Color by Linear Neighborhood Embedding

Guoping Qiu and Jian Guan

School of Computer Science, The University of Nottingham

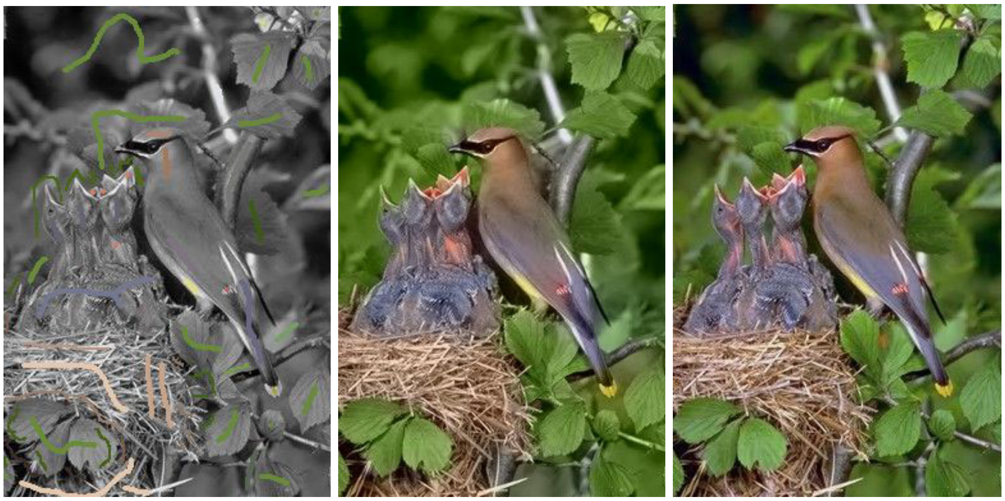

Fig. 1. A monochrome image is scribbled with some initial colors (left), these colors are then propagated to the whole image using our color by linear neighborhood embedding method (middle). For reference, the original image is shown on the right.

Abstract - **This paper presents a new semiautomatic method for adding colors to monochrome images. The new method, termed color by linear neighborhood embedding (CLNE) first computes the geometry of local image patches in the monochrome image and then transmits the geometry to the chrominance. Following computational methods of the locally linear embedding (LLE) algorithm, we compute the geometry of the monochrome image by solving a constrained least squares problem and embed the computed geometry in the chrominance images by solving a linearly constrained quadratic optimization problem, both in closed forms. We also show that the computed geometry of different color channels are highly correlated thus empirically supporting methods that embed the geometry of the monochrome in the chrominance for colorization.**

I. INTRODUCTION

Adding color to monochrome image has a long history, and recently there has been renewed effort to develop tools and algorithms for automating this laborious process [1, 2]. The work of [1], which also partially inspired this paper, used a simple premise that neighboring pixels that have similar intensities should have similar colors. Closely related to [1], [2] exploited the fact that the (scalar) luminance channel faithfully represents the geometry of the whole (vectorial) image and formalized the premise of [1] by inpainting using the gradients from the provided monochrome image thereby transmitting the geometry among the color channels.

Another recent work which has also partially motivated this paper is the locally linear embedding (LLE) algorithm [3, 4]. LLE characterizes the local geometry of a high dimensional manifold by linear coefficients that reconstruct each data point from its neighbors. The local geometry in a high dimensional space is then transmitted to a low dimensional space by (global) embedding. The LLE algorithm is an elegant method due to the fact that both the local geometry in the high dimensional space and the embedding in the lower dimensional space can be computed in closed forms using standard methods for solving constrained least squares problems and sparse eigenvector problems.

In this paper, we also exploit the fact that luminance channel faithfully represents the geometry of the whole image, but we use a computational approach similar to LLE to compute the characteristics of the local geometry of the monochrome image provided. We then transmit the luminance's geometry to the chrominance using a method similar to the embedding process of the LLE algorithm, which uses the colors scribbled on by the user as the linear constraints to solve a quadratic optimization problem.

Our colorization algorithm, which we term color by linear neighborhood embedding (CLNE) and LLE may be described by following analogies. The LLE computes the local geometry of manifold patches in a high dimensional space and CLNE computes the geometry of local spatial neighborhood pixels in the provided monochrome image. The LLE embeds the geometry of the high dimensional space in a low dimensional space and CLNE embeds the geometry of the monochrome image in the chrominance images. The computational methods of CLNE follow closely those used by LLE but under different constraints.

II. LOCAL PIXEL GEOMETRY

Any two-dimensional image can be described as the sum of a set of points. Due to optical and other system aberrations, neighboring points (pixels) are correlated. A general imaging model can be expressed in terms of the light field that reaches the camera's imaging plane and the system's point spread function. Assume $\boldsymbol{x} = (x, y) \in \boldsymbol{R}^2$and $\xi = (\xi, \eta) \in \boldsymbol{R}^2$ are pixel coordinates in the imaging and object planes respectively, then an image, $\boldsymbol{I}(\boldsymbol{x})$ is formed as

$$I(\boldsymbol{x}) = \int h(\boldsymbol{x} - \xi) E(\xi) d\xi \tag{1}$$

where $h(\ )$is the point spread function (***PSF***) of the imaging system, $E(\ )$ is the light field that would have reached the imaging plane. For color image, different color channels will have different sensor responses to illumination wavelengths and there will be additions integrations over illumination wavelengths and over time. A true imaging model would have to consider noise as well. We omit these details in the imaging model since they should not affect our discussion in this paper.

From the imaging formation model, it is clear that neighboring pixels would have been correlated by the ***PSF***. $h(\boldsymbol{x})$ will normally have a limited support, i.e., it will only have nonzero values in the vicinity of $\boldsymbol{x}$ and zero everywhere else, also $h(\boldsymbol{x})$ is normalized, i.e., integration of $h(\boldsymbol{x})$ over the entire space will result in unity.

A. Within Channel Local Pixel Geometry

To characterize the neighborhood pixels correlation within the same color channel, we can use a linear combination of a pixel's neighbors to represent the pixel itself:

$$G(\boldsymbol{x}) = \sum_{\boldsymbol{m} \in N_{\boldsymbol{x}}} w_{\boldsymbol{x}}(\boldsymbol{m}) I(\boldsymbol{x} - \boldsymbol{m}) \tag{2}$$

where $\boldsymbol{m} = (m, n) \in \boldsymbol{R}^2$ denotes a two dimensional grid, $N_{\boldsymbol{x}}$ is a local pixel neighborhood centered around $\boldsymbol{x}$. For a given image, $\boldsymbol{I}(\boldsymbol{x})$, we can compute the pixel neighborhood relation coefficients, $w_{\boldsymbol{x}}(\boldsymbol{m})$'s, by optimizing following cost function

$$\boldsymbol{\varepsilon}(\boldsymbol{W}) = \sum_{\boldsymbol{x}} \left\| I(\boldsymbol{x}) - \sum_{\boldsymbol{m} \in N_{\boldsymbol{x}}} w_{\boldsymbol{x}}(\boldsymbol{m}) I(\boldsymbol{x} - \boldsymbol{m}) \right\|^2 \tag{3}$$

Optimizing (3) is a constrained least squares problem, i.e., minimizing $\boldsymbol{\varepsilon}(\boldsymbol{W})$ subject to two constraints. The first constraint is that $w_{\boldsymbol{x}}(\boldsymbol{m}) = 0$ for $\boldsymbol{m} \notin N_{\boldsymbol{x}}$, and the second is that all neighborhood relation coefficients for each pixel, $w_{\boldsymbol{x}}(\boldsymbol{m})$'s, sum to 1. $w_{\boldsymbol{x}}(\boldsymbol{m})$'s characterize the geometry of a local neighborhood of pixels.

We recognize that (3) is very similar to the first step in the locally linear embedding (LLE) algorithm developed for high dimensionality reduction [3, 4]. The local neighborhood relation coefficients that best reconstruct each pixel can be computed in closed form, and we follow the method suggested in [4] by solving a linear system of equations. Since in our case, finding the coefficients does not have a unique solution, we again follow the method of [4] to recondition the problem before solving the system of linear equations. For algorithm details, readers are referred to the excellent tutorial article [4].

B. Between Channel Local Pixel Geometry

For different color channels, the ***PSF*** can be roughly assumed to be the same. It is therefore reasonable to expect that the local pixel relation coefficients, $w_{\boldsymbol{x}}(\boldsymbol{m})$'s, computed in one color channel will be similar to those computed from another color channel. To test the validity of this assumption, we have used many natural color photographs and computed the local pixel relation coefficients in separate color channels. We represented the color images in the ***YUV*** color space. For each image, we then computed the local relation coefficients on each channel, i.e., by solving following constrained least squares problems

$$\boldsymbol{\varepsilon}(Y\boldsymbol{W}) = \sum_{\boldsymbol{x}} \left\| Y(\boldsymbol{x}) - \sum_{\boldsymbol{m} \in N_{\boldsymbol{x}}} Yw_{\boldsymbol{x}}(\boldsymbol{m}) Y(\boldsymbol{x} - \boldsymbol{m}) \right\|^2$$

$$\boldsymbol{\varepsilon}(U\boldsymbol{W}) = \sum_{\boldsymbol{x}} \left\| U(\boldsymbol{x}) - \sum_{\boldsymbol{m} \in N_{\boldsymbol{x}}} Uw_{\boldsymbol{x}}(\boldsymbol{m}) U(\boldsymbol{x} - \boldsymbol{m}) \right\|^2$$

$$\boldsymbol{\varepsilon}(V\boldsymbol{W}) = \sum_{\boldsymbol{x}} \left\| V(\boldsymbol{x}) - \sum_{\boldsymbol{m} \in N_{\boldsymbol{x}}} Vw_{\boldsymbol{x}}(\boldsymbol{m}) V(\boldsymbol{x} - \boldsymbol{m}) \right\|^2$$

We then computed the normalized cross correlation between the local relation coefficients of different channels as

$$\rho_{YU} = \frac{1}{|\boldsymbol{x}|} \sum_{\boldsymbol{x}} \frac{Yw_x \bullet Uw_x}{\|Yw_x\| \|Uw_x\|}, \rho_{YV} = \frac{1}{|\boldsymbol{x}|} \sum_{\boldsymbol{x}} \frac{Yw_x \bullet Vw_x}{\|Yw_x\| \|Vw_x\|}, \rho_{VU} = \frac{1}{|\boldsymbol{x}|} \sum_{\boldsymbol{x}} \frac{Vw_x \bullet Uw_x}{\|Vw_x\| \|Uw_x\|}$$

where $|\boldsymbol{x}|$ represents the total number of pixels in the image. **Table 1** shows the values of these correlation measures for eight testing images

Table 1, Cross correlations of local pixel relation coefficients across color channels

Images	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
ρ_{YU}	**0.83**	**0.82**	**0.80**	**0.89**	**0.79**	**0.81**	**0.81**	**0.85**
ρ_{YV}	**0.81**	**0.83**	**0.81**	**0.90**	**0.81**	**0.78**	**0.77**	**0.85**
ρ_{VU}	**0.90**	**0.87**	**1.00**	**0.90**	**0.89**	**0.87**	**0.87**	**0.91**

From Table 1, it is seen that the local embedding coefficients across color channels are highly correlated with an averaged cross correlation coefficient above **0.85**, which indicates that the relation of a pixels and its neighbors in different color channels follow a very similar pattern. This is also in agreement with other authors work, see e.g. [2] and reference therein, which states that the luminance faithfully represents the geometry of the whole image.

Based on this highly correlated property of the cross channel local pixel relation coefficients, we develop a colorization method that embeds the local pixel relation coefficients computed from the luminance channel to the chrominance channels, in a way similar in spirit to the locally linear embedding algorithm for embedding the geometry of the high dimensional manifolds in lower dimensional ones [3, 4].

III. COLOR BY NEIGHBORHOOD EMBEDDING

For a given monochrome image, ***Y***, to be colorized, we use it directly as the luminance channel (***Y***) in a ***YUV*** color space to construct a color version of the image. We first compute the local pixel neighborhood relations of ***Y*** by solving a constrained least squares problem similar to the first step of the LLE algorithm. We then compute the ***UV*** signals by assuming that the ***UV*** channels having the same local pixel neighborhood relations as the ***Y*** channel. This is achieved by solving a linearly constrained quadratic optimization problem similar to the embedding step of LLE with user scribbled colors as constraints. Our color by linear neighborhood embedding (CLNE) algorithm consists of following steps.

Step 1: Computing luminance geometry

For the given monochrome image $\boldsymbol{Y}(\boldsymbol{x})$, we solve the constrained least squares problem (4) to obtain the local pixel relation coefficients, $w_{\boldsymbol{x}}(\boldsymbol{m})$'s. The constraints and the closed form solution to this problem were discussed in the previous section.

$$\boldsymbol{\varepsilon}(\boldsymbol{W}) = \sum_{\boldsymbol{x}} \left\| Y(\boldsymbol{x}) - \sum_{\boldsymbol{m}\in N_x} w_x(\boldsymbol{m}) Y(\boldsymbol{x} - \boldsymbol{m}) \right\|^2 \qquad (4)$$

Step 2: Embedding the colors

Once the local pixel relation coefficients, $w_{\boldsymbol{x}}(\boldsymbol{m})$'s, have been computed from the given monochrome image, they are then fixed and used to (globally) embed the colors scribbled on by the user. The two chromaticity signals, $\boldsymbol{C}(\boldsymbol{x}) = (\boldsymbol{U}(\boldsymbol{x}), \boldsymbol{V}(\boldsymbol{x}))$, are generated by solving the following linearly constrained optimization problems

$$\min_{\boldsymbol{C}} \Phi(\boldsymbol{C}(\boldsymbol{x})) = \sum_{x} \left\| \boldsymbol{C}(\boldsymbol{x}) - \sum_{\boldsymbol{m}\in N_x} w_x(\boldsymbol{m}) \boldsymbol{C}(\boldsymbol{x} - \boldsymbol{m}) \right\|^2 \qquad (5)$$

The cost function, similar to (4), is based on locally linear reconstruction errors, but this time the weights, $w_x(\boldsymbol{m})$'s (computed from **Step 1**) are fixed while optimizing the chromaticity $\boldsymbol{C}(\boldsymbol{x})$ subject to the color constraints scribbled on by the user, i.e., at locations $\boldsymbol{x}_i$, $\boldsymbol{C}(\boldsymbol{x}_i) = (\boldsymbol{u}_i, \boldsymbol{v}_i)$, $i = 1, 2, \ldots$ This problem has some well-known solutions, we again follow the method of LLE under different constraints.

It is worthy mentioning that [1] solved a similar problem but with different $w_x(\boldsymbol{m})$'s to ours. The optimization approach of [1] implemented the intuitive premise "pixels with similar intensities should have similar colors". Our approach is more formal in the sense that (5) and (4) have exactly the same form and the geometry of the monochrome computed using (4) is transmitted (unaltered) onto the chromas using (5).

IV. RESULTS

We have implemented our CLNE algorithm in Matlab. As discussed in the text, we followed the methods suggested by the authors of LLE [3, 4] (but under different constraints) to solve both the local relation coefficients (4) and the color embedding (5) problems in closed forms. The initial colors were scribbled on by the user similar to the way it was done in the work of [1, 2]. Fig.2 shows examples of the results of our colorization algorithm and that of [1]. More examples of our method are shown in Figures 1 and 3. These results show that the algorithm is quite effective.

V. CONCLUDING REMARKS

Our method is closely related to the method of [1] which partially inspired our work. Computationally, our method is more similar to LLE [3, 4]. Unlike [1] and [2] we computed the local pixel neighborhood geometry of the provided monochrome image by solving a constrained least squares problem. We also empirically showed that the (computed) geometries of the local pixel patches of different color channels were highly correlated hence justifying the embedding of the geometry from the monochrome to the chrominance images. Our work suggested an alternative approach to semiautomatically colorizing monochrome images, also, we provided empirical evidence that justified our method as well as supported methods of previous authors [1, 2].

REFERENCES

[1] A. Levin, D. Lischinski and Y. Weiss, "Colorization using optimization", SIGGRAPH 2004

[2] G. Sapiro, "Inpainting the colors", IMA Preprint Series #1979, Institute for Mathematics and Its Applications, University of Minnesota, May 2004

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding", Science 290, 2323 – 2326, 2000

[4] L. K. Saul and S. T. Roweis, "An introduction to locally linear embedding", http://www.cs.toronto.edu/~roweis/lle/

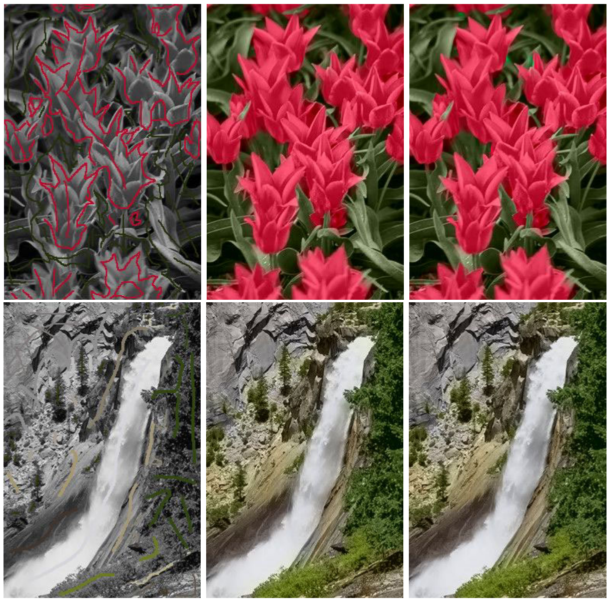

Fig. 2, Examples of colorization results. Monochrome images with user scribbled colors (left column), results of our CLNE method (middle column) and results of the method of [1] (right column).

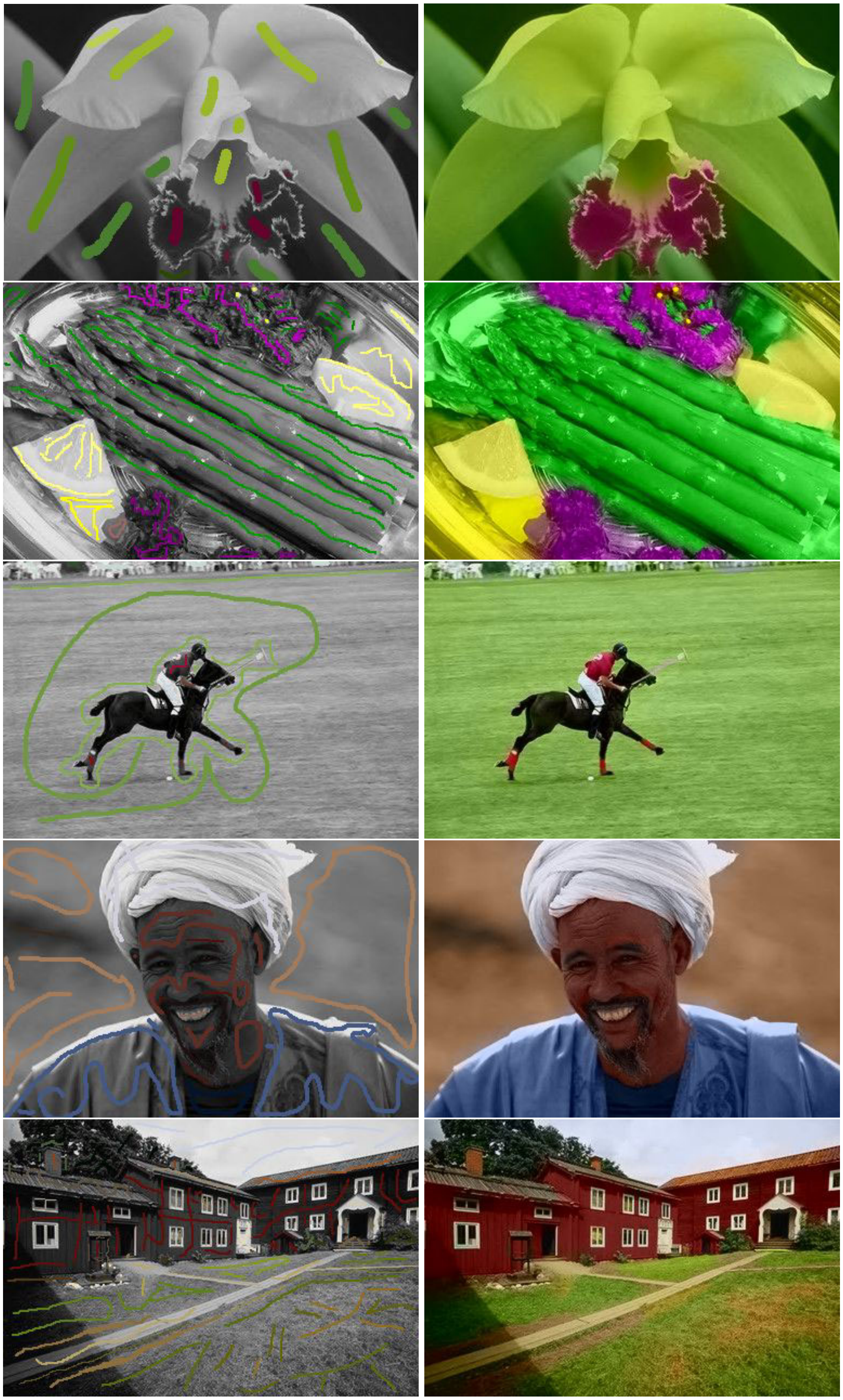

Fig. 3, More examples of our algorithm. Left column: original monochrome with user scribbled initial colors. Right column: colorization results by CLNE.