SUMMARY

Color transfer is a simple process to change a color tone in one image (source) to look like another image (target). In transferring colors between images, there are several issues needed to be considered including partial color transfer, trial-and-error, and multiple target color transfer. Our approach enables users to transfer colors partially and locally by letting users select their regions of interest from image segmentation. Since there are many ways that we can transfer colors from a set of target regions to a set of source regions, we introduce the region exploration and navigation approach where users can choose their preferred color tones to transfer one region at a time and gradually customize towards their desired results. The preferred color tones sometimes can come from more than one image; therefore our method is extended to allow users to select their preferred color tones from multiple images. Our experimental results have shown the flexibility of our approach to generate reasonable segmented regions of interest and to enable users to explore the possible results more conveniently.

key words: color transfer, probabilistic segmentation, Gaussian mixture models, image segmentation, image exploration

1. Introduction

The ability to make changes or manipulate the image content is important in the image editing task, especially the color content in an image which is important to the human interpretation. Therefore, there has been research work in the past to manipulate color information in the image.

Color transfer is one simple idea with the goal to change a color tone in one image (source) to look like another image (target). Reinhard et al. [1] introduced a simple color transfer technique that can transfer colors between two images. The algorithm first converts pixels in the RGB color space to the lab color space introduced by Ruderman et al. [2]. Then, a color distribution of each source and target image is modeled by the Gaussian model, i.e., mean (μ) and standard deviation (σ). Color transfer is performed for each source pixel (Cs) by a linear transformation on each channel as the following equation:

\[ C_s' = \frac{\sigma_t}{\sigma_s} (C_s - \mu_s) + \mu_t \] (1)

where subscripts s and t refer to the source and target images, respectively. The result pixels (Cs’) are finally converted back to the RGB color space.

Since the work of Reinhard et al., there has been a lot of research work proposed in the area related to color transfer trying to improve the quality of transferred results. However, while performing the task of color transfer between images, several issues need to be considered.

- Whole vs. partial: Sometimes users may want to edit color of the source image partially.
- Users do not always know exactly what colors that they want for their final image while they are using color transfer tools.
- Single vs. multiple targets: Sometimes preferred colors may come from multiple target images.

Our approach addresses the above issues to make the process of transferring colors between images more convenient. Our approach enables users to transfer colors locally by defining regions of interest. Since image segmentation is considered a challenging problem, we are not expecting our segmentation algorithm to work well in all types of images. However, we are aiming for user convenience instead; therefore, we focus on an automatic segmentation algorithm which allows the users to control the possible number of segmented regions and be able to interactively pick and choose regions of interest or merge several regions into one bigger region that they want to work with.

When we divide source and target images into regions, the number of possible combinations from transferring a set of color tones from the target image regions into the source image regions is usually large. To produce all possibilities exhaustively and let users choose the one that they like is normally impractical and undesirable. Thus, we introduce the regional exploration and navigation approach where the users can choose their preferred color tones one region at a time and gradually customize towards their desired results. The preferred color tones might not be entirely in one target image; therefore, our method is extended to allow users to select their preferred color tones from multiple images.

The rest of the paper is organized as follows: Sect. 2 presents related work in the color transfer between images. Section 3 describes our approach on image segmentation, region merging, color transfer between regions, region exploration and navigation which includes transferring colors from multiple target images. Section 4 shows the experimental results using our approach. Finally the conclusions are given in Sect. 5.
2. Related Work

Global color transfer  Early work in color transfer focuses on changing colors of the whole image. Reinhard et al. [1] modeled colors in the image by Guassian models. Cheng and Hsia [3], Kotera [4], Xiao and Ma [5], Abadpour and Kasaei [6] used the principal component analysis (PCA) to model image colors. Pité et al. [7] proposed a nonlinear color transfer technique using N-dimensional probability density function transfer. This is an iterative technique that gradually changes an image histogram of a source image to match that of a target image. Liu [8] introduced another nonlinear technique based on multiple regression analysis. These global color transfer methods usually work well when images have simple color tones. With complex color analysis. These global color transfer methods usually work well partially. Tai et al. [9] presented a region-based color transfer technique based on a probabilistic segmentation. Xiang et al. [10] continued the local idea with more target images. They also introduced additional rules for region mapping and some evaluation criteria. Chang et al. [11] used predefined basic colors to classify pixels and group them as local regions. Both source and target images are segmented according to the basic colors. Each corresponding pair of basic color regions then have color transfer performed using a linear function. In another technique, Greenfield and House [12] had a similar idea with palette colors which are derived from image pixels instead of pre-defined colors and are not fixed or limited to the number of palette colors as in [11]. Recent work by Wu et al. [13] presented a content-based method for transferring the color patterns between images. Their method takes advantage of high-level scene content analysis to help in transferring colors between regions. These local color transfer methods define regions automatically and transfer colors locally. However, the segmented regions sometimes are not what the users expect. Also the number of segmented regions cannot be easily controlled. Once the regions in the source and target images are defined, the next step is to find correspondences between them to transfer color. Rule-based approaches are used to create mapping between regions automatically. However, users might not always know what they need; therefore, having ways to allow users to explore their imagination might be more beneficial.

Interactive color transfer  Interactive techniques aim to add more flexibility for the users in defining regions of interest when they perform color transfer. Interactive selection tools such as swatches, control points, scribbles, strokes, and brushes [1], [14]–[19] are applied in the interactive color transfer algorithms. Luan et al. work [15] used a brush tool to multi-label regions on both source and target images. It used brush colors to distinguish region correspondences as well as special color for the region a user wants to keep untouched from the color transfer process. Konushin et al. [14] also provided a stroke tool for labeling. However, its segmentation process uses cellular automata, which starts the process from the pixels around the provided strokes, then the algorithm iteratively evolves the states of the pixels next to the stroke boundaries and spreads out to cover the whole image. Dong and Xu [20] continued the Konushin et al. work in the re-coloring part by using the same Growcut technique [21]. Maslennikova and Vezhnets’s work [22] was also an interactive local color transfer. However, it did not define an object or a specific area to take a transfer effect. Instead, all pixels in a source image had their values changed by the weighting technique. Each source pixel found a distance between its color and the mean color of the marked area, and used the inverse of distance as its weight.

Interactive color transfer techniques offer more control to users to define their regions of interest. Sometimes, the defining or selection process might be error-prone for less skilled users, i.e., they might draw strokes in the wrong place and these algorithms generate undesirable regions. Once the regions are selected both in the source and target images, the color transfer happens between them. There is a tight coupling between the defining region process and the color transfer process. If users are not satisfied with the results, they might need to repeat the process many times until they are satisfied. Another interactive approach introduced by Shapira et al. [23] gave another point of view of interactive image editing. The exploration method is a semi-automatic and iterative method that generates results for a user as choices. Hence, a user can navigate, evaluate, and select a desired result. The algorithm can respond to the user selection by generating a new set of results due to the current selection.

3. Approach

3.1 Image Modeling by GMM

Since our aim is to help users to be able to edit colors in an image partially. Our method is based on the work by Tai et al. [9] on modeling an image as GMM. In a color image I with N pixels, the joint color distribution of a whole image is modeled by a mixture of M Gaussian models which can be calculated as:

\[
p(x|\Theta) = \sum_{i=1}^{M} a_i p(x|\lambda_i)
\]  

where \( x \) is a pixel in \( I = \{x_1, x_2, \ldots, x_I\} \); \( x_j \in \mathbb{R}^3 \) and the parameters \( \Theta = a_1, \ldots, a_M, \lambda_1, \ldots, \lambda_M \) where \( a_i \) denotes the weight of each Gaussian and \( \lambda_i \) is its respective Gaussian parameters. The prior weight \( a_i \) must satisfy \( a_i \geq 0; i = 1, \ldots, M \) and \( \sum_{i=1}^{M} a_i = 1 \).

Each \( p(x|\lambda_i) \) is a probability density function and can be computed as below:

\[
p(x|\lambda_i) = \frac{1}{(2\pi)^{3/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2} (x-\mu)^T \Sigma_i^{-1} (x-\mu)}
\]  

where \( x = (x_1, x_2, x_3) \) is the pixel in the image, \( \mu \) is the mean of the Gaussian, and \( \Sigma \) is the covariance matrix.
where $\lambda_i$ is the parameter set of $G(\lambda_i)$ that includes the mean vector $\mu_i$ and the co-variance matrix $\Sigma_i$ (which contains the standard deviation vector $\sigma_i$).

Expectation-maximization (EM) [24], [25] is a widely used method for estimating the parameter set ($\Theta$) of the models in GMM using unlabeled data (pixels). The algorithm of EM iterates between two steps until converged. E-step calculates the expectation of the log-likelihood over all possible assignments of pixels. $p^{i,x}$ is probability that a pixel $x$ belongs to the Gaussian $i$. The probabilities are subject to $\sum_{i=1}^{M} p^{i,x} = 1$ and can be derived by:

$$p^{i,x} = \frac{a_i p(x|\lambda_i)}{\sum_{i=1}^{M} a_i p(x|\lambda_i)} \quad (4)$$

M-step maximizes the expectation according to the current parameters. The new parameters are updated for the next iteration. For region $i$, the new values $\tilde{a}_i$, $\tilde{\mu}_i$, and $\tilde{\Sigma}_i$ are re-estimated as follow:

$$\tilde{a}_i = \frac{\sum_{n=1}^{N} p^{i,x_n}}{N} \quad (5)$$

$$\tilde{\mu}_i = \frac{\sum_{n=1}^{N} x_n p^{i,x_n}}{\tilde{a}_i N} \quad (6)$$

$$\tilde{\Sigma}_i = \frac{\sum_{n=1}^{N} p^{i,x_n}(x_n - \tilde{\mu}_i)(x_n - \tilde{\mu}_i)^T}{\tilde{a}_i N} \quad (7)$$

The EM for an image is modified by considering the spatial information [9], [16], [26] in order to keep changes consistent among neighborhood pixels. The initial values in the EM algorithm $a_i$, $\mu_i$, and $\Sigma_i$ are calculated by the k-means clustering algorithm. The algorithm requires the number of cluster $k$ at the beginning step and returns $k$ clusters as regions of the image. The k-means clustering technique is used as hard segmentation for an image. A cluster $i$ is a region which is used to calculate its mean $\mu_i$ and standard deviation $\sigma_i$ from its member pixels. The number of pixels in region $i$, $|R_i|$, is also calculated as the prior weight $a_i$ for region $i$ by: $a_i = |R_i|/N$.

### 3.2 Region Merging

We define regions from Gaussian mixture models. Figure 1 (d) shows typical probabilistic segmentation results of the image in Fig. 1 (a). Sometimes we will not get good quality segmentation results due to several factors such as a random step in the initialization process and the number of clusters set is less than the color tones in the image. One way to obtain a better quality segmentation result is to increase the number of initial $k$ clusters ($i.e.$ greater than 20), so that the algorithm can produce many intermediate regions as shown in Fig. 1 (b).

In our region merging step, a user then considers similar colors between intermediate regions and merges them interactively by pick-and-choose method. The new probability from merging two intermediate regions $i$ and $j$ is assigned to region $i'$ by:

$$p^{i',x} = \frac{a_i p(x|\lambda_i) + a_j p(x|\lambda_j)}{a_i + a_j} \quad (8)$$
$p^{f \cdot x_n} = p^{l \cdot x_n} + p^{l \cdot x_n}$ \hspace{1cm} (8)

where $p^{l \cdot x_n}$ is the new probability of the region $i$. Since the number of regions is reduced and $p^{l \cdot x_n}$ is changed, the rest of parameters in all remaining regions need to be re-calculated as in Eq. (5)–(7).

Figure 1 (c) shows example results from merging intermediate regions to form regions of interest. Compare between Fig. 1 (c) and Fig. 1 (d) we can see that we can obtain better segmentation results from our region merging step.

3.3 Color Transfer between Regions

Once we have our regions of interest both in the source and target images and assume that we know which region in the target image we wish to transfer to which region in the source image, then we transfer colors between them locally or partially. Since each region is represented by its own Gaussian model, color transfer between regions can be performed simply by the following equation:

$$x'_{Rs} = \frac{\sigma_{Rs}}{\sigma_{Rt}}(x_{Rs} - \mu_{Rs}) + \mu_{Rt}, \hspace{1cm} (9)$$

where $x_{Rs}$ is an arbitrary source pixel in a source region, $R_s$ while $R_t$ is a target region. Similar to the color transfer equation by Reinhard et al. [1], the resultant pixel $x'_{Rs}$ is derived separately from each color channel. To expect the good result from transferring color between regions using this simple color transfer equation, we should select or merge regions (from Sect. 3.2) that contain simple color tones.

3.4 Region Exploration and Navigation Approach

From the probabilistic segmentation results (Sect. 3.1) and the region merging step (Sect. 3.2), we may have many regions in both source and target images. If the numbers of the regions in the source and target images are $m$ and $n$ respectively, the total number of possibilities $P$ to map colors from target regions to source regions by allowing colors of the source regions to be kept unchanged would be $P = (n + 1)^m$.

Assume that a source image has 3 regions and the target image has 2 regions as shown in Fig. 2 (a), to generate all possible mapping combinations between source and target regions would produce 27 possible results as shown in Fig. 2 (b). However, it is usually not practical to let the program generate all possibilities exhaustively and let users browse all options and select the most preferred version.

Figure 2 (c) illustrates the concept of our region exploration and navigation approach. It focuses on a user’s desire which is not limited by the rules. Users have the ability to navigate and explore the color transfer results region by region and work gradually until they are satisfied with the results. Region exploration and navigation framework is an iterative process. Each iteration considers only one source region versus each target region. The flow of the exploration is straightforward, e.g., exploring just one round from the first source region until the last one. Otherwise, the flow can be repeated by navigating back and forth along source regions many times until a user is satisfied.

For each iteration, a user can evaluate the color-transferred results in one region and select the preferred one, or just keep exploring to the next region without any change. If a user selects one of the results (surrounded by the dark-framed rectangle) as shown in Fig. 2 (c), colors of the considered source region will be transferred according to the selection, and the source image will be updated for the next iteration. A user can make any change at any stage of the exploration process, and may obtain desired results from any stage while exploring.

Our framework is not only limited to one target image, but it can also be extended to transfer colors from multiple target images as users’ preferred colors may be in several images as shown in Fig. 2 (d). There are even more possibilities when dealing with multiple target images as shown in Fig. 2 (e). However, Fig. 2 (f) illustrates that our approach extends to several target images and still allows users to navigate and explore the color transfer results region by region in the same framework as a single target image.
4. Results

We perform color transfer experiments using our approach and also compare results with several existing color transfer methods. Our approach demonstrates the flexibility of generating reasonable segmented regions of interest and enables users to explore and navigate the possible results more conveniently. We also show the examples of permutation of color tones within an image itself and the extension to transfer colors from multiple target images. Note that for images used in our paper, we set initial number of clusters $k = 20$ as suggested by [9] and we obtained reasonable intermediate regions in all of our experiments.

The first experiment is to transfer colors from a single target image to a source image. We provide results from two different target images. Figure 1 (a) and 1 (c) are the source image and its segmented and merged regions. Figure 3 (a) and 3 (b) are the first target image and its segmented and merged regions. The source image has 4 regions and the first target image has 5 regions. Hence, there are 1,296 possible combinations to transfer color between them. However, our exploration and navigation approach offers region-by-region color transfer between one source region and one target region at a time, as shown in Fig. 4. Each row of Fig. 4 shows possible results of transferring colors from target regions to one source region. We can evaluate the intermediate results and select the ones that we prefer. Figure 5 (a) shows the final result from the selected intermediate results (images with the frame in Fig. 4) after the exploration and navigation step. We can also generate many other preferred results as shown in Fig. 5 (b)–(e).

For the second target image, Fig. 3 (c) and 3 (d) are the image and its 9 segmented and merged regions. We use the same source image previously used. Therefore, there are 10,000 possible transferred results. Using our approach, Fig. 6 (a)–(e) show examples of the transferred results that we can obtain.

The second experiment is to compare our method to other color transfer techniques. Figure 7 (a) and 7 (b) are source and target images. Figure 7 (c) is the result generated by the global color transfer method [1]. The color tone in the source image is changed as a whole. Partial color editing is difficult to perform in this case. Figure 7 (d) is generated from the local color transfer method [9] where a rule-based approach is used to create mapping between source and target regions automatically. The difficulty of this approach is the defining of such rules that reflect the user preferences because most of the time users do not know what colors that they want for their final image. Figure 7 (e) is the result from using the interactive color transfer method [15]. The regions both in the source and target images must be
selected manually by the brush selection tool. If the result is not satisfied, the user needs to repeat the whole process again. Figure 7 (f)–(j) show the source’s and target’s merged regions using our method. Figure 7 (k)–(o) show our results after we explore and navigate through various intermediate results and choose the preferred ones.

Sometime users may not want to transfer color tones from another image, but they simply want to see how the image would look like if color tones are being rearranged within itself. Therefore, for the third experiment, we concern about the rearrangement or permutation of color tones within the image itself. Using our framework, the source
Fig. 7 Comparison of our method with three existing color transfer methods: (a) a source image; (b) a target image; (c) the result by the global color transfer method [1]; (d) the result by the local color transfer method [9]; (e) the result by the interactive color transfer method [15]; (f)–(h) the source’s merged regions from our method; (i)–(j) the target’s merged regions from our method; (k)–(o) several results from our method. (a)–(e) are images from [15].

Fig. 8 Color transfer within one image. Several results from different rearrangements of color tones.

and target images are the same image in this case. We simply perform this task by segmenting and merging regions first. Then we explore and navigate through different results. For the results in Fig. 8, we use Fig. 3 (a) to be our source and target images. Figure 8 shows different rearrangements of color tones obtained by our approach.

The fourth experiment is to show that our approach can handle the multiple target images similar to the single target image. Figure 9 (a)–(e) show one source image with three different colors of tulips and its merged regions. Figure 9 (f)–(n) show three target flower images and their merged regions for each target flower. There are 2,401 possible combinations to transfer colors between source and target regions. Figure 9 (o)–(q) are the intermediate results from transferring colors from each target flower to the specified source tulips. Finally, Fig. 10 shows other possible results of color transfer from these target flowers.

5. Conclusion

In this paper, we present a new color transfer technique that aims to make the color transfer process more convenient. Our technique allows users to do local or partial color transfer based on probabilistic segmentation along with region merging to define regions of interest. Since there are many ways to transfer a set of colors from one image to another image, we introduce the region exploration and navigation approach to allow users to explore and navigate the color transfer results region by region and gradually work towards satisfying results. In our method, we can also extend our transferred colors to come from multiple target images with the same framework.

Currently, our approach focuses on transferring colors between images. In the future, we are interested to extend
Fig. 9  Color transfer with multiple target images: (a) a source image; (b)–(e) the source’s merged regions; (f) the first target image; (g) and (h) the first target’s merged regions; (i) the second target image; (j) and (k) the second target’s merged regions; (l) the third target image; (m) and (n) the third target’s merged regions; (o) an intermediate result by transferring color between (d) and (g); (p) an intermediate result by transferring color between (c) and (j); (q) an intermediate result by transferring color between (b) and (m). Source and target images are from https://pixabay.com.

Fig. 10  Possible results of color transfer from multiple target images with other alternative user selections.

our work to transfer colors between videos. More efficient segmentation and region merging algorithms will be needed to handle frames in videos. It would be interesting to come up with techniques to allow users to explore and navigate their intermediate results while they are transferring colors between videos. Also the extension to handle multiple target
videos will be needed for color transfer between videos.

References

[1] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” IEEE Computer Graphics & Applications, vol.21, no.5, pp.34–41, 2001.
[2] D.L. Ruderman, T.W. Cronin, and C.-C. Chiao, “Statistics of cone responses to natural images: implications for visual coding,” Journal of the Optical Society of America A, vol.15, no.8, pp.2036–2045, 1998.
[3] S.-C. Cheng and S.-C. Hsia, “Fast algorithms for color image processing by principal component analysis,” Journal of Visual Communication and Image Representation, vol.14, no.2, pp.184–203, 2003.
[4] H. Kotera, “A scene-referred color transfer for pleasant imaging on display,” in Proceedings of the 2005 IEEE International Conference on Image Processing, pp.5–8, 2005.
[5] X. Xiao and L. Ma, “Color transfer in correlated color space,” in Proceedings of the 2006 ACM international conference on Virtual Reality Continuum and its Applications, pp.305–309, 2006.
[6] A. Abadpour and S. Kasaei, “An efficient PCA-based color transfer method,” Journal of Visual Communication and Image Representation, vol.18, no.1, pp.15–34, 2007.
[7] F. Pitié, A.C. Kokaram, and R. Dahyot, “N-dimensional probability density function transfer and its application to colour transfer,” in Proceedings of the 10th IEEE International Conference on Computer Vision, pp.1434–1439, 2005.
[8] C.C. Liu, “A global color transfer scheme between images based on multiple regression analysis,” International Journal of Innovative Computing, Information and Control, vol.8, no.1(A), pp.167–186, 2012.
[9] Y.-W. Tai, J. Jia, and C.-K. Tang, “Local color transfer via probabilistic segmentation by expectation-maximization,” in CVPR 2005: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.747–754, 2005.
[10] Y. Xiang, B. Zou, and H. Li, “Selective color transfer with multi-source images,” Pattern Recognition Letters, vol.30, no.7, pp.682–689, 2009.
[11] Y. Chang, S. Saito, and M. Nakajima, “A framework for transfer colors based on the basic color categories,” in CGI 2003: Proceedings of the Computer Graphics International, pp.176–181, 2003.
[12] G.R. Greenfield and D.H. House, “Image recoloring induced by palette color associations,” Journal of WSCG, vol.11, no.1, pp.3–7, 2003.
[13] F. Wu, W. Dong, Y. Kong, X. Mei, J.-G. Paul, and X. Zhang, “Content-based color transfer,” Computer Graphics Forum, vol.32, no.1, pp.190–203, 2013.
[14] V. Konoshin and V. Vezhnevets, “Interactive image colorization and re-coloring based on coupled map lattices,” Proceedings of International Conference on Computer Graphics & Vision, pp.231–234, 2006.
[15] Q. Luan, F. Wen, and Y.-Q. Xu, “Color transfer brush,” in PG 2007: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, pp.465–468, 2007.
[16] J. Wang and M.F. Cohen, “An iterative optimization approach for unified image segmentation and matting,” in ICCV: Proceedings of the 10th IEEE International Conference on Computer Vision, pp.936–943, 2005.
[17] T. Welsh, M. Ashikhmin, and K. Mueller, “Transferring color to greyscale images,” in SIGGRAPH 2002, Computer Graphics Proceedings, pp.277–280, 2002.
[18] C.-L. Wen, C.-H. Hsieh, B.-Y. Chen, and M. Ouhyoung, “Example-based multiple local color transfer by strokes,” Computer Graphics Forum, vol.27, no.7, pp.1765–1772, 2008.
[19] Y. Hwang, J.-Y. Lee, I.S. Kweon, and S.J. Kim, “Color transfer using probabilistic moving least squares,” in CVPR 2014: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.3342–3349, 2014.
[20] Y. Dong and D. Xu, “Interactive local color transfer based on coupled map lattices,” in CAD/Graphics 2009, Proceedings of the 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp.146–149, 2009.
[21] V. Vezhnevets and V. Konouchine, “Growcut–interactive multi-label N-D image segmentation by cellular automata,” Proceedings of International Conference on Computer Graphics & Vision, pp.150–156, 2005.
[22] A. Maslennikova and V. Vezhnevets, “Interactive local color transfer between images,” Proceedings of International Conference on Computer Graphics & Vision, pp.146–149, 2007.
[23] L. Shapira, A. Shamir, and D.C. Or, “Image appearance exploration by model-based navigation,” Computer Graphics Forum, vol.28, no.2, pp.629–638, 2009.
[24] J. Blimes, “A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models,” Technical report, TR-97-021, University of California, Berkeley, 1998.
[25] A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” Journal of the Royal Statistical Society, series B, vol.39, no.1, pp.1–38, 1977.
[26] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” ACM Transactions on Graphics, vol.23, no.3, pp.689–694, 2004.
[27] Q. Luan, F. Wen, D.C. Or, L. Liang, Y.Q. Xu, and H.Y. Shum, “Natural image colorization,” SR 2007 Rendering Techniques, pp.309–320, 2007.

Somchai Phathathanachuanchom is a Ph.D. student in Faculty of Information and Communication Technology at Mahidol University, Thailand. In 2002, he received his Master degree in Communication and Media Engineering from Offenburg University, Germany. His research focuses on image processing.

Rawesak Tanawongsuwan received his Ph.D. degree in Computer Science from Georgia Institute of Technology, USA in 2003. He is currently an instructor in the Faculty of Information and Communication Technology, Mahidol University, Thailand. His main research interests include computer vision, image processing and computer graphics.