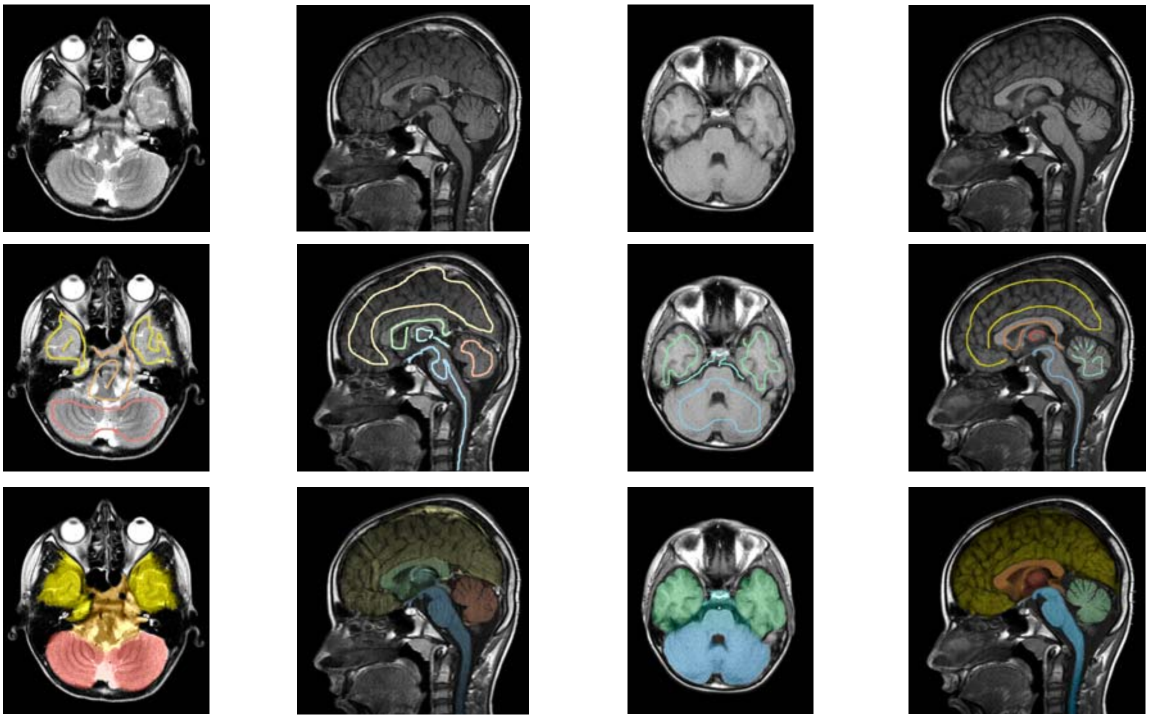


Fig.13. Magnetic Resonance Imaging colorization example. In rows: original grey scale images, scribbles images and our colorization result.

(a) (b) (c)

Fig.14. Colorization examples of images acquired with X-Rays: (a) grey scale image, (b) scribbles image and (c)our colorization result.

4. CONCLUSIONS

In the future work, we will experiment with new fast distance transformation algorithms to make our algorithm more efficient. The optimisation 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.

BIBLIOGRAPHY

[1] REINHARD E., ASHIKHMIN M., GOOCH B, and SHIRLEY P., Colour 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 colour to greyscale images. Proc. ACM SIGGRAPH Conf., 2002, pp. 277-280.

[3] PAN Z., DONG Z., and ZANG M., A new algorithm for adding colour 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 grey-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 grey-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.