

# Gray-Scale Image Colorization Using Various Affinity Functions

Imron Rosyadi

mas.imron.rosyadi@gmail.com

*Electrical Engineering Study Program, Engineering Department, Faculty of Science and Engineering Jenderal Soedirman University  
Jl. Mayjend. Soengkono Km. 5 Blater Purbalingga, 53371*

**Abstract**— In this paper, we have proposed, implemented, and compared some affinity functions for an image colorization algorithm. The colorization quality of the proposed affinity functions was just slightly behind the original functions, while one of the proposed functions performed faster than the original affinity function. We also implemented the colorization algorithm for coloring an Indonesian historical image.

*Keyword*— colorization, affinity function, optimization.

## INTRODUCTION

Colorization is image or video processing mechanism that converting from grayscale to color image/video. It is used in many areas, for example in entertainment world when it is needed to give color to old movie or to cartoon sketch. Colorization is also used in sensor measurement for medicine or biological application. Color output will give user more understanding than its grayscale image output.

Colorization processes assign three-dimensional pixel values to an image which originally has one-dimension pixel values. Since different colors can possess same luminance or gray scale value but may vary in hue or saturation value, the problem of colorizing gray-scale images is actually ill-posed and has no "correct" solution. The other difficulties in colorization process are its expensive and time-consuming process. Hence, good colorization algorithm should give good color result and be fast enough to be executed.

Many methods have been proposed for colorization grayscale images. Some of them (Levin et al. (2004), Yinggo Qu et al. (2006), Yatziv and Sapiro (2006)) using optimization process that need weighting function called affinity function. These affinity functions contribute in colorization result quality and processing time.

## RESEARCH METHOD

In this paper, we proposed, tested, and compared some affinity functions based on its quality and computation time. For those purposes, we used algorithm proposed by Levin et al. (2006). Image output quality was measured using PSNR (Peak Signal to Noise Ratio). We also measure the colorization computation time using each affinity function. Then, we choose the best affinity function based on the PSNR and processing time and use it to color some grayscale historical images.

Historical images are images that contain any historical information and are related with other historical sources like historical documents, historical artifacts, and historical books. We choose historical images because they have an important role in describing and explaining history of human beings. Unfortunately, their grayscale seems to 'hide' any important historical information. Colorization of grayscale historical images can expand our historical perspective. In this report, prior color information in the colorization process of historical images is investigated from other historical sources or determined through common sense (green for grass, blue for the ocean, etc.).

Levin et al. (2006) algorithm works in YUV color space. Y is the monochromatic luminance channel, which will be referred as intensity, while U and V are the chrominance channels to encode the color. Using this color space, the problem of colorization is become to estimate the U and V component value of the input gray-scale image from the only known intensity value Y.

Because colorization is an ill-posed problem, many assumptions are introduced in the previous works to overcome the ambiguity in colorization process. The algorithm premises nearby pixels in space that have similar grey levels also have similar colors. It is assumed that there is little difference between the color value of a given sampling point and the weighted summary of its neighbor points (Levin et al. (2006)). Based on this assumption, the colorization problem is formulated to minimizing the following objective function,

$$J(U) = \sum_r \left( U(r) - \sum_{s \in N(r)} w_{rs} U(s) \right) \dots \quad (1)$$

where  $U(r)$  is color at pixel  $r$ ,  $w_{rs}$  is a weighting function that sums to one (also known as affinity function), large when  $Y(r)$  is similar to  $Y(s)$ , and small if the two intensities are different. The affinity function intuitively means that when the intensity is constant the color should be constant and when the intensity is an edge the color should be an edge (Levin et al. (2006)).

The colorization processes can be described as follows. If we have a set of location  $r_i$  where the colors are specified by the user. Then with  $U(r_i)=u_i$  and  $V(r_i)=v_i$ , we minimize  $J(U)$  and  $J(V)$  subject to these constraints. Since the cost function is quadratic and the constraints are linear, the optimization problem yields a large-sparse system of linear equations, which may be solved using many standard methods.

Affinity function in equation (1) describes how strongly adjacent pixels in image "stick together". In applications such as segmentation, edge detection, and noise reduction, affinity function is measurement between pixels of same image. In motion segmentation or tracking, the affinity function related with pixels from neighboring frames (Omer and Werman (2006)). The affinity functions were referred as weighting function and contribute both in colorization quality and computation time.

In practice, most applications calculate pixels affinity as a simple function of the Euclidean distance between the pixels' features as,

$$w(d) \propto e^{-\frac{d^2}{\sigma^2}} \quad \dots \dots \dots (2)$$

where  $d$  is the Euclidean distance in some feature space and  $\sigma$  is a normalization factor (Omer and Werman (2006)).

The Euclidean distance  $d$  can be represented as,

$$d = Y(r) - Y(s) \quad \dots \dots \dots (3)$$

where  $Y(r)$  is the intensity of particular pixel, and  $Y(s)$  is the intensity in a neighborhood pixel of  $r$ .

Levin et al. (2004) used two weighting functions as below,

$$w_{rs} \propto e^{-(Y(r)-Y(s))^2/2\sigma_r^2} \quad \dots \dots \dots (4)$$

and

$$w_{rs} \propto 1 + \frac{1}{\sigma_r^2} (Y(r) - \mu_r)(Y(s) - \mu_r) \quad \dots \dots \dots (5)$$

where  $\mu_r$  and  $\sigma_r^2$  are the mean and variance of the intensities of the window around  $r$ .

In this paper, we proposed and applied two affinity functions to enhance the colorization quality and the computation time and compare them to the original affinity functions proposed by Levin et al. Our proposed functions are,

$$w_{rs} \propto e^{-(Y(r)-Y(s))^2/2} \quad \dots \dots \dots (6)$$

and,

$$w_{rs} \propto \begin{cases} \frac{1}{|Y(r)-Y(s)|+\epsilon} & |Y(r)-Y(s)| < T_w \\ 0 & \text{otherwise} \end{cases} \quad \dots \dots \dots (7)$$

where  $T_w$  is set as the standard deviation of intensity values of sampling point, and  $\epsilon$  is a very small number that close to zero. Here we used  $\epsilon$  equal to 0.0001. In this paper we refer equation (4), equation (5), equation (6), and equation (7) as affinity function #1, affinity function #2, affinity function #3, and affinity function #4, respectively.

To compare the colored images, we use PSNR which is defined as follows:

$$(PSNR) = 10 \log_{10} \frac{3M_1 M_2 (2^{L_I}-1)^2}{\sum_{i,j} |I-I^*|^2} \quad \dots \dots \dots (8)$$

where  $L_I$  is bit quantization,  $M_1$  and  $M_2$  are size of image,  $I$  and  $I^*$  are two compared images. In this report we used one bit quantization.

## EXPERIMENT RESULTS AND DISCUSSION

### A. Affinity Function Comparison

In order to evaluate the proposed method, colorization experiments using some standard test images were performed. We use 8 images standard test images. Some of them have special characteristics, for examples, *Milkdrop* (256x256 pixels) which has little change of colors among adjacent pixels, *Pepper* (256x256 pixels) which has various colors but with clear transition in edge pixels, and *Baboon* (512x512 pixels) which includes various colors with blur transition in edge pixels. The original images and its grayscale images are shown in Figure 1 and Figure 2, respectively.



Figure 1. Original images used in experiments: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)

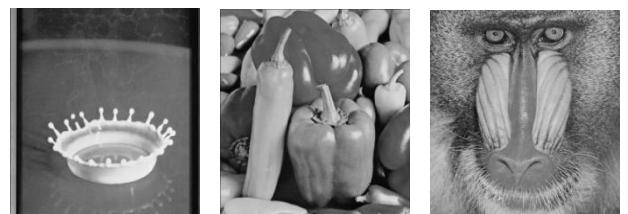


Figure 2. Grayscale of the test images: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)

After marked or scribbled as in Figure 3, for four affinity functions the results are shown in Figure 4, Figure 5, Figure 6, and Figure 7, respectively. From the figures we can see that good results are generated by algorithm even from relatively small marked pixels. The results should be better if careful color choosing and enough strokes are performed.



Figure 3. Mark (scribbles) on the grayscale test images: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)



Figure 4. Colored image by affinity function #1: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)



Figure 5. Colored image by affinity function #2: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)



Figure 6. Colored image by affinity function #3: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)



Figure 7. Colored image by affinity function #4: *Milkdrop* (left), *Pepper* (middle), and *Baboon* (right)

TABLE 1. PSNR OF COLORED TEST IMAGES USING FOUR AFFINITY FUNCTIONS

	Affinity Function #1	Affinity Function #2	Affinity Function #3	Affinity Function #4
Milkdrop	16.25 dB	16.14 dB	16.14 dB	16.00 dB
Pepper	24.97 dB	24.35 dB	24.32 dB	24.15 dB
Baboon	15.63 dB	15.25 dB	15.26 dB	15.83 dB
Hato	18.42 dB	18.85 dB	18.86 dB	18.13 dB
House	15.82 dB	15.80 dB	15.80 dB	15.71 dB
Lena	27.37 dB	24.02 dB	23.98 dB	20.41 dB
Parrot	37.06 dB	27.06 dB	27.04 dB	20.65 dB
Boat	15.86 dB	15.75 dB	15.75 dB	16.26 dB
average	21.42 dB	19.65 dB	19.64 dB	18.39 dB

TABLE 2. COMPUTATION TIME AVERAGE OF TEST IMAGES COLORIZATIONS USING FOUR AFFINITY FUNCTIONS

	Affinity Function #1	Affinity Function #2	Affinity Function #3	Affinity Function #4
Milkdrop	8.9072	10.6580	4.3697	11.9211
Pepper	8.6578	11.0833	4.1519	9.5346
Baboon	10.0553	12.8261	4.8428	15.1170
Hato	11.8919	14.5816	5.5918	14.4583
House	8.1172	10.0097	3.7985	10.7974
Lena	9.1569	11.2518	5.9273	18.2102
Parrot	14.7274	17.5423	7.6503	30.0211
Boat	15.3051	19.4267	8.1547	57.7761
average	10.8524	13.4224	5.5609	20.9795

To measure algorithm output quality, we used two parameters: PSNR result and computation time. PSNR and computation time comparison between many colored test images with their affinity functions are shown in Table 1 and Table 2. PSNR between affinity functions is only slightly different. The original affinity functions work slightly better than our proposed functions. However, the proposed affinity function #3 performed two fold faster than the original affinity function.

#### B. Effect of Prior Color Information

For the test images, instead using scribble to give prior color information to the grayscale image, we also gave prior color information using randomly generated pixels. The color image version of a grayscale image is used as source of the prior color information.

Figure 8 shows colored images by giving prior information from 0.01% until 10% of pixels on affinity function #3. We can see that the PSNR of images will be increase if initial correct rate is increase. Random colored marking pixels are shown in Figure 9. The colored images after randomly marked are shown in Figure 10. Colored images quality of this random color choosing seems better than scribble color choosing because of its uniform distribution characteristic and the color chosen are exactly same with the original image color. That's why 0.1% correct color pixels in a gray image will have good result with high PSNR.

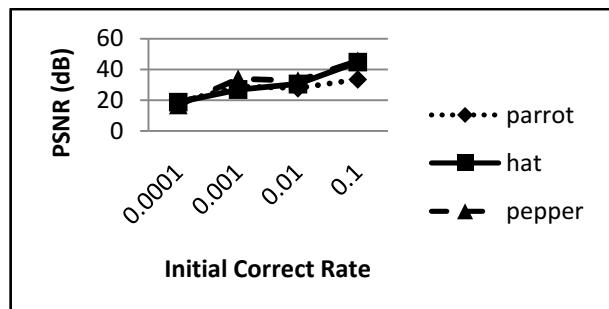


Figure 8. Random marking with 10% initial correct pixels



Figure 9. Random marking with 10% initial correct pixels



Figure 10. Colored image with 0.01% (up-left), 0.1% (up-right), 1% (bottom-left) and 10% (bottom-right) initial correct pixels, using affinity function #1

Figure 11. Indonesian grayscale historical images: *Suharto02*Figure 12. Indonesian color historical image: *Army*

### C. Historical Image Colorization

In historical images colorization, we sampled three grayscale images related with Indonesian history, as shown in Figure 11. The image was taken around October 1965 showing Maj. Gen. Suharto in some events after unsuccessful communist party *coup d'état*.

For scribbling the first image, we used other historical images in Figure 12 which shown Indonesian army entered Timor Timur at year 1975. Based on image in Figure 12, we scribbled the image in Figure 11. We used assumption that no change in military uniform from October 1965 until 1975, the military emblems was always in gold color, and Indonesian has brown skin. Figure 13 shows the scribbled image.



Figure 13. Scribbled grayscale historical image



Figure 14. Colored historical images using affinity function #3

Figure 14 show the historical image after colorization using affinity function #3. By these color image, we can get ‘warm’ and ‘natural’ images than their original grayscale image.

#### CONCLUSION AND FUTURE WORK

In this paper, we have proposed, implemented, and compared colorization based optimization algorithm using various affinity functions. Our method successfully colored test images. We compared output quality using PSNR and computation time. PSNR between original affinity functions and our proposed functions are slightly

different, while our affinity function #3 is executed two times faster than the original affinity function.

We also evaluated relation between percentages of correct color pixels in gray scale image with its colored result using proposed affinity function #3. More prior color information will give better result. To have output with PSNR between 25 and 30 dB, we just need 0.1% exact color pixels. Implementation of the algorithm for coloring historical images also has been performed with satisfactory result.

In future work we can develop more sophisticated affinity function, tuning the optimization technique faster, and exploration other propagation schemes that treat hue and saturation channels differently.

#### REFERENCE

- Levin, A., Lischinski, D., and Weiss, Y., “Colorization using Optimization”, in *Proc. ACM SIGGRAPH Conf.*, 2004, pp. 689-694.  
Yinggo Qu, Tien-Tsin Wong, Phen-Ann Heng, “Manga Colorization”, in *Proc. SIGGRAPH*, Computer Graphics Proceedings, 2006, pp 1214-1220.  
Yatziv, L., and Sapiro, G., “Fast Image and Video Colorization using Chrominance Blending”, in *IEEE Transactions on Image Processing*, vol. 15, no. 5, May 2006, pp. 1120-1129.  
Omer, I., and Werman, M., “The Bottleneck Geodesic: Computing Pixel Affinity”, in *Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, June 2006, pp. 1901-1907.