Robust Recursive Envelope Operators for Fast Retinex

Doron Shaked, Renato Keshet
HP Laboratories Israel
HPL-2002-74(R.1)
March 11, 2004*

E-mail: [dorons,renato]@hpli.hpl.hp.com

The effort to make efficient Retinex engines motivated the research reported in this paper. In most Retinex-type algorithms the computational engine is some (possibly non-linear) variant of a large support low-pass filter. The presented work relies on the fact that recursive (2D IIR) filtering is an efficient method to implement such large support linear space invariant filters. We propose a generalization of recursive filters to non-liner operators, and their use in the Retinex core.

In particular, we pay attention to two kinds of non-linearities, which are important in the context of Retinex type algorithms. First, the output should be an upper or lower envelope of the original image, and secondly, the operator should be robust in the sense that it does not use information from across an edge in the image. In an effort to reach stable algorithm parameters, namely parameters that would perform reasonably well for all images, we developed scale invariant operators, in which the nature of the obtained envelope is invariant to the size of the input image.

Operators presented in this paper provide Retinex enhancement that is comparable to the best alternatives, and can be computed in about half the time.

* Internal Accession Date Only
HP Laboratories Israel, Technion City, Haifa 32000, Israel

1. Introduction

Retinex is a theory modeling visual perception [19,20,21,22,23]. Retinex type algorithms are used for dynamic range compression, colored illumination correction, or general image enhancement [1,7,9,11,12,13,15,16,17,18,21,22,23,27,28,30,31,32]. Retinex theory is based on the physical imaging model: The perceived scene image, *S*, is a multiplication between the illumination, *L*, shed on visible surfaces and their respective reflectance[1], *R*.

$$S = R \cdot L. \tag{1}$$

The underlining assumption behind Retinex-like algorithms is that the illumination is an artifact. Therefore it is estimated and either removed completely: $\hat{S}_1 = S / L = R$, or partially: $\hat{S}_2 = S / f(L)$, where $f(L)$ is some function of the illumination [17,27,30]. Consequently, the estimation of *L* from *S* is the main algorithmic and computational issue for Retinex type algorithms.

Unfortunately, it is impossible (mathematically ill-defined) to estimate *L* from *S*, unless something else is known about either *L* or *R*. Indeed, most illumination estimation algorithms share the basic Retinex assumption that edges in the scene are edges in the reflectance, whereas illumination changes slowly. Thus, in most Retinex type algorithms, the estimated *L* is a smooth version of the input *S*. Consequently, in most implementations one can identify some kind of strong smoothing filter in the algorithm. Further characterization of the desired illumination estimate, *L*, however, leads to the following additional requirements:

1. ***The envelope requirement:*** [12,13,17,23,27] In a scene with no directly visible light sources, the reflectance is smaller than one, or in other words, surfaces cannot reflect more light than what is shed on them. Thus, the illumination *L* is a smooth local envelope rather than a local average, i.e., $L > S$ everywhere.
2. ***The robustness requirement:*** [7,9,18,31] *L* is piecewise smooth rather than smooth. This corrects the classical Retinex paradigm, and stems from a more accurate description of real world scenes. Typical scenes with illumination discontinuities are backlit or flash scenes, scenes with shadows, and mixed indoor-outdoor scenes. The piecewise smooth illumination model is also good for scenes with visible light sources and specularities. As elsewhere, ‘robustness’ implies that local average is not affected by outliers (in our case pixels across an edge). Thus, the estimated illumination is discontinuous in locations where the input image *S* has strong discontinuities. In those locations, the discontinuities in *L* are preferably similar to those in *S*.

[1] The reflectance is a function of the surfaces’ albedo, orientation with respect to the light source, and the observer, and more.

Let us focus now on the implementation. The computational bottleneck in Retinex algorithms is usually the smoothing filter, which is required to have a large support kernel[2]. The problem of designing an efficient smoothing filter is especially difficult when non-linear variants such as the envelope and robustness are required.

In order to devise a fast method, let us start by considering alternative implementations for convolution, and compare them in terms of computational complexity. Let us assume that the image and the kernel have square supports of size *NxN* and *KxK* respectively (with $N \geq K$).

Method	Complexity
Space domain	$\mathrm{O}\left(N^2 \cdot K^2\right)$
Frequency domain	$\mathrm{O}\left(N^2 \cdot \log(N)\right)$
Kernel decomposition	$\mathrm{O}\left(N^2 \cdot \log(K)\right)$
Pyramidal methods	$\mathrm{O}\left(N^2\right)$
Recursive filters	$\mathrm{O}\left(N^2\right)$

Table 1 – Complexity of Different methods for computing a linear convolution

- *Space domain* refers to direct convolution.
- *Frequency domain* refers to 2D FFT.
- *Kernel decompositions* is based on dyadic filters [4].
- *Pyramidal methods* are based on the Laplacian pyramid [5], and are linear in the input complexity. Nevertheless, for practical purposes we have to consider their constant, which is relatively high since they require fixed support convolutions in each pyramid layer.
- *Recursive filters* are 2D extensions of signal-processing's IIR filters [24], and were introduced to image-processing by Deriche [6].

For many of the above methods, one can find a robust/envelope version, i.e., a similar algorithm that returns an envelope rather than a linear averaging, or one that produces piece-wise smooth results. For example, of spatial envelopes can be obtained through classical mathematical morphology [26], and spatial robust filters are described in [29], and analyzed in [2,10]. Frequency domain, being the eigenspace of linear methods, is not easily amenable to the non-linear robust or envelope versions. Nevertheless, the slope transform [8] is the morphologic equivalent to the Fourier transform. As for Efficient Kernel Decompositions, J. McCann decomposed an envelope filter for his Retinex algorithm [13,23], where the result of each convolution is clipped to be above the input image, and R. Sobol added robustness by making each dyadic average a function of the two pixel values [27]. Pyramidal envelopes were suggested in [17], where the envelope was enforced at each level, and robustness was incorporated in [18], where robust metrics replaced the mean square measure in the energy functional.

[2] For image quality the support of the LP kernel has to be larger than the largest expected object, and thus, in the order of the size of the input image.

In this paper, we introduce a robust envelope algorithm based on Recursive Filtering. The algorithm is developed in the context of a Retinex type algorithm. We also elaborate on ways to determine stable filter parameters, and show that by making the filter scale invariant[3] it can be stabilized. We conclude this paper by suggesting a scheme for a scale-independent version of robust recursive envelope filters.

In Section 2 we review Recursive Filtering, and in Section 3 we generalize it to be a robust envelope filter. In Section 4, we make the filter stable by making it scale independent. In Section 5, we present Retinex results using the proposed algorithm, and compare it to alternatives. Section 6 is a Summary.

2. Recursive Filters

This section reviews recursive filters, however before we go on to describe them let us adopt a standard notation *S* and *L* for their respective input and the output. We chose it for compatibility with the Retinex notation in (1).

Recursive filters were introduced to image processing by Deriche [6]. They are 2D extensions of signal processing IIR filters. Let us consider a causal 1D direct convolution

$$L_i = \sum_{n=0}^{N-1} S_{i-n} \cdot K_n \tag{2}$$

When *N* is large, the computational complexity of the direct convolution might be prohibitive. On the other hand, the general formulation of the causal 1D IIR filter is

$$L_i = \sum_{n=0}^{m-1} S_{i-n} \cdot b_n + \sum_{n=1}^{n-1} L_{i-n} \cdot a_n \tag{3}$$

and it is easy to see that the simplest recursive filter

$$L_i = \alpha \cdot L_{i-1} + (1-\alpha) \cdot S_i \tag{4}$$

is equivalent to

$$L_i = \sum_{n=0}^{\infty} S_{i-n} \cdot K_n , \qquad \text{with} \qquad K_n = (1-\alpha) \cdot \alpha^n \tag{5}$$

where K_n is the an exponential kernel having an infinite support. Notice however, that the effective support is in the order of 1/α. Unlike the direct formulation, in the IIR formulation it is possible to change the size of the effective support of the filter with no

[3] A filter is scale invariant when its output for a scaled input is a scaled version of the original output (or at least very nearly so).

additional computations. One can also design IIR filters with a profile other than exponential [24].

When non-causal filters are required (as is the situation in images), we can implement $K_n = \frac{1-\alpha}{1+\alpha}\cdot\alpha^{|n|}, \quad \forall n$, as a sum of two recursive filters: a causal and an anti-causal filter. Define

$$\begin{aligned} L_i^+ &= \alpha\cdot L_{i-1} + (1-\alpha)\cdot S_i \\ L_i^- &= \alpha\cdot L_{i+1} + (1-\alpha)\cdot S_{i+1} \end{aligned} \tag{6}$$

and then

$$L_i = \frac{1}{1+\alpha}\left(L_i^+ + L_i^-\right) \tag{7}$$

In image processing one usually requires 2D separable filters, i.e., filters for which $K(i,j) = K_x(i)\cdot K_y(j)$. In this case, implementation is separable, namely, one first filters each row using K_x, and then each column of the output using K_y.

3. From Recursive Linear Filtering to Robust 2D Envelopes

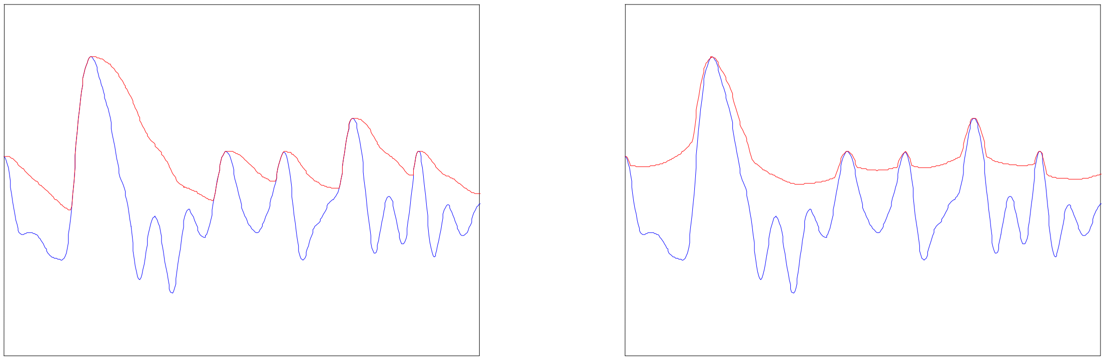
In this section, we derive a scheme for a robust 2D envelope version of recursive filtering. We derive it gradually: In Subsection 3.1 we develop a 1D recursive envelope filter, then in Subsection 3.2 we make it robust, and finally in Subsection 3.3 we revisit the generalization to 2D in the robust envelope context.

3.1 Envelope Filters

The first step to do is to force the envelope condition $L_i \geq S_i, \ \forall i$. This can be done directly on the recursive formulation (6) resulting in

$$\begin{aligned} L_i^+ &= \max\left\{\alpha\cdot L_{i-1} + (1-\alpha)\cdot S_i, \quad S_i\right\} \\ L_i^- &= \max\left\{\alpha\cdot L_{i+1} + (1-\alpha)\cdot S_{i+1}, \ S_i\right\} \end{aligned} \tag{8}$$

We see that now by construction $L_i^+ \geq S_i$, $L_i^- \geq S_i$, and thus from (7) also, $L_i \geq S_i$.



a b

Figure 1: A 1D signal in Blue, and its causal envelope in (a), and the average (7) of its causal and anti-causal envelopes in (b).

Figure 1a presents a signal *S* in Blue, and its causal envelope L_i^+ in Red. Figure 1b presents *S* and *L*, which portrays some undesirable characteristics around locations where the envelope limitation is being enforced. At those locations, the output *L* is almost identical to *S*, and consequently the reflectance *R* (1) will be saturated.

The solution that we propose to this problem is to change the non-causality paradigm. Instead of two parallel causal and anti-causal filters, use a cascade of the same filters, namely:

$$\begin{aligned} L_i^+ &= \max\left\{\alpha \cdot L_{i-1} + (1-\alpha)\cdot S_i, \quad S_i\right\} \\ L_i &= \max\left\{\alpha \cdot L_{i+1}^+ + (1-\alpha)\cdot S_{i+1}, \; S_i\right\} \end{aligned} \tag{9}$$

Figure 2 presents a signal *S* in Blue, and the cascaded causality envelope *L* of (9) in Black, which is much more like an envelope. Figure 2 also presents two other envelopes created with different α parameters, lower α values correspond to 'more elastic' (and thus lower) envelopes.

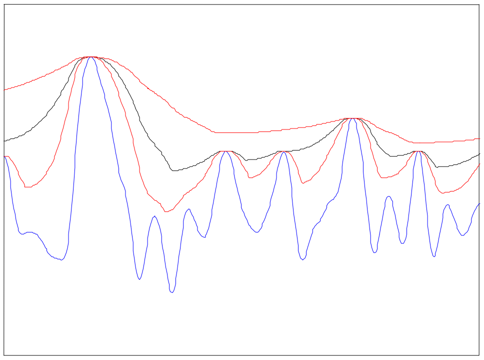


Figure 2: A 1D signal in Blue and three consecutive envelopes in Red and Black, the latter corresponding to the same α value as the envelopes in Figure 1.

It has to be noted that the proposed envelope, *L,* has some disadvantages:

1. It is not symmetric, and flipping the input signal will not result in a flipped envelope. This asymmetry is depicted in Figure 3, where the Red envelope of the Blue signal does not match the Black flipped envelope of the flipped input signal.
2. There is a derivative discontinuity artifact in the valleys of the envelope. It is more visible in the more elastic envelopes (lower αs). This artifact is less noticeable in images. This is especially true in parameter ranges used for the Retinex application.
3. Although parallel causality L obtained by substituting (8) into (7) is not what one can expect from an envelope, it seems very close to what we will try to achieve in the next subsection, i.e. robust envelopes. You may note the way L 'drops' at locations where S has sharp discontinuities, and how it 'floats' over small discontinuities in S. This 'short cut robustness' was nevertheless found to be less desirable relative to the alternative of the robust L as will be detailed in the following subsection.

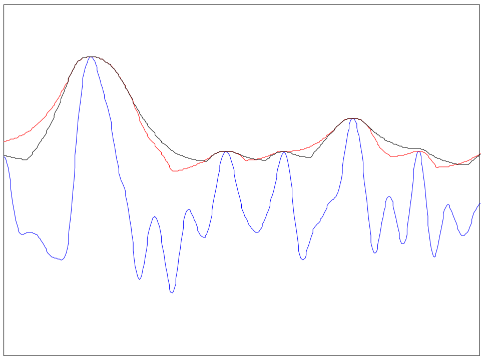


Figure 3: Asymmetry of the consecutive envelope, a 1D signal in Blue, its envelope in Red, and the flipped envelope of its flipped version in Black.

3.2 Robust Envelopes

The next step is to make the envelopes robust, namely, in locations where the input has sharp discontinuities, the envelope should have similar discontinuities, and in locations where the input has small discontinuities or is altogether smooth, the envelope should remain smooth and follow the envelope at the rate prescribed by α.

Reformulating the above in terms of recursive filters parameterized by α, we have that the envelope should have a constant α everywhere except on sharp discontinuities, where α should be smaller. Thus, α should be a function of the local input gradient. When the gradient is low the new gradient-dependent parameter, $\alpha(\nabla S)$, should approximately obtain the original value α_0, and when the gradient has large negative value, $\alpha(\nabla S)$ should be smaller. Note that when the gradient has large positive values we do not want $\alpha(\nabla S)$ to be small. In this case, discontinuities will occur due to the envelope requirement, otherwise, when the envelope is higher than the input, a smaller α value will bring the enveloped down, creating a discontinuity that is inverted relative to the input (and thus undesirable)[4].

Making algorithms robust by turning constant parameters to be a function of the input has been implemented elsewhere [2,9,30], and most notably [3]. Figure 4 illustrates a Huber type of robust profile.

$$\alpha(\nabla S)=\begin{cases}\alpha_0 & \nabla S \geq -\dfrac{1}{T} \\ -\dfrac{\alpha_0}{\nabla S \cdot T} & \nabla S < -\dfrac{1}{T}\end{cases} \tag{10}$$

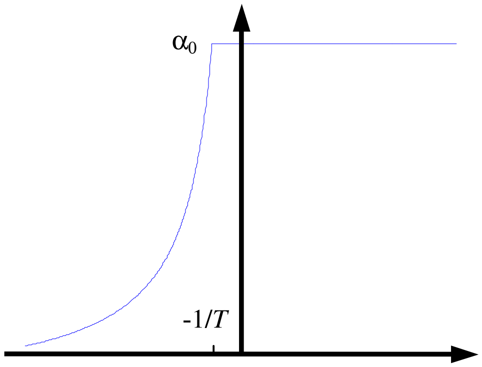


Figure 4: A Huber influence function for α.

The robust version of (7) is thus

$$\begin{aligned} L_i^+ &= \max\{\alpha(\nabla S_i)\cdot L_{i-1} + (1-\alpha(\nabla S_i))\cdot S_i,\ S_i\} \\ L_i &= \max\{\alpha(-\nabla S_{i+1})\cdot L_{i+1}^+ + (1-\alpha(-\nabla S_{i+1}))\cdot S_{i+1},\ S_i\} \end{aligned} \tag{11}$$

Figure 5 presents L_i^+ of (11). It is the robust version of Figure 1a. The dotted vertical lines indicate local minima of the input gradient below the threshold *–1/T* (10). Note that in regions where the threshold was not crossed, output is very similar to that of Figure 1a, elsewhere, the difference depends on the gradient.

[4] A related issue is exemplified by an input region of low values surrounded on both sides by high values such that on one side the change is sharp, and on the other side it is smooth. The envelope cannot both maintain the sharp discontinuity and be smooth over the mild transition. To solve this, we let smoothness take precedence over robustness. The precedence issue will become even more delicate in the 2D case, see Subsection 3.3.

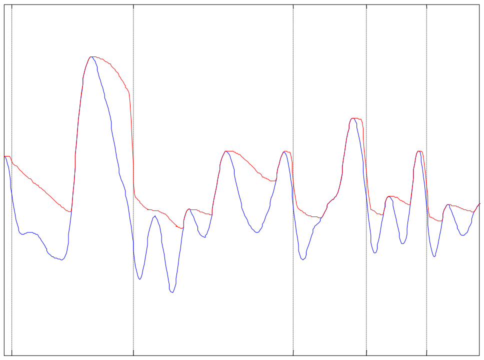


Figure 5: Robust causal envelope

3.3 Extension to 2D

The last stage in generalizing recursive filters to robust envelopes is to revisit the generalization of linear recursive filters to 2D. Figures 6b and 6c are respectively the non-robust and robust envelopes (separable application of (9) and (11)) of the input image in Figure 6a. The envelope required by the Retinex algorithm is such that major structures such as the shadow and highlights are preserved in the envelope and can thus be corrected for, but details are removed. Making the envelope 'posterized' leaves the depth and details in the reflectance image. In that respect, the robust envelope is much better than the non-robust one.

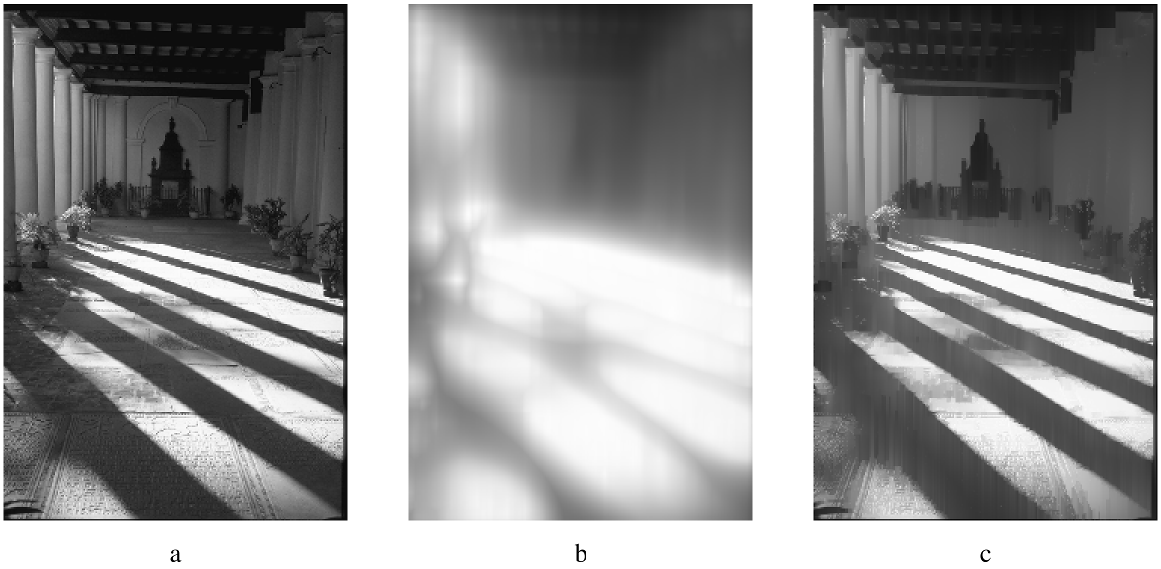


Figure 6: Input image in (a) with non-robust and robust envelopes in (b) and (c) respectively.

However, when one takes a closer look at the images, an inherent problem of the generalization to 2D becomes apparent. Figures 7a, and 7b are a zoom in on the lower column shadow of the images in Figures 6a and 6c respectively. The required loss of detail of the envelope is accompanies by artifacts in the Y direction

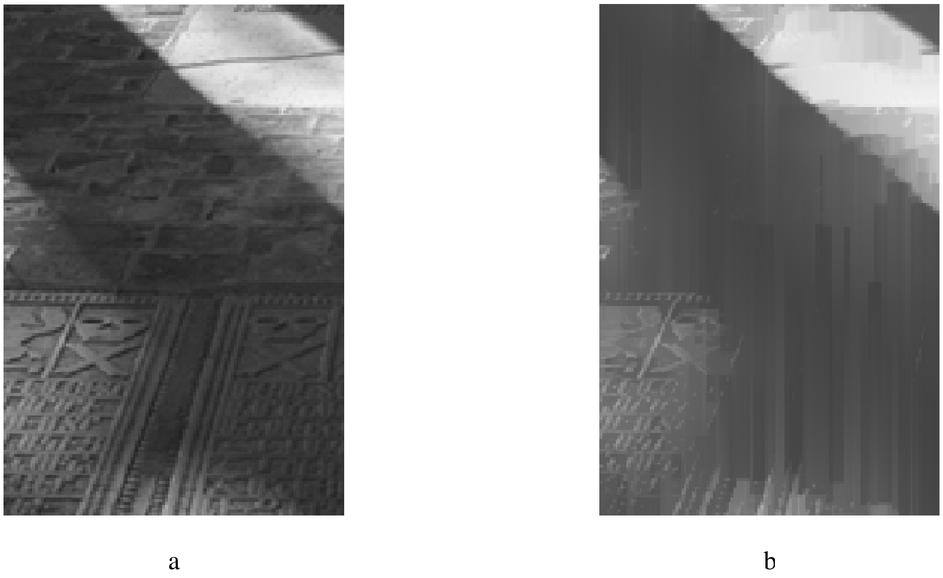


Figure 7: Zoom in on the input image in (a) and the robust envelopes in (b).

One explanation for the artifacts is that the 1D filter in the Y direction is applied along each of the image columns, and is independent of the information in nearby columns. The non-robust envelope is not affected by this problem since the X direction filter produces smooth outputs, which means that nearby columns of its output are similar. Combining this with the fact that similar inputs to the Y filter result in similar outputs, amounts to a smooth 2D output. Neither of the above 'regularizing' features is true for the robust case.

One can further analyze the difference between the robust and the non-robust envelope cases, by considering the recursive filter in terms of information flow. In each of the recursive passes (we have a forward pass and a backward pass in each dimension) information flows with the recursion along a row or column. Thus in a 1D filter every pixel gets information from pixels preceding it (during the forward pass) and from pixels following it (during a backwards pass). In a 2D filter, the information flows along rows and then along columns. Since the column filter operates on the result of the row filter, every pixel 'has access' to information from all the other pixels in the image. Note however, that the information from one pixel to the other flows in a single predetermined path – first along the row of the source, and then along the column of the destination pixel.

In the robust case, information does not flow freely. It is rather blocked at locations where the gradient of the input is below a threshold. In the 1D case, this is not a problem because that is what robustness is all about – we do not want to consider information from across an edge. Nevertheless, in the 2D case two pixels might be in the same segment (i.e. not across an edge from each other), even though the row + column path connecting them might cross edges. Thus any flow block on the single path, connecting two pixels could prevent them from considering each other, where in fact they should, as long as there is even a single unblocked path connecting them.

In Subsection 3.1, we determined that 1D Robust Envelopes should be implemented as cascaded recursive filters rather than in parallel. This opens the way to dimension interleaving described in block diagram in Figure 8. Interleaving is the simplest way to increase information flow between pixels.

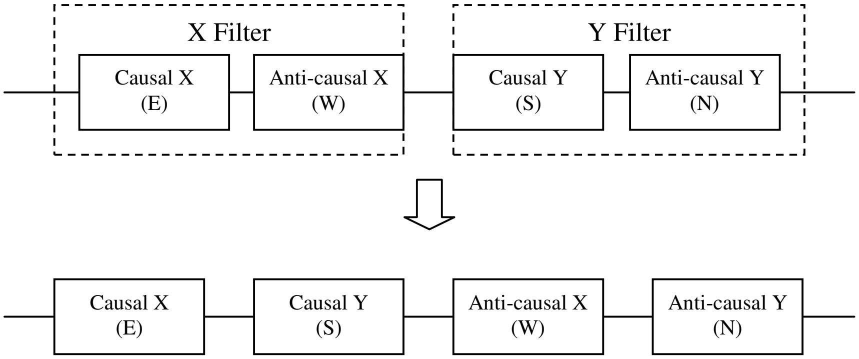


Figure 8:Dimention interleaving.

In the sequel, we use a compass notation referring to different causal/non-causal and Y/X directions. For example, the first line in Figure 8 would be referred to as an E,W,S,N flow, whereas the second line would be called an E,S,W,N flow.

While not necessarily a good quantitative model, information flow diagrams presented in Figure 9 are a good way to illustrate the problem and evaluate alternatives. In each of the eight flow diagrams in Figure 9, the Red zone represents information and the bold lines represent information flow barriers. Letting the information flow in a direction sequence detailed on the left of each row of diagrams results in the information being barred from a zone represented by the shaded zone. Each of the four diagrams on the first row in Figure 9 is thus the result of an E,W,S,N flow sequence, whereas each of the diagrams on the second line is the result of an E,S,W,N flow sequence.

Though somewhat simplistic, the information flow barriers illustrated in Figure 9 are common in natural images. It is evident that while helpful, using the interleaved sequence (E,S,W,N) as opposed to the original sequence (E,W,S,N) is not a full solution to the problem. Alternatives exist; however, they come with a computational cost. One such alternative involves repetition: e.g., ‘E,S,W,N,E’, ‘E,S,W,N, E,S’, etc.

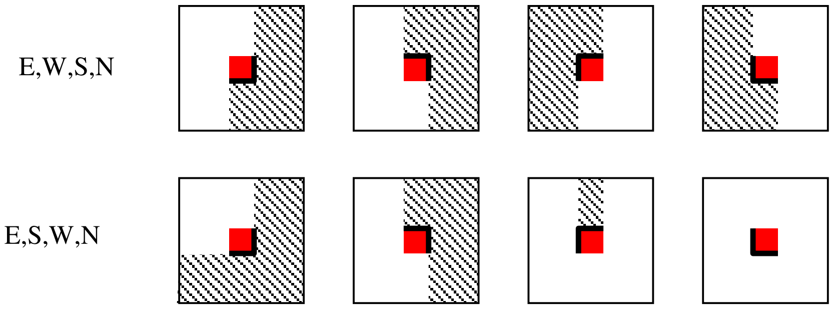


Figure 9: Information flow illustration for simple dimension interleaving.

Another alternative is to turn the passes into “intrinsic” 2D passes, i.e., instead of getting information from a single pixel, each pixel gets information from two pixels, one in each dimension. Let us consider, as an example, one such 2D pass, which we denote by SE, in accordance with the compass notation:

$$\begin{aligned} L^{SE}_{i,j} = \max\{&\alpha(\nabla_x S)\cdot L_{i-1,j} + (1-\alpha(\nabla_x S))\cdot S_{i,j}, \\ &\alpha(\nabla_y S)\cdot L_{i,j-1} + (1-\alpha(\nabla_y S))\cdot S_{i,j}, \quad S_i\} \end{aligned} \tag{12}$$

Four such passes can be cascaded to create a sequence, such as ‘SE,SW,NW,NE’. The resulting 2D schemes are in the spirit of classical sequential processing schemes, proposed in [25] for various computer vision algorithms.

Figure 10 illustrates the information flow of some alternatives in a more complex barrier system.

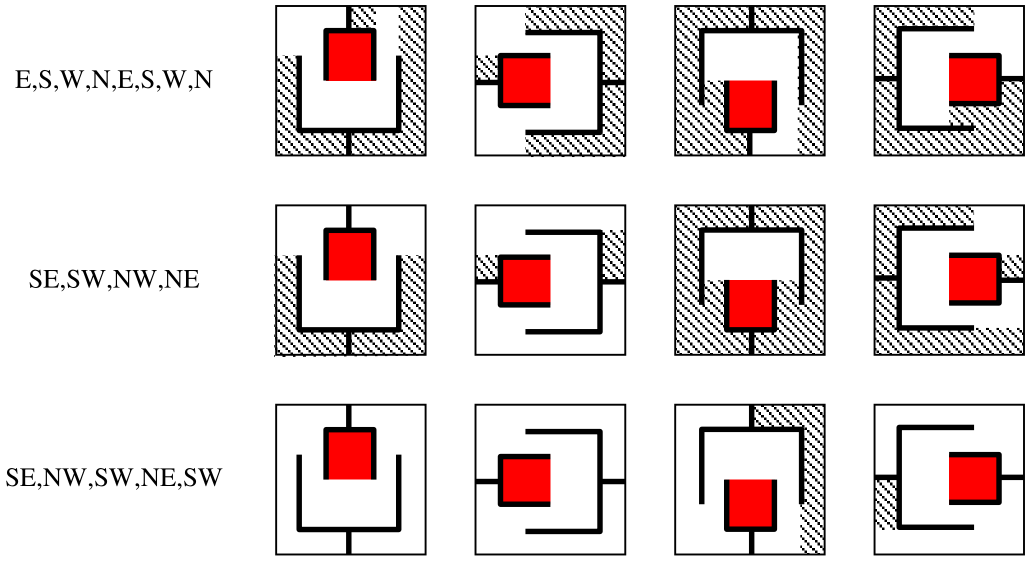


Figure 10: Information flow illustration for combinations of interleaving, repetition, and 2D schemes.

Information flow artifacts should be evaluated both in terms of quality and computational complexity. For example, the 8 1D passes described in the first row of Figure 10 are more complex than the 4 non interleaved 2D passes in the second row of the same figure. Furthermore, according to the barrier model of Figure 10 they will usually result in more artifacts. We conclude that in general 2D schemes perform better than 1D schemes on the trade-off between quality and complexity.

Artifacts such as in Figures 9 and 10 are indeed visible in images with corresponding edge structures. However, since information barriers are not absolute, artifacts in complex schemes are usually fuzzy. In the rest of the paper, we use the 5 interleaved 2D pass scheme described in the third row of Figure 10.

4. Scale Invariance and Parameter Tuning

This section deals with the following problem: Given two images of different sizes, one being a decimated version of the other, how can one make sure that the envelope of the former is approximately equal to the decimation of the envelope of the latter?

Scale invariance is important because of two reasons: First, the Retinex paradigm [19] is scale invariant, and relates only to the composition of colors in the visible field of view and not to their scale. More importantly however, we were motivated by the need to find stable parameters for the algorithm. By stable parameters, we mean a set of parameters, which will perform reasonably well for all natural images. While examining images at different scales, we found that it is typically possible to find a good set of parameters for small images. However, tuning the parameters becomes harder as images grow larger, and is practically impossible above a certain scale. On the other hand, in the scale invariant version of the algorithm (presented below), parameters are stable, and the algorithm performs equally well at all scales.

The proposed recursive operators are scale dependent by virtue of three components relating to the scale of the image (or more accurately the size of its pixels):

- The parameter α determines the size of the LPF in pixels.
- The recursion is defined on pixels.
- The gradient ∇S is defined on pixels.

In subsection 4.1, we redefine the robust exponent $\alpha(\nabla S)$ in order to make the algorithm scale independent. The new definition results with the desired scale independent robust envelopes, with the exception of sporadic artifacts due to rare configurations of high-scale details in the input image. Subsection 4.2 proposes a pre-process that eliminates those artifacts.

4.1 Redefining the Robust Exponent

The robust exponent, $\alpha(\nabla S)$, is a function of the local gradient of the image, such as the function defined in (10) and depicted in Figure 4. In this subsection, we modify it to be scale independent. The following subsections address each of the three scale dependent components detailed above, and modifies the robust exponent function in order to eliminate it.

4.1.1 Removing Parameter Dimensionality

Let us first consider the parameter α_0, of $\alpha(\nabla S)$ in (10), which was set to correspond to a prescribed effective kernel support of the original recursive linear filter (4). Let us therefore see what it takes to make α_0 scale invariant in the linear case. In order for a kernel to be scale invariant it has to change its size with the input image. Therefore the value K_j of the kernel at the j^{th} location will be a function of the size, N, of the image. For the kernel to be scale invariant, the coordinate j is scale by N. In other words, we define a new kernel $K_{j/N}$, which is equal to the original K_j for some desired image size N_0. Since the kernels are Laplacian $K_j = C \cdot \alpha_0^{|j|}$ we have that $K_{j/N} = C_N \cdot \alpha_0^{\frac{N_0 \cdot |j|}{N}} = C_N \cdot \alpha_0(N)^{|j|}$, with

$$\alpha_0(N) = \sqrt[N]{\alpha_0^{N_0}} \tag{13}$$

where N_0 is some reference image size (in our case N_0=256).

4.1.2 Removing Recursion Dimensionality

Assuming a 2:1 decimation rate, an edge with a two-pixel interface transition in the high-resolution image has a single-pixel interface transition in the low-resolution image. In the low-resolution, the filtering recursion crosses the same edge in a single step rather than in two consecutive ones, and nevertheless the total effect has to be the same. If $\alpha(\nabla S)$ were a constant, as in the linear case, then the above would be true by virtue of the previous subsection. The robustness of the filter, caused by the variation in $\alpha(\nabla S)$, requires however further considerations in order to gain scale invariance at those regions where $\alpha(\nabla S)$ is not constant. In this subsection, we consider the 'robust' part of $\alpha(\nabla S)$. In (10) we chose the Huber profile from several alternatives in [3]. None of which is however scale invariant.

Consider a hypothetical edge stretched over three pixels in the high resolution image, inducing gradients $\nabla x_i^1 = g_1$ and $\nabla x_{i+1}^1 = g_2$. The same edge stretches over two pixels in the low-resolution image, inducing a gradient $\nabla x_{i/2}^2 = g_0$.

We naturally have

$$g_0 = g_1 + g_2 \tag{14}$$

Assuming in addition that all edges have passed the threshold (i.e. g_1, $g_2 < -T$), scale invariance requires

$$\alpha(g_0) = \alpha(g_1) \cdot \alpha(g_2) \tag{15}$$

From (14) and (15) we have that the robust exponent function should itself have an exponential profile

$$\alpha(g) = K \cdot e^{\beta \cdot g} \tag{16}$$

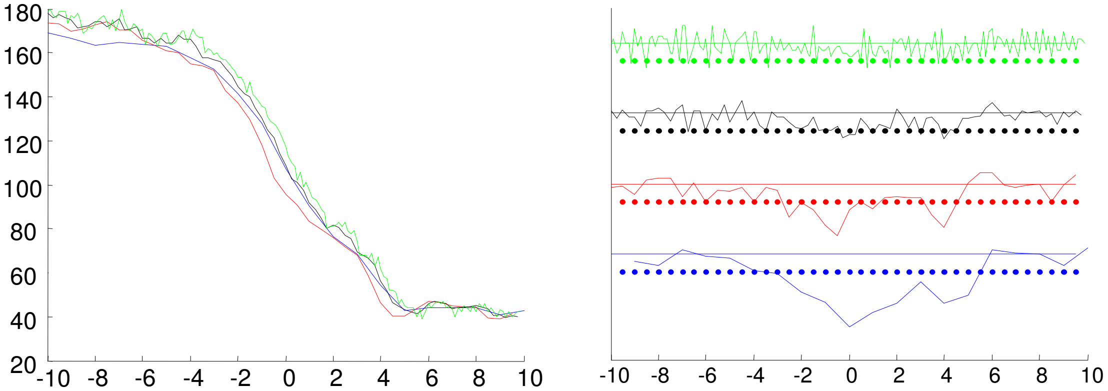
and, since $\alpha(0)=\alpha_0$, then $K=\alpha_0$. The remaining parameter β, parameterizes the desired degree of robustness, starting from non-robust, with $\beta=0$, and up.

4.1.3 Removing Gradient Dimensionality

We now reach the stage where we have to select the threshold *T*. We will use a rule of thumb determining that edges with total variation of less than 10 gray levels (in the standard gray scale *[0,255]*) are detail edges, and regions surrounded by edges with larger total variation represent illumination or object edges.

What makes the above statement scale independent is the use of the term total variation instead of gradient. Consider the example illustrated in Figure 11, where a cross section of an edge is plotted in Figure 11a at four scales (the X axis is in the coarse scale units). Figure 11b illustrates the gradient functions of the four cross sections. Where for each scale the solid straight line is the corresponding X axis, and the dotted line is a *T=-5* threshold. At larger scales, the edge stretches over a larger number of pixels, and consequently the gradient at each location is smaller. Ultimately, edges due to fine details or noise become dominant in the gradient function, whereas the gradient at the real edge is doomed to drop below any threshold. The total variation of the edge (the difference between its left and right sides in Figure 11a) is however, constant throughout the scale space, and thus scale independent.

The above proposition to determine the relevance of an edge to robustness according to its total variation rather than local gradient is conditioned on our ability to tell an edge from other structures (e.g. light gradations as those across columns in Figure 6a). Fortunately, we found empirically that for the Retinex application, using a threshold on the gradient of natural images performed consistently well at small scales (approximately 256x256 pixels).



a

b

Figure 11: Gray values (a) and gradients (b) of an edge in various scales.

We conclude that the span of 2 pixels in scale N_0=256 is a good basis for a practical definition of total variation in our application.

Note however that one cannot simply replace the gradient with a gradient at scale N_0 if one wants to preserve the edge details of the original image scale $N>N_0$. The base scale N_0 should rather be used only in the threshold. Thus, we define two different gradients, the standard pixel based gradient ∇S, and the scaled gradient $\nabla_N S$.

$$\begin{aligned} \nabla S &= S(i) - S(i-1) \\ \nabla_N S &= S(i+\Delta_N^+) - S(i-\Delta_N^-) \end{aligned} \tag{17}$$

where

$$\Delta_N^+ + \Delta_N^- = \left\lfloor {}^{N}\!/_{N_0} \right\rfloor, \quad \text{and} \quad \Delta_N^+ + 1 \ge \Delta_N^- \ge \Delta_N^+ \tag{18}$$

Considering also (13) and (16), the scale independent robust exponent is

$$\alpha(\nabla S, \nabla_N S, N) = \begin{cases} \sqrt[N]{\alpha_0^{N_0}} & \nabla_N S \ge -T \\ \sqrt[N]{\alpha_0^{N_0}} \cdot e^{\beta \cdot \min\{\nabla S, 0\}} & \nabla_N S < -T \end{cases} \tag{19}$$

4.2 Pre-Filter

It should be noted that, although all the components of the robust exponent in (19) have been modified with scale independence in mind, it is not strictly scale independent due to small-scale details. Namely, scale independence has been mainly achieved by adopting a base scale (N_0=256). To make the algorithm strictly scale independent all the details above the base scale will have to be ignored. However, we do not want to ignore all of them because it will ruin image quality. Let us therefore analyze the cases where the proposed method fails and eliminate only those small-scale details necessary to eliminate the scale artifacts.

Figure 12 depicts a typical failure of the scale independent recursion. A segment of an image appears on the left and the proposed scale-invariant output in the middle. The output of the first 2D pass (12), shown to the right, practically points at the local origin of the artifacts.

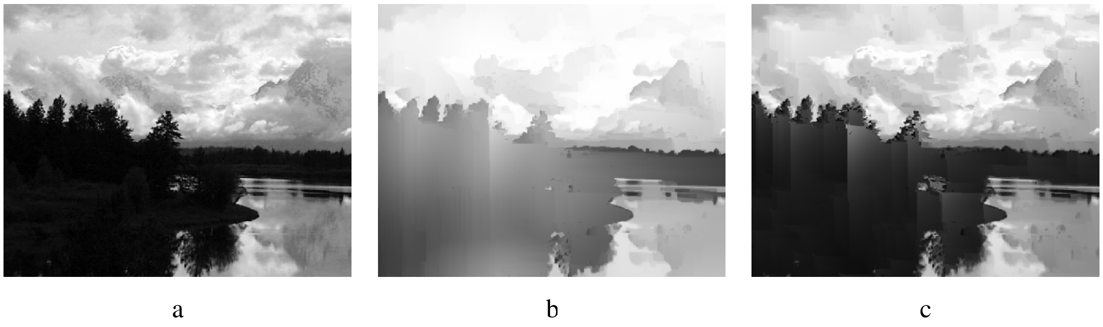


Figure 12: A typical artifact of the scale invariant algorithm: (a) Input, (b) output, (c) output of first 2D pass.

Analyzing the origins of the artifacts, we isolated two problematic high-scale configurations, the roots of which can be traced to our definition (17) of the scaled gradient $\nabla_N S$. The bright-dot configuration in Figure 13a$_1$ is a bright spot whose gray-difference from the background should have defined it as an edge but is, nevertheless, not recognized as such since it is missed altogether by $\nabla_N S$. The Blue arrow points at the pixel (just right off the dot) where the scaled gradient fails to detect the gradient (there is a symmetric location just in front of the dot). The Cyan dots mark the locations Δ_- and Δ_+ from (17) according to which $\nabla_N S = 0$. Consequently, the bright dot's brightness information flows freely to fill the dark segment as in Figure 13a$_2$. The dark-canyon configuration in Figure 13b$_1$ is essentially the dual of the bright-dot configuration.

We propose to solve both problems using the dual morphological grayscale filters opening and closing [26]. Considering the image as a surface, the 'close' filter literally closes all the holes and scratches into which a structuring element cannot penetrate, whereas 'open' is the dual operation. Thus, using a KxK flat structuring element, where

$$K = \left\lfloor \frac{N}{N_0} \right\rfloor \tag{20}$$

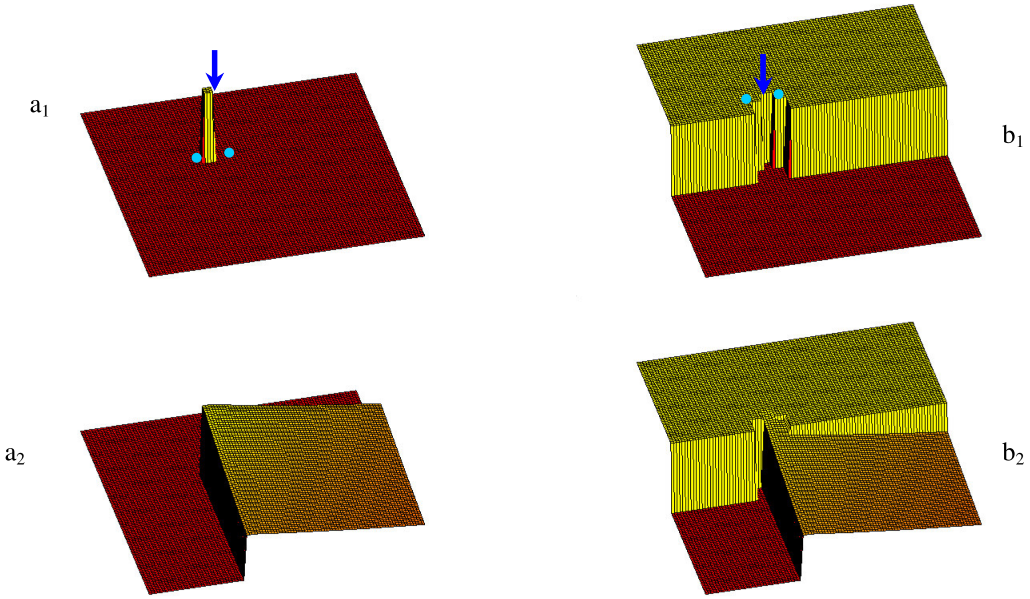


Figure 13: Bright-dot (a), and Dark-Canyon (b), configurations (1), and artifacts (2).

‘open’ eliminates bright-dot configurations, whereas close eliminates and dark-canyons. It should be noted that there are fast algorithms for both grayscale filters [14].

Notice however that, if we replace the input image by a filtered version, we get an envelope of the filtered version, which is not necessarily an envelope of the input. Thus, we need to augment the algorithm by a maximum operation between the output and the input – ensuring the envelope condition.

We conclude that, if the input image is first filtered by the appropriate open and close filters, we can safely apply the proposed envelope filters. Figure 14 depicts the scale independent robust envelope obtained by the proposed algorithm.

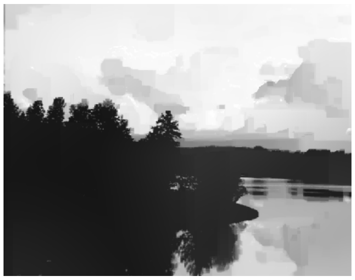


Figure 14: Scale independent robust envelope of Figure 12a. Small-scale artifacts removed.

5. Results

In this section, we present and analyze the results of the proposed algorithm. Its details and parameters as well as those of the corresponding Retinex correction are as follows:

Retinex -Pre:	$S = \max\{R,G,B\}$	
Prefilter:	Morphological Opening followed by morphological Closing.	
Recursive Filter:	NW, SE, SW, NE, NW.	
α_0:	0.945	
N_0:	200	
β:	0.15	
T:	7	
PostFilter:	Maximum	
Retinex-Post:	$\{R,G,B\} \leftarrow 255\,\{R,G,B\} / L^{(1-1/\gamma)}$	(see [17] for details)
γ:	2	

Figure 15 presents an input image in (a), its gamma correction in (b), the robust envelope obtained by the proposed algorithm in (c), and the resulting Retinex correction in (d). The gamma correction of (b) was tuned to result in the same general brightness of (d). As the reader can see, the details in the Retinex output are significantly more contrasted.

In our terminology, the benefits of the Retinex algorithm are expressed in the envelope's ability to be both flat where contrast has to be maintained, and to follow the input's edges, where the contrast can be traded-off for reduction in the overall dynamic range. This image is especially difficult in that aspect, due to the fractal interface between the shaded foreground and the illuminated background.

Next, we compare the proposed recursive envelope filters to an alternative method that yields similar robust envelopes. The pyramidal robust envelope described in [Rx] is a pyramidal solution to a variational formulation of the robust envelope extraction problem. Similar to the proposed recursive filters, it has linear complexity (see Table 1 in the Introduction), however, due to the iterative nature of differential equation solved in each of the pyramidal layers, the constant is relatively large. For images of 1M pixels, the execution time for the proposed algorithm is about half that of the pyramidal method. It has to be noted that about half of this time is spent on the morphological pre-process.

Figure 16 shows the envelopes and the corresponding Retinex enhancements for the proposed algorithm and the pyramidal Retinex, side by side. It is evident that, in both methods, envelopes are, as required, smooth and robust. Nevertheless, a few differences should be noted.

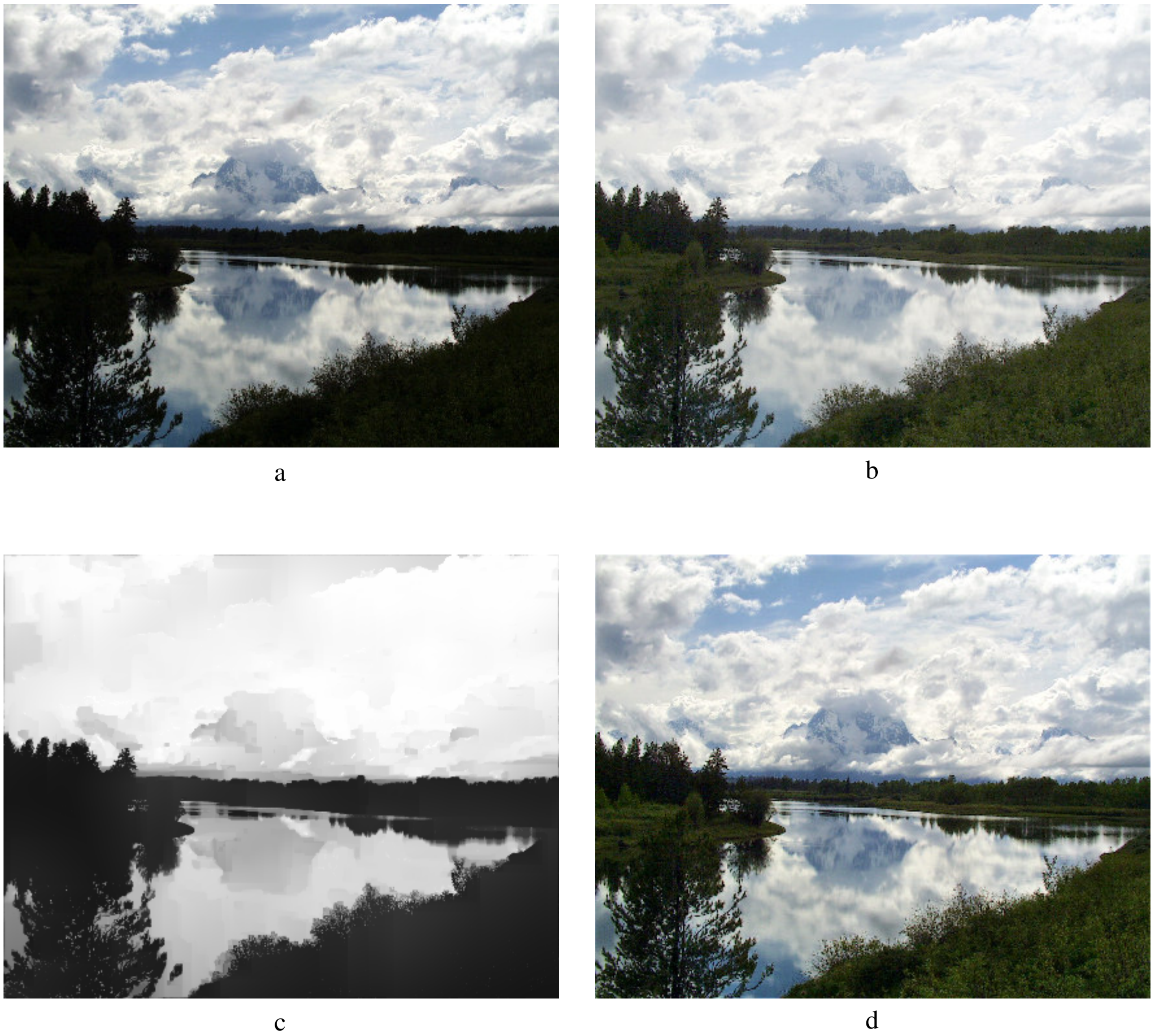


Figure 15: An input image (a), its gamma correction (b), the robust envelope obtained by the proposed algorithm (c), and the resulting Retinex correction (d).

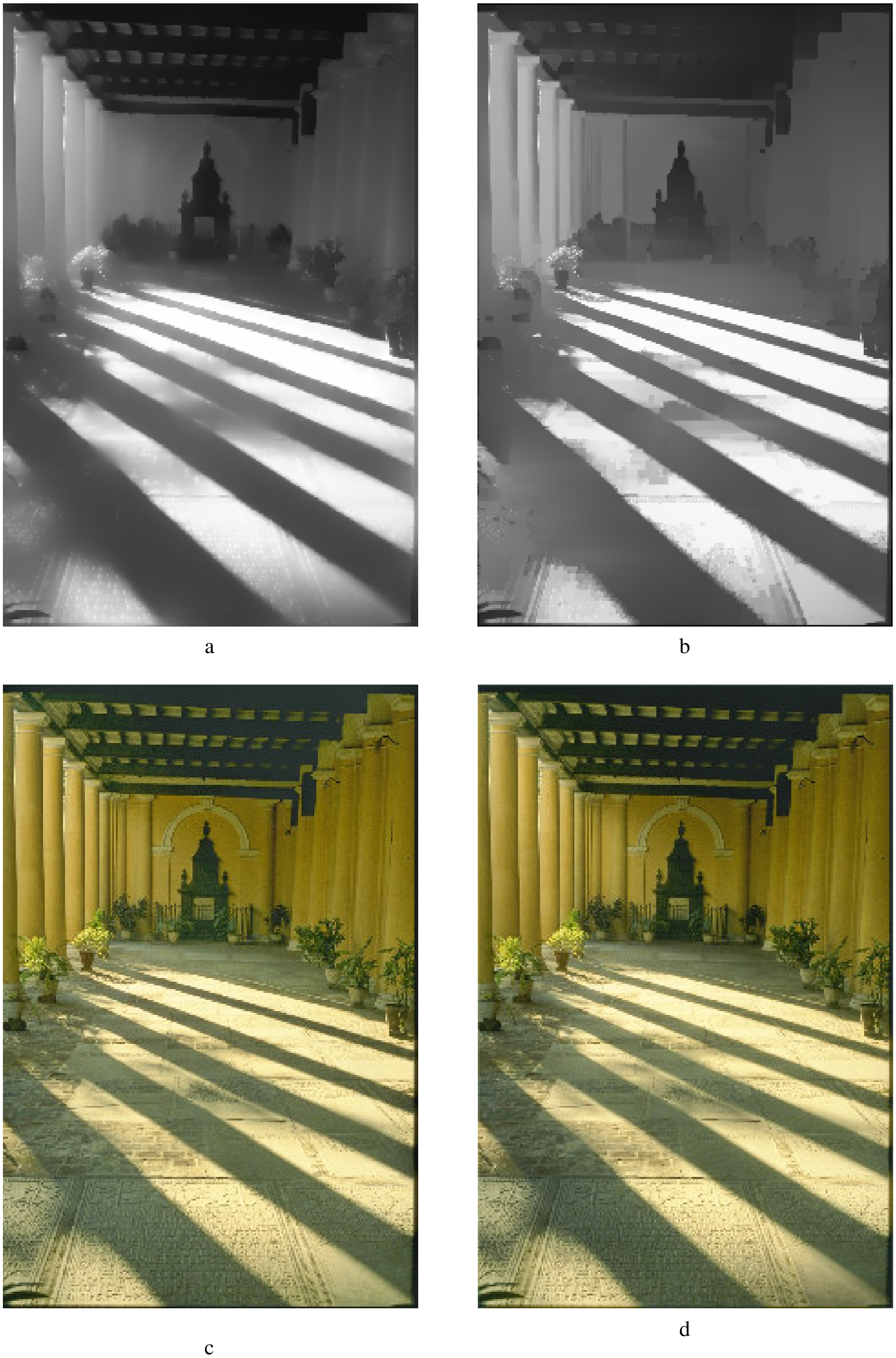


Figure 16: Pyramidal envelope (a) and corresponding Retinex correction (c).
The proposed envelope (b) and corresponding Retinex correction (d).

Let us first characterize the scale invariance of the two algorithms:

The pyramidal Retinex is scale invariant in the global sense. Namely, the nature of the envelope image does not depend on the size of the input image. However, it is not locally scale invariant, meaning that features of different size in the image are treated differently, and specifically, the smaller a feature is, the more it is treated as detail rather than as a feature. The image in Figure16 is a good example where the illumination for each of the column shadows is corrected differently according to its width.

The proposed recursive filter is locally scale invariant up to a certain 'threshold scale'. The threshold scale is dependent on the size of the image, thus maintaining the global scale invariance of the algorithm. Whereas in scales which are clearly above the threshold this seems as an advantage, in features whose scales are around the threshold scale this might introduce some artifacts, e.g., in the sporadic details visible on the highlighted floor in Figure 16b.

In addition, an 'information-flow' type artifact (see Section 3.3) is marginally visible in the main shadow of Figure 16b[5]. Furthermore, artifacts due to the gray threshold, *T*, are visible on the profile of the highlighted columns, where the intensity profile has a gradually increasing slope, which is bound to break the threshold at some point or the other. Note that since it is a zero sum game, the more we see depth in the envelope the less it is evident in the enhanced image. Thus, the highlighted columns, which broke threshold T, have a reduced 3D sense compared to the pyramidal enhancement, whereas the other columns have a better 3D sense due to their flatter envelope.

6. Summary

In this paper, we have generalized recursive (2D-IIR) filters to non-liner operators. Particularly, we addressed two non-linearities: Envelope operators, and robust operators. Their combination was used in the context of Retinex type image enhancement. We have further modified the robust envelope operators to be scale independent, making sure that the output of scaled inputs will be correspondingly scaled versions of each other. We have argued, and shown, that scale invariant operators have a more stable set of parameters, namely for a fixed set of parameters, the operators perform reasonably on a wide variety of natural images. Naturally, superior performance will usually be obtained when parameters are tuned for a particular image.

The comparison of the proposed envelope extraction algorithm to alternatives places it at comparable quality, and improved efficiency.

[5] Not visible in the printed version

8. References

[1] A. Blake, "Boundary Conditions of Lightness Computation in Mondrian World", *Computer Vision Graphics and Image Processing*, Vol. 32, pp. 314-327, 1985.

[2] D. Barash, "A Fundamental Relationship Between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation", to appear in *IEEE Trans. on PAMI*.

[3] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust Anisotropic Diffusion", *IEEE Trans. on Image Proc.*, Vol. 7, pp. 421-432, 1998.

[4] P. J. Burt, "Fast Filter Transforms for Image Processing", *Computer Graphics and Image Processing*, Vol. 6, pp. 20-51, 1981.

[5] P. J. Burt, and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Trans. on Communication*, Vol. 31, pp. 532-540, 1983.

[6] R. Deriche, "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector", *Int. J. of Computer Vision*, Vol. 1, pp. 167-187, 1987.

[7] J. M. DiCarlo, and B. A. Wandell, "Rendering High Dynamic Range Images", in *Proc. SPIE Vol. 3965*, pp. 392-401, 2000.

[8] L. Dorst, and R. van den Boomgaard, "Morphological signal processing and the slope transform", Signal Processing, Vol. 38, pp. 79-98, 1994.

[9] F. Durand, and J. Dorsey, "Fast Bilateral Filtering for the Display of High Dynamic Range Images", preprint in < http://graphics.lcs.mit.edu/~fredo/DurandBilateral.pdf >.

[10] M. Elad, "On the Bilateral Filter and Ways to Improve It", to appear in *IEEE Trans. on Image Processing*.

[11] O. D. Faugeras, "Digital Image Color Processing Within the Framework of a Human Visual System", *IEEE Trans. on ASSP*, Vol. 27, pp. 380-393, 1979.

[12] J. Frankle, and J. McCann, "Method and Apparatus for Lightness Imaging", *US Patent # 4,384,336*, 1983.

[13] B. Funt, F. Ciurea, and J. McCann, "Retinex in Matlab", in *Proc. of IS&T 8th Color Imaging Conference*, pp. 112-121, 2000.

[14] J. Gil, and R. Kimmel, "Efficient Erosion, Dilation, Opening, and Closing Algorithms", in *Proc. Int. Symp. Mathematical Morphology and its Applications to Image and Signal Processing* V, 2000.

[15] B. K. P. Horn, “Determining Lightness from an Image”, *Computer Graphics and Image Processing*, Vol, 3. pp. 277-299, 1974.

[16] D. J. Jobson, Z. Rahman, and G. A. Woodell, “A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observation of Scenes”, *IEEE Trans. on Image Proc.*, Vol. 6, 1997.

[17] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, “A Variational Framework for Retinex”, *Hewlett Packard Technical Report HPL-1999-151*, June 1999.

[18] R. Kimmel, D. Shaked, M. Elad, and I. Sobel, “Space Sensitive Color Gamut Mapping: A Variational Approach”, *Hewlett Packard Technical Report HPL-2000-50*, April 2000.

[19] E. H. Land, “The Retinex Theory of Color Vision”, *Scientific American*, Vol. 237, pp. 108-128, 1977.

[20] E. H. Land, “Recent Advances in the Retinex Theory and Some Implications for Cortical Computations: Color Vision and the Natural Image”, in *Proc. National Academy of Science USA*, Vol. 80, pp. 5163-5169, 1983.

[21] E. H. Land, “An Alternative Technique for the Computation of the Designator in the Retinex Theory of Color Vision”, *Proc. National Academy of Science USA*, Vol. 83, pp. 3078-3080, 1986.

[22] E. H. Land, and J. J. McCann, “Lightness and the Retinex Theory”, *J. Optical Soc. of America A*, Vol. 61, pp. 1-11, 1971.

[23] J. McCann, “Lessons Learned from Mondrians Applied to Real Images and Color Gamuts”, in *Proc. IS&T/SID 7th Color Imaging Conference*, pp. 1-8, 1999.

[24] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Prentice Hall, New-Jersey, 1975.

[25] A. Rosenfeld, and J. L. Pfaltz, “Sequential Operations in Digital Picture Processing", *J. of the ACM*, Vol. 13, pp. 471-494, 1966.

[26] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.

[27] R. Sobol, "Improving the Retinex algorithm for rendering wide dynamic range photographs" in *Proc. of SPIE Vol. 4662*, Jan. 2002.

[28] T. G. Stockham Jr, “Image Processing in the Context of a Visual Model”, *Proc. of the IEEE*, Vol. 60, 828--842, 1972.

[29] C. Tomasi, and R. Manduchi, “Bilateral Filtering for Gray and Color Images”, in Proc ICCV, 1998.

[30] J. Tumblin, J. K. Hodgins, B. Guenter, “Two Methods for Display of High Contrast Images”, *ACM Trans. on Graphics*, Vol. 18, pp. 56-94, January 1999.

[31] J. Tumblin, and G. Turk, “Low Curvature Image Simplifiers (LCIS): A Boundary Hierarchy for Detail-Preserving Contrast Reduction”, in *SIGGRAPH*, pp. 83-90, 1999.

[32] D. Terzopoulos, “Image Analysis Using Multigrid Relaxation Methods”, *IEEE Trans. on PAMI*, Vol. 8, 129-139, 1986.