International Journal of Computer Vision 52(1), 7–23, 2003

A Variational Framework for Retinex

RON KIMMEL
Computer Science Department, Technion—I.I.T., Technion City, Haifa 32000, Israel
ron@cs.technion.ac.il

MICHAEL ELAD
The Computer Science Department, Stanford University, Gates 2B, Room 282, Stanford 94305-9025, CA, USA
elad@sccm.stanford.edu

DORON SHAKED AND RENATO KESHET
HP Laboratories Israel, Technion City, Haifa 32000, Israel
doron.shaked@hp.com
renato.keshet@hp.com

IRWIN SOBEL
HP Laboratories, 1501 Page Mill Rd., Palo Alto, CA 94304, USA
Irwin.sobel@hp.com

Received October 12, 2001; Revised October 4, 2002; Accepted October 8, 2002

Abstract. Retinex theory addresses the problem of separating the illumination from the reflectance in a given image and thereby compensating for non-uniform lighting. This is in general an ill-posed problem. In this paper we propose a variational model for the Retinex problem that unifies previous methods. Similar to previous algorithms, it assumes spatial smoothness of the illumination field. In addition, knowledge of the limited dynamic range of the reflectance is used as a constraint in the recovery process. A penalty term is also included, exploiting a-priori knowledge of the nature of the reflectance image. The proposed formulation adopts a Bayesian view point of the estimation problem, which leads to an algebraic regularization term, that contributes to better conditioning of the reconstruction problem.

Based on the proposed variational model, we show that the illumination estimation problem can be formulated as a Quadratic Programming optimization problem. An efficient multi-resolution algorithm is proposed. It exploits the spatial correlation in the reflectance and illumination images. Applications of the algorithm to various color images yield promising results.

Keywords: variational models, multi-resolution, quadratic programming, illumination removal, image enhancement, dynamic range compression, reflectance

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 possibility of removing illumination effects of back/front lighting, and enhancing shots that include spatially varying illumination such as images that contain indoor and outdoor zones.

Recovering the illumination from a given image is known to be a mathematically ill-posed problem, and algorithms proposed in the literature for its solution vary in their way of overcoming this limitation. The Retinex methodology was motivated by Land's landmark research of the human visual system (Land, 1977). Through his experiments it was shown that our visual system is able to practically recognize and match colors under a wide range of different illuminations, a property that is commonly referred to as the *Color Constancy Phenomenon*. As a matter of fact, Land's findings indicated that even when retinal sensory signals coming from different color patches under different illuminations are identical, subjects were able to name the surface reflectance color (Land, 1977). The ability to extract the illumination image is sufficient but not necessary to achieve this property.

In this paper we define the Retinex reconstruction problem for gray-level images through physically motivated considerations. The proposed formulation is shown to be a mathematically well-posed problem. A variational expression is obtained by defining the optimal illumination as the solution of a Quadratic Programming (QP) optimization problem. It is shown that different previous algorithms are essentially solutions to similar variational problems. We introduce an efficient algorithm based on QP solvers and the fact that the unknown illumination is spatially smooth. Our algorithm uses a multi-resolution reconstruction of the illumination with few relaxation iterations at each resolution layer.

We apply and compare the proposed algorithm in two color spaces. The first operates in the RGB space, in which each spectral channel is processed separately. The second is the HSV color space in which only the Value (V) channel is processed. Both methods produce pleasing results in terms of dynamic range compression. In addition color corrections can be obtained as a by-product of the Retinex algorithm in the RGB space. The resulting reflectance image usually appears to be over-enhanced. A relaxation algorithm for this effect is proposed. Tests on images from various sources produce pleasing images, and support the assumption that the results of the proposed formulation are similar to those of the human visual system.

This paper is organized as follows: In the next section we review several different Retinex algorithms. Some of those were motivated by assumptions based on the color constancy process in the human visual system. Section 3 presents the proposed formulation along with an efficient numerical algorithm for the illumination reconstruction. Uniqueness and convergence of the solution are also discussed in Section 3. Section 4 extends the proposed formulation to color images. In Section 5 we show a possible application that corrects the illumination component and then adds it back to the image. In Section 6 we apply the method to different images and demonstrate the algorithm's performances and the effects of its free parameters. Section 7 gives concluding remarks.

2. Previous Work

The first Retinex algorithms proposed by Land et al. were of random walk type (Land, 1983; Land and McCann, 1971). Subsequent algorithms (Jobson et al., 1997a, 1997b; Land, 1986) use Homomorphic Filters (Faugeras, 1979; Stockham Jr., 1972). Yet another group of Retinex algorithms is based on solving a Poisson equation (Blake, 1985; Funt et al., 1992; Horn, 1974; Terzopoulos, 1986). Retinex algorithms by McCann et al. (Frankle and McCann, 1983; Funt et al., 2000; McCann, 1999), are an iterative multi-resolution type of non-linear filter.

A first step taken by most algorithms is the conversion to the logarithmic domain by $s = \log S, l = \log L, r = \log R$, and thereby $s = l + r$. This step is motivated both numerically, preferring additions over multiplications, and physiologically, referring to the sensitivity of our visual system (Land, 1977). The different Retinex algorithms usually have the same flow chart as shown in Fig. 1, and the difference between them concentrates on the actual estimation of the illumination image.

The main motivation of the subsequent overview of the above-mentioned Retinex algorithm families is to find similarities in the apparently different approaches, and thereby to motivate the proposed approach. Thus, the description of algorithm families should not be interpreted as a detailed description of any specific Retinex algorithm but rather as a generalized description of the family.

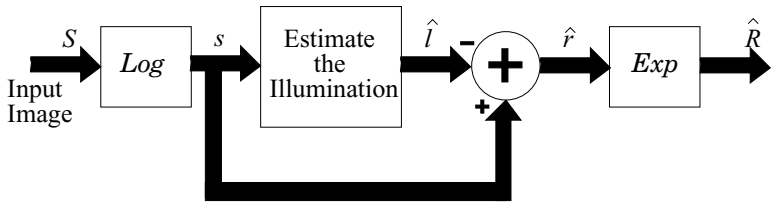

Figure 1. The general flow chart of Retinex algorithms.

2.1. *Random Walk Algorithms*

A random walk is a discrete time random-process in which the 'next pixel position' is chosen randomly from the neighbors of the current pixel position. Random walk type Retinex algorithms are nonlinear variants of the following basic formulation (Brainard and Wandell, 1986): A large number of walkers are initiated at random locations of an input image s, adopting the gray-value of their initial position. An accumulator image A that has the same size as s is initialized to zero. As the walkers walk around, they update A by adding their values to each position they visit. Finally, the reflectance image is obtained by normalizing the accumulator image, i.e., its value at each location divided by the number of walkers visited it. By using many walkers with long paths, it is easily verified (Papoulis, 1991) that each accumulator value assimptotically converges to a Gaussian average of its neighbors. The non-linearities added to this basic version are:

- As random walks cross strong gradients (larger than a predetermined threshold) the corresponding walker updates its value by adding it to the gradient value. Note that if the threshold is very large the output r is practically a low pass version of the input image, and correspondingly the difference l between the input and the output images contains all the image details. Conversely, if the threshold is very small, the details remain in the output image, and the difference l is a smooth version of the input. In practice, the threshold is very small—so as to include 'illumination gradients' only.
- walker values can not exceed 255 (or whatever other value for White), thus making sure no illumination is whiter than White.

2.2. *Homomorphic Filtering*

Homomorphic Filtering type Retinex algorithms (Faugeras, 1979; Jobson et al., 1997a, 1997b; Land, 1986; Stockham Jr., 1972) share the following basic motivation: The reflectance image corresponds to the sharp details in the image (i.e. edges), whereas the illumination image is expected to be spatially smooth, a reasonable guess for l is a low-pass version of s, where the low pass is usually obtained as a convolution with a wide Gaussian kernel.

2.3. *Poisson Equation Solution*

Following the above reasoning, since the illumination is expected to be spatially smooth, its derivative should be close to zero everywhere. On the other hand, by the assumption that the reflectance is piece-wise constant, its derivative is expected to vanish almost everywhere, and get high values along the edges. Thus, if we take the derivative of the sum $s = l + r$ and clip out the high derivative peaks, we can assume that the clipped derivative signal corresponds only to the illumination.

Poisson Equation type Retinex algorithms (Blake, 1985; Horn, 1974; Terzopoulos, 1986) rely on Land's Mondrian world model. The Mondrian model boils down to the above assumption on the reflectance as a piece-wise constant image. Applying the Laplacian, and the following clipping operation

$$\tau(\Delta s) = \begin{cases} \Delta s & \text{where } |\Delta s| < T \\ 0 & \text{otherwise,} \end{cases}$$

we get the following Poison equation

$$\Delta \hat{l} = \tau(\Delta s).$$

As to the solution of the resulting Poisson equation, Horn (1974) suggested an iterative procedure which effectively inverts the Laplacian operator. Similar to the previous methods, a low-pass filter is applied in order to solve the above equation. Blake (1985) introduced an improvement to Horn's method. He proposed to extract the discontinuities from the image gradient magnitude instead of the Laplacian and thereby came

up with better boundary conditions that deal with less trivial scenarios along the image boundary. An additional algorithmic improvement by Funt et al. (1992) uses the curl to assure integrability.

2.4. McCann's Algorithm

McCann et al. (Frankle and McCann, 1983; Funt et al., 2000; McCann, 1999) proposed variants of an algorithm that can be equivalently written as follows: The illumination image $\hat{l}_0$ is initialized to be s, the original image. The algorithm performs the following iterative procedure,

$$\hat{l}_{n+1} = \max\left\{\frac{\hat{l}_n + s}{2}, \frac{\hat{l}_n + D_n[\hat{l}_n]}{2}\right\}$$

where D_n is a translation operator, shifting the image by the nth element of a sequence of spirally decaying translation vectors $\{d_n\}$, as shown in Fig. 2. The size of the first displacement is set to be half the minimum between the image width and height.

Let us link this procedure to the previous methods. If we remove the max operation we get the simplified version

$$\hat{l}_{n+1} = \frac{\hat{l}_n + D_n[\hat{l}_n]}{2}.$$

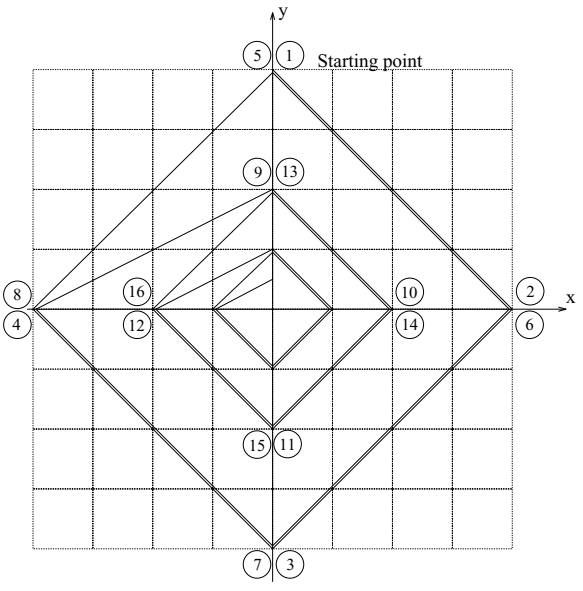

Figure 2. The sequence of displacement vectors for the D_n operator in McCann algorithm.

This is a simple averaging operation that smoothes the image. Actually, it is possible to show that with the displacements shown in Fig. 2, the effective smoothing kernel approaches a Gaussian.

The non-linear (max) operation inside the loop forces the illumination image to satisfy the constraint $\hat{l} \geq s$. Incorporating the physical nature of reflecting objects which reflect only part of the incident light. Thus, the reflectance is restricted to the range $R \in [0, 1]$, and $L \geq S$, which implies $l \geq s$.

A multi-resolution version is also proposed in Funt et al. (2000) and McCann (1999). A Gaussian pyramid is constructed for the given image s. The algorithm starts at the coarsest level, and the size of the displacements for D_n are one pixel in each direction for each resolution. The multi-resolution version is significantly faster, yet produces lower quality results compared to the original version.

2.5. Summary of Previous Work

The discussion in this section suggests that the previous seemingly different algorithms are actually very similar. They are all based on the spatial smoothness assumption of the illumination l. All the above algorithms apply various, potentially nonlinear, smoothing operators to s in order to extract $\hat{l}$. Some methods add more assumptions about the reflectance, such as its limited range, or its Mondrian form. Eventually, 'skinning' the illumination from the given image yields the reflectance image, which is expected to be free of non-uniform illumination, have a reduced dynamic range, and be a more pleasing image.

3. The Variational Framework

3.1. Functional Definition

We start by listing the known information about the illumination image.

1. The first important assumption about the illumination is its spatial smoothness.
2. We also know that, since R is restricted to the unit interval, we can add the constraint $L \geq S$. Since the log function is monotone, we also have $l \geq s$.
3. By setting $l = \text{Const}$, where Const is any constant above the maximal value of s, we get a trivial solution that satisfies the two previous assumptions. We

therefore add the assumption that the illumination image is close to the intensity image s, i.e., it minimizes a penalty term of the form dist(l, s), e.g., the L_2 norm $(l-s)^2$.

4. The reflectance image $r = s - l$ can be assumed 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). Note that since $r + l = s$, spatially smooth r contradicts spatially smooth l. In practice adding this penalty term kicks in mainly on sharp edges and handles situations where the illumination is not smooth (as well as cases of direct light sources and specularities).
5. We can assume that the illumination continues smoothly as a constant beyond the image boundaries. This is an artificial assumption required for boundary conditions that would have minor effect on the final results.

Collecting all the above assumptions into one expression we get the following penalty functional

$$\begin{aligned} \text{Minimize:} \quad F[l] &= \int_\Omega (|\nabla l|^2 + \alpha(l-s)^2 \\ &\quad + \beta|\nabla(l-s)|^2)\, dx\, dy \\ \text{Subject to:} \quad & l \geq s, \quad \text{and} \quad \langle \nabla l, \vec{n} \rangle = 0 \text{ on } \partial\Omega, \end{aligned} \tag{1}$$

where Ω is the support of the image, $\partial\Omega$ its boundary, and $\vec{n}$ is the normal to the boundary. 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. This choice of smoothness penalty is natural, if we keep in mind that minimizing $\int(|\nabla l|^2)\, dx\, dy$ translates into the Euler-Lagrange (EL) equation $\Delta l = 0$. Its steepest descent solution is a Gaussian smoothing operation with increasing variance of the initial condition. As mentioned in the previous section, several authors proposed Gaussian related smoothing of s for the illumination reconstruction.

The second penalty term $(l-s)^2$ forces a proximity between l and s. The difference between these images is exactly r, which means that the norm of r should be small (i.e., R tends to White). This term is weighted by the free parameter α. The main objective of this term is a regularization of the problem that makes it better conditioned. Notice that, in addition, we force the solution l to be $l \geq s$. In practice this penalty term should be weak in order not to pull l down too much towards s. Note that in contrast to gradient penalty terms which apply mainly on edges, this term applies potentially anywhere, and α should therefore be very small.

The third term represents a Bayesian penalty expression. It forces r to be a 'visually pleasing' image. This term weighted by the free parameter β penalizes gradients in r and forces it to be spatially smooth. Note that more complicated Bayesian expressions may be used allowing 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.

The problem we have just defined has a Quadratic Programming (QP) form (Bertsekas, 1995; Luenberger, 1987). The necessary and sufficient conditions for its minimization are obtained via the Euler-Lagrange equations

$$\begin{aligned} &\forall (x, y) \in \Omega \\ &\quad \times \left\{ \begin{array}{c} \frac{\partial F[l]}{\partial l} = 0 = -\Delta l + \alpha(l-s) - \beta\Delta(l-s) \\ \text{and} \quad l > s \\ \text{or} \\ l = s \end{array} \right\} \end{aligned} \tag{2}$$

Note that the differential equation does not have to hold when $l = s$.

An interesting side-effect of the proposed formulation is an invariance property to a specific yet popular transformation of the input image, known as the gamma-correction. In most imaging systems the linear sensory data is passed through a Look-Up-Table in order to brighten the values. Such typical transformation is the Gamma-correction, where $S_{out} = S_{in}^{1/\gamma}$ (assuming $0 \leq S_{in}, S_{out} \leq 1$).

In general, if indeed such transformation takes place, it has to be removed prior to any Retinex algorithm and redone prior to viewing of the results. However, as we step into the log-domain, the Gamma-correction becomes a multiplication of $s = \log S$ by the constant $1/\gamma$. Based on the conditions in Eq. (2), if l_{opt} is the optimal solution for a specific image s, then l_{opt}/γ satisfies these conditions for an input image s/γ with $r/\gamma = (s - l_{opt})/\gamma$. This means that instead of inverting the Gamma-correction, applying Retinex and re-applying Gamma correction, one may simply apply Retinex to Gamma-corrected images.

3.2. Numerical Solution

The minimization problem is QP with respect to the unknown image l. Many algorithms for solving such problems are known in the literature (Bertsekas, 1995; Luenberger, 1987). In this paper we chose to focus on the Projected Normalized Steepest Descent (PNSD) algorithm, accelerated by a multi-resolution technique.

3.2.1. Projected Normalized Steepest Descent. The PNSD algorithm requires the application of a Normalized Steepest Descent (NSD) iteration that minimizes the functional $F[l]$, followed by a projection onto the constraints. A NSD iteration has the format

$$l_j = l_{j-1} - \mu_{\text{NSD}} \cdot G,$$

where l_j and l_{j-1} are the illumination images at step j and $j-1$, respectively, G is the gradient of $F[l]$, and μ_{NSD} is the optimal line-search step size. In our case, Eq. (2), the gradient of $F[l]$ is given by:

$$G = -\Delta l_{j-1} + (\alpha - \beta\Delta)(l_{j-1} - s),$$

and μ_{NSD} is given by

$$\mu_{\text{NSD}} = \frac{\int_\Omega |G|^2}{\int_\Omega (\alpha|G|^2 + (1+\beta)|\nabla G|^2)}$$

Observe that, by integration by parts, $\int |\nabla G|^2 = -\int G\Delta G$ up to boundary conditions.

An alternative approach is the Steepest Descent (SD) algorithm, where μ_{NSD} is replaced by a constant value μ_{SD}, such that

$$\mu_{\text{SD}} \in \left(0, \frac{2}{\lambda_{\max}\{-(1+\beta)\Delta + \alpha I\}}\right),$$

where $\lambda_{\max}\{A\}$ refers to the greatest eigenvalue of the linear operator A. This alternative method saves computations at the expense of a slightly slower convergence.

Finally, projecting onto the constraint $l \geq s$ is done by $l_j = \max(l_j, s)$.

Notice that G can be calculated by

$$G = -G_A + \alpha(l_{j-1} - s) - \beta(G_A - G_B),$$

where

$$G_A \triangleq \Delta l_{j-1},$$
$$G_B \triangleq \Delta s.$$

Similarly, μ_{NSD} is given by

$$\mu_{\text{NSD}} = \frac{\mu_A}{\alpha\mu_A + (1+\beta)\mu_B},$$

where

$$\mu_A \triangleq \int_\Omega |G|^2,$$
$$\mu_B \triangleq \int_\Omega |\nabla G|^2.$$

We approximate the Laplacian by a linear convolution with the kernel κ_{LAP}

$$\kappa_{\text{LAP}} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

and the integrations are approximated by summations

$$\int_\Omega |G|^2 \approx \sum_n \sum_m G[n,m]^2$$
$$\int_\Omega |\nabla G|^2 = -\int_\Omega G\Delta G \approx -\sum_n \sum_m G[n,m](G^* \kappa_{\text{LAP}})[n,m],$$

where $G[m,n] = G(m\Delta x, n\Delta y)$. In order to handle the boundary conditions, defined in Eq. (1), the above convolution is applied on an expanded version of the image G. The extension is done by replicating the first and last columns and rows. After the convolution, the additional rows and columns are removed.

3.2.2. Multi-Resolution. The PNSD algorithm usually converges slowly (Bertsekas, 1995; Luenberger, 1987). Instead of general acceleration schemes, we use the fact that the unknown image l is assumed to be smooth. Specifically, we apply a multi-resolution algorithm that starts by estimating a coarse resolution image l, expands it by interpolation and uses the result as an initialization for the next resolution layer. This way, few iterations at each resolution are enough for convergence.

Summarizing the above, a proposed algorithm for the solution of Eq. (1) involves the following steps,

1. **Input:** The input to the algorithm is an image s of size $[N, M]$, and two parameters α and β.

2. **Initialization:** Compute a Gaussian pyramid of the image s. This pyramid is constructed by smoothing the image with the kernel κ_{PYR},

$$\kappa_{\mathrm{PYR}} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix},$$

and decimating by 2:1 ratio. The process is repeated p times and produces a sequence of images $\{s_k\}_{k=1}^{p}$. The image s_1 is the original image s, and s_p is the one with the coarsest resolution in this pyramid. Define the numerical inner product

$$\langle G, F\rangle = \sum_{n=1}^{N}\sum_{m=1}^{M} G[n,m]F[n,m],$$

and the numerical Laplacian at the kth resolution as

$$\Delta_k G = G * k_{\mathrm{LAP}} 2^{-2(k-1)}.$$

Set $k = p$, i.e., start at the coarsest resolution layer, and set the initial condition $l_0 = \max\{s_p\}$.

3. **Main Loop:** For the kth resolution layer,

- **Calculate** $G_B \triangleq \Delta_k s_k$.
- **For** $j = 1, \ldots, T_k$ **Do:**

(a) Calculate gradient:

$$G_A \triangleq \Delta_k l_{j-1},$$
$$G \leftarrow G_A + \alpha(l_{j-1} - s_k) - \beta(G_A - G_B).$$

(b) Calculate μ_{NSD}

$$\mu_A \triangleq \langle G, G\rangle,$$
$$\mu_B \triangleq -\langle G, \Delta_k G\rangle,$$
$$\mu_{\mathrm{NSD}} \leftarrow \mu_A/(\alpha\mu_A + (1+\beta)\mu_B).$$

(c) Complete NSD iteration

$$l_j \leftarrow l_{j-1} - \mu_{\mathrm{NSD}} \cdot G,$$

(d) Project onto the constraints

$$l_j = \max\{l_j, s_k\}.$$

- **End** j **Loop**;

The above loop solves the intermediate problem

$$\text{Minimize:} \quad F_k[l] = \int_{\Omega_k} (|\nabla l|^2 + \alpha(l - s_k)^2 + \beta|\nabla(l - s_k)|^2)\, dx\, dy$$
$$\text{Subject to:} \quad l \geq s_k \quad \text{and} \quad \langle \nabla l, \vec{n}\rangle = 0 \text{ on } \partial\Omega,$$

4. **Update the next resolution layer:** If $k > 1$, the result l_{T_k} is up scaled (2:1 ratio) by pixel replication into the new l_0, the initialization for the next resolution layer. The resolution layer is updated $k = k-1$, and the algorithm proceeds by going again to Step 3. If $k = 1$, the result l_{T_1} is the final output of the algorithm.

3.3. *Relation to Previous Methods*

Let us revisit the algorithms described in Section 2 and analyze them in light of the proposed formulation. First, by setting $\alpha = \beta = 0$, and removing the constraint $l \geq s$ we get Homomorphic filtering. Adding back $l \geq s$ we become similar to random walk algorithms and the McCann algorithm.

The Poisson Equation approach seems to be unrelated directly to our formulation. However, if we let $\alpha(x, y) = \tau(\Delta s)$ and set the second distance term to $\int \alpha(x, y)(l - s)$, keeping the constraint $l \geq s$, we get that the optimal illumination should satisfy the equation

$$\Delta l = \tau(\Delta s), \tag{3}$$

subject to $l \geq s$, which is identical (up to the constraint) to Horn's formulation.

3.4. *Uniqueness and Convergence*

In this section we prove the uniqueness of the solution to Eq. (1), and the convergence of the proposed numerical algorithm. The following theorem shows that the convexity of the problem guarantees existence and uniqueness of the solution.

Theorem. *The variational optimization problem P, given by*

$$\text{Minimize:} \quad F[l] = \int_{\Omega} (|\nabla l|^2 + \alpha(l - s)^2 + \beta|\nabla(l - s)|^2)\, dx\, dy$$
$$\text{Subject to:} \quad l \geq s, \quad \text{and} \quad \langle \nabla l, \vec{n}\rangle = 0 \textit{ on } \partial\Omega,$$

with $\alpha > 0$ and $\beta \geq 0$, has a unique solution.

The proof is given in the Appendix.

Regarding the convergence of the numerical scheme, the core of the proposed algorithm is the Projected Normalized Steepest Descent (PNSD) algorithm, which is known to converge for convex optimization problems, such as our case (Bertsekas, 1995; Luenberger, 1987). The pyramidal shell of the algorithm can be considered as an efficient method for creating a good initialization for the highest resolution layer stage. We found that few iterations at the finer resolution layer are sufficient for effective convergence.

4. Color Images

Thus far we dealt with a single channel. In this section, we apply our method to color images. When we process color images the traditional approach is to deal with each color channel separately. We refer to channel-by-channel processing as 'RGB Retinex'. Treating the R, G, and B channels separately usually yields a color correction effect. For example, RGB Retinex on a reddish image is expected to modify the illumination in such a way that the red hue is removed so that the resulting image is brightened and corrected. Therefore, for some images, RGB Retinex actually improves the colors. Nevertheless, in other cases, such color correction can cause color artifacts that exaggerate color shifts, or reduce color saturation.

Another approach is to map the colors into a different color space, such as HSV, apply the Retinex correction only to the intensity layer, and then map back to the RGB domain. We refer to this method as the 'HSV Retinex'. Color shifts in such cases are less-likely. A major advantage is that we have to process a single channel. We refer to Barnard and Funt (1998) for further analysis of color constancy issues in a Retinex algorithm.

5. Alternative Illumination Correction

The reflectance image obtained by the Retinex process is sometimes an over-enhanced image. It might be argued that (i) the human visual system merely reduces the dynamic range of scenes rather than removing the illumination altogether, namely, shaded areas are definitely perceived as such. (ii) removal of all the illumination exposes noise that might exist in darker regions of the original image.

We propose adding a corrected version of the reconstructed illumination back to the reconstructed reflectance image. Figure 3 describes this operation. The proposed scheme computes the illumination image $L = \exp(l)$ from the intensity image $S = \exp(s)$, and the reflectance image $R = S/L$, as discussed in previous sections. Then, we 'tune up' the illumination image L by a Gamma Correction operation with a free parameter γ, obtain a new illumination image L', and multiply it by R, that gives the output image $S' = L' \cdot R$. The Gamma correction is performed by

$$L' = W \cdot \left[\frac{L}{W}\right]^{\frac{1}{\gamma}}, \tag{4}$$

where W is the White value (equal to 255 in 8-bit images).

The final result S' is given, therefore, by

$$\begin{aligned} S' &= L' \cdot R = \frac{L'}{L} S \\ &= W \frac{(L/W)^{1/\gamma}}{L} S = \frac{S}{(L/W)^{1-1/\gamma}}. \end{aligned} \tag{5}$$

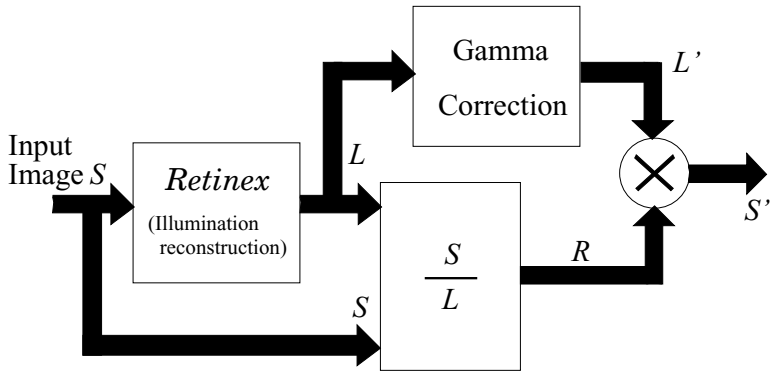

Figure 3. Returning part of the illumination to the reflectance image.

For $\gamma = 1$, the whole illumination is added back, and therefore $S' = S$. For $\gamma = \infty$, no illumination is returned, and we get $S' = R \cdot W$, which is the same reflectance image, R, as obtained by the original retinex, stretched to the interval $[0, W]$. The later case can also be considered as tuning up the illumination to a maximal valued uniform illumination W.

Adding part of the illumination to the final image can also be found in the homomorphic filtering approach. In Oppenheim and Schafer (1975, ch. 10), the proposed linear filter for the illumination calculation in the log domain, removes high-pass spatial components of s, yet also attenuates the low-pass components by a factor of γ_i (where i stands for *illumination*). This is analog to a gamma correction of the illumination with $\gamma = \gamma_i$, since Eq. (5) can be written as

$$\frac{S'}{W} = \left(\frac{L}{W}\right)^{1/\gamma} \cdot R, \tag{6}$$

and therefore

$$\begin{aligned} s' - w &= \frac{1}{\gamma}(l - w) + r \\ &= \frac{1}{\gamma}(\text{low-pass components}) \\ &\quad + (\text{high-pass components}). \end{aligned} \tag{7}$$

6. Results

In our experiments we applied the numerical algorithm of Section 3 to several test images, two of which are shown in Fig. 4, other examples may be viewed at Kimmel et al. (1999). All results correspond to $\alpha = 0.0001$ and $\beta = 0.1$, unless indicated differently. Four resolution layers were used with $T_k = 1, 2, 3,$ and 4 iterations at each layer, 1 iteration at the finest ($k = 1$) and 4 at the coarsest resolution ($k = 4$). This idea of using more iterations at coarser scales is known by the name 'Cascadic Multigrid' (Bornemann and Deuflhard, 1996) and is used in order to gain convergence at coarser scales where complexity is much lower. All images appear in color in the electronic version.

Note that since we do not know the origin of the images we work with, we assume them to be linear up to Gamma-correction, and as explained earlier, we do not need to invert this transformation as our algorithm is in fact invariant to it.

Figure 5 demonstrates the invariance to gamma-correction. It shows the results of the RGB-retinex process in two different paths: (i) Apply RGB retinex directly on the input image; and (ii) Apply inverse-gamma-correction to the input image ($\gamma = 2.2$), then apply RGB-retinex as usual, and finally apply gamma-correction (again $-\gamma = 2.2$) to better view the results. As can be seen, the results are very similar. Differences are due to the discretization effects and the small number of iterations used.

Figure 4. The original images used in the experiments.

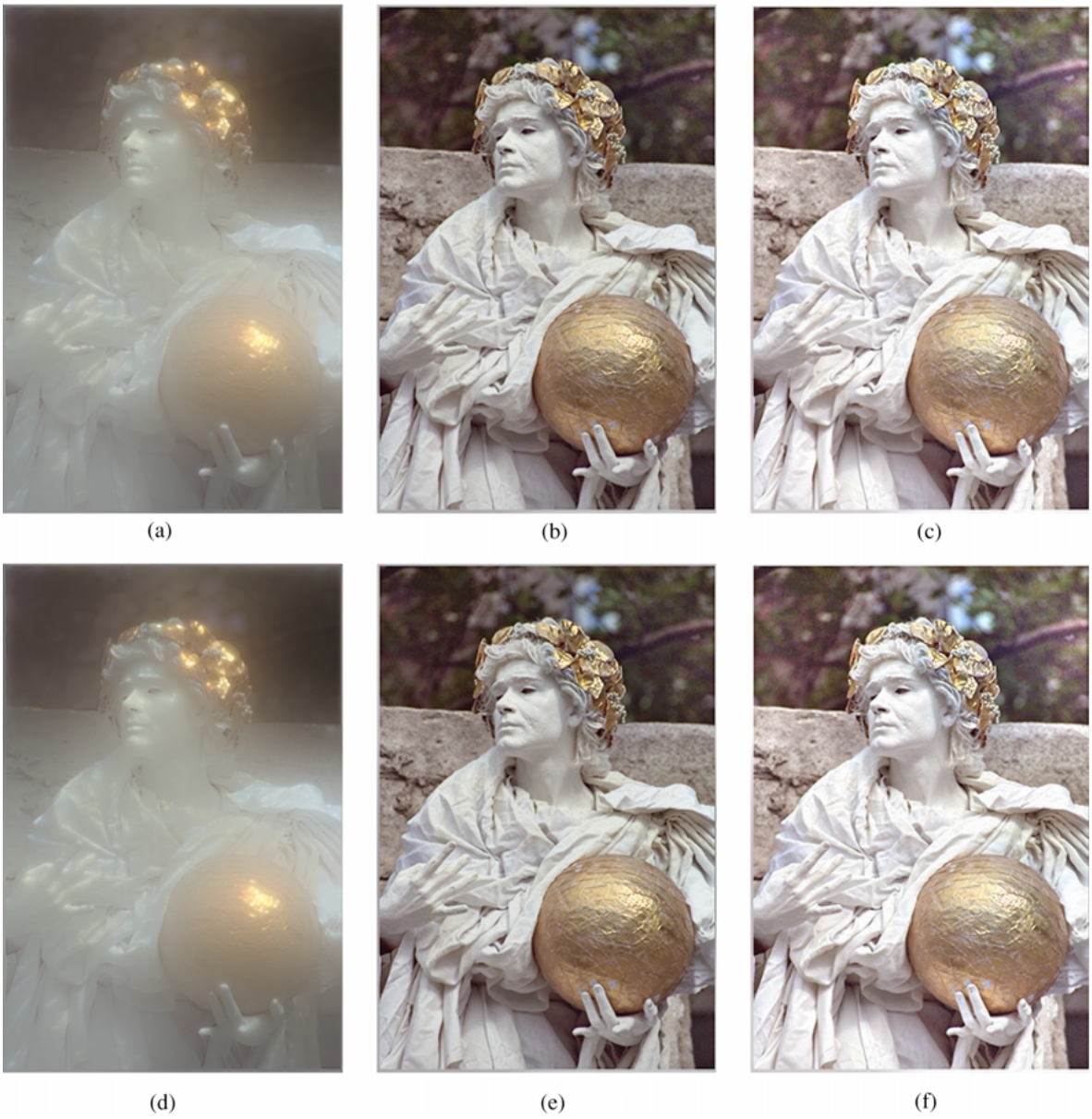

Figure 5. Invariance to gamma-correction: (i) Top row—direct application of the Retinex procedure, with illumination in (a) output with $\gamma = 3$ illumination return in (b), and output with $\gamma = \infty$ illumination return in (c); (ii) Bottom row—inverse gamma correction before and gamma correction after the Retinex procedure using $\gamma = 2.2$, with illumination in (d), output with $\gamma = 3$ illumination return in (e), and output with $\gamma = \infty$ illumination return in (f).

In the next test, we apply the RGB and the HSV Retinex algorithms to two input images. The results are shown in Figs. 6 and 7.

The second test (Fig. 8) presents the influence of the β and T values on the reconstructed reflectance image. Our goal is to show that the algorithm is stable in the choice of these parameters which can change in a wide range with minor effects on the outcome. We apply the HSV retinex process, with fixed $\gamma = 3$. The β values change from 1e-5 to 1 with relatively minor effect on the output quality. As for the number of iterations, the number of iterations per resolution

Figure 6. RGB retinex results: Illumination in (a) output with $\gamma = 3$ in (b), and $\gamma = \infty$ (i.e. reflectance image) in (c); HSV retinex results: Illumination in (d) output with $\gamma = 3$ in (e), and $\gamma = \infty$ (i.e. reflectance image) in (f).

level is $T \cdot k$, where $k = 1$ is the finest resolution level. We vary the value of T between $T = 2$ and 32 with slight apparent differences.

In Fig. 9, we restore the illumination through Gamma correction and add it back to the reflectance image. We compare illumination correction with $\gamma = \{2,\ 6,\ 24\}$ to standard Gamma correction on the image. Corresponding γ values were tuned to approximate the overall mean brightness of the illumination corrected images. Again, HSV retinex process was used.

Finally, we demonstrate the convergence of the proposed numerical algorithm. Figure 10 shows the values of $F[l]$, the functional in Eq. (1), as a function of the number of iterations, and the influence of using the multi-resolution methodology. For this simulation we use the RGB-retinex that applies the optimization process to each of the three layers separately. In the first part of the experiment we use the non-pyramidal approach and plot the value of $F[l]$ as a function of the number of iterations. We refer to each iteration as a

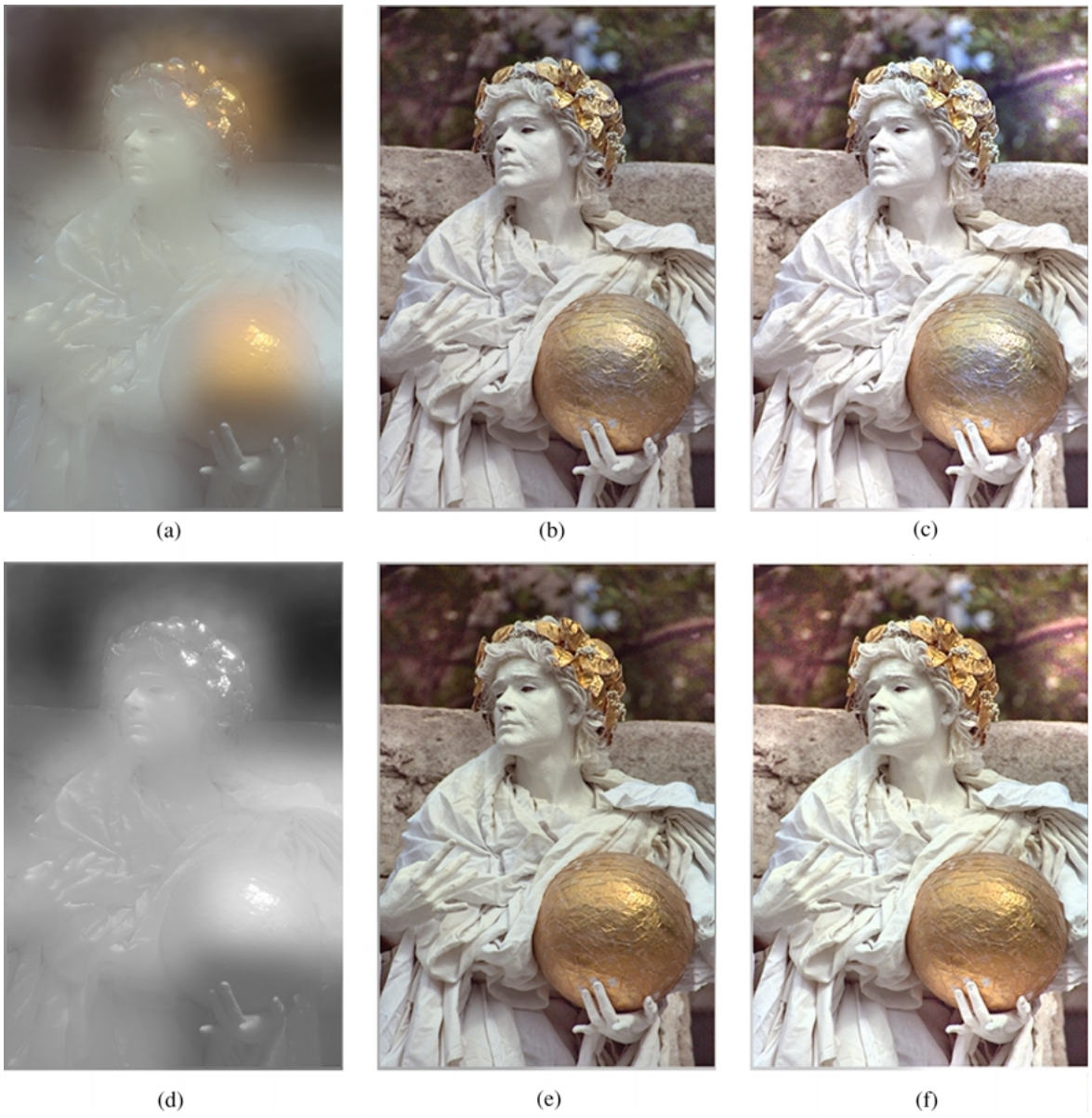

Figure 7. RGB retinex results: Illumination in (a) output with $\gamma = 3$ in (b), and $\gamma = \infty$ (i.e. reflectance image) in (c); HSV retinex results: Illumination in (d) output with $\gamma = 3$ in (e), and $\gamma = \infty$ (i.e. reflectance image) in (f).

single unit of operation. We see convergence after 10 such operations (iterations). When applying the same algorithm with N layer pyramidal approach, each single iteration becomes one iteration in the original resolution, 2 iterations on the next resolution later, and generally, k iterations on the kth layer. However, the kth layer is $0.5^{2(k-1)}$ smaller in size, thus reducing by the same factor the computational complexity. To summarize, for N-layer pyramid we have that one iteration is equivalent to

$$\sum_{k=1}^{N} k \cdot 0.25^{k-1} = \frac{1}{0.75^2}\left(1 - (1 + 0.25 \cdot N)0.25^N\right) \leq 1.78 \text{ Operations.}$$

For 2-layer pyramid the above factor becomes 1.5 while 4-layer pyramid gives 1.75 operations/iteration. Plotting $F[I]$ results as a function of the number of

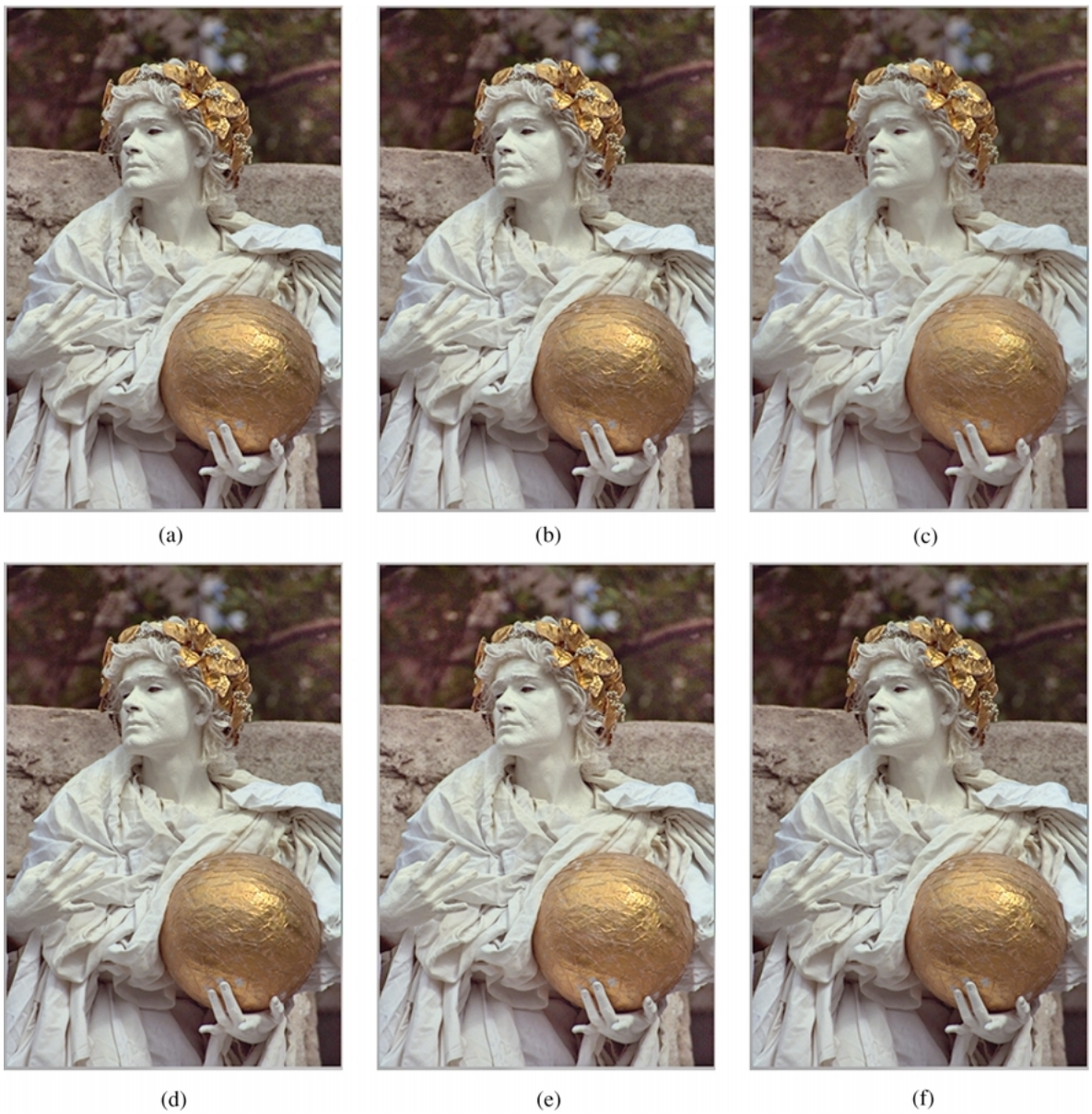

Figure 8. The influence of β and T: Constant $T = 2$ and $\beta = 0.00001$ in (a), $\beta = 0.1$ in (b), and $\beta = 10$ in (c); Constant $\beta = 0.1$ and $T = 2$ in (d), $T = 8$ in (e), and $T = 32$ in (f).

operations, we see that the convergence with the pyramidal method is much faster, getting to near-steady-state after 2 operations only. We also see that increasing the number of resolution layers improves the overall convergence. Note that in this analysis we do not take into account the construction of the pyramids, as we have found this to be redundant. We also do not take into account the initialization, which tends to be much simpler in the pyramidal approach.

We conclude that

1. As we see in Figs. 6 and 7, both the RGB and the HSV Retinex algorithms provided the desired dynamic range compression. The output images are indeed enhanced versions of the original

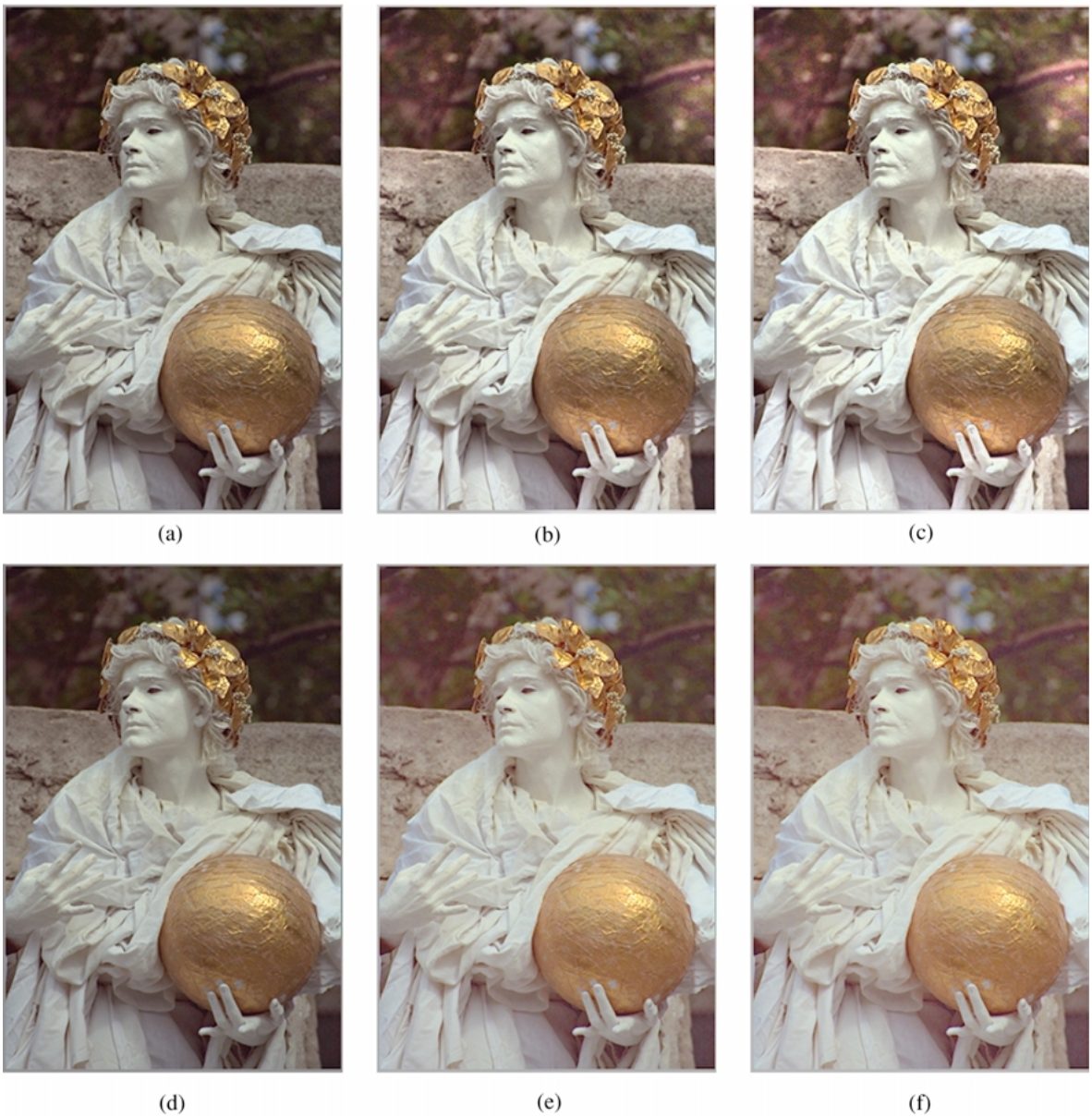

Figure 9. The influence of γ (illumination return) compared to standard Gamma correction; The proposed algorithm with $\gamma = 2$ in (a), $\gamma = 6$ in (b), and $\gamma = 24$ in (c). Standard Gamma-correction with γ values tuned to fit corresponding output of the proposed algorithm: $\gamma = 1.4$ in (d), $\gamma = 2.1$ in (e), and $\gamma = 2.4$ in (f).

one, although in some versions they are over-enhanced.

2. The illumination feedback through Gamma correction seems to improve both the RGB and the HSV Retinex results. However, they have different effects: In the RGB Retinex this process restores some of the colors, whereas in the HSV Retinex, the result is merely darker.
3. When we compare the RGB and the HSV Retinex algorithms our preference depends on the input image. Generally speaking, for images with colored illumination, the RGB usually performs better, whereas for images with a milder illumination hue, the HSV is better.
4. The Retinex approach obviously performs better than a simple Gamma correction. The latter indeed

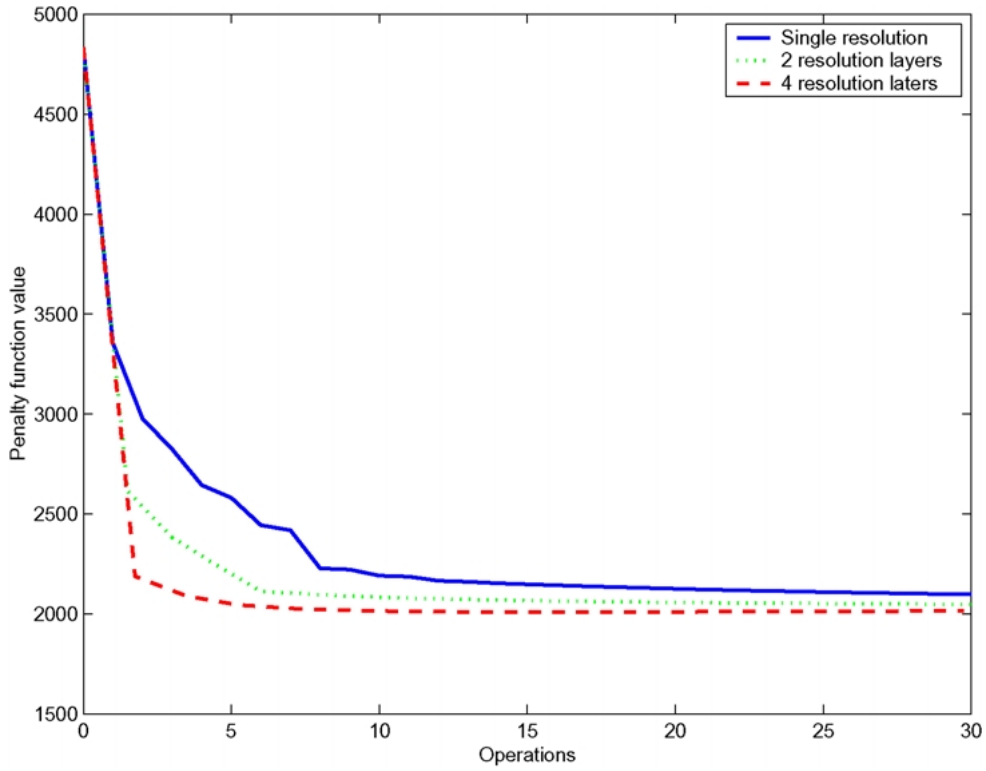

Figure 10. The functional value $F[l]$ as a function of the number of operations, and the influence of the multi-resolution method.

improves the overall illumination of the image, but also decreases details contrast and flattens the objects. Retinex, on the other hand, usually increases both detail contrast and depth sense of the image, as well as improving the overall illumination.

5. The proposed algorithm is robust to the choice of its parameters.
6. The proposed numerical method converges very fast to its steady-state solution which is also the minimizer of the defined functional in Eq. (1). Also, the pyramidal method is found to speed-up convergence.

7. Concluding Remarks

In this paper we surveyed several algorithms for image illumination correction and dynamic range compensation, based on a common motivation known as the Retinex theory. We have shown that in spite of their different formulations, these algorithms can be derived from the same variational principle.

We introduced a comprehensive Retinex analysis, motivated by the different Retinex algorithms. Our variational approach provides solid mathematical foundation, that yields efficient and robust numerical solutions.

We introduced a fast multi-resolution solution to the corresponding variational problem, resulting in an algorithm whose computational complexity amounts to less than 11 convolutions of the full size image with a 3×3 kernel plus a few addition algebraic operations per pixel. The advantages of the proposed algorithm are:

1. Computational efficiency.
2. Good image quality.
3. Parameter robustness. It was shown that for a wide range of the involved parameters, the output quality is practically the same.

As part of the proposed Retinex enhancement algorithm, we proposed a new method to control the overall brightness of the image. Traditionally, after removing a non-uniform illumination via the Retinex, standard point operations like the γ-correction are required. According to the proposed method, the

overall illumination correction is coupled with partial illumination removal. Instead of removing the illumination from the original image, the illumination is corrected via a standard point operation like the γ-correction, and returned to the reflectance image. Thus, dark regions in the image which have been poorly illuminated are better illuminated, as if the actual illumination conditions in the image were improved.

Appendix: Uniqueness of the Solution

Theorem. *The variational optimization problem P, given by*

$$\text{Minimize:} \quad F[l] = \int_\Omega (|\nabla l|^2 + \alpha(l-s)^2 + \beta\,|\,\nabla(l-s)|^2)\,dx\,dy$$

$$\text{Subject to:} \quad l \geq s, \quad \text{and} \quad \langle \nabla l, \vec{n}\rangle = 0 \text{ on } \partial\Omega,$$

with $\alpha > 0$ and $\beta \geq 0$, has a unique solution.

Proof: First, we show that the functional $F[l]$ is purely convex. The Hessian of the quadratic functional $F[l]$ is given by

$$\frac{\partial^2 F[l]}{\partial l^2} = -(1+\beta)\Delta + \alpha I,$$

where I is the identity operator. The multiplication of the Laplacian operator by the negative value $-(1+\beta) < -1$ yields a positive semi-definite operator $-(1+\beta)\Delta \geq 0$. Since $\alpha > 0$, $\alpha I > 0$, i.e., it is positive definite. Therefore, the Hessian is also a positive definite operator. Thereby, the functional $F[l]$ is a strictly convex functional (Bertsekas, 1995; Luenberger, 1987). If $\alpha = 0$, the Hessian is semi-positive definite, and the convexity of $F[l]$ is not strict.

Define the set $C = \{l \,|\, l \geq s \text{ and } \langle \nabla l, \vec{n}\rangle = 0 \text{ on } \partial\Omega\}$ such that the constraints of P are equivalent to requiring $l \in C$. For every $l_1,\ l_2 \in C$, $\forall \theta \in [0,1]$, we have $\theta l_1 + (1-\theta) l_2 \in C$, or in other words, C is a convex set. This is true since C is the intersection of two convex sets (one per each original constraint).

Let us denote the minimum of the functional $F[l]$ as $\hat{l}_{opt}$. This solution is unique since $F[l]$ is strictly convex. If $\hat{l}_{opt} \in C$ than $\hat{l}_{opt}$ is the solution of P, and therefore we get a unique solution as the theorem claims.

On the other hand, if $\hat{l}_{opt} \notin C$, the solution to P is obtained on the boundary of the constraint set $C = \{l \,|\, l \geq s\}$. We prove this property by contradiction. Assume that the solution is given as $l_0 \in \text{Interior}\{C\}$. Define $l_1 = (1-\theta)\, l_0 + \theta \hat{l}_{opt}$ for $\theta \in (0,1)$. Due to the convexity of $F[l]$, it is clear that $F[l_1] < (1-\theta)F[l_0] + \theta F[\hat{l}_{opt}] < F[l_0]$. Since $l_0 \in C$, for θ sufficiently close to zero it can be guaranteed that $l_1 \in C$ as well. This way we get l_1 as a better solution, which contradicts our assumption. Thus, the solution for P is obtained on the boundary of C.

Let us now assume that two solutions are possible, and prove that this assumption leads to a contradiction. The two optimal solutions l_1 and l_2 must satisfy the following set of conditions

1. The solutions should be feasible: $l_1,\ l_2 \in C$.
2. Based on the previous results, the solutions should be on the boundary of C: $l_1,\ l_2 \notin \text{Interior}\{C\}$.
3. The functional value of the two solutions should be the same: $F[l_1] = F[l_2]$.
4. The solutions are optimal: $\forall l \in C,\ \ F[l] > F[l_1]$.
5. The solutions should not be equal to $\hat{l}_{opt}$, i.e., $F[l]$: $l_1,\ l_2 \neq \hat{l}_{opt}$.

Since C is convex, $\forall \theta \in (0,1), l_0 = (1-\theta)l_1 + \theta l_2 \in C$. Moreover, by the strict convexity of $F[l]$, we have that $F[l_0] = F[(1-\theta)l_1 + \theta l_2] < (1-\theta)F[l_1] + \theta F[l_2] = F[l_1]$ and again, we got a better solution l_0. This contradicts the previous assumptions, and therefore, there is a unique solution to P. □

References

Barnard, K. and Funt, B. 1998. Investigations into multy-scale Retinex, In *Proc. Color Imaging in Multimedia 98*, Derby.

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

Blake, A. 1985. Boundary conditions of lightness computation in mondrian world. *Computer Vision Graphics and Image Processing*, 32:314–327.

Blake, A. and Zisserman, A. 1987. *Visual Reconstruction*. The MIT Press: Cambridge, MA.

Bornemann, F. and Deuflhard, P. 1996. The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75:135–152.

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

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

Frankle, J. and McCann, J. 1983. Method and apparatus for lightness imaging, US Patent no. 4, 384,336.

Funt, B.V., Ciurea, F., and McCann, J. 2000. Retinex in Matlab. In *Proc. of IS&T/SID Eighth Color Imaging Conference*, pp. 112–121.

Funt, B.V., Drew, M.S., and Brockington, M. 1992. Recovering shading from color images. In *Proc. European Conference on Computer Vision (ECCV'92)*, pp. 124–132.

Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis ans Machine*, 6:721–741.

Horn, B.K.P. 1974. Determining lightness from an image. *Computer Graphics and Image Processing*, 3:277–299.

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

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

Kimmel, R., Elad, M., Shaked, D., Keshet, R., and Sobel, I. 1999. A variational framework for Retinex. *Hewlett Packard Laboratories* TR no. HPL-1999-151.

Lagendijk, R.L. and 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. Amer.*, 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. Nat. 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. Nat. Acad. Sci. USA*, 83:3078–3080.

Land, E.H. and 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*. 2nd edn. Addison-Wesley: Menlo-Park, CA.

Marroquin, J., Mitter, J., and Poggio, T. 1987. Probabilistic solution for ill-posed problems in computational vision. *J. of the American Statistical Assoc.*, 82:76–89.

McCann, J. 1999. Lessons learned from mondrians applied to real images and color gamuts. In *Proc. IS&T/SID 7th Color Imaging Conference*, pp. 1–8.

Oppenheim, A.V. and Schafer, R.W. 1975. *Digital Signal Processing*. Prentice Hall: NJ.

Papoulis, A. 1991. *Probability, Random Variables, and Stochastic Processes*. 3rd edn. McGraw-Hill: New York, pp. 345–346.

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

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