Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

J. Vis. Commun. Image R. 14 (2003) 369–388

JOURNAL OF VISUAL Communication & IMAGE Representation

www.elsevier.com/locate/yjvci

Reduced complexity Retinex algorithm via the variational approach ☆

M. Elad,[a,*] R. Kimmel,[b] D. Shaked,[c] and R. Keshet[c]

[a] *Computer Science Department, Stanford University, Stanford, CA 94305, USA*
[b] *Computer Science Department, Technion–Israel Institute of Technology, Haifa 32000, Israel*
[c] *Hewlett-Packard Laboratories, Israel*

Received 10 December 2002; accepted 6 June 2003

Abstract

Retinex theory addresses the problem of separating the illumination from the reflectance in a given image, and thereby compensating for non-uniform lighting. In a previous paper (Kimmel et al., 2003), a variational model for the Retinex problem was introduced. This model was shown to unify previous methods, leading to a new illumination estimation algorithm. The main drawback with the above approach is its numerical implementation. The computational complexity of the illumination reconstruction algorithm is relatively high, since in the obtained Quadratic Programming (QP) problem, the whole image is the unknown. In addition, the process requirements for obtaining the optimal solution are not chosen a priori based on hardware/software constraints. In this paper we propose a way to compromise between the full fledged solution of the theoretical model, and a variety of efficient yet limited computational methods for which we develop optimal solutions. For computational methods parameterized linearly by a small set of free parameters, it is shown that a reduced size QP problem is obtained with a unique solution. Several special cases of this general solution are presented and analyzed: a Look-Up-Table (LUT), linear or nonlinear Volterra filters, and expansion using a truncated set of basis functions. The proposed solutions are sub-optimal compared to the original Retinex algorithm, yet their numerical implementations are much more efficient. Results indicate that the proposed methodology can enhance images for a reduced computational effort.

Keywords: Retinex; Illumination; Quadratic programming; Look-Up-Table; Volterra filters; Gamma correction

☆ This research was carried out in Hewlett-Packard laboratories, Israel.
* Corresponding author.
E-mail address: elad@sccm.stanford.edu (M. Elad).

1047-3203/$ - see front matter
doi:10.1016/S1047-3203(03)00045-2

1. Introduction

Retinex theory deals with compensation for illumination effects in images. The primary goal is to decompose a given image S into two different images, the reflectance image R, and the illumination image L, such that at each point (x, y) in the image domain $S(x, y) = R(x, y) \cdot L(x, y)$. The benefits of such a decomposition include the ability to remove illumination effects of back/front lighting, enhance photos that include spatially varying illumination such as images that contain indoor and outdoor zones, and correct the colors in images by removing illumination induced color shifts.

Recovering the illumination from a given image is known to be a mathematically ill-posed problem. In order to alleviate this problem, additional assumptions on the unknowns are required. The most commonly used assumption is that the spatially smooth parts of S originate from the illumination image, whereas edges in S are due to the reflectance in the image (Blake, 1985; Brainard and Wandell, 1986; Faugferas, 1979; Horn, 1974; Jobson et al., 1997a,b; Land, 1977, 1983, 1986; Land and McCann, 1971; PersComm, 1998; Stockham, 1972; Terzopoulos, 1986).

In a previous paper (Kimmel et al., 2003), a new variational based Retinex formulation to the Retinex problem was introduced and compared to other state-of-the-art methods. This formulation takes into account the illumination smoothness assumption. In addition, it exploits the known limited range of the reflectance image, and the fact that this image, being the process output, should be visually pleasing. The new formulation is shown to have a Quadratic Programming structure, which guarantees an existing unique solution. It is also shown that different previous Retinex algorithms are essentially solutions to similar variational problems.

One important drawback with the new variational approach is its numerical implementation. The unknown to be recovered in the obtained QP optimization problem is the reflectance or the illumination image. Thus, the number of unknowns is the number of pixels in the treated image, which is typically a very large number. Solving such a problem requires an iterative algorithm, where each iteration includes both radiometric and spatial operations. Such a process is known to be computationally demanding, even if efficient QP solvers, as the one proposed in Kimmel et al. (2003), are used.

Another problem with the above formulation is that the induced numerical process for obtaining the optimal solution is not constrained by software/hardware considerations. For example, in case where the illumination reconstruction system is restricted to a linear filter of pre-specified size, followed by a general LUT (Look-Up-Table) operation, the algorithm cannot take this constraint into account in the reconstruction process.

In this paper we propose to exploit the same variational formulation in order to define an optimal system with a pre-specified structure. A general framework for such a solution is constructed for general processes controlled by a limited set of parameters. The Retinex problem in this case translates into a search for optimal values for these small number of parameters.

For structures controlled linearly by a set of free parameters, we show that a reduced size QP problem is obtained, which guarantees a unique solution. Several special cases of this general solution are presented and analyzed:

1. Expressing the unknown illumination by a truncated set of basis functions.
2. A general linear filter with a pre-specified kernel size.
3. A general nonlinear Volterra filter with pre-specified kernel size and order.
4. A full or polynomial Look-Up-Table (LUT).
5. A Gamma-correction process.

For all these cases, the resulting solution is sub-optimal compared to the original Retinex method, yet the numerical implementations are more efficient. Nevertheless, we show that these solutions succeed in enhancing the input image for the above choices.

This paper proposes a novel approach to tuning an image processing algorithm. We solve a reduced size QP problem which optimizes for the parameters of a hardware implementation or alternatively an efficient computational algorithm that will actually perform the image processing efficiently. Other efficient algorithms for adaptive image processing exist, however they are usually not a result of a rigorous problem formulation and optimization. One example of an efficient adaptive image enhancement is tone-mapping algorithm by Holm (1996) which proposes to tailor a tone-map to an input image according to parameters extracted from a histogram of a thumbnail of an image (rather than the histogram of the full image). Another example is an efficient algorithm, which much like the Retinex reduces the dynamic rage of images (Durand and Dorsey, 2002). The algorithm is formulated as a nonlinear filter on the full image. However, it is implemented as an interpolation between a set of convolutions of down-sampled images. This is, indeed, a significant algorithmic improvement aimed specifically at efficient convolution hardware/software. It is, however, different from the method proposed in this paper in that it is specific to a unique computational tool (linear convolutions) whereas we propose a generic method to determine parameters in a large class of efficient implementations. Furthermore, we determine those parameters via optimization with respect to a goal formulated, in our case, as a dynamic range compression problem.

This paper is organized as follows: In the next section, we briefly present the variational Retinex formulation, as presented in Kimmel et al. (2003). Section 3 shows the proposed method for extracting sub-optimal solutions to the exact Retinex problem, while preserving the QP structure. In Section 4, we discuss the properties of the new reduced size QP problem. Results are given in Section 5, with concluding remarks in Section 6.

2. The variational Retinex formulation

Our starting point is the Retinex formulation as presented in Kimmel et al. (2003). The variational formulation for Retinex relies on the following assumptions (Kimmel et al., 2003):

1. The illumination is spatially smooth.
2. The reflectance image, R, is restricted to the unit interval ($0 \leqslant R \leqslant 1$), and therefore, $L \geqslant S = L \cdot R$.

3. The illumination image is close to the input image, S, enhancing the local contrast of the reflectance image R.
4. The reflectance image R is likely to have a high prior probability (Blake and Zisserman, 1987; Geman and Geman, 1984; Lagendijk and Biemond, 1991; Marroquin et al., 1987). One of the simplest prior functions used for natural images assigns high probability to spatially smooth images (Lagendijk and Biemond, 1991).

Define $s = \log(S)$, $r = \log(R)$, and $l = \log(l)$. If we integrate all the above assumptions into one expression we get the following penalty functional:

$$\begin{aligned} &\text{Minimize}: \quad F[l] = \int_{\Omega} (|\nabla l|^2 + \alpha (l-s)^2 + \beta |\nabla (l-s)|^2)\, \mathrm{d}x\, \mathrm{d}y \\ &\text{Subject to}: \quad l \geqslant s, \end{aligned} \tag{1}$$

where Ω is the support of the image. α and β are free non-negative real parameters. In the functional $F[l]$, the first penalty term ($|\nabla l|^2$) forces spatial smoothness on the illumination image. The second penalty term $(l-s)^2$ forces a proximity between l and s. The third term is the penalty expression for the prior. This term forces r to be spatially smooth. Note that more complicated prior penalty expressions may be used allowing for sharp edges, textures, $1/f$ behavior, etc. (Blake and Zisserman, 1987; Geman and Geman, 1984; Lagendijk and Biemond, 1991; Marroquin et al., 1987). As long as this expression is purely quadratic, the above minimization problem remains fairly simple.

Since the numerical implementation is applied on sampled images, we can rewrite the above problem using discrete notations. As we shall see in the next section, a discrete representation lends itself to the definition of optimal pre-specified system structure.

Let us define the vectors $\underline{l}$, $\underline{r}$, and $\underline{s}$ as the column-stack lexicographic ordering of the illumination, reflectance, and original images, respectively. The matrices D_x and D_y stand for a horizontal and vertical discrete first derivative operations. Thus, the variational problem transforms into

$$\begin{aligned} &\text{Minimize}: \quad F[\underline{l}] = \|D_x \underline{l}\|^2 + \|D_y \underline{l}\|^2 + \alpha \|\underline{l} - \underline{s}\|^2 \\ &\qquad\qquad\qquad + \beta(\|D_x(\underline{l} - \underline{s})\|^2 + \|D_y(\underline{l} - \underline{s})\|^2) \\ &\text{Subject to}: \quad \underline{l} \geqslant \underline{s}. \end{aligned} \tag{2}$$

The above problem (in both representations) has a Quadratic Programming (QP) form (Bertsekas, 1995; Luenberger, 1987). In Kimmel et al. (2003) it was shown that the Hessian of the function $F[\underline{l}]$ is positive definite if $\alpha > 0$. As such, this problem is strictly convex and has a unique solution.

An interesting interpretation of the above functional is obtained for the case where $\beta \gg \alpha$. As it turns out, such a choice for the parameters leads to a Gamma-correction solution. More details about this anomaly can be found in Appendix A.

3. General simplification for the Retinex problem

Solving for the optimal illumination image $\underline{l}$, as defined by Eq. (2), requires an iterative algorithm.

Assume that due to hardware or computational limitations, there is a pre-defined procedure P that we are willing to apply on $\underline{s}$ in order to get $\hat{\underline{l}}$. We further assume that this operation is governed by a relatively small number, $N_{\underline{\theta}}$, of parameters denoted by $\underline{\theta}$. Thus, $\hat{\underline{l}} = P\{\underline{\theta}, \underline{s}\}$.

In order to get a good quality estimate $\hat{\underline{l}}$, which will imitate the solution of the exact variational Retinex, we define the optimal parameter set $\underline{\theta}$ as the solution of the problem:

$$\begin{aligned}\text{Minimize :} \quad & F_\theta[\underline{\theta}] = F[P\{\underline{\theta},\underline{s}\}] = \|D_x P\{\underline{\theta},\underline{s}\}\|^2 + \|D_y P\{\underline{\theta},\underline{s}\}\|^2 \\ & \quad + \alpha\|P\{\underline{\theta},\underline{s}\} - \underline{s}\|^2 + \beta(\|D_x(P\{\underline{\theta},\underline{s}\} - \underline{s})\|^2 \\ & \quad + \|D_y(P\{\underline{\theta},\underline{s}\} - \underline{s})\|^2) \\ \text{Subject to :} \quad & P\{\underline{\theta},\underline{s}\} \geqslant \underline{s}.\end{aligned} \tag{3}$$

If the operation P is linear with respect to the parameter set $\underline{\theta}$, it can be rewritten as $P\{\underline{\theta},\underline{s}\} = M\{\underline{s}\} \cdot \underline{\theta}$, where $M\{\underline{s}\}$ is a matrix of size $[L_x \times L_y \times N_{\underline{\theta}}]$, with $L_x \times L_y$ the size of the image. $M\{\underline{s}\}$ is a possibly nonlinear function matrix of the image $\underline{s}$. This special case is important since then Eq. (3) becomes

$$\begin{aligned}\text{Minimize :} \quad & F_\theta[\underline{\theta}] = \|D_x M\{\underline{s}\}\underline{\theta}\|^2 + \|D_y M\{\underline{s}\}\underline{\theta}\|^2 + \alpha\|M\{\underline{s}\}\underline{\theta} - \underline{s}\|^2 \\ & \quad + \beta(\|D_x(M\{\underline{s}\}\underline{\theta} - \underline{s})\|^2 + \|D_y(M\{\underline{s}\}\underline{\theta} - \underline{s})\|^2) \\ \text{Subject to :} \quad & M\{\underline{s}\}\underline{\theta} \geqslant \underline{s}\end{aligned} \tag{4}$$

and this problem has again a Quadratic Programming form. In order to assure that the function $F_\theta[\underline{\theta}]$ is strictly convex, we have to verify that the Hessian of $F_\theta[\underline{\theta}]$ is positive definite (Bertsekas, 1995). The Hessian is given by

$$\frac{\partial^2 F_\theta[\underline{\theta}]}{\partial \underline{\theta}^2} = M^{\mathrm{T}}\{\underline{s}\}\left[\alpha I + (1+\beta)\left(D_x^{\mathrm{T}} D_x + D_y^{\mathrm{T}} D_y\right)\right] M\{\underline{s}\}. \tag{5}$$

The term $D_x^{\mathrm{T}} D_x + D_y^{\mathrm{T}} D_y$ is the Laplacian operator (Kimmel et al., 2003). Thus, if $\alpha > 0$ and $M\{\underline{s}\}$ is full rank (meaning that its columns are linearly independent), then the Hessian is positive definite, the functional is strictly convex, and there is a unique solution.

Let us explore several possibilities for the construction of the matrix M:

1. *Basis functions*. Since the illumination is known to be spatially smooth, it can be spanned by relatively small number of smooth basis functions. One such possibility is to use a truncated Fourier basis. Each such basis function is a complete image of size $L_x \times L_y$, ordered lexicographically into a single column in the matrix M. Note that in this case, M is not a function of $\underline{s}$.
2. *Linear filter*. In some situations we might be forced to use a linear space invariant filter, using a $(2K+1) \times (2K+1)$ kernel, for the construction of $\hat{\underline{l}}$. Let us define a global displacement operation $\mathrm{Disp}_{[i,j]}\{\underline{s}\}$, which displaces image $\underline{s}$ by $[i, j]$ (in the

two axes). This definition must assume a specific boundary condition (e.g. for the condition mentioned earlier, one shift left causes the rightmost column of the image to be replicated and represent the new entering column from the right). Thus, M is built by

$$M\{\underline{s}\} = \left[\mathrm{Disp}_{[-K,-K]}\{\underline{s}\}, \ldots, \mathrm{Disp}_{[0,0]}\{\underline{s}\}, \ldots, \mathrm{Disp}_{[K,K]}\{\underline{s}\}\right],$$

i.e., by displacing $\underline{s}$ to all possible positions in a block of $(2K+1) \times (2K+1)$. A linear combination of these columns is a simple linear space invariant convolution, as required.

A bias can be added to the linear filter by adding one more column to the matrix $M\{\underline{s}\}$, containing ones. This way, each pixel in the estimated illumination image is created by a weighted average of the local neighborhood, added to a prespecified optimal bias value.

3. *Full Look-Up-Table*. One of the simplest, and therefore, computationally appealing, possible operations on an image is a Look-Up-Table (LUT). In the general case, LUT is a map that assigns an output value to each input gray-value. Assuming an 8 bit input, the 256 output values are the parameters of this operation. It is not trivial to see how a LUT operation falls into the linear structure $M\{\underline{s}\}\underline{\theta}$. To see that, let us define an indicator operation $\mathrm{Ind}_v\{\underline{s}\}$ as

$$\mathrm{Ind}_v\{\underline{s}\} = \begin{cases} 1, & s[i,j] = v, \\ 0, & s[i,j] \neq v, \end{cases}$$

i.e., all pixels in the image which are equal to v are set to 1, and the remaining pixels are zeroed. Using this operation, the LUT operation can be modeled as the multiplication of the following sparse matrix:

$$M\{\underline{s}\} = [\mathrm{Ind}_0\{\underline{s}\}, \mathrm{Ind}_1\{\underline{s}\}, \ldots, \mathrm{Ind}_{255}\{\underline{s}\}]$$

by a vector $\underline{\theta}$ representing the 256 output gray-values. In this case, M uses $\underline{s}$ in a nonlinear manner.

4. *Polynomial Look-Up-Table*. If 256 unknowns are hard to get, or if the desired LUT should be smooth, a polynomial approximation of it can be used instead. The operation on the input image $\underline{s}$ will be

$$\hat{l}[i,j] = \sum_{k=0}^{N-1} \theta_k s^k[i,j]. \tag{6}$$

In this case, the unknown vector will be the coefficients of the above sum, and the matrix M is represented by

$$M\{\underline{s}\} = \left[\underline{s}^0, \underline{s}^1, \ldots, \underline{s}^{N-1}\right],$$

where the operation $\underline{s}^k$ is applied per entry.

5. *Volterra filtering*. A possible nonlinear extension to the linear filter is the Volterra filer. Instead of linearly weighting gray values of a neighborhood, Volterra filter proposes linear weight of polynomials of the gray values. The resulting matrix M turns out to combine the mathematical machinery of both the linear filter and the

polynomial LUT approximation. For example, for a simple 2-pixels spatial Volterra operation of up to second -order polynomials we get

$$M\{\underline{s}\} = \Big[\mathrm{Disp}_{[1,0]}\{\underline{s}\}, \mathrm{Disp}_{[0,0]}\{\underline{s}\}, \mathrm{Disp}_{[0,0]}\{\underline{s}\}^0, \ldots, \mathrm{Disp}_{[0,0]}\{\underline{s}\}\mathrm{Disp}_{[1,0]}\{\underline{s}\}, \mathrm{Disp}_{[0,0]}\{\underline{s}\}^2, \mathrm{Disp}_{[1,0]}\{\underline{s}\}^2\Big],$$

where $\mathrm{Disp}_{[0,0]}\{\underline{s}\}^0$ is a vector of ones.

Other possibilities can be formulated using this approach, and in particular, a combination of the above options is also possible.

It is important to note that while choosing a linear structure of the form $P\{\underline{\theta}, \underline{s}\} = M\{\underline{s}\} \cdot \underline{\theta}$ is limiting, we see that it leads to a diverse set of options, commonly used as processing-blocks in image processing systems (especially hardware ones). Thus, not only we have gained some simplification with respect to the computational complexity due to this choice, but we also gain the ability to perform the Retinex-correction process in hardware with optimized parameters of linear filtering, LUT, and more.

We also note that the approach taken above (both the general and the subsequent linear) can be posed as the original variational Retinex method as posed in (2) with an additional constraint of the form $\hat{\underline{l}} = P\{\underline{\theta}, \underline{s}\}$. Thus this additional constraint limits the solution space and therefore yields suboptimal result. As we shall see next, this loss in output quality comes with a gain in stability speed of implementation. The limited solution space could be interpreted as a variant of regularization that stabilizes the problem and its numerical solution.

4. Properties of the reduced Retinex problem

In all the above options for the choice of $M\{\underline{s}\}$, the QP problem becomes

$$\begin{aligned} &\text{Minimize}: && F[\underline{\theta}] = \frac{1}{2}\underline{\theta}^{\mathrm{T}}\mathbf{H}\{\underline{s}\}\underline{\theta} + \underline{\theta}^{\mathrm{T}}\underline{Q}\{\underline{s}\} + \mathrm{Const} \\ &\text{Subject to}: && M\{\underline{s}\}\underline{\theta} \geqslant \underline{s}, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \mathbf{H}\{\underline{s}\} &= M^{\mathrm{T}}\{\underline{s}\}\Big[\alpha I + (1+\beta)\Big(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\Big)\Big]M\{\underline{s}\}, \\ \underline{Q}\{\underline{s}\} &= -2M^{\mathrm{T}}\{\underline{s}\}\Big[\beta\Big(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\Big) + \alpha I\Big]\underline{s}, \\ \mathrm{Const} &= \underline{s}^{\mathrm{T}}\Big[\beta\Big(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\Big) + \alpha I\Big]\underline{s} \end{aligned} \tag{8}$$

and both $\mathbf{H}$ and $\underline{Q}$ are relatively small. For a typical problem treating images of size 1000×1000 pixels, the original Retinex procedure requires the recovery of 1e6 unknown pixels and the Hessian of the QP is of size 1e6 $\times$ 1e6 entries. Going to the new approach with 1000 parameters (a reasonable number for the options discussed in the previous section), the number of unknowns is 1000 and the Hessian is of size

1000×1000 entries. Moreover, for the $M(\underline{s})$ proposed here, the condition-number of the new Hessian is far better than the original one, leading to a better numerical stability.

In order to solve the above QP problem, we need to compute $\mathbf{H}\{\underline{s}\}$, $\underline{Q}\{\underline{s}\}$, and the set of constraints. The constant value appearing in the penalty function has no impact on the solution and thus can be omitted. Here are some comments about this computational process.

For the computation of $\mathbf{H}$, we have to apply a linear operator

$$\left[\alpha I + (1+\beta)\left(D_x^{\mathrm{T}} D_x + D_y^{\mathrm{T}} D_y\right)\right] \tag{9}$$

on each of the images representing the columns of $M^{\mathrm{T}}\{\underline{s}\}$. The final stage is an application of an inner product between these images and the images in $M\{\underline{s}\}$. If the columns of $M\{\underline{s}\}$ are obtained by pure global displacement (as in the linear filter case), then, instead of computing the filtering results per row, we can simply generate them by displacing the filtered result for the center column.

For the case where $M\{\underline{s}\}$ is built by basis functions, this matrix does not depend on $\underline{s}$. Therefore, the matrix $\mathbf{H}\{\underline{s}\}$ can be computed off-line once.

Computing $\underline{Q}$ is done by applying the filter $[\alpha I + \beta(D_x^{\mathrm{T}} D_x + D_y^{\mathrm{T}} D_y)]$ on the input image $\underline{s}$, and again, performing an inner product with the columns of $M^{\mathrm{T}}\{\underline{s}\}$. A second option is to use the matrix

$$\left[\alpha I + (1+\beta)\left(D_x^{\mathrm{T}} D_x + D_y^{\mathrm{T}} D_y\right)\right] M\{\underline{s}\}$$

as obtained from the computation of $\mathbf{H}$, and perform an inner product with the input image $\underline{s}$.

In general, the constraint set $M(\underline{s})\underline{\theta} \geqslant \underline{s}$ has $L_x \times L_y$ inequality constraints. This may become prohibitive, especially because we would like to avoid the actual storage of the matrix M.

As it turns out, a very effective shortcut can be used in order to prune the number of constraints for the LUT design in both the full and the polynomial approximation. In these two cases, the number of different constraints is smaller or equal to 256, since for all the pixels getting gray value v, all the corresponding constraints are identical. Pruning the constraints-set is done by first finding all the existing gray values in the image s, and then creating for each one of them a scalar inequality constraint. For example, if all (8 bits) gray values are occupied, then for the full LUT design we have the constraint

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdot \\ \cdot \\ \cdot \\ \theta_{255} \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 255 \end{bmatrix}.$$

This constraint requires the resulting LUT to be bounded from below by the unity LUT operation.

For the same case, the third-order polynomial LUT will have the constraint

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1^0 & 1^1 & 1^2 & 1^3 \\ 2^0 & 2 & 2^2 & 2^3 \\ . & . & . & . \\ . & . & . & . \\ 255^0 & 255^1 & 255^2 & 255^3 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 1 \\ . \\ . \\ . \\ 255 \end{bmatrix}.$$

For the other cases where such simple pruning is not possible, an interesting problem is how to efficiently prune redundant constraints from such a set, either as an accurate or as an approximated process. This problem is left for future research. In our simulations we exploited the fact that constraints for neighboring pixels are expected to be similar. Therefore, we simply decimated the constraints list.

Another interesting option with respect to the LUT-based approaches is to enforce monotonicity on the results. If desired, this property can be forced as a set of additional inequality constraints. Monotonicity is guaranteed if the first derivative of the obtained LUT is non-negative. For the complete LUT design, this requirement is formulated by

$$\begin{bmatrix} -1 & 1 & 0 & . & . & . & 0 \\ 0 & -1 & 1 & 0 & . & . & 0 \\ 0 & 0 & -1 & 1 & 0 & . & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & 0 & -1 & 1 & . \\ 0 & 0 & . & . & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ . \\ \theta_{255} \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}.$$

In the polynomial approach we require that the derivative of (6) is non-negative for all θ.

$$\frac{\mathrm{d}\hat{l}}{\mathrm{d}s} = \sum_{k=1}^{N-1} k\theta_k s^{k-1} \geqslant 0,$$

which implies

$$\begin{bmatrix} 1\cdot 0^0 & 2\cdot 0^1 & 3\cdot 0^2 & . & . & (N-1)\cdot 0^{N-2} \\ 1\cdot 1^0 & 2\cdot 1^1 & 3\cdot 1^2 & . & . & (N-1)\cdot 1^{N-2} \\ 1\cdot 2^0 & 2\cdot 2^1 & 3\cdot 2^2 & . & . & (N-1)\cdot 2^{N-2} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 1\cdot 255^0 & 2\cdot 255^1 & 3\cdot 255^2 & . & . & (N-1)\cdot 255^{N-2} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ . \\ . \\ . \\ \theta_{N-1} \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \\ . \\ 0 \end{bmatrix}.$$

Assuming that the image $\underline{s}$ is given in the Log domain, Gamma correction is merely the multiplication of $\underline{s}$ by a constant value, $1/\gamma$. Thus, such a case turns out to be a special case of the polynomial approximated LUT, using a single column in the matrix M, i.e., $M\{\underline{s}\} = \underline{s}$. Note that in this case, the constraint is $\underline{s}\theta_0 \geqslant \underline{s}$, which is equivalent to $\theta_0 \geqslant 1$.

All the above refers to treating the image in the 8 bit gray-value domain. If the algorithms are to be applied in the Log domain, each input gray value x is replaced by $\log(1+x)/8$. Note that for the LUT operations, this change implies a change of entries in the constraint matrix.

As to the overall complexity of the resulting algorithm, it generally depends on the specific method used (essentially the choice of $M(\underline{s})$ and the number of the parameters in the assumed model). Generally speaking, the complexity obtained is of the order of one iteration of the original Retinex algorithm or below, and thus expected to be solved much faster.

5. Results

In this section we present several examples to demonstrate the quality of the proposed reduced complexity approach. Throughout this section we apply an RGB Retinex algorithm, i.e., each color layer is treated separately (Kimmel et al., 2003). For all the shown results we have used the parameters $\alpha = 0.01$, $\beta = 1\mathrm{e}-5$.

We start by showing the test images, and the full fledged variational-based Retinex algorithm results (Fig. 1) that serve as reference.

Next, Fig. 2 shows the results of a full LUT for the two input images. Fig. 3 shows the obtained LUT for the three color components. As can be seen, the look-up-tables are above the identity function, which means that the reflectance image turns out to be a brighter version of the input image, as expected.

Fig. 1. Full fledged variational-based Retinex results: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

Fig. 2. Optimal complete Look-Up-Table results: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

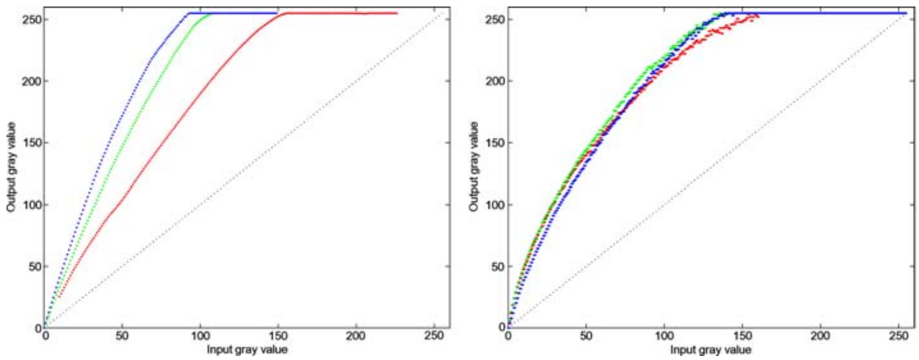

Fig. 3. Optimal complete Look-Up-Table results: (left) the LUT for the first image (“child”); (right) the LUT for the second image (“Houses”).

Note that when comparing the results of the full fledged Retinex to the full LUT in the first image, we get the impression that the LUT result is better. This is because Retinex algorithms in general attempt to recover the reflectance image, thereby enhancing contrast in the dark tones. In order to get a visually pleasing output, some of the illumination should be returned to the image (Kimmel et al., 2003). In these simulations the original Retinex succeeded better in recovering details, which would have resulted in a better visual quality if the obtained reflectance results were to be used in the appropriate enhancement algorithm (Kimmel et al., 2003).

Fig. 4 shows the results of a polynomial approximated LUT of sixth order. Fig. 5 shows the resulting LUT for the three color components. A close resemblance can be seen between these results and the ones obtained by the full LUT. Note that the LUT are not monotone for the first image, but the descending part of the LUT corresponds to gray values that do not exist in the image, and thus, the fact that the LUT is not monotone should not be disturbing.

Our last example in the family of LUT operations is Gamma correction, in which a single parameter is determined. Figs. 6 and 7 show the results and the resulting

Fig. 4. Optimal polynomial Look-Up-Table results: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

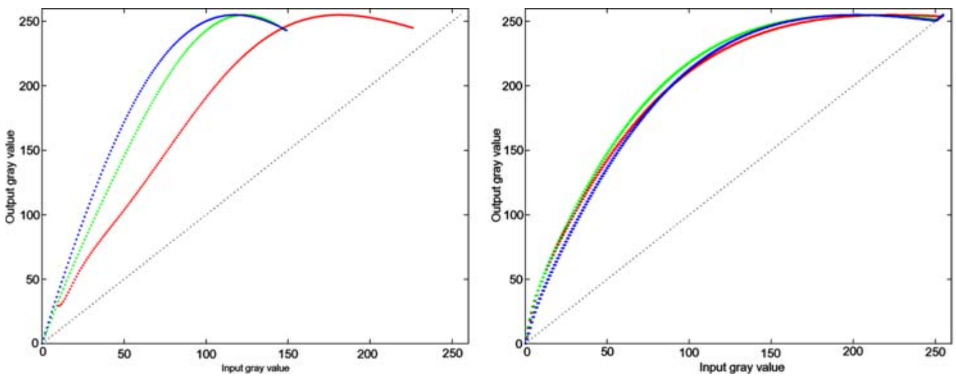

Fig. 5. Optimal polynomial Look-Up-Table results: (left) the LUT for the first image ("child"); (right) the LUT for the second image ("Houses").

Fig. 6. Optimal Gamma Look-Up-Table results: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

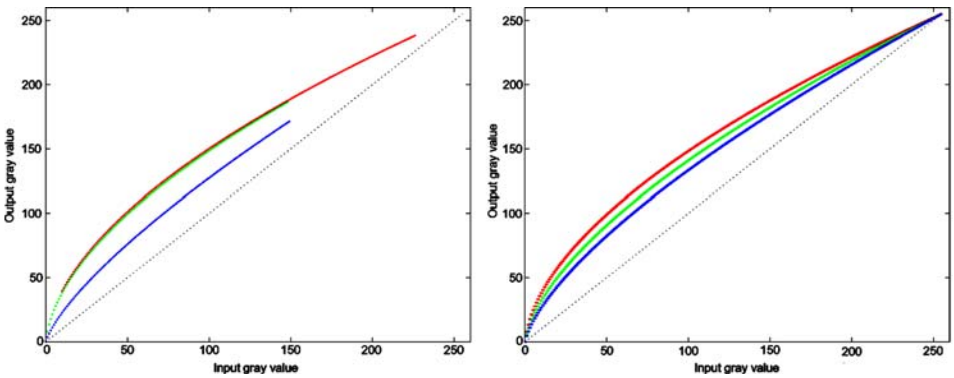

Fig. 7. Optimal Gamma Look-Up-Table results: (left) the LUT for the first image (“child”); (right) the LUT for the second image (“Houses”).

LUT. Again, we see that the results resemble the ones obtained the previous two methods.

Fig. 8 shows the results obtained for a 5×5 linear kernel, combined with a bias value. Fig. 9 shows the actual kernels for the three colors. Notice that the illumination image is obtained from the input image by blurring, and therefore, the reflectance image has a sharpening effect. In this simulation we chose to decimate the set of constraints by a factor 10:1, i.e., for every 10 constraints, only the first was

Fig. 8. Optimal linear filter + bias results: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

used. This way we simplified the QP solution, relaying on the expected spatial smoothness of these inequalities.

As a final example we show the results of the basis functions approach. We chose to use a two-dimensional DCT basis functions, using the L^2 functions taken from the square of $L \times L$ starting from the origin. In the following simulation $L = 5$. Fig. 10 shows the results obtained for this method.

All these simulations were done using Matlab v6.0, run on a Pentium-III Windows machine with 500 MHz processor and 200 MB RAM. The images described here are of size 256×256. The original Retinex algorithm built on multi-resolution solver approach (see (Kimmel et al., 2003) for more details) takes 16 s[1]—this is a highly efficient numerical scheme, implying that a regular iterative solver is expected to take much longer. The simplified parametric algorithm with a direct or a polynomial LUT requires 2 s, the Gamma LUT takes on the average 0.7 s, the linear + bias parameterization requires 1.5 s, and the DCT method requires 2.1 s. We should stress, however, that the proposed parametric methods are expected to be far more efficient when hardware, DSP, or even plain C-code implementation is considered. Also, when discussing running the same algorithm for a set of images we expect to see a further speedup since some preparation of the matrices involved could be done only once.

As a last point in this section we return to the claim made in Section 3 about the choice $\beta \gg \alpha$ causing the solution to become a pure Gamma correction. Appendix A gives an explanation for this property and Appendix B shows that indeed, for such a

[1] All run-times reported here are average ones.

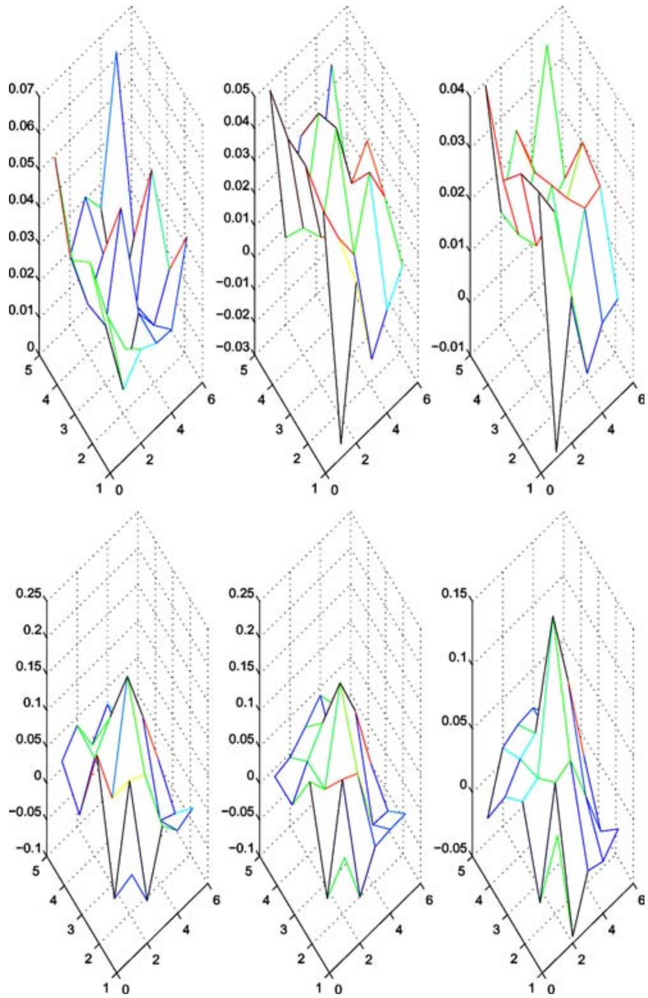

Fig. 9. Optimal linear filter + bias results: (Up) the kernels for the first image ("child"); (down) the kernels for the second image ("Houses").

choice of parameters, the above methods tend to give close to Gamma-correction results.

6. Concluding remarks

In this paper we presented several methods for reducing the complexity of the variational Retinex method, as described in Kimmel et al. (2003), while restricting the solution to have a pre-specified structure. Systems based on a Look-Up-Table, linear or nonlinear filtering, and expansion by arbitrary basis functions are shown to be

Fig. 10. Optimal combination of 2D DCT basis functions: (left) source image; (middle) estimated reflectance image; (right) estimated illumination image.

special cases of the proposed methodology. This way, instead of searching for an unknown illumination image, the newly defined problem focuses its search on a very small number of free parameters and thereby controlling the chosen system. It was shown that the proposed approach yields reasonable quality output with very efficient numerical implementations.

An alternative lesson from the obtained results is a characterization of the heuristics we want to employ in order to get a Retinex imitation. For example, using a LUT it was found (see Fig. 3) that simple dynamic range stretching, followed by Gamma correction is the effective solution. As another example, the linear filter approach (see Fig. 9) indicated that Retinex effect is obtained by sharpening, followed, again, by Gamma correction.

Appendix A. The Retinex algorithm for $\beta \gg \alpha$—theory

Let us look at the variational expression (Eq. (2)), and assume that $\beta \gg \alpha$. If we ignore the inequality constraints, the optimal image $\underline{l}$ should satisfy

$$\left[(1+\beta)\left(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right) + \alpha I\right]\hat{\underline{l}} = \left[\beta\left(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right) + \alpha I\right]\underline{s}. \tag{10}$$

Using the assumption $\beta \gg \alpha$ we get that

$$\hat{\underline{l}} \approx \left[(1+\beta)\left(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right)\right]^{-1}\left[\beta\left(D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right)\right]\underline{s} = \frac{\beta}{1+\beta}\underline{s},$$

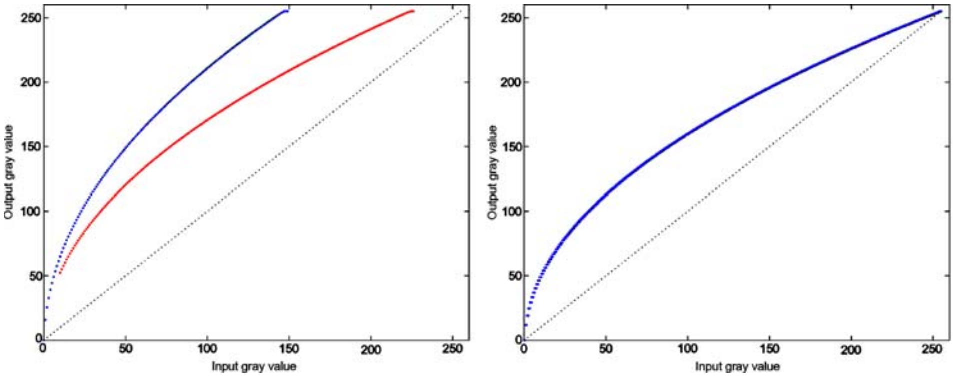

Fig. 11. Optimal LUT designed for $\beta \gg \alpha$: (left) the results as a LUT mapping for the image "Child"; (right) the results as a LUT mapping for the image "Houses."

where we have assumed that the Laplacian operator is invertible. Since this is not true, the term αI stands as an algebraic regularization to the inverted matrix.

An interesting property of this solution is that it also satisfies the constraints, and therefore, this is the solution for the original optimization problem. We should remember that the images are in the Log domain, and their values are negative.[2] Thus, multiplying by a positive fraction smaller than 1, we get that $(1+\beta)^{-1}\beta\underline{s}$ is indeed higher than $\underline{s}$, as required. This result implies that

$$\underline{\hat{r}} = \underline{s} - \underline{\hat{l}} \approx \frac{1}{1+\beta}\underline{s} \Rightarrow \underline{\hat{R}} = \exp\{\underline{\hat{r}}\} = \frac{\underline{S}}{\underline{\hat{L}}} \approx \underline{S}^{1/(1+\beta)},$$

which is exactly the Gamma-correction operation $\underline{S}^{\frac{1}{\gamma}}$. Therefore, if the required Gamma for a given image is known, it may give an indication as to the required value of the parameter β.

Note that in Eq. (10), adding a constant ϵ to the solution does not impact the correctness of this equation, since

$$\left[D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right]\left[\epsilon + \frac{\beta}{1+\beta}\underline{s}\right] = \left[D_x^{\mathrm{T}}D_x + D_y^{\mathrm{T}}D_y\right]\frac{\beta}{1+\beta}\underline{s}.$$

The constant ϵ must be such that it does not contradict the constraint

$$\epsilon + \frac{\beta}{1+\beta}\underline{s} \geqslant \underline{s} \rightarrow \epsilon \geqslant \frac{1}{1+\beta}\mathrm{Max}(\underline{s}) \geqslant \frac{1}{1+\beta}\underline{s}.$$

Among all the possible values of ϵ, preferred values are those which cause the illumination image to be as close as possible to the input image. This is true if α is not

[2] We assume that the input image is normalized to the range $[0, 1]$.

zero (even if it is very small). This implies that ϵ should be the smallest possible value which still satisfies the constraint. For the case where $\mathrm{Max}(\underline{s}) = 0$ (in the Log domain), we get that $\epsilon = 0$. For cases where $\underline{s}$ does not fill the entire dynamic range, ϵ can be chosen as a negative value that will improve the proximity between $\underline{s}$ and $\hat{\underline{l}}$. Therefore

$$\epsilon = \min\left\{\frac{1}{1+\beta}\mathrm{Max}(\underline{s}), 0\right\}.$$

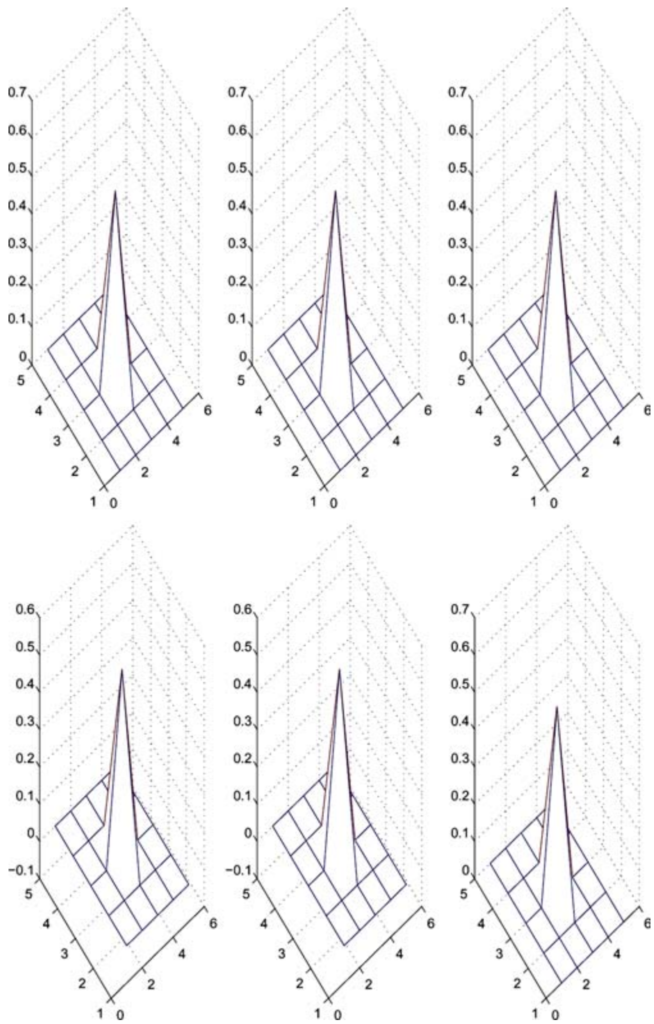

Fig. 12. Optimal linear filter + bias for $\beta \gg \alpha$: (Up) the obtained kernels for the image "Child"; (down) the obtained kernels for the image "Houses."

Effectively, the value of ϵ plays a role of stretching before the Gamma correction, since

$$\begin{aligned}\underline{\hat{r}} &= \underline{s} - \underline{\hat{l}} = -\epsilon + \frac{1}{1+\beta}\underline{s} = \frac{1}{1+\beta}\underline{s} - \min\left\{\frac{1}{1+\beta}\mathrm{Max}(\underline{s}), 0\right\} \\ &= \min\left\{\frac{1}{1+\beta}[\underline{s} - \mathrm{Max}(\underline{s})], \frac{\beta}{1+\beta}\underline{s}\right\}, \\ \underline{\hat{R}} &= \min\left\{\left[\frac{\underline{S}}{\mathrm{Max}(\underline{S})}\right]^{1/(1+\beta)}, [\underline{S}]^{1/(1+\beta)}\right\}.\end{aligned}$$

Appendix B. The Retinex algorithm for $\beta \gg \alpha$—results

In this appendix we show through several examples that indeed we get a Gamma correction for the choice $\beta \gg \alpha$. More specifically, in the following simulations, we have chosen $\alpha = 1\mathrm{e} - 6$, $\beta = 1$. Obviously, following the results of the previous appendix, the results are supposed to be very close to exact Gamma correction with effective Gamma value of $\gamma = 1 + \beta = 2$.

Fig. 11 shows the obtained look-up-tables for the full LUT case. For both images, we can see that the optimal LUT has a shape of a stretching, followed by Gamma correction.

Fig. 12 shows the obtained kernels in the application of linear filter + bias approach. In this case, instead of performing a spatial sharpenning operation, the kernels are chosen to be a simple unit operation multiplied by some constant.

Fig. 13 shows the results of the linear + bias approach by plotting the output versus the input as a Look-Up-Table. Again, we see that the results are an effective LUT having the shape of a Gamma correction, as expected.

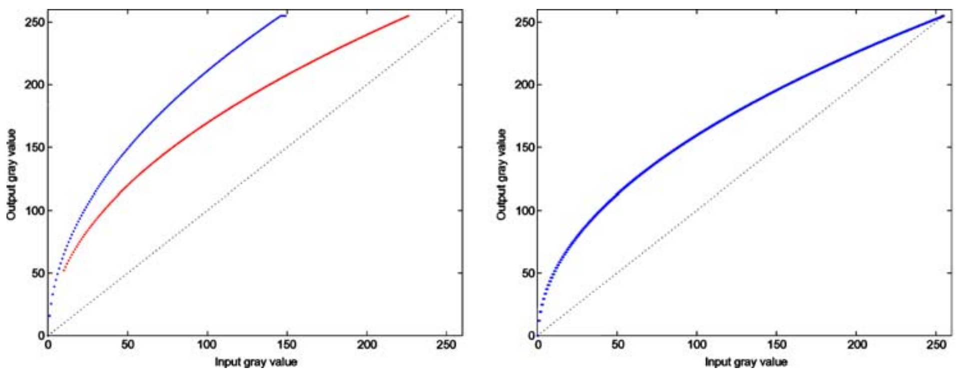

Fig. 13. Optimal linear filter + bias for $\beta \gg \alpha$: (left) the results as a LUT mapping for the image "Child"; (right) the results as a LUT mapping for the image "Houses."

In all these examples, the measured Gamma in these graphs was found to be 2, as expected.

References

Bertsekas, D.P., 1995. Non-Linear Programming. Athena Scientific, Belmont, MA.

Blake, A., 1985. Boundary conditions of lightness computation in Mondrian world. Comput. Vision Graphics Image Process. 32, 314–327.

Blake, A., Zisserman, A., 1987. Visual Reconstruction. MIT Press, Cambridge, MA.

Brainard, D.H., Wandell, B., 1986. Analysis of the Retinex theory of color vision. J. Opt. Soc. Am. A 3, 1651–1661.

Durand, F., Dorsey, J., 2002. Fast bilateral filtering for the display of high dynamic range images. ACM Trans. Graphics 21 (3), 257–266.

Faugferas, O.D., 1979. Digital image color processing within the framework of a human visual system. IEEE Trans. ASSP 27, 380–393.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6, 721–741.

Holm, J., 1996. Strategy for Pictorial Digital Image Processing (PDIP). In: Proceedings of the Color Imaging Conference, November, pp. 194–201.

Horn, B.K.P., 1974. Determining lightness from an image. Comput. Graphics Image Process. 3, 277–299.

Jobson, D.J., Rahman, Z., Woodell, G.A., 1997a. Properties and performance of the center/surround retinex. IEEE Trans. Image Proc. 6, 451–462.

Jobson, D.J., Rahman, Z., Woodell, G.A., 1997b. A multiscale Retinex for bridging the gap between color images and the human observation of scenes. IEEE Trans. Image Proc. 6.

Kimmel, R., Elad, M., Shaked, D., Keshet (Kresch), R., Sobel, I., 2003. A variational framework for Retinex. Int. J. Comput. Vision 52 (1), 7–23.

Lagendijk, R.L., Biemond, J., 1991. Iterative Identification and Restoration of Images. Kluwer Academic Publishing, Boston, MA.

Land, E.H., 1977. The Retinex theory of color vision. Sci. Am. 237, 108–128.

Land, E.H., 1983. Recent advances in the retinex theory and some implications for cortical computations: color vision and the natural image. Proc. Natl. Acad. Sci. USA 80, 5163–5169.

Land, E.H., 1986. An alternative technique for the computation of the designator in the Retinex theory of color vision. Proc. Natl. Acad. Sci. USA 83, 3078–3080.

Land, E.H., McCann, J.J., 1971. Lightness and the Retinex theory. J. Opt. Soc. Am. 61, 1–11.

Luenberger, D.G., 1987. Linear and Non-Linear Programming, second ed. Addison-Wesley, Menlo-Park, CA.

Marroquin, J., Mitter, J., Poggio, T., 1987. Probabilistic solution for ill-posed problems in computational vision. J. Am. Statist. Assoc. 82, 76–89.

McCann, J.J., Sobel, I.,1998. Experiments with Retinex, HPL Color Summit.

Stockham Jr., T.G., 1972. Image processing in the context of a visual model. Proc. IEEE 60, 828–842.

Terzopoulos, D., 1986. Image analysis using multigrid relaxation methods. IEEE Trans. PAMI 8, 129–139.