

Image Colorization Using Anisotropic Diffusion

A.M.S. Rahma*

A. A. Hussein**

Received on :23/6/2005

Accepted on :9/3/2006

Abstract

Adding colors to all pixels in monochrome image are difficulty performed. It requires considerable user intervention and remains a tedious, time-consuming, expensive task with no certain results.

The previous methods have not the wanted color-scale and the harmony in colors used.

This paper presents semi-automatic image colorization, where the user only needs to annotate the image with a few desirable color scribbles depending on the user's choice and the indicated colors are automatically diffused to produce a fully colorized image.

The execution depending on the geometry and structure of the monochrome luminance input, given by its gradient information, The color is then diffused by solving a partial differential equation (Anisotropic Diffusion Equation) that annotate a few color scribbles provided by the user, with firmness in the original gradient information in image.

The approach suggests an semi- automatic method that minimizes the amount of human work and the results are of high efficiency from the previous methods.

الخلاصة

إضافة الألوان لجميع نقاط الصورة الأحادية اللون يعتبر من العمليات الصعبة الإنجاز والتي تتطلب تدخل المنفذ بشكل كبير وتكون مملة وبالعلة الكلفة في الوقت والجهد وغير مضمونة النتائج. الطرق الآلية السابقة لا توفر التدرج اللوني المطلوب والتجانس في الألوان المنفذة. يتناول البحث عمليات التلوين للصور الرقمية أحادية اللون بمساعدة الحاسب الآلي، حيث يؤثر المستخدم الصورة بقليل من الألوان على المناطق المراد تلوينها وبصورة غير نظامية وبالألوان المرغوب بها وحسب اختيار المنفذ، الألوان المؤشرة يتم انتشارها بصورة تلقائية لتنتج صورة ملونه بالكامل. التنفيذ يعتمد على هندسة وهيكلية الصورة المدخلة (أحادية اللون) وباعتماد على حل المعادلات التفاضلية الجزئية (الانتشار النقطي الشمولي) والتي يتم اعتمادها بنشر الألوان المؤشرة بواسطة المنفذ مع ثبات التدرج اللوني في الصورة.

البحث يقترح عملية للتلوين الشبه التلقائي التي تقلل كمية الجهد المطلوب من قبل المنفذ والنتائج تكون ذات كفاءة أعلى من الطرق السابقة.

Keywords

Image colorization, Partial Differential equations, Anisotropic Diffusion

1. Introduction

Colorization is a computer-assisted process of adding color to a monochrome image or movie. Wilson Markle in 1970 describes a computer-

assisted process invented for adding color to black and white movies or TV programs [1]. The term is now used generically to describe any technique for adding color to monochrome stills.

*Computer Science Dept. University of Technology.

**Computer Center. Babylon University.

Grayscale image colorization can find its applications in black and white photo editing, classic movies colorization, and scientific illustrations. Colorization can increase dramatically the visual appeal of grayscale images and perceptually enhance scientific illustrations [2].

Gonzalez and Wintz [1987] describe a simple approach for pseudo coloring grayscale images. Pseudo coloring, which is a common technique for adding color to grayscale images such as X-ray, scanning electron microcopy or other imaging modalities in which color information does not exist, can be used to enhance the detectability of detail within the image. This method does not appeal to common users, since it requires lots of interaction, experience and rendering time [3].

BlackMagic, commercial software for colorizing still images provides the user with useful brushes and color palettes [4].

Welsh *et al* [2002] describe a semi-automatic technique for colorizing a grayscale image by transferring color from a reference color image. They examine the luminance values in the neighborhood of each pixel in the target image and transfer the color from pixels with matching neighborhoods in the reference image. This technique works well on images where differently colored regions give rise to distinct luminance clusters, or possess distinct textures [5]. While this technique has produced some impressive results, note that the artistic control over the outcome is quite indirect.

Colorization in general is an active and challenging area of research with a lot of interest in the image editing community. The problem of colorizing a gray-scaled image involves assigning three-dimensional (RGB) pixel values to an image whose elements (pixels) are characterized only by one feature

(luminance).

Since different colors may carry the same luminance in spite of differences in hue and/or saturation, the problem of colorizing gray-scaled images has no inherently “correct” solution. Due to these ambiguities, human interaction usually plays a large role in the colorization process [6].

This paper presents semi-automatic image colorization. In this work the artist only needs to annotate the image with a few color scribbles, and the indicated colors are automatically diffused to produce a fully colorized image. The color is diffused by solving a partial differential equation that diffuses a few color scribbles provided by the user, with firmness in the original gradient information in image.

The remainder of the paper is organized as follows: In Section two, we discuss the color spaces. In section three, we discuss the mathematical background. Then, we describe our image-colorizing algorithm in Section four. Section five will show results of the algorithm and compare them with the original color image. Conclusion will be given in Section Six.

2.Color Spaces

A color space is a mathematical representation of a set of colors. The three most popular color models are RGB (used in computer graphics); YIQ, YUV, or YCbCr (used in video systems); and CMYK (used in color printing). All of the color spaces can be derived from the RGB information supplied by devices such as cameras and scanners.

2.1.RGB Color Space

The red, green, and blue (RGB) color space is widely used throughout computer graphics. Red, green, and blue are three primary additive colors

(individual components are added together to form a desired color) [7]. The RGB color space is the most prevalent choice for computer graphics because color displays use red, green, and blue to create the desired color. Therefore, the choice of the RGB color space simplifies the architecture and design of the system. Also, a system that is designed using the RGB color space can take advantage of a large number of existing software routines, since this color space has been around for a number of years.

The main disadvantage of RGB color space in applications involving natural images is highly correlated between its components. Also, processing an image in the RGB color space is usually not the most efficient method [8].

For these and other reasons, many video standards use luminance and two color difference signals. The most common are the YUV, YIQ, and YCbCr color spaces. Although all are related, there are some differences.

2.2.YUV color space

The YUV model is a bit different from the other colorimetric models. It is basically a linear transformation of RGB image data and is most widely used to encode color for use in television transmission. Y specified gray scale or luminance. The U and V components correspond to the chrominance (color information).

The basic equations to convert between RGB and YUV are [7]:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= -0.147R - 0.289G + 0.436B \\ &= 0.492 (B - Y) \\ V &= 0.615R - 0.515G - 0.100B \\ &= 0.877(R - Y) \end{aligned}$$

and The equations to convert between YUV and RGB are [7]:

$$\begin{aligned} R &= Y + 1.140V \\ G &= Y - 0.395U - 0.581V \end{aligned}$$

$$B = Y + 2.032U$$

For digital RGB values with a range of 0–255, Y has a range of 0–255, U a range of 0 to ± 112 , and V a range of 0 to ± 157 .

We will work on the YUV color space which has been chosen because it promptly provides the luminance value (channel Y) which is a crucial datum for our procedure. It also grants a more faithful modeling of human eye [8].

3.Mathematical Background

3.1.Gradient operators

First-order derivatives of a digital image are based on various approximations of the 2-D gradient. The gradient of an image $f(x, y)$ at location (x, y) is defined as the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

It is well Known from vector analysis that the gradient vector point in the direction of maximum rate of change of f at coordinates (x, y) .

An important quantity in edge detection is the magnitude of this vector, denoted ∇f , where

$$\nabla f = \text{mag}(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2} \dots (1)$$

This quantity gives the maximum rate of increase of $f(x, y)$ per unit distance in the direction of ∇f . It is a common (although not strictly correct) practice to refer to ∇f also as the *gradient*.

Computation of the gradient of an image is based on obtaining the partial derivatives $\partial f / \partial x$ and $\partial f / \partial y$ at every pixel location [9].

3.2.Anisotropic diffusion

Perona-Malik formulation

Diffusion algorithms remove noise

from an image by modifying the image via a partial differential equation (PDE). Consider applying the isotropic diffusion equation (the heat equation) given by $\partial I(x,y,t)/\partial t = \text{div}(\nabla I)$, using the original (degraded/noisy) image $I(x,y,0)$ as the initial condition, where $I(x,y,0): \mathbb{R}^2 \rightarrow \mathbb{R}^+$ is an image in the continuous domain, (x,y) specifies spatial position, t is an artificial time parameter, and where ∇I is the image gradient. Modifying the image according to this isotropic diffusion equation is equivalent to filtering the image with a Gaussian filter [10].

Perona *et al* [11] replaced the classical isotropic diffusion equation with

$$\frac{\partial I(x,y,t)}{\partial t} = \text{div} [g(\|\nabla I\|) \cdot \nabla I]$$

Where $\|\nabla I\|$ is the gradient magnitude, and $g(\|\nabla I\|)$ is an “edge-stopping” function.

This function is chosen to satisfy $g(x) \rightarrow 0$ when $x \rightarrow \infty$ so that diffusion is “stopped” across edges.

3.3. Perona-Malik Discrete formulation

Perona *et al* [1990] discretized their anisotropic diffusion equation as follows [Black *et al* 1998]:

$$I_s^{t+1} = I_s^t + \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} g(\|\nabla_{s,p}\|) \cdot \nabla_{s,p} \dots \dots \dots (2)$$

I_s^t Where is a discretely sampled image, s denotes the pixel position in a discrete, two-dimensional (2-D) grid, and t now denotes discrete time steps (iterations). The constant $\lambda \in \mathbb{R}^+$ is a scalar that determines the rate of diffusion, η_s represents the spatial neighborhood of pixel s , and $|\eta_s|$ is the number of neighbors (usually 4, except at the image boundaries). Perona *et al* [1990] linearly approximated the image gradient

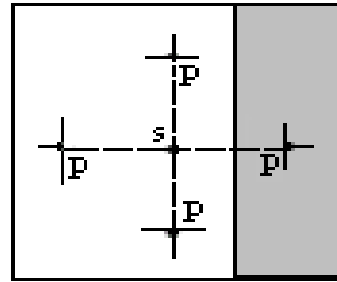
(magnitude) in a particular direction as

$$\nabla_{I_{s,p}} = I_p - I_s, p \in \eta_s \dots \dots (3)$$

Consider, for example, the image region illustrated in Figure 1. The intensity values of the neighbors of pixel s are drawn from two different populations, and in estimating the “true” intensity value at s we want to include only those neighbors that belong to the same population. In particular, the pixel labeled p is on the wrong side of the boundary so I_p will skew the estimate of I_s significantly [10].

In our algorithm, the pixels skew the estimate of I_s is equal to Zero.

In this paper, we introduce equation (4) which similar to equation (2) to a great extent with some modifications for colorization an image.



4. Modified color Diffusion using gradient and boundary conditions

Let $Y(x,y): \Omega \rightarrow \mathbb{R}^+$ be the given monochromatic image defined on the region Ω . The goal is to compute $U(x,y): \Omega \rightarrow \mathbb{R}^+$ and $V(x,y): \Omega \rightarrow \mathbb{R}^+$. We assume that Colors are given in a region Ω_c in such that $|\Omega_c| \ll |\Omega|$. This information is provided by the user via color strokes in editing type of applications. The goal is from the knowledge of Y in Ω and U, V in Ω_c to diffusion the color information (U, V) into the rest of Ω . Following the description updating equation (2) to compute U (and similarly V):

$$U_s^{j+1} = \frac{\lambda}{|\eta_k|} \sum_{p \in \eta_k} g(\nabla I_{s,p}) \cdot \nabla U_{s,p}^j \dots\dots\dots (4)$$

Procedure of color Diffusion

The procedure of color Diffusion is described as follows:

Step1.Read threshold to compute gradient operator.

Step2.Input grayscale image.

Step3.Scribbles the input image with a few colors, $\Omega_c < \Omega$.

Step4.convert the input image to YUV color space.

Step5.Copmute gradient operator for $Y(x,y)$.

Step6.Compute $g(\nabla I_{s,p})$ by applies equation (3) on the results from computing gradient operator for $Y(x,y)$, in which $g(\nabla I_{s,p})$ equals zero or one.

Step7.computes $|\eta_s|$ for each pixel that represents the number of neighbors (usually 4, except at the image boundaries)

Step8. Modified colors diffusion By iteration of equation (4) after computing the above steps until arriving at the stability of this equation. Where λ equals to one and t refers to the number of iteration.

Fig.1. Local neighborhood of pixels at a boundary (intensity discontinuity).

5. 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.

Figure 2 shows some examples of colored still images representing comparison between our method and "Transferring color to grayscale images" by [Welsh et al \[2002\]](#) method. This method describes a semi-automatic technique for colorizing a grayscale image by transferring color from a reference color image. This technique works well on images where differently colored regions give rise to distinct luminance clusters, or possess distinct textures.

While this technique has produced some impressive results, note that the artistic control over the outcome is quite indirect.

In figure 3 our method is compared with previous methods. Figure 3(a) compares our method with "Fast colorization of gray image" by [Di Blasi et al \[2003\]](#) method in "Fast colorization of gray image" method colorizing images by transferring colors between a source image (colored) to a destination image (gray-scaled). This method works only in the case of homogeneous images and this method may fail to automatically delineate all the correct boundaries, such as the intricate boundary between the hair and the forehead, or the low contrast boundary between the lips and the face. Consequently, the colorization achieved with this method is not satisfactory.

In figure 3(b) compares our method with "Grayscale image matting and colorization" by [Chen et al \[2004\]](#) in "Grayscale image matting and colorization" method. First, the source grayscale image is split into different objects using the grayscale image matting algorithm [12]. Then, the objects are colorized using color-transferring technique [5]. Finally, the Colorized objects are composite using alpha

blending to reach the ultimate colorization.

The above method is user intervention, a tedious, time-consuming, and expensive. In our method notice we provide an automatic method to help minimize the amount of human work required for this task and the results are of high quality in addition to an easily usage.

Figure 4 shows how our method can be coloring one or more object in image, note the bird in first row: the car in the middle row and the rose in the last row are colored while the background remains grayscale.

Our method does not color the other object in the image, since colors are not diffused across intensity boundaries.

6.Conclusions

Despite considerable progress in image processing since 1970, colorization remains a manually intensive and time-consuming process. In this paper we have formulated a new, general, fast, reduce tedious manual work and user-friendly approach to the problem of colorizing grayscale images. This paper suggests a method that helps graphic artists colorize an image with less manual effort. Our technique empowers the user to first select desired color and then scribble this color choice on the image or object in image that we need colorized.

In our framework, the artist does not need to explicitly delineate the exact boundaries of objects. Instead, the artist colors a small number of pixels in selected frames and the algorithm diffuses these colors in a manner that respects intensity boundaries. We have shown that excellent colorizations can be obtained with a surprisingly small amount of user effort.

7.References

- [1] **Burns, G.**, "Colorization. Museum of Broadcast Communications", Encyclopedia of Television. Available on-line: <http://www.museum.tv/archives/etv/index.html>.
- [2] **Chen, T., Wang, Y., Schillings, V. and Meinel, C.**, "Grayscale image matting and colorization", in Proceeding of ACCV 2004, Jan 27-30, 2004, Jeju Island, Korea, pp. 1164-1169.
- [3] **Gonzalez, R. C. and Wintz, P.**, "Digital Image Processing", Addison-Wesley Publishin, 1987.
- [4] **NeuralTek**, 2003. BlackMagic photoColorization Software, version 2.8. Available on-line: <http://www.timebrush.com/blackmagic>.
- [5] **Welsh, T., Ashikhmin, M., AND Mueller, K.**, "Transferring Color to grayscale images", in Proceedings of ACM SIGGRAPH, 2002, pp. 277.280
- [6] **Di Blasi G. and Reforgiato Recupero D.**, "Fast Colorization of Gray Images", University of Catania, Catania, Italy, 2003
- [7] **Diepold, K., And Oelbaum, T.**, "Digital Video", Technische Universität München, 2004. Available on-line: <http://www.ldv.ei.tum.de/page19>
- [8] **Sangwine, S.J. and Horne, R.E.N.**, "The Colour Image Processing Handbook", Published by Chapman & Hall, 1998.

[9] **Gonzalez, R.C. and Woods, R.E.**, “Digital Image Processing ”, Published by Prentice Hall, second edition, 2002.

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

[11] **Perona, P. and Malik, J.**, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. Pattern Anal. Machine Intell.* Vol.12, no. 7 pp. 629–639, July 1990.

[12] **Chuang, Y. Y., Curless, B., Salesin, D. and Szeliski, R.**, "A bayesian approach to digital matting", in proceedings of Computer Vision and Pattern Recognition, 2001, pp. 264-271.

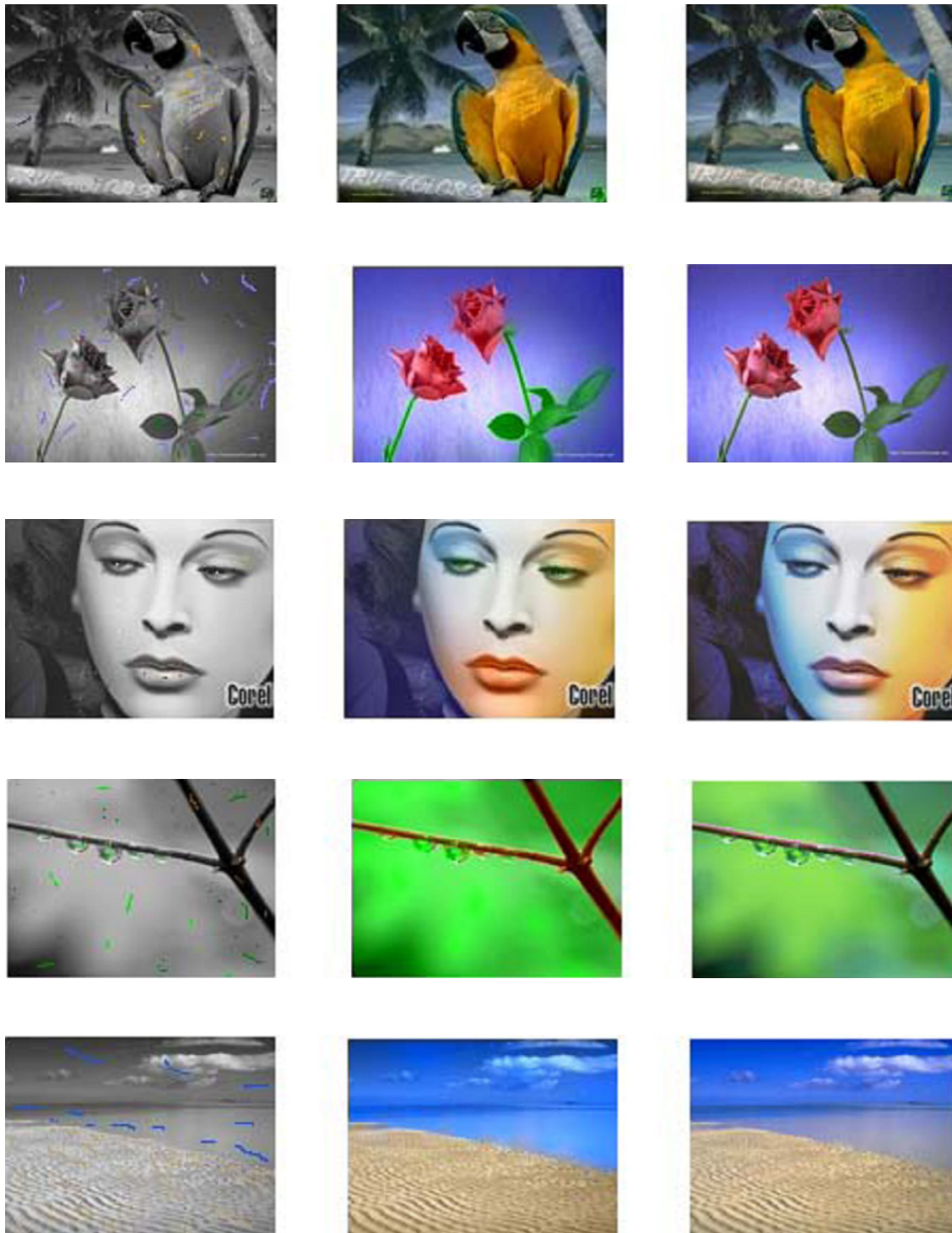


Fig 1: Still Image Colorization . First column is the monochromatic image with color strokes with colors from the original data , second column is the colorized image automatically obtained from our technique , and followed (last column) is the original image.



Fig 2 : Some examples of colored still images representing compares between our algorithm technique and “ Transferring color to Grayscale images “ algorithms by Welsh et al 2002 . First column is a grayscale image marked with some color scribbles by the user , second column is colorized image obtained from our algorithm technique , and followed (last column) is the colorizes image obtained from “ Transferring color to Grayscale images “ algorithm .



Fig 3 (a) : Comparison between our algorithm technique and “ Fast colorization of gray images” by Di Blasi et al 2003 . A grayscale image marked with some color scribbles by the user (left) , our algorithm produces a colorized image (middle), the colorized image obtained from “ Fast colorization of gray images” algorithm is shown on the right.

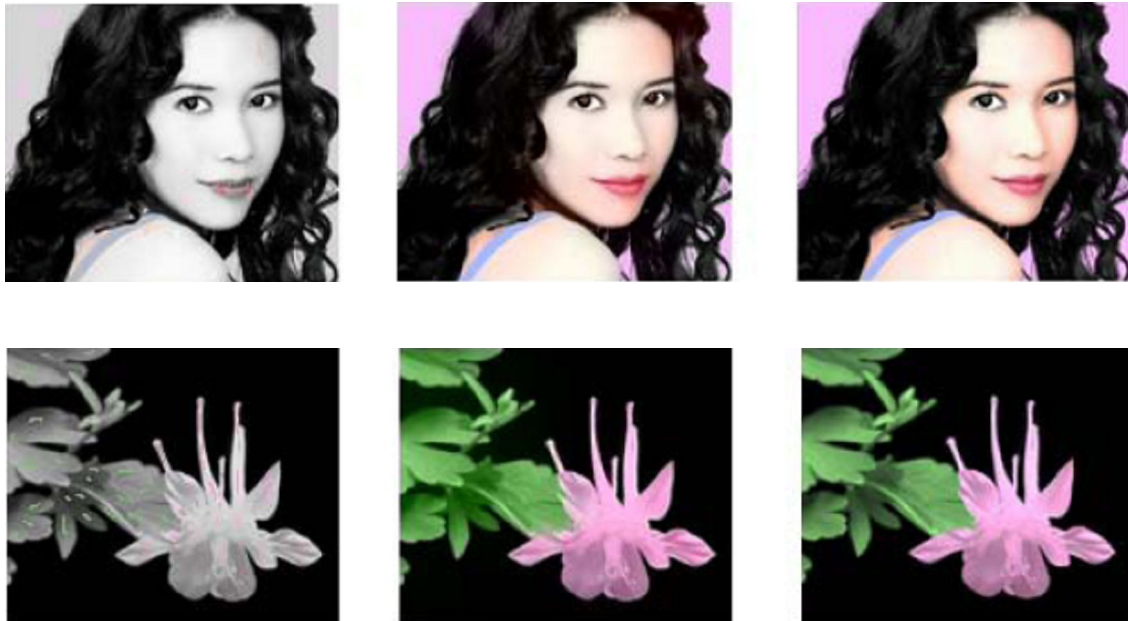


Fig 3 (b) : Comparison between our algorithm technique and “ Grayscale image matting and colorization” by Chent et al 2004 . First column is a grayscale image marked with some color scribbles by the user , second column is a colorized image obtained from our algorithm technique , and followed (last column) is the colorized image obtained from “ Grayscale image matting and colorization” .

Fig 3: Some examples of colored still images representing comparisons between our algorithm technique and the previous work .

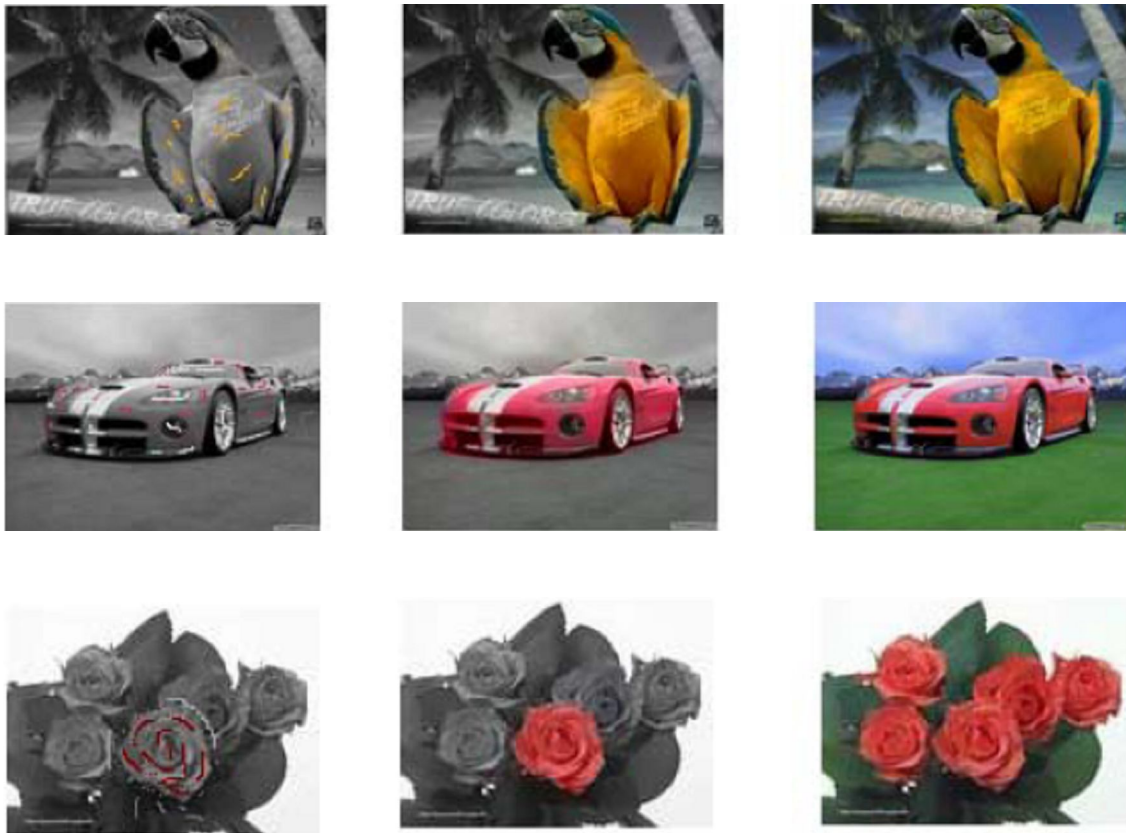


Fig 4: coloring of one or more object in still image . First column is the input image is the object marked with some color scribbles by the user , second column is the resulting image , and followed (last column) is the original image .