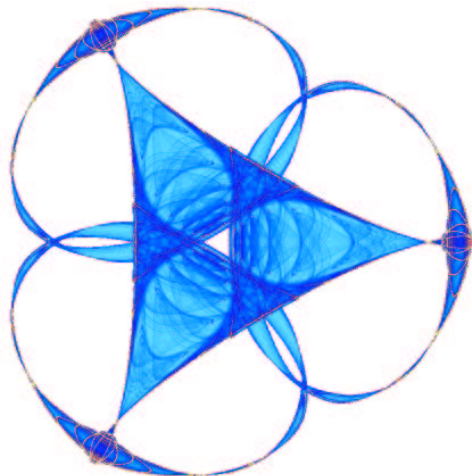# INPAINTING THE COLORS

By

Guillermo Sapiro

IMA Preprint Series # 1979

(May 2004)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

Inpainting the Colors[*]

Guillermo Sapiro
Electrical and Computer Engineering and
Digital Technology Center
University of Minnesota
Minneapolis, MN 55455
guille@ece.umn.edu

May 21, 2004

Abstract

A framework for automatic image colorization, the art of adding color to a monochrome image or movie, is presented in this paper. The approach is based on considering the geometry and structure of the monochrome luminance input, given by its gradient information, as representing the geometry and structure of the whole colored version. The color is then obtained by solving a partial differential equation that propagates a few color scribbles provided by the user or by side information, while considering the gradient information brought in by the monochrome data. This way, the color is inpainted, constrained both by the monochrome image geometry and the provided color samples. We present the underlying framework and examples for still images and movies.

1   Introduction

As explained in [14],[1] *colorization* is a term introduced by Wilson Markle in 1970 to describe the computer assisted process he invented for adding color to black and white movies [6]. The term is generically used now to describe the process of adding color to monochrome still images and movies. The value of color in art, and colorization in particular, is sometimes controversial, and some of the relevance of this for image processing is addressed and commented in [7, 14]. Colorization is

---

[1]This work motivated us to use the concepts of inpainting, as detailed below, in the interesting challenge of colorization.

in general an active and challenging area of research with a lot of interest in the image editing community. In addition to the original (controversial) intensions of coloring old movies, colorization has applications such as color changing (editing) and compression. The latter comes from the fact that as shown in this paper, with the luminance information and just some samples of the color (much less than the ordinary sub-sampling in common compression schemes), the color components of the data can be faithfully recovered. This has implications also in wireless image transmission, where lost image blocks can be recovered from the available channels [19].

Classically, colorization is done by first segmenting the image and then assigning colors to each segment. This is not only a very time consuming process, but as shown in [14], can lead to significant errors, particularly in fuzzy boundaries. For movies, colorization also requires the tracking of these regions, adding computational complexity and the typical difficulties when tracking non-rigid objects.

Our framework is motivated by two main bodies of work, one dealing with the geometry of color images and one dealing with image inpainting, the art of modifying an image in a non-detectable work. Caselles *et al.* [7] and Chung and Sapiro [8] (see also [22]) have shown that the (scalar) luminance channel faithfully represents the geometry of the whole (vectorial) color image. This geometry is given by the gradient and the level-lines, following the mathematical morphology school. Moreover, Kimmel [12] proposed to align the channel gradients as a way of denoising color images, and showed how this arises naturally from simple assumptions. This body of work brings us to the first component of our proposed technique, to consider the gradient of each color channel to be given (or hinted) by the gradient of the given monochrome data.[2] The second component of our framework comes from inpainting [3]. In addition to having the monochrome (luminance) channel, the user provides a few strokes of color, that need to be propagated to the whole color channels, clearly a task of inpainting. Moreover, since from the concepts described above, information on the gradients is also available (from the monochrome channel), this brings us to the inpainting technique described in [1] (see also [2]), where we have interpreted inpainting as recovering an image from its gradients, these ones obtained via elastica-type interpolation from the available data. Recovering an image from its gradients is of course a very old subject in image processing and was studied for example in [11] for image denoising (see also [15]) and in [18] for a number of very interesting image editing tasks. Combining both concepts we then obtain that colorizing reduces to finding images (the color channels) provided their gradients (which are derived from the monochrome data) and constrained to color strokes provided by the user. Below we present partial

---

[2]This monochrome image becomes the luminance of the reconstructed color data.

differential equations for doing this, which in its simplest form, is just a Poisson equation with Dirichlet boundary conditions. This puts the problem of colorizing in the popular framework of solving image processing problems via partial differential equations [13, 17, 21].

1.1 Additional comments on colorization prior art

Before concluding this introduction and going into the technical details, we should point out some relevant works in the literature. For more details and in particular description of some commercial products which heavily rely on user intervention, see [14].

Markle and Hunt [16] original work represents the trend mentioned above of segmenting, tracking, and color assignment. Welsh *et al.* [24] present a semi-automatic technique for colorizing a grayscale image by transferring color from reference data. The idea is to transfer color from neighborhoods in the reference image that match the luminance in the target data. There is then an underlying assumption that different colored regions give rise to distinct luminance, and their approach works properly only when this is not violated, otherwise requiring significant user intervention. The results reported by the authors are quite impressive, although the technique intrinsically depends on the user to find proper reference data. More details on the problems with this work are reported in [14], which as said above, inspired our own work. Levin *al.*, as done in this work, assume that in addition to the monochrome data, the user scribes some colors in the image. First, in contrast with the work in [24], this gives much more control to the user, both in the selection of the desired colors (without having to search in image databases), and in the correction of possible errors of the automatic algorithm. This last step is very important, since we are "inventing" information (the color), and then it is expected that the algorithm will make decisions that the user would like to change. Therefore, the proposed technique has to intrinsically allow for that. As in our approach, in [14] this is easily done by adding strokes (color constraints). In [14], the color is added following the simple premise that neighboring pixels having similar intensities in the monochrome data should have similar colors in the chroma channels. This premise is formalized in our work, following [7, 8], by using the gradients from the monochrome provided image, thereby transmitting the geometry among the channels. This premise is materialized in [14] by a discrete variational formulation that penalizes for the difference between a pixel color value and the weighted average of the colors in its neighborhood.[3] The weights are provided by the monochrome data. Intrinsic to their approach is the concept

---

[3]This is a discrete analogue of classical penalty functions of the types used in color image processing, e.g., [23].

of neighborhood, which forces in the case of movies to compute optical flow. We avoid this by using the spatial and temporal gradient. We therefore proposes a simpler algorithm, which uses the full gradient information as suggested by the color image geometry works in [7, 8] and the inpainting and editing works from image gradient in [1, 2, 11, 18].

2 Inpainting colors from gradients and boundary conditions

We start with the description of the proposed algorithm for still images. Let $Y(x, y) : \Omega \rightarrow I\!\!R^+$ be the given monochromatic image defined on the region Ω. We will work on the $YCbCr$ color space (other color spaces could be used as well), and the given monochromatic image becomes the luminance Y. The goal is to compute $Cb(x, y) : \Omega \rightarrow I\!\!R^+$ and $Cr(x, y) : \Omega \rightarrow I\!\!R^+$. We assume that colors are given in a region Ω_c in Ω such that $|\Omega_c| << |\Omega|$ (otherwise, simple interpolation techniques would be sufficient). This information is provided by the user via color strokes in editing type of applications, or automatically obtained for compression (selected compressed regions) or wireless (non lost and transmitted blocks) applications. The goal is from the knowledge of Y in Ω and Cb, Cr in Ω_c to inpaint the color information (Cb, Cr) into the rest of Ω.

Following the description in the introduction, Cb (and similarly Cr) is reconstructed from the following minimization problem:

$$\min_{Cb} \int_\Omega \rho(\| \nabla Y - \nabla Cb \|) d\Omega, \tag{1}$$

with boundary conditions on Ω_c, $\nabla := \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$ is the gradient operator, and $\rho(\cdot) : I\!\!R \rightarrow I\!\!R$. The basic idea is to force the gradient (and therefore the geometry) of Cb to be as the geometry of the given monochromatic image Y while preserving the given values of Cb at Ω_c. Note that although here we consider these given values as hard constraints, they can also be easily included in the above variational formulation in the form of soft constraints. This can be particularly useful for compression and wireless applications, where the given data can be noisy, as well as for editing applications where the user only provides color hints instead of color constraints. For ease of the presentation, we continue with the assumption of hard constraints. In [5] we discussed a number of robust selections for ρ in the case of image denoising, while in [1] we used the L_1 norm, $\rho(\cdot) = |\cdot|$, following the work on Total Variation [20]. Of course, the most popular, though not robust, selection is the L_2 norm $\rho(\cdot) = \cdot^2$, which leads via simple calculus of variation to the following

necessary condition for the minimizer in (1):

$$\Delta Cb = \Delta Y, \qquad (2)$$

with corresponding boundary conditions on Ω_c and Δ being the Laplace opera-
tor given by $\Delta := \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$. This is the well known Poisson equation with
Dirichlet boundary conditions.

Equations (1) and (2) can be solved very efficiently by a number of well devel-
oped *Poisson solvers*, e.g., see [9], making our proposed algorithm very simple and
computationally efficient. Note that in contrast with the work in [14], our formula-
tion is continuous, and the vast available literature on numerical implementations
of these equations accurately handles their efficient solution.

To conclude the presentation, we need to describe how to address the coloriza-
tion of movies. Although optical flow can be incorporated as in [14], it would be
nice to avoid its explicit computation. We could implicitly introduce the concept of
motion in the above variational formulation, though we opt for a simpler formula-
tion. Following the color constancy constraint often assumed in optical flow, and if
the gradient fields and motion vectors of all the movie channels are the same, then
of course we can consider $\frac{\partial Y}{\partial t} = \frac{\partial Cb}{\partial t} = \frac{\partial Cr}{\partial t}$, where t is the time coordinate in the
movie. Therefore, equations (1) and (2) are still valid constraints for the movie case
(Ω is now a region in (x, y, t) and Ω_c are 2D spatial strokes at selected frames), as
long as we consider three dimensional gradients and Laplace operators given by
$\nabla := \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t} \right)$, $\Delta := \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial t^2} \right)$, respectively. Anisotropy between
the spatial and temporal derivatives can be easily added to these formulations as
well.

2.1   Comments on different variational formulations

Equations (1) and (2) represent just one particular realization of our proposed
framework. For example (see also comments in the concluding remarks section),
we could constraint the color normalized gradients to follow the luminance nor-
malized gradient. In this case, the variational formulation becomes

$$\min_{Cb} \int_\Omega \rho \left(\frac{\nabla Y}{\| \nabla Y \|} \cdot \nabla Cb - \| \nabla Cb \| \right) d\Omega, \qquad (3)$$

with corresponding boundary conditions. From calculus of variations, the corre-
sponding Euler-Lagrange equation is (for an L_2 norm)

$$\mathrm{div} \left(\frac{\nabla Cb}{\| \nabla Cb \|} \right) = \mathrm{div} \left(\frac{\nabla Y}{\| \nabla Y \|} \right), \qquad (4)$$

which is once again solved using standard efficient numerical implementations [1, 2] (div stands for the divergence). The concepts above transmit to movies as with equations (1) and (2).

3   Examples

In Figure 1 we present the first example. For comparison, we use color from the original image to provide the color strokes on the monochromatic input. The original image is then provided first, followed by the monochromatic image with the color strokes, and followed by the result of our colorization algorithm. Note that the colorized image is visually almost identical to the original image. In Figure 2 we present a number of additional examples for still images. On the first row we show the input monochromatic image with the used color strokes overlayed on them. The result of our algorithm is provided in the second row.[4] Note that as in image inpainting, the original image is not available, and therefore every "reasonable" and visually pleasant result should be considered acceptable. A movie example is presented in Figure 3 for a few colorized frames from the movie Shrek 2. The first column shows to original frames, while the colorized ones are presented on the right. These are obtained by a few random strokes on each frame, using colors from the original movie.

4   Concluding remarks

A simple colorization framework was introduced in this papers. The technique is based on combining concepts from image inpainting with studies on the geometry of color images. Particular realizations of this framework were described, while others are certainly possible. For example, we could use the gradients and optical flow of the monochromatic image and video to explicitly provide the inpainting direction needed in the algorithm introduced in [3]. We could also follow ideas put forward in [4] and colorize in a decomposition domain, using for example ideas from [10] to colorize the texture component, considering color as the "style." More interesting probably is to understand what kind of information is needed in the chroma channels for error controlled colorization. This not only will help in editing images, directing the user to the crucial regions to provide the strokes, but also in the use of colorization for compression and wireless image transmission. These topics will be the subject of future research and will be reported elsewhere.

---

[4]The images are obtained from the Berkeley database.

References

[1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and grey levels," *IEEE Trans. Image Processing* **10**, pp. 1200-1211, August 2001.

[2] C Ballester, V. Caselles, and J.Verdera, "Disocclusion by joint interpolation of vector fields and gray levels," *SIAM Multiscale Modelling and Simulation* **2**, pp. 80-123, 2003.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," *Computer Graphics* (SIGGRAPH), July 2000.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Processing* **12**, pp. 882-889, 2003.

[5] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing* **7:3**, pp. 421-432, 1998.

[6] G. Burns, *Colorization. Museum of Broadcast Communication: Encyclopedia of Television.*

[7] V. Caselles, B. Coll and J-M. Morel, "Geometry and color in natural images," *Journal of Mathematical Imaging and Vision* **16**, pp. 89-105, 2002.

[8] D. H. Chung and G. Sapiro, "On the level-lines and geometry of vector-valued images," *IEEE Signal Processing Letters* **7**, pp. 241-243, September 2000.

[9] J. Demmel, *Lecture Notes on Numerical Linear Algebra,* Berkeley Lecture Notes in Mathematics, Mathematics Dept, UC Berkeley.

[10] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," *Proc. SIGGRAPH*, 2001.

[11] C. Kenney and J. Langan, "A new image processing primitive: Reconstructing images from modified flow fields," pre-print, June 1999.

[12] R. Kimmel, "A natural norm for color processing," *Proc. of Third Asian Conf. on Computer Vision*, Hong Kong, January 8-11, 1998.

[13] R. Kimmel, *Numerical Geometry of Images: Theory, Algorithms, and Applications*, Springer-Verlag, New York, 2004.

[14] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics*, Aug 2004.

[15] M. Lysaker, S. Osher, and X. Tai, "Noise removal using smoothed normals and surface fitting," *UCLA CAM Report* **03-03**, January 2003

[16] W. Markle and B. Hunt, "Coloring a black and white signal using motion detection," *US Patent* **1291260**, 1987.

[17] S. J. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision, and Graphics,* Springer-Verlag, Ney York, 2003.

[18] P. Perez, M. Gangnet, and A. Blake, "Poisson image editing," *Proc. SIG-GRAPH*, 2003.

[19] S. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Processing* **12**, pp. 296-303, 2003.

[20] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* **60**, pp. 259-268, 1992.

[21] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*, Cambridge University Press, January 2001.

[22] G. Sapiro and D. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Trans. on Image Processing* **5**, pp. 1582-1586, 1996.

[23] B. Tang, G. Sapiro, and V. Caselles, "Color image enhancement via chromaticity diffusion," *IEEE Trans. Image Processing* **10**, pp. 701-707, May 2001.

[24] T. Welsh, M. Ashikhmin, and K. Mueller, "Transfering color to grayscale images," *ACM Transactions on Graphics* **21**, pp. 277-280, 2002.
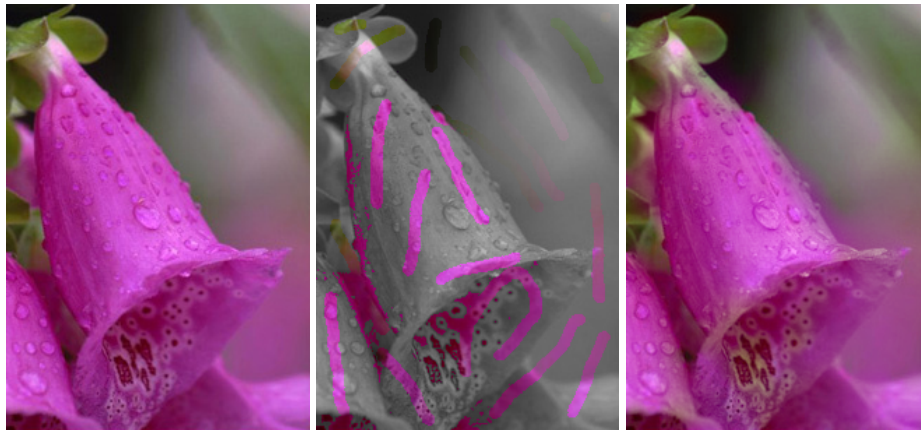
Figure 1: *Still image colorization. The original image is presented first, followed by the monochromatic image with color strokes with colors from the original data, and followed by the colorized image automatically obtained from our technique, which is visually almost identical to the original data. (This is a color figure.)*
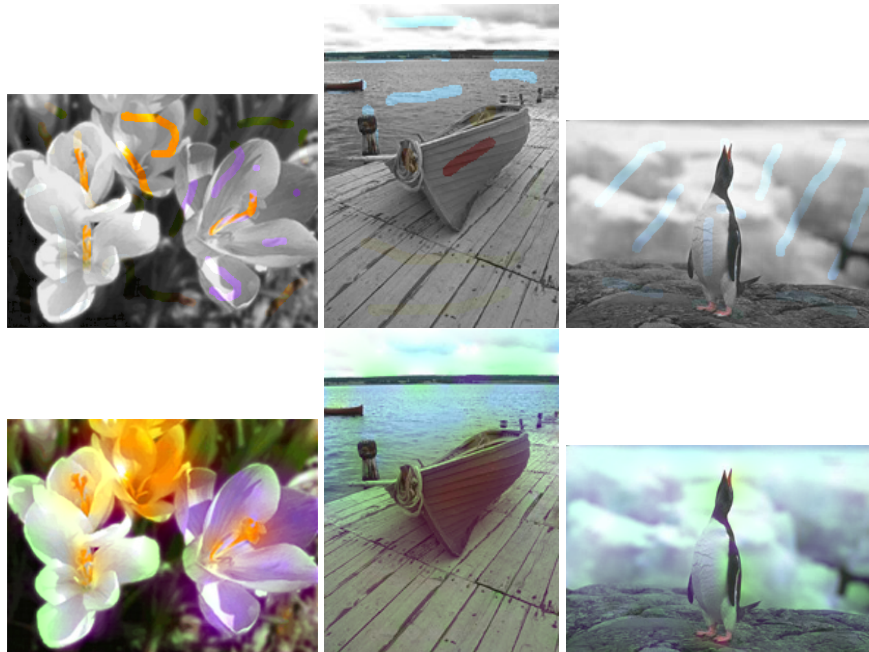
Figure 2: *Still images colorization. The monochromatic images with the color strokes are presented in the first row, followed in the second row by the results of our colorization technique. When the color has drifted too much, the user can easily add strokes to repair this. (This is a color figure.)*

Figure 3: *Movie colorization. Colorized results are on the second column and original frames on the first one (never available to the editor/receiver of course). (This is a color figure.)*