

A Computationally Efficient Technique for Image Colorization

Adrian Pipirigeanu, Vladimir Bochko, and Jussi Parkkinen

Department of Computer Science and Statistics, University of Joensuu, 80101
Joensuu, Finland

vbochko@uwasa.fi, parkkine@cs.joensuu.fi

Abstract. In this paper, the fast technique for image colorization is considered. The proposed method transfers colors from the color image (source) to the gray level image (target). For the source image, we use the segmented uniformly colored regions (dielectric surfaces) under single color illumination. This method maps the gray level image into the color space by means of parametrical mapping learnt using PCA and principal components regression. The experiments show the method's feasibility for colorizing the objects, and textures, as well.

1 Introduction

The demand for colorization is increasing, especially in the movie industry, television, the medical industry, computer graphics and scientific visualization. Colorization is still expensive, complicated and time-consuming and requires human participation.

There are several successful colorization schemes [11], [8]. These methods consider two images: a color image (source) and a gray level image (target). Then color is transferred from the source to the target if pixels match in both images. The methods implement matching by using achromatic information - intensity, pixel neighborhood statistics, and texture features. In Welsh's method, the user defines the color regions (swatches) used in colorization [11]. In Levin's method, the interactive colorization is improved by using color scribbles [8]. Color scribbles are used for gray level images and then color propagates to the neighbor pixels via optimization. Further improvement was made in the Horiuchi and Tominaga work, where the appropriate color seeds are automatically selected and propagate to adjacent pixels [5]. Their colorization scheme, incorporating the spatial pixel location and luminance value matching, is efficiently used in color image coding. The automatic colorization of gray level images is proposed by Vieira et al. [10]. They eliminate human participation in selection of the source images by using content-based image retrieval methods. A colorization approach where colorized images look realistic was proposed by Drew and Finlayson [4]. The colorized images are obtained by matching the gradient perceived in the original gray level image. In this case, the input color may mismatch the synthetic colors.

For methods [11], [8] [5], [10], [3] realistic colorization may be difficult because they are not based on a physical model. Therefore, the other methods incorporating physical image properties are proposed, [6], [1]. Horiuchi and Kotera consider colorization of gray level images based on a diffuse-only reflection model [6]. Their approach makes colorization more realistic. Abadpour and Kasaei propose the efficient PCA based colorization method [1]. For the segmentation defined by the user, the method generates the color vector related to the gray level pixel value as the function of the gray level pixel value, the mean of the gray level image, the mean vector and the first eigenvector of color image. The experimental results show that the PCA based method is superior to Welsh's method for several test images. However, methods [6], [1] are suitable for color images described by the incomplete physical model presented either by diffuse reflection (shadow and color) or by color and highlight. For example, Fig. 8 from [1] illustrates this and indicates that PCA is good for analysis of the linear or low degree nonlinear data structure. Our method works well with the incomplete and complete physical models because it explores nonlinear correlation between principal components. We propose modification of the authors' algorithm presented in [2]. To make the algorithm more computationally efficient, we replace the regression based on machine learning by closed form solution. The method works with segmented uniformly colored regions (dielectric surfaces) under single color illumination. The algorithm colorizes objects, and textures, as well.

In Section 2, we describe methodology of our approach. The experimental results are presented in Section 3. Finally, we give our conclusions in Section 4.

2 Methodology

2.1 Data Model for Colorization

Our design scheme for colorization includes a source, that is an RGB color image, and a target, that is a gray level image. The colors are then transferred from the source to the target. We assume that the source image has a roughly constant hue. Then, the physical approach suggests that the dichromatic reflection model describes data in color space [7]. The model considers the data structure as a linear combination of two linear clusters related to highlight and body reflection. Then, the first two eigenvectors span the data space. Alternatively, we consider the data structure as a single nonlinear cluster [2]. In this case, the intrinsic dimensionality of the data is one. To describe the data structure we still need the first two eigenvectors but the second principal component is not required because we approximate it using the first principal component. We assume that there is a single-valued mapping between these components. Thus, in addition to the two eigenvectors, we have to find the single-valued function of the first principal component approximating the second principal component. On the one hand, it is important for colorization to reduce the data dimensionality to one component where the color information is discarded. Then we simply replace the component with the gray level image. On the other hand, the first two eigenvectors and the mapping function contain information about color. This is

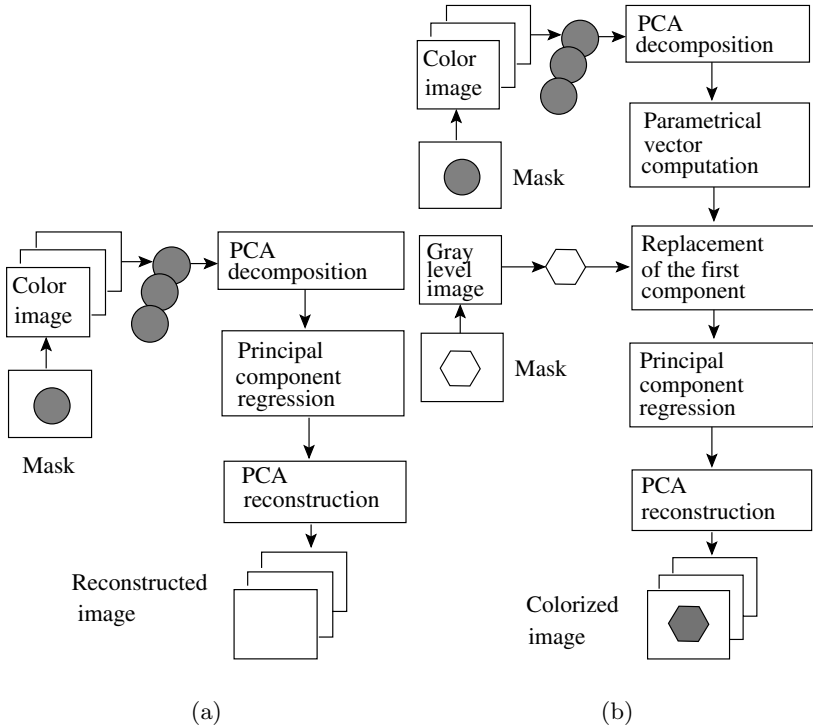


Fig. 1. a) A flow chart for image analysis. b) A flow chart for image colorization.

used in our colorization scheme because we have a straightforward way to embed color into the gray level image.

2.2 Regressive PCA

In this section, we consider an approach that combines two techniques, these being PCA and principal components regression. We call this approach regressive PCA (RPCA). A similar technique referred to as principal components regression (PCR) is used for enhancing regression accuracy in the feature space defined by the first principal components [9]. RPCA occupies the intermediate place between the design schemes dealing with linear and (high degree) nonlinear data structures, due to its limitation of assuming a single-valued mapping between components. Unlike [2], the main point of this paper is to use an analytic closed form solution. As a result, we expect to reduce a demand for computation time. The computation time issue is especially important when the volume of data being colorized is large.

The most suitable way for us is to represent the RGB color image as a vector $\mathbf{x} = (x_1, x_2, x_3)^T$, where the vector elements are random variables and T denotes the transposition. We consider that x_1 is a red component, x_2 is a green component and x_3 is a blue component. For computing, instead of the random

variables we use the sample or observation sets. We obtain these sets by stacking or concatenating the pixels values along rows or columns in each component image to arrange them as long vectors. Then these vectors are the elements in \mathbf{x} . For the vector \mathbf{x} , the elements are the row vectors while for the transposed vector \mathbf{x}^T , the elements are the column vectors.

Thus, we assume that \mathbf{x} is a three-dimensional observed vector

$$\mathbf{x} = (x_1, x_2, x_3)^T, \tag{1}$$

For PCA, the vector \mathbf{x} in Eq. 1 is first centered by subtracting its mean $\boldsymbol{\mu} = E(\mathbf{x})$,

$$\mathbf{x} \leftarrow \mathbf{x} - \boldsymbol{\mu}, \tag{2}$$

where the symbol \leftarrow represents substitution when the value of the right-hand side is computed and substituted in \mathbf{x} . The covariance matrix is then defined as

$$\mathbf{C} = E(\mathbf{x}\mathbf{x}^T), \tag{3}$$

where E denotes an expectation operator. PCA involves an eigen decomposition as follows:

$$\mathbf{C} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T, \tag{4}$$

where $\mathbf{U} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$, $\mathbf{e}_i = (e_{i1}, e_{i2}, e_{i3})^T$ are the eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix of the sorted eigenvalues of the covariance matrix, $\boldsymbol{\Lambda} = \text{diag}|\lambda_1, \lambda_2, \lambda_3|$, $\lambda_1 > \lambda_2 > \lambda_3$. Thus, the first two principal components are given by:

$$\mathbf{y}_2 = \mathbf{U}_2^T \mathbf{x}, \tag{5}$$

where $\mathbf{y}_2 = (y_1, y_2)^T$ and $\mathbf{U}_2 = (\mathbf{e}_1, \mathbf{e}_2)$. Then, polynomial regression is used to approximate the second principal component by the first principal component as follows:

$$\hat{y}_2 = f(y_1, \mathbf{w}) = \sum_{i=1}^M w_i \Phi_i(y_1) = \mathbf{w}^T \boldsymbol{\Phi}(y_1), \tag{6}$$

where \hat{y}_2 is the approximated y_2 principal component, $f()$ is a mapping function, $\boldsymbol{\Phi}$ represents the polynomial basis functions, $\boldsymbol{\Phi} = (\Phi_1, \dots, \Phi_M)^T$, and \mathbf{w} is a vector of the polynomial coefficients or a parametrical vector, $\mathbf{w} = (w_1, \dots, w_M)^T$. M is an order of the polynomial used for approximation (in our case $M = 3$). The vector \mathbf{w} is learnt from the regression approximation of y_2 by y_1 . The polynomial basis functions $\boldsymbol{\Phi}$ are considered $\Phi_j = y^j$, $j = 1, \dots, M$, so the equation Eq. 6 becomes

$$\hat{y}_2 = (w_1, w_2, w_3)(y_1, y_1^2, y_1^3)^T. \tag{7}$$

The parametric vector minimizing the approximation error is as follows:

$$\mathbf{w} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T y_2. \tag{8}$$

In Eq. 8, we use $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}^T$, where \leftarrow means substitution, and the samples of y_2 are arranged as a column. The term $\boldsymbol{\Phi}^\dagger = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T$ is the Moore-Penrose pseudoinverse of the matrix $\boldsymbol{\Phi}$.

The real and approximated components are an estimate of $\hat{\mathbf{y}}_2$, $\hat{\mathbf{y}}_2 = (y_1, \hat{y}_2)^T$. Finally, since \mathbf{U} is an orthogonal matrix satisfying $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, where \mathbf{I} is an identity matrix, reconstruction is defined as

$$\hat{\mathbf{x}} = \mathbf{U}_2\hat{\mathbf{y}}_2 + \boldsymbol{\mu}. \quad (9)$$

2.3 System Diagram

We consider two flow charts related to image analysis and colorization. The image analysis is important in order to examine how reconstruction is made in comparison with the original images. Fig. 1 shows the flow charts.

For image analysis, the color image is given. In addition, the user manually defines the mask corresponding to the region or the object which is used as a source image in colorization. These regions should contain shadows, colors and highlights, as well. PCA is used only for the masked regions. After PCA decomposition, principal components regression is implemented. We approximate the second principal component by the first one and reconstruct the image using PCA. Finally, we obtain the reconstructed image that is an image with synthetic colors.

For image colorization, the source image (color image) and the target image (gray level image) are given. The user manually defines the masks for both images. After selecting the mask area and PCA decomposition of the color image, we compute the parametrical vector Eq. 8. Then, the selected region of the gray level image replaces the first principal component y_1 . The single requirement here is that the gray level image should have the same dynamic range as the first principal component. Therefore, we scale the gray level image so that its minimum and maximum pixel values are equal to the minimum and maximum pixel values of the first principal component. Then we find the second principal component using polynomial regression Eq. 7 and reconstruct the image. As a result, we obtain the colorized image.

One problem arises in colorization when the first eigenvector changes its direction. In this case, the colorized image has negative contrast. To avoid this, we use the following empirical approach. First, we measure the average C_{avg} , minimum C_{min} and maximum C_{max} for the first principal component and I_{avg} , I_{min} and I_{max} for the intensity image $I = R + G + B$. Then, we compute

$$\text{sign}(C_{avg} - (C_{max} + C_{min})/2) = \text{sign}(I_{avg} - (I_{max} + I_{min})/2). \quad (10)$$

If Eq. 10 is not true, the pixel values of the gray level image are multiplied by -1 . This provides a positive contrast of the resultant image.

3 Experiments

We conducted experiments with several images of fruits and vegetables. They were obtained with a Canon EOS 40D reflex camera in laboratory conditions (a



Fig. 2. Test images. The first row: Apple, Green Stem, Oranges and Tomato. The image Green Stem is rotated. The second row: Avocado & Banana, Onion, Red Pepper and Yellow Pepper.

lightbox with daylight). Although two light sources were used, the data structure in the color space was similar to one light source and well defined by the physical model. Then, the images in RAW format were converted into JPEG format. The images are Apple, Green Stem, Oranges, Tomato, Avocado & Banana, Onion, Red Pepper and Yellow Pepper. The image size (width and height) except for the Green Stem is 1280×853 pixels. The size of the Green Stem image is 853×1280 . Fig.2 shows the test images.

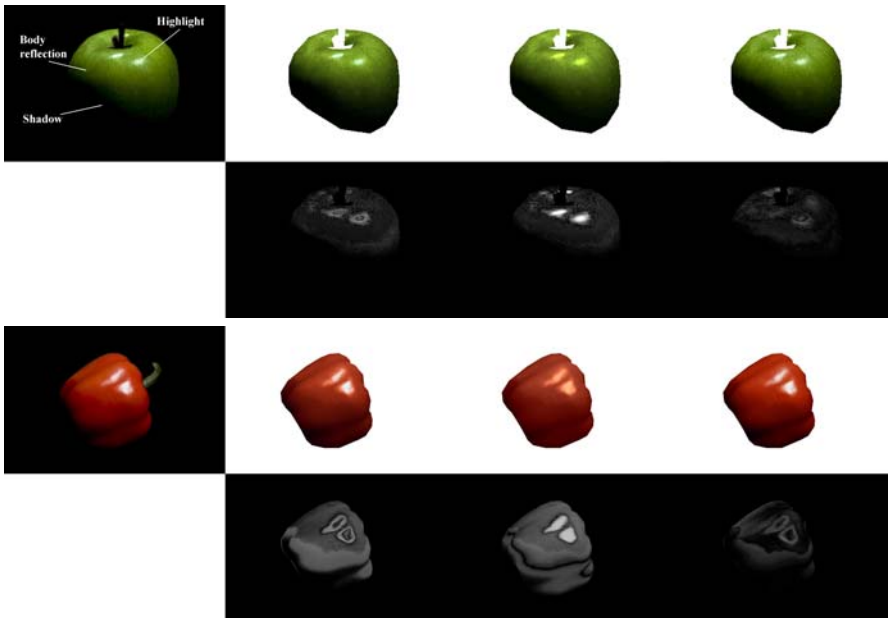


Fig. 3. Apple and Red Pepper. The original images (first column). The images are partially covered by a semi-transparent mask. PRPCA (second column), PCA (third column) and RRPCA (fourth column) reconstruction and the error maps.

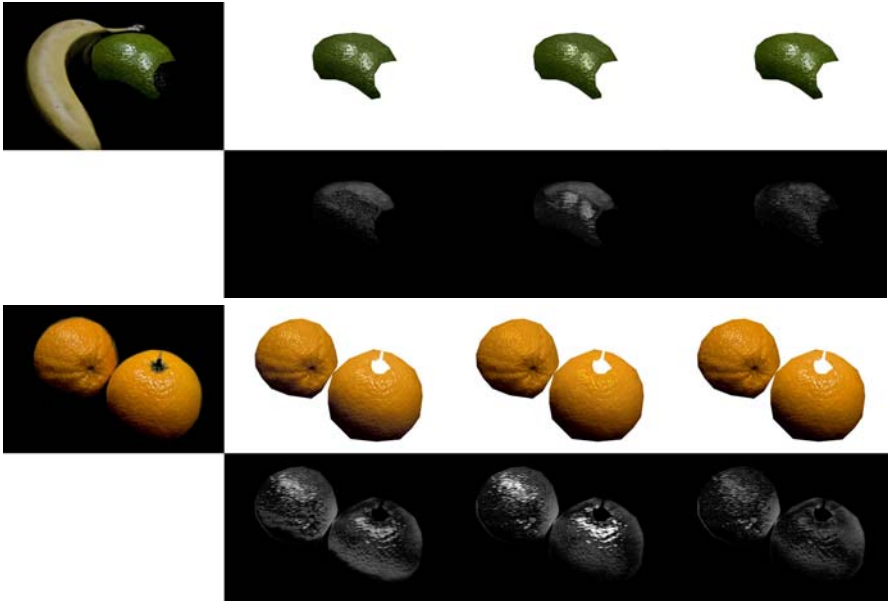


Fig. 4. Avocado & Banana and Orange. The original images (first column). The images are partially covered by a semi-transparent mask. PRPCA (second column), PCA (third column) and RRPCA (fourth column) reconstruction and the error maps.

In order to evaluate the performance, we compare the proposed method with standard PCA and RPCA based on RBF regression [2]. The parameters for the RBF neural network are 10 iterations and 7 hidden neurons.

First, we test the algorithm for accurate reproduction of color images. For selected image regions PCA, RPCA based on polynomial regression (PRPCA) and RPCA based on RBF regression (RRPCA) are used. The color differences between the original images and the reconstructed images are measured by using

Table 1. Average and maximum S-CIELAB ΔE

Image	PRPCA		PCA		RRPCA	
	ΔE_{avg}	ΔE_{max}	ΔE_{avg}	ΔE_{max}	ΔE_{avg}	ΔE_{max}
Apple	1.78	43.16	2.80	164	1.22	17.18
Avocado	5.14	58.2	7.32	127.88	3.45	69.6
Banana	3.29	137.39	3.32	146.62	3.15	129.7
Green Stem	4.29	41.69	6.60	53.04	2.94	38.26
Onion	1.86	33.1	3.43	56.72	0.89	22.04
Oranges	4.25	73.37	5.38	154.66	2.67	53.99
Red Pepper	6.43	41.52	9.16	72.46	2.05	19.88
Tomato	5.92	37.83	7.08	81.91	3.05	31.82
Yellow Pepper	1.15	30.60	1.64	69.74	0.66	27.39

the S-CIELAB ΔE [12]. We also use an error map. For the error map, the error values greater than 100 are truncated and the remaining values are scaled in the range [0-255]. Fig.3 illustrates the reconstruction results and the error maps made by using three methods. Only parts of the color objects are used. The regions used in colorization contain shadow, color and highlight. PRPCA and RRPCA visually reproduce images very close to the original images. The error maximum for PRPCA and RRPCA is around the boundaries of highlight areas. It was found that RRPCA gives the best results. PCA has large error values in the highlight regions due to its lack of approximation of the nonlinear data structure. The visual reproduction of highlights by PCA is also not very good.

Similar results are obtained for color textures. Fig.4 shows texture surfaces and their reproduction along with an error map. This is a case of multiple highlights. These images are more difficult for approximation in comparison with the objects. One can see that the error values in color and color-shadow regions are increased. PRPCA has visual color and highlight reproduction close to RRPCA. Table 1 summarizes the color differences computed for the test images. RRPCA is superior to the other methods. The error rate of the proposed method (PRPCA) is better than PCA. The Banana image does not have the highlight areas, therefore the RPCA methods have results comparable with PCA.

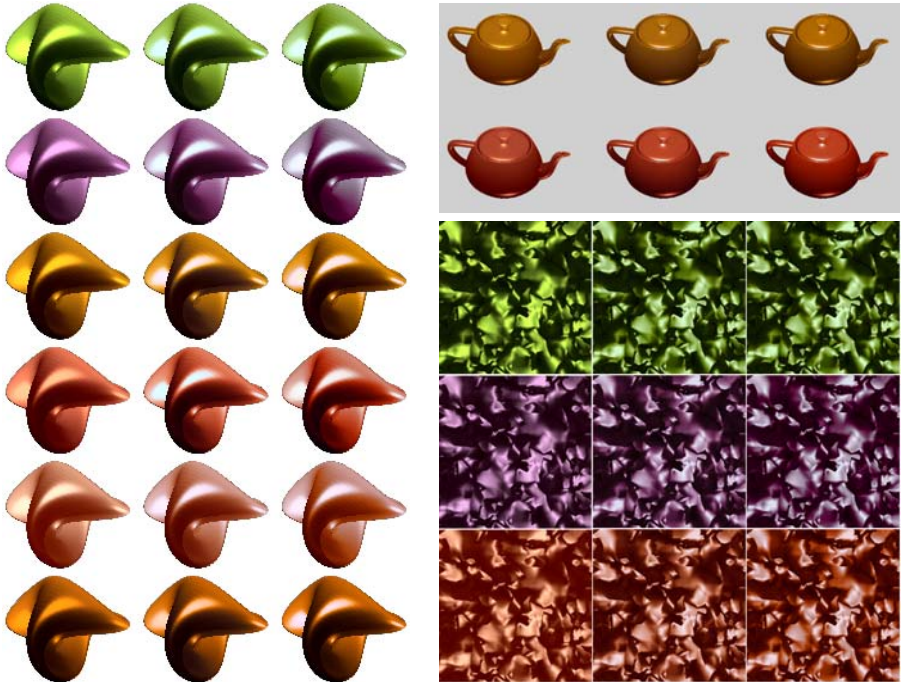
In addition, we compare the methods by using computation time. The methods are implemented using Matlab 7.0, Intel Core 2 Duo CPU at 2 GHz and 2GB of RAM. Table 1 shows the computational time. PCA is fastest. Since PRPCA combines PCA and principal components regression, its computation time is longer than PCA computation time. But the method is still fast. RRPCA using machine learning is, however, much more computationally demanding in comparison with PCA and PRPCA.

Finally, we conducted the experiments with image colorization. We use the colors of the fruits and vegetables described above. For colorization, we take the synthetic 3D object, the synthetic teapot and the synthetic texture image.

Fig.5 shows the colorized images. Sometimes the highlight areas in the objects and texture are narrowed, since the scaling procedure affects the intensity values of gray level images. PCA results are not very good. The colorized images have

Table 2. Computational time, s

Image	PRPCA	PCA	RRPCA
Apple	0.285	0.162	15.068
Avocado	0.116	0.067	6.544
Banana	0.081	0.046	4.575
Green Stem	0.049	0.029	2.938
Onion	0.088	0.052	5.119
Oranges	0.391	0.197	20.149
Red Pepper	0.206	0.113	10.56
Tomato	0.255	0.124	12.73
Yellow Pepper	0.063	0.037	3.534



(a) The 3D object. For rows from top to bottom we use: Apple, Onion, Oranges, Red Pepper, Tomato and Yellow Pepper

(b) The teapot and texture. For rows (teapot) from top to bottom we use: Oranges and Red Pepper. For rows (texture) from top to bottom we use: Apple, Onion and Tomato

Fig. 5. The colorization results. For each object and texture the PCA resultant images are given in the first column, the PRPCA resultant images are given in the second column and the RRPCA resultant images are given in the third column.

poor contrast and color reproduction. The images colorized by PRPCA have good contrast and highlights. The images are sharp and clear. RRPCA also has good highlight reproduction and contrast. Colors are, however, usually more vivid with better appearance. Thus, PRPCA is a tradeoff between PCA and RRPCA. Since visual color reproduction by PRPCA is close to RRPCA and its computation time close to PCA, PRPCA is suitable for practical use.

4 Conclusions

In this paper, we considered the fast technique for image colorization. The technique using PCA and polynomial regression does not require segmentation of the image into the surface color and highlight regions. The visual colorization results are similar to the best method using RBF regression, while computation

time is short and close to PCA. The technique is simple for implementation and can be used for colorizing objects and textures, as well.

References

1. Abadpour, A., Kasaei, S.: An efficient PCA-based color transfer method. *Journal of Visual Communication and Image Representation* 18, 15–34 (2007)
2. Bochko, V., Parkkinen, J.: A spectral color analysis and colorization technique. *IEEE Computer Graphics and Applications* 26, 74–82 (2006)
3. Chen, T., Wang, Y., Schilings, V., Meinel, C.: Grayscale image matting and colorization. In: *Proc. ACCV 2004*, pp. 1164–1169 (2004)
4. Drew, M.S., Finlayson, G.D.: Realistic colorization via the structure tensor. In: *Proc. ICIP*, pp. 457–460 (2008)
5. Horiuchi, T., Tominaga, S.: Color image coding by colorization approach. *EURASIP Journal on Image and Video Processing* 8, 1–9 (2008)
6. Horiuchi, T., Kotera, H.: Colorization for monochrome image based on diffuse-only reflection model. In: *Proc. AIC 2005*, pp. 353–356 (2005)
7. Klinker, G.J., Shafer, S.A., Kanade, T.: A physical approach to color image understanding. *International Journal of Computer Vision* 4, 7–38 (1990)
8. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. *ACM Trans. Graphics* 23, 689–694 (2004)
9. Shawe-Taylor, J., Cristianini, N.: *Kernel methods for pattern analysis*. Cambridge University Press, Cambridge (2004)
10. Vieira, L.F.M., do Nascimento, E.R., Fernandes Jr., A., Carceroni, R.L., Vilela, D.R., de Araújo, A.A.: Fully automatic coloring of grayscale images. *Image and Vision Computing* 25, 50–60 (2007)
11. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to grayscale images. *Proc. ACM Siggraph* 20, 277–280 (2002)
12. Zhang, X., Wandell, B.: A Spatial extension of CIELAB for digital color image reproduction. In: *Proc. Soc. Information Display Symp. Technical Digest*, vol. 27, pp. 731–734 (1996)