A statistical approach for learning invariants: application to image color correction and learning invariants to illumination

B. Bascle, O. Bernier, V. Lemaire

Orange / France Telecom R & D

Abstract. This paper presents a new approach for automatic image color correction, based on statistical learning. The method both parameterizes color independently of illumination and corrects color for changes of illumination. The motivation for using a learning approach is to deal with changes of lighting typical of indoor environments such as home and office. The method is based on learning color invariants using a modified multi-layer perceptron (MLP). The MLP is odd-layered. The middle layer includes two neurons which estimate two color invariants and one input neuron which takes in the luminance desired in output of the MLP. The advantage of the modified MLP over a classical MLP is better performance and the estimation of invariants to illumination. The trained modified MLP can be applied using look-up tables (LUTs), yielding very fast processing. Results illustrate the approach.

1 Introduction

The apparent color of objects in images depends on the color of the light source(s) illuminating the scene. Because of this color constancy problem, image processing algorithms using color, such as color image segmentation or object recognition algorithms, tend to lack robustness to illumination changes. Such changes occur frequently in images (shadows, lights on/off, varying sunlight). To deal with this, a color correction scheme that can compensate for illumination changes is needed.

2 Illumination correction - state of the art

Color in images is usually represented by a triband signal, for instance Red-Green-Blue (RGB). As discussed in the introduction, this signal is sensitive to changes in illumination. However, image processing techniques need to be robust to such changes. Therefore color needs to be parameterized independently of illumination. This can be done by parameterizing color with one or two parameters or by correcting the triband signal. A number of color parametrization and color correction schemes have been described in the literature [9]. This section describes a number of approaches that work on a single image. Table 1 summarizes their pros and cons.

Examples of directly correcting the triband signal are diagonal color correction (such as gray world and white patch) and non-diagonal color correction [6]. They are both linear, and cannot model non-linearities. They also rely on limiting assumptions (known image mean for gray world, known maximum value for each channel for white

Table 1. Comparison of color correction approaches that work on a single image.

approach	principle of the approach	local / global	cons	pros
estimation of illuminant color [1]	neural network estimates illuminant chromaticity from image uv histogram	global	hypothesis of same illuminant for whole image, further processing needed for image correction	illuminant explicitly identified
estimation of ratio-based color invariants such as $(R_{x_1}G_{x_2})/(G_{x_1}R_{x_2})$ [2]	analytic color invariants	local / pixel-wise	original/ corrected images can't be reconstructed from invariant images	fast
luminance correction in the parametric HSV color space [3]	simple analytic color correction	local / pixel-wise	completely local, relatively sensitive to illumination changes	very fast (LUT implementation possible)
color transfer [4]	normalization by mean and variance in $l\alpha\beta$ color space	global	limited to global changes in illumination	fast
intrinsic image by entropy minimization [5]	finds an axis invariant to illuminant color by entropy minimization, then projects image perpendicularly to axis	global	need for few colors and many illuminations in image to determine invariant axis, not fast	works for any illuminants
diagonal color correction (gray world, white patch)	linear	global	limiting assumptions (gray mean, white max), non-linearities not modeled	very fast
non-diagonal color correction [6]	PCA-based linear correction	local / pixel-wise	illuminants must be known	fast (LUT implementation possible)
enhancement of dark images using modified luminance-based multi-scale retinex [7]	multi-scale convolution (linear)	large neighborhoods	aim is color correction for visual effect, performance for background subtraction unknown	fairly fast (3 frames / sec for 640x480 images), any type of lighting (blueish, reddish, etc ...)
color correction using a "classic" MLP [8]	statistical learning of non-linear color correction transform by MLP	local / pixel-wise with learnt global a priori about lighting	trained for a given rear projection environment & specific lighting conditions, does not estimate color invariants	could be very fast (using LUTs)
color correction using a trained modified MLP (this paper)	statistical learning of non-linear color correction transform by MLP $+$ statistical learning of 2 color invariants	local / pixel-wise with learnt global a priori about the type of lighting	trained for a range of lighting conditions (here lighting customary in home and office environments, e.g. whitish & yellowish)	very fast using LUTs (3.75 ms per frame or 266 frames per second for 320x240 images), trained for a range of illumination variations

patch, illuminants known for [6]). They are very fast and can be implemented using LUTs for even greater speed.

In [8] a neural network is used to learn the color correction needed in a specific rear projection environment. It does not estimate color invariants. It also is trained for specific lighting conditions.

An example of mono-band parametrization of color is hue (from hue-saturation-value, a.k.a. HSV) [3]. An example of bi-band color parameterization are chrominances uv (from the YUV color space) [3] and the ab values from the CIE Lab color space [3]. These three color representations (H, uv or ab) are analytical and thus do not require learning. They are fast pixel-wise methods. They have a certain robustness to illumination changes, but this robustness is limited. Color transfer [4] is a method with a similar philosophy, normalizing color by its mean and variance in $l\alpha\beta$ space. It is global and fast, but limited to global changes in illumination.

An approach for estimating color invariants from images consists in calculating ratios of RGB components at a given pixel (R/B) or between neighboring pixels (such as $(R_{x_1}G_{x_2})/(G_{x_1}R_{x_2})$) [2]. This method is also pixel-wise and thus fast. These invariants are also very robust to illumination changes. However, a lot of information about the original signal is lost and reconstructing it from the invariants is difficult.

A more sophisticated method has been proposed by [5]. It estimates a mono-band invariant and is based on a physical model of image formation. It works globally from the whole image. In $(log(R/B), log(G/B))$ color space, an axis invariant to illuminant color is determined by entropy minimisation. Projecting the image perpendicularly to the axis gives corrected colors. The approach does not require learning and applies to any type of illuminant, but is relatively slow. It also requires that the image contains relatively few different colors and many changes of illumination for each color.

Yet another approach consists in explicitly estimating the color of the illuminant [1]. A neural network estimates the chromaticity of the illuminant from the histogram of chromaticity of the whole image. The method works globally from the whole image and supposes there is only one illuminant for the entire image.

Another method is [7]. It is a bit out of the scope of this paper, since it aims at the enhancement of dark images for visual effect, and does not give information about performance for color correction. However, it gives a benchmark about speed, since the authors aimed at fast processing. This will be discussed in section 4.5.

3 A statistical approach to measure color invariants

3.1 A modified multi-layer perceptron: motivation

The motivation of this work is twofold: (1) to parameterize color compactly and independently of illumination by two invariants (2) to do it in real-time. Firstly, two parameters are needed to parameterize color with enough degrees of freedom to reconstruct a triband signal, given a luminance (or a gray level signal). Secondly, real-time processing (25/30 images per second for video) is also necessary for some applications. For this, slow methods such as [1] and [5] are unsuitable. Pixel-wise approaches are more suited. Among those, hue-Saturation, uv (from YUV) and ab (from the CIE Lab color space)

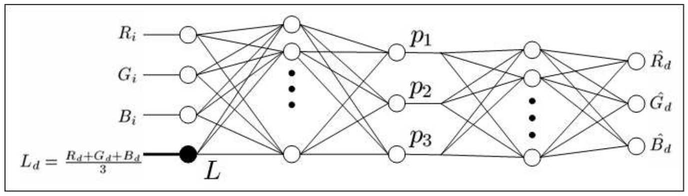

Fig. 1. A classical MLP with 4 inputs can be used to perform color correction. (R_i, G_i, B_i) is the input color. (R_d, G_d, B_d) is the desired output color, corresponding to the same color seen under a different illumination. $L_d = \frac{R_d+G_d+B_d}{3}$ is the luminance of the desired output and is a direct function of the illumination.

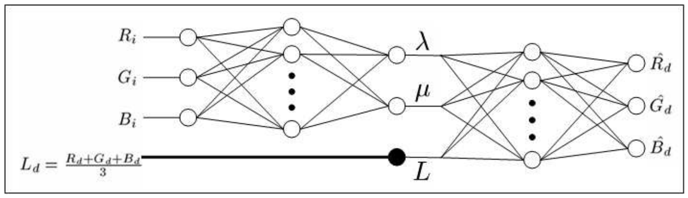

Fig. 2. A modified MLP for color correction and color invariant learning. λ and μ are the color parameters invariant to illumination that the MLP is trained to estimate. $(\hat{R}_d, \hat{G}_d, \hat{B}_d)$ are the actual outputs of the network. Bias neurons are omitted from this figure.

lack robustness to illuminations changes. [2] is robust to these, but reconstructing an image from the invariant(s) is difficult. A new fast approach is needed.

In practice, a limited range of illuminants are available in indoor environments. It is therefore interesting to use learning methods to find a color parameterization invariant to the "usual" illumination changes. This also provides a priori information about the illuminants, making the color correction global, which is, as Land showed [10]), necessary to perform correct illuminant correction. In practice, the lighting usually found in home and offices comes from fluorescent lights, incandescent light bulbs and natural sunlight from windows. They tend towards the whitish and yellowish areas of the spectrum (very few bluish or reddish lights). These are the illuminants that our approach deals with.

Our learning method of choice is neural networks and more specifically multi-layer perceptrons (MLPs) for their ease of use and adaptability. A classic MLP with 4 input neurons and 3 output neurons can be used for color correction under varying illuminations (see fig. 1). The fourth input, a context input, is the luminance L of the expected output and is a direct function of the illumination. This fourth input neuron prevents the mapping to be learnt by the MLP from including one-to-many correspondences (the different corrected colors corresponding to the same input color with different illuminations) and thus makes it solvable. If the MLP contains a bottleneck layer with 3 neurons, then these perform a re-parameterization of RGB space. However the three color parameters estimated by the 3 neurons (called here $p_1 p_2 p_3$) have no reason to be invariant to illumination.

To force the MLP to code color independently of illumination, the architecture of the traditional MLP is modified (see fig. 2). The entry point L of the MLP (fourth input neuron) is moved to the bottleneck layer of the network so that it becomes the third and last neuron of this layer. This displaced entry makes our MLP different from a trivial compression network. The two other neurons of the bottleneck layer have outputs (λ, μ). During training, the network learns to reconstruct the corrected color (R_d, G_d, B_d) from (λ, μ) and the desired output luminance $L_d = \frac{R_d+G_d+B_d}{3}$. Thus it learns to ignore the luminance of the input (R_i, G_i, B_i) and learns to estimate two color characteristics (λ, μ) that are invariant to illumination.

The approach does not require any camera calibration or knowledge about the image. However, it supposes that the illuminants are of the type commonly found in indoor environments.

3.2 Training the modified multi-layer perceptron

As shown in fig. 2, the modified MLP includes 5 layers (this could be generalized to an odd number of layers). The input and output layers have 3 neurons each (plus an additional bias), for RGB inputs and outputs. The middle layer includes 3 neurons (excluding bias): their outputs are called λ, μ and L. The second and fourth layers have arbitrary numbers of neurons (typically between 3 and 10 in our experiments). The links between neurons are associated to weights. Neurons have sigmoid activation functions. The network includes biases and moments [11].

A database of images showing the same scenes under different illuminations is used to train the modified MLP. The illuminations are typical of indoor environments such as home and office.

A classic MLP training scheme based on backpropagation is applied. A pixel is randomly sampled at each iteration from the training set. Its RGB values before and after an illumination change (from real images) are used as input (R_i, G_i, B_i) and desired output (R_d, G_d, B_d) to the network. Propagation and back-propagation are then performed, with one modification: as mentioned above, the output L of the third neuron of the third layer is forced to the value of the luminance corresponding to the desired output color.

3.3 Use of the modified multi-layer perceptron

The trained modified MLP can be used to correct color images. Each image pixel is propagated through the first half of the trained network to find the invariants λ and μ. An arbitrary luminance L is imposed on the pixel by forcing the output of the third neuron of the third layer to L. The output of the trained network then gives the corrected color. If a constant luminance L is used for all pixels in the image, an image corrected for shadows and for variations of illumination across the image and between images is obtained. The color correction can be tabulated for fast implementation. The approach could be easily extended to a greater number of inputs and outputs or different inputs/outputs than RGB. For instance, YUV or HSV, or redundant characteristics such as RGBYUVLab could be used as inputs and outputs.

4 Image correction results

4.1 Experimental conditions and database

The network was trained using 546000 pixels, randomly sampled from 91 training images (6000 pixels per image), taken by 2 webcams (Philips ToUCam Pro Camera and Logitech QuickCam Zoom). The training images are of indoor scenes viewed under different illuminations typical of home and office environments. Testing was performed on other images taken by the 2 webcams used for training and by a third webcam, not used for training, a Logitech QuickCam for Notebooks Pro.

In practice, using 8 neurons in the second and fourth layers of the MLP gives good performance. A gain of 1.0 was used, with a momentum factor of 0.01 and a learning rate of 0.001. Pixels that were too dark (luminance $\leq$ 20) or too bright / saturated (luminance $\geq$ 250) were not used for training.

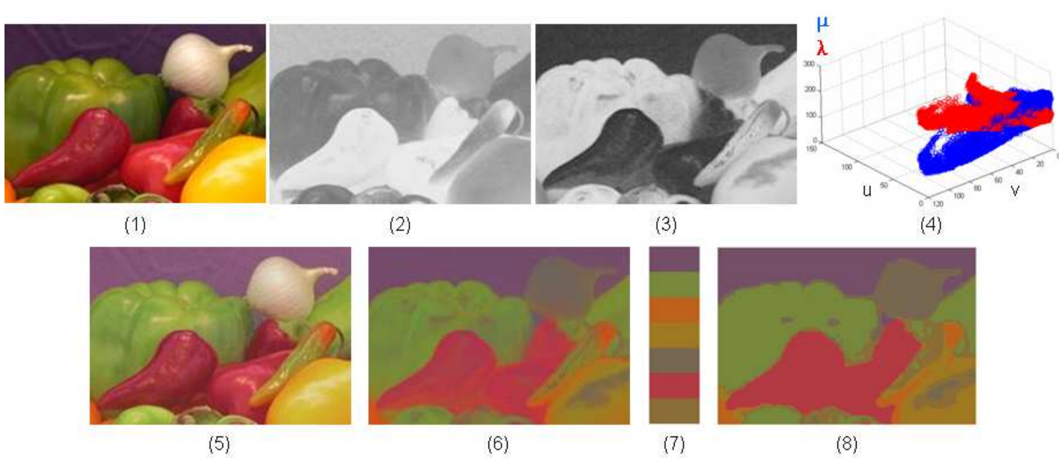

Fig. 3. Example of color correction learnt by the modified MLP. (1) original image (unknown illumination). (2) and (3) invariants λ and μ estimated by the MLP. (4) locus of the invariants in the uv space. (5) corrected image with pixel luminance inputs set to values proportional to pixel luminances in the original image (plus a constant). (6) corrected image with the pixel luminance inputs set to a constant value for all pixels. (7) 7 color peaks found by mean shift [12] in the corrected image (6). (8) resulting image segmentation.

4.2 Comparison with a "classical" multi-layer perceptron

Table 2 shows that the modified MLP (fig. 2) performs better in reconstructing target images than a classic MLP (fig. 1). The reconstruction is done given the expected luminances L_d of the pixels of the desired target image.

Table 2. Mean error between reconstructed and target images for a "classical" MLP and the modified MLP presented in this article. The mean error was calculated using 748 320x240 test images (not in the training set). The error is averaged over the three color components (R,G,B).

	for a classical MLP	for the modified MLP
mean error (in pixel values $\in [0, 255]$)	10.47	5.54
relative mean error	4.11%	2.17 %

4.3 Invariant estimation by the modified MLP

Figure 3 shows the two invariants (λ, μ) learnt by the modified MLP and calculated on an image (see part (1) of fig 3) of unknown illumination. The two invariants are seen in parts (2) and (3) of the figure. Objects of similar color to the human eye have similar values of λ and μ. Part (4) of fig. 3 shows the locus of the invariant values (λ, μ) in the image as a fonction of the chrominance values (u, v) (from YUV color space) of the image pixels. The locii of the two invariants are not identical, and thus we have two invariants and not only one. Part (6) of figure 3 shows the corrected image estimated for a constant luminance input over the image. Much of the influence of shading and variations of illumination across the image is removed, apart from specularities (white saturated areas) which are mapped to gray by the network. Areas of similar color in the original image (despite shading and illumination) have much more homogeneous color in the corrected image. This is further shown by performing mean-shift based color segmentation [12] on the corrected image. Seven areas of uniform color are readily identified and segmented (see part (7) and (8) of fig. 3) in the corrected image. They correspond roughly to what is expected by a human observer. This example illustrates that our modified MLP successfully learns a parameterization of color by two parameters that are invariant to illumination.

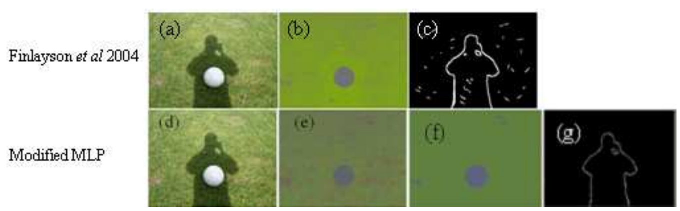

Fig. 4. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and the whole-image color correction method of Finlayson *et al* [5]. Application to shadow detection. Example I. (a) and (d) show the original image. (b) is the invariant image obtained using the method of [5] and (c) shows the shadow edges estimated from (b). (e) shows the corrected image estimated using the modified MLP, (f) and (g) the results of mean shift color segmentation [12] from (e) and (h) the shadow edges estimated from (g).

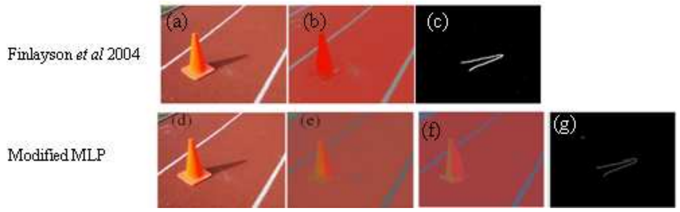

Fig. 5. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and the whole-image color correction method of Finlayson *et al* [5]. Application to shadow detection. Example II. (a), (b), (c), (d), (e), (f), (g) and (h) illustrate the same steps as in fig. 4.

4.4 Comparison with other color correction methods from the literature

Figures 6, 4 and 5 compare our color correction approach with other color correction approaches.

Figure 6 compares our approach to HSV-based color correction and applies it to color-based background subtraction. The two first images of the first and second columns of the figure show that our color correction scheme is indeed robust to changes in illumination, since there is much less difference between the images after correction than before. Figure 6 also shows that the correction performed in this paper compares favorably with an HSV-based color correction (which consists in taking an RGB color to hue-saturation-value space, setting its value/luminance to a constant, then going back to RGB space to get the corrected color).

Figures 4 and 5 illustrate that our correction is of similar quality to that of Finlayson et al [5] (briefly described in the introduction of this paper). The application of color correction is the detection of shadow contours (which can be used for shadow removal, as shown in [5]). Even though it might be less robust to large light changes or unusual light changes (such as turning on a blue or red light), our method is faster, being pixel-wise.

4.5 Performance of a LUT implementation of the trained modified MLP

Color correction by the modified MLP can be tabulated, making it one the fastest possible color correction approaches. Execution time using LUTs is 3.75 ms for an entire 320x240 image, on a Pentium4 3GHz. This way, color correction can be used as a first step in video-rate image processing, without using a large part of the frame processing time (40ms). This LUT implementation is possible because the approach is pixel-wise.

An HSV correction scheme could be as fast (using LUTs), but it would be less performant, as illustrated by fig. 6. A color correction scheme based on [5] would be of equal performance, as illustrated on examples by fig. 4 and 5. It could deal with more changes of illumination, since our approach is limited to the type of frequently found indoor lighting the modified MLP was trained for. However, working globally on the image, it could not be implemented as a LUT, and would thus be slower. The approach of [7] (briefly described in section 2), which performs good-quality color enhancement

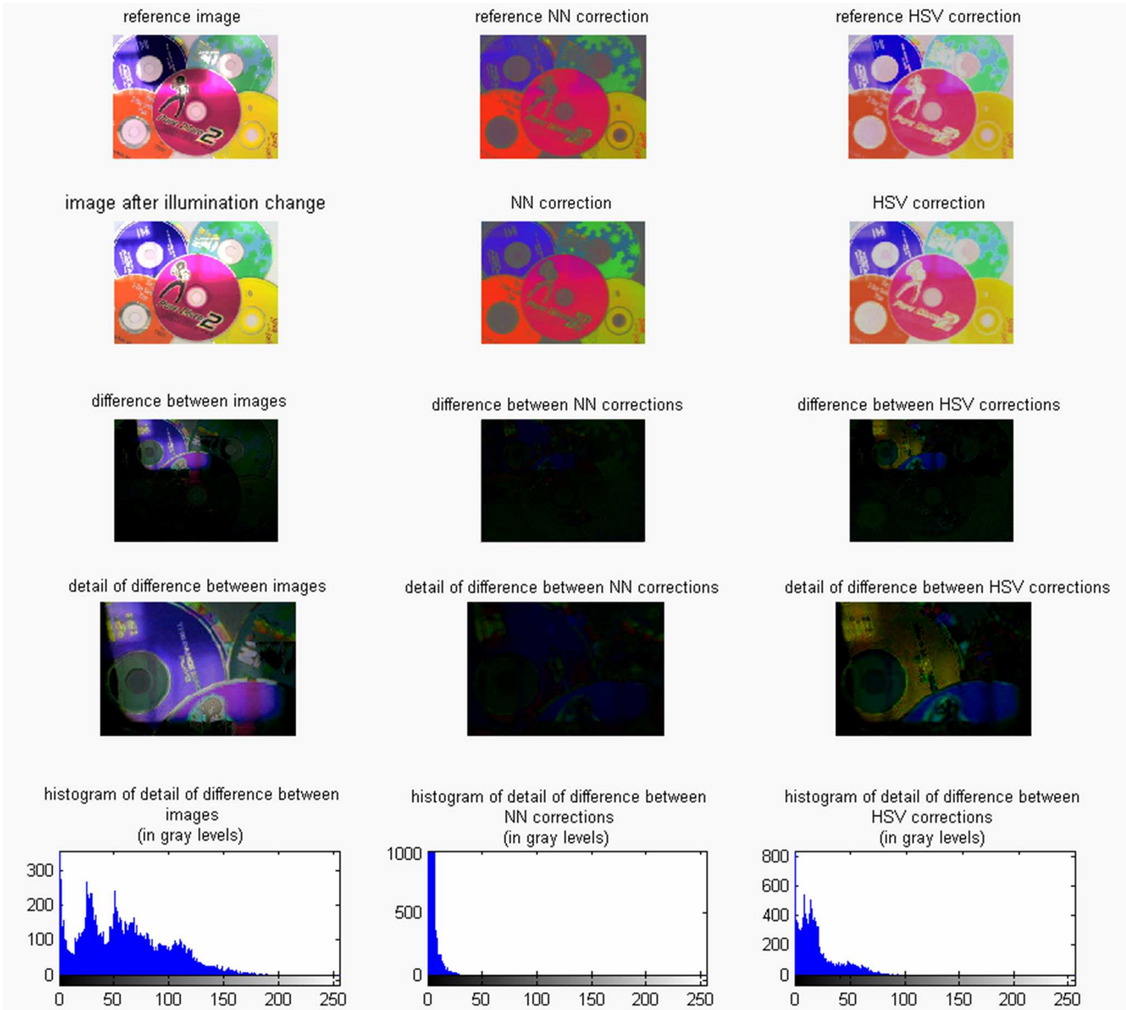

Fig. 6. Comparison of the pixel-wise color correction by the modified MLP presented in this paper and pixel-wise HSV-based color correction, HSV being the well known hue-saturation-value color space.

at good speed, is slower than our approach (3 frames per second on a Pentium4 2.26GHz for a 640x480 image).

5 Conclusion

This paper presents a new neural network-based approach to estimating image color independently of illumination. A modifi ed multi-layer perceptron is trained to estimate two color invariants and an illumination- corrected color for each input color. The network is trained for typical indoor home and offi ce lighting (fluorescents and light bulbs) and outdoor natural light, using two webcams. Such statistical training gives the approach a good compromise between generality (being able to handle different types of illuminants) and discrimination power (being able to discriminate between different colors). Experiments with lighting changes and another webcam show that the training seems to have good generalization properties. Once learning has been achieved, color correction is very fast using look-up tables, so that color correction can be performed as a part of image pre-processing before applying other image processing algorithms (such as background subtraction or color-based image segmentation).

References

1. Funt, B., Cardei, V., Barnard, K.: Neural network colour constancy and specularly reflecting surfaces. In: Proc. of AIC Color 97, Kyoto, Japan. (1997)
2. Gevers, T., Smeulders, A.: Color based object recognition. Pattern Recognition (1999)
3. Gonzalez, R., Woods, R.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc. (2001)
4. Reinhard, E., Ashikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer Graphics and Applications (2001)
5. Finlayson, G., Drew, M., Lu, C.: Intrinsic images by entropy minimization. In: Proc. 8th European Conf. on Computer Vision (ECCV'04), Prague, pp 582-595. (2004)
6. Funt, B., Jiang, H.: Non-diagonal colour correction. Proc. International Conference on Image Processing (ICIP 2003), Barcelona (2003)
7. Tao, L., Asari, V.: Modified luminance based msr for fast and efficient image enhancement. Proc. of 32nd Applied Imagery Pattern Recognition Workshop (AIPR'03) (2003)
8. Yin, J., Cooperstock, J.: Color correction methods with applications to digital projection environments. Journal of the Winter School of Computer Graphics (2004)
9. Barnard, K., Martin, L., Coath, A., Funt, B.: A comparison of computational color constancy algorithms, part 2; experiments with images. IEEE Transactions on Image Processing (2002)
10. Land, E., McCann, J.: Lightness and retinex theory. J. Opt. Soc. Am. (1971)
11. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press (1996)
12. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. of IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR 2000). (2000)
13. Tappen, M., Freeman, W., Adelson, E.: Recovering intrinsic images from a single image. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. (2005)
14. Luo, Q., Khoshgoftaar, T.: Efficient image segmentation by mean shift clustering and MDL-guided region merging. In: Proc. of 16th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'04). (2004)
15. Gu, I.H., Gui, V.: Colour image segmentation using adaptive mean shift filters. In: Proc. of . Int. Conference on Image Processing (ICIP'01). (2001)
16. Tominaga, S.: Color coordinate conversion via neural networks. Proc. of International Conf. on Colour Imaging in Multimedia (CIM'98) (1998)
17. Vrhel, M., Trussell, H.: Color scanner calibration via a neural network. Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 99) (1999)
18. Rosenberg, C., Minka, T., Ladsariya, A.: Bayesian color constancy with non-gaussian models (NIPS 2003)