

Lightness and Brightness Computations by Retinex-like Algorithms

Thesis for the M.Sc. Degree

by

Eran Borenstein



Under the Supervision of
Prof. Shimon Ullman

Faculty of Mathematics and Computer Science
The Weizmann Institute of Science

Submitted to the Feinberg Graduate School of
the Weizmann Institute of Science

Rehovot 76100, Israel

December 31, 1999

Abstract

This work deals with lightness and brightness computations and, in particular, the use of retinex-like algorithms for these computations. The retinex algorithm, proposed originally by E.Land, was essentially a one-dimensional computation. Previous extensions to two-dimensions had severe shortcomings, and owing to lack of an efficient solution they were not tested on large-scale images. In this work we produced efficient two-dimensional retinex-like computations, and studied their application to test images. Theoretical and empirical results were then used to analyze the limitations of retinex-like computations.

Contents

1	Introduction	3
1.1	Scope of the thesis	6
1.2	Structure of thesis	6
2	Basic definitions and perceptual phenomena	8
2.1	Basic terms	8
2.2	Brightness phenomena	9
2.3	Examples	9
3	Algorithms for reflectance and brightness computations	16
3.1	Retinex algorithms	16
3.1.1	The Laplacian operator	18
3.2	Linear algorithms	19
4	Analysis of the Retinex computation	22
4.1	The Laplacian operator in finite discrete space	22
4.2	The inverse of L	23
4.3	How to make the problem well posed	25
4.4	Horn's iterative method converges	30
5	Computing the inverse Laplacian efficiently	32
5.1	Computing the solution efficiently	32
5.2	The structure of the inverse operator	35
5.3	Additional application	39
6	Limitations of retinex-like algorithms	40
6.1	Using retinex-like scheme for brightness computations	40
6.2	The threshold operator	41
6.3	General operators in retinex like algorithms	42
6.4	Towards a better Non linear operator	43
6.5	conclusions	44
7	Results and Conclusions	49
7.1	Results	49
7.2	Conclusions	54
7.3	Future work	56

A	Proof for equivalence	58
B	The Fourier transform of the Laplacian kernel	61

1 Introduction

Light intensities in an image carry information about the reflectance properties of surfaces from which light was reflected. Under simple conditions, such as a scene composed of uniform surfaces, lighter surfaces will reflect more light than darker surfaces and will also be perceived as darker. However, in general there is a complex relationship between light intensity, surface reflectance, and perceived brightness. Two examples for these complex relationships are shown in Figures 1 and 2. Figure 1 shows the complex relationship between light intensity and surface reflectance or brightness. The regions marked by the arrows have the same intensities but they differ in perceived brightness and surface reflectance, and the assumption that simple relation exists between light intensity, perceived brightness and surface reflectance does not hold. The indirect relationship between light intensity and perceived brightness is also illustrated in Figure 2. The two squares in the image have the same light intensity but we see the left square as brighter than the right square.

These complex relationships raise two main problems. The first is more objective, or physical in nature: given the light intensities in the image, how to recover the reflectance properties of the viewed surfaces. The second is a perceptual problem: given the light intensities in the image, how to predict the perceived brightness of each region in the image. I will refer to the first as the *Reflectance problem* and to the second as the *Brightness problem*. These problems still do not have complete and satisfactory solutions. Reflectance computation is difficult because image intensities confound the effect of illumination and reflectance and it is difficult in general to separate the two. Brightness is complicated because the perceived brightness of a region depends on intensities in large parts of the image surrounding the region in addition to intensities in the region itself (see for example [14]).

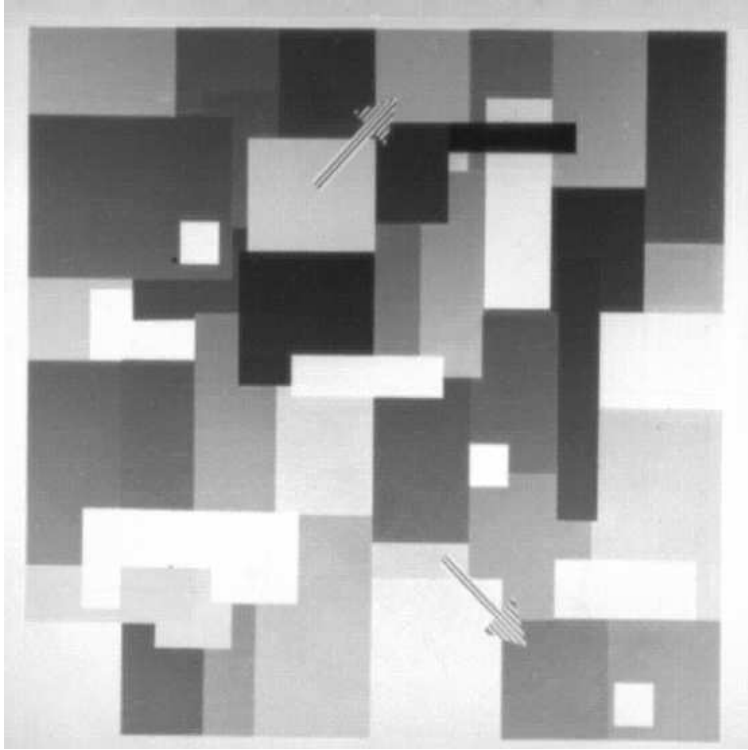
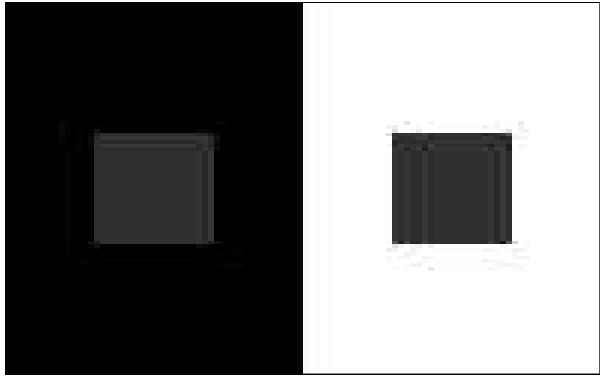
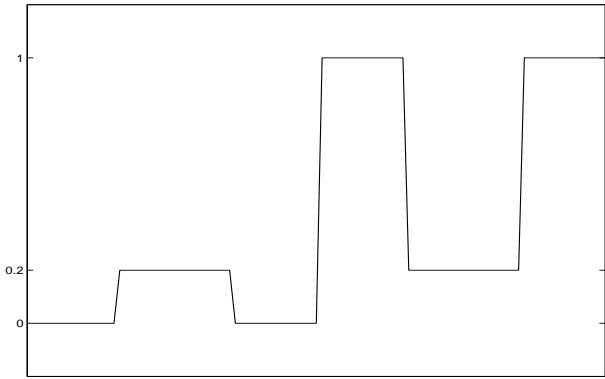


Figure 1: The two patches tagged by the arrows have the same intensity but they differ in reflectance and in perceived brightness. (Taken from [2]).



(a)



(b)

Figure 2: (a) - Simultaneous contrast display. (b) Normalized 1-D profile of the display intensity. Although the box on the left side has the same intensity as the box on the right, it looks brighter.

1.1 Scope of the thesis

In this work I will analyze two popular models for brightness and reflectance computations, the retinex model and the center-surround linear filter. I will suggest improvements over previous computations, but also show the inherent limitations of these two families of models. The focus of the work is on the retinex-like algorithms. These algorithms include the one dimensional retinex algorithm by Land and McCann [3], its extension to the two dimensional case by E.Land, and the two dimensional retinex algorithm by Horn [7]. Retinex-like algorithms were originally proposed as models for both reflectance and brightness computations. We will analyze some aspects of their properties and develop an improved computation. Then, we will examine the behavior of the computation and show that it can provide a useful reflectance model under simple conditions, but it is inadequate as a brightness model, and cannot deal with perceptual phenomena related to simultaneous contrast. We will examine briefly an alternative scheme, the linear filter model that can deal better with simultaneous contrast but has other crucial limitations for both reflectance and brightness computations. As we will show, a modified version of Horn's algorithm can give a model for the brightness problem in limited cases including simultaneous contrast. According to this model, the perceived brightness is not necessarily an attempt to recover a physical reflectance map but it may result from an attempt to increase the dynamic range of the perceived image.

1.2 Structure of thesis

The first two chapters provide introduction of the problem and review of previous approaches and results. In **chapter 2**, useful terms such as brightness, reflectance map and intensity will be defined. A number of basic phenomena in brightness perception that are relevant to the models under consideration will also be described briefly. **Chapter 3** reviews the basics of the retinex-like algorithms in general and the model of Horn in particular. Then, there is a brief discussion about linear algorithms and their ability to give adequate models for the reflectance and brightness problems. An analysis of the retinex computation is given in **Chapter 4**. This analysis uses a matrix presentation of the discrete Laplacian operator. The convergence of iterative methods for the retinex computation will be proven using this presentation.

This presentation also gives us in **Chapter 5** an efficient way to compute the inverse of the Laplacian operator and to examine its structure. **Chapter 6** discusses the general limitations of retinex-like algorithms and proposes a modified version of Horn's algorithm that can make the retinex-like computation exhibit simultaneous contrast phenomena. **Chapter 7** shows some results and gives our final conclusions with possible future directions of research.

2 Basic definitions and perceptual phenomena

Before we continue to examine the brightness and reflectance problems, it will be useful to list briefly some basic terms that are used throughout this work. Then, we will also describe two basic brightness phenomena known as **Brightness constancy** and **Simultaneous contrast** and finally we will show examples.

2.1 Basic terms

- **Quanta**: A light beam can be modeled by a stream of energy packets, called **Quanta**. Whenever light is absorbed by any sensor (e.g., the rods in the eye), the resulting events can be described as functions of **Quanta** absorbed by that sensor. That is, one quanta can be either wholly absorbed or not be absorbed at all. (Cornsweet p.13).
- **Intensity** of light can be defined as **Quanta** per second per unit area (Cornsweet p.234). An image presents measurements of light intensities on the image plane.
- The **Reflectance** is the property of a surface that describes the amount of light it reflects as a function of the amount of light illuminating it. Reflectance is usually wave-length and direction dependent, that is, for each specific wave-length and direction there is a specific value of reflectance. The surfaces considered here have **Lambertian** reflectance, namely the reflectance is the same in all directions. The **Reflectance map** associates each pixel in the image a reflectance value between zero and one. Under the Lambertian reflectance model, the intensity of surfaces, can be approximated by the following scalar product: $I(x, y) = R(x, y)L(x, y)\cos(\theta)$ where $I(x, y)$ is for the intensity, $R(x, y)$ for reflectance, $L(x, y)$ for the illumination magnitude and θ is the angle between the surface normal and the illumination direction.
- **Brightness** is defined as that aspect of perception of a patch of light that varies as its **Intensity** varies. (Cornsweet 234). In grey-level images this is the perception associated with a region that makes it appear as brighter or darker than other regions.

2.2 Brightness phenomena

In the introduction, we mentioned two perceptual phenomena, **Simultaneous contrast - SC**, and **Brightness constancy - BC**. In this section we will describe these phenomena in more details and then give some examples.

BC is observed when the brightness of object remains fairly constant despite large changes in illumination. For example, we know that a paper that looks white in daylight will also look white under almost any room illumination. One explanation for this constancy is that the visual system tries to recover the reflectance of objects, and since reflectance is independent of illumination the brightness should not change when the illumination changes.

SC is observed when the brightness of one object is perceived differently, depending upon the surrounding regions. For example, when a piece of grey paper is placed over a white paper it appears darker than the same piece placed over a black paper. SC is related to BC, since it appears to be the case where the visual system “fails” to recover the correct reflectance map whereas BC is associated with good approximation to the reflectance map.

2.3 Examples

In this section, intensity images along with their brightness images are presented for two purposes. One is to show examples where brightness images are very different from the intensity images, and the other is to provide a set of test images for the algorithms which will be proposed and studied later in this work.

- The pattern in figure 1 is called a **Mondrian** which was introduced by Land and McCann in [2]. These stimuli, named after the Dutch painter Piet Mondrian, consists of an array of patches, each having a uniform reflectance value. Mondrians can demonstrate BC. The two patches marked with the arrows have different reflectance. The upper one reflects more light than the lower one. However, when the Mondrian is illuminated such that the two patches have the same intensity, the upper patch is still brighter than the lower patch (see figure 1). This is a demonstration of BC since even for illumination that makes the two patches almost identical in intensities, their brightness does not change, they still look different, and their perceived brightness is similar to their relative reflectance.

- Figure 2a is an example of a **Simultaneous contrast** display. When two patches with identical intensity are placed on different backgrounds, the patch on the dark background looks brighter than the patch on the bright background. This image is a classic demonstration for SC. The physical reflectance distribution along the x-axis is shown in figure 2b.
- The pattern in figure 3 is called a **Mach-band** after Ernst Mach who first described it in 1865 (Cornsweet 277). The actual intensity distribution along the x axis is plotted below the mach-band. Note the difference between the intensities profile and the brightness profile. In regions A,B where the intensity transition starts, stripe B appears brighter than its background and stripe A appears darker than its background. The intensity at B is not higher than its right background and the intensity at A is not lower than its left background.
- The image in figure 4a is called a **Cornsweet edge** after T.M.Cornsweet who first discussed this phenomenon. Cornsweet edge demonstrates SC since two identical parts placed one beside the other, seem to have different intensity. The cross section of this stimuli (plotted below) shows that in the middle of the image where the two identical parts meet, there is a sharp gradient of intensities, while anywhere else the gradient is smaller. It is assumed that the visual system ignores small gradients and therefore each half has uniform brightness. Sharp gradients are not ignored, and therefore the gradient in the middle of the image makes the left half brighter than the right half.
- The upper image in figure 5 is called **Benussi ring**. In this image, a ring with a uniform intensity is placed over a background divided into two parts, one dark and the other bright. The brightness of the ring is almost uniform although each half is placed over a different background. This shows that the brightness is not a simple relation between the intensity of a patch and its background as might be inferred from the simultaneous contrast display. When a mid-line that separates the ring is added, as shown in the lower image, it is actually a variation of the simultaneous contrast display and the two separated parts of the ring are perceived with different brightness. This perceptual separation takes place even for a very thin mid-line.

- Figure 6 demonstrates the **staircase** stimuli. The brightness of each stair is not uniform although each one of them has a uniform intensity.
- Figure 7 demonstrates the **Windmill** image. In this image, the intensity at (x, y) changes as the function $\arctan(y - y_0, x - x_0) \text{ modulo } \frac{\pi}{2}$. In other words, in each of the four quarters the reflectance changes smoothly from 0 to 1 as we go clockwise along a circle. Since the intensities change smoothly everywhere except the \hat{x}, \hat{y} axes where the modulo takes affect, each quarter should have uniform brightness according to the assumption that the system ignores small gradients. However, in this case the small gradients are perceived by the visual system.

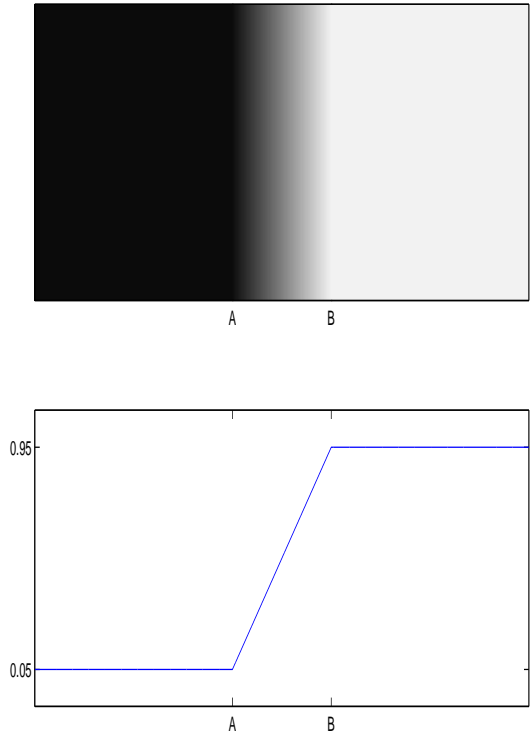


Figure 3: Mach band and its 1-D profile. Notice the illusion of a dark bar in area A and a bright bar in area B.

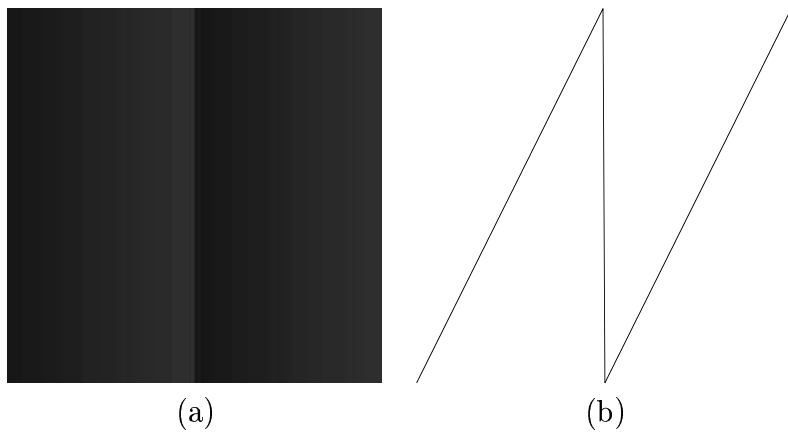
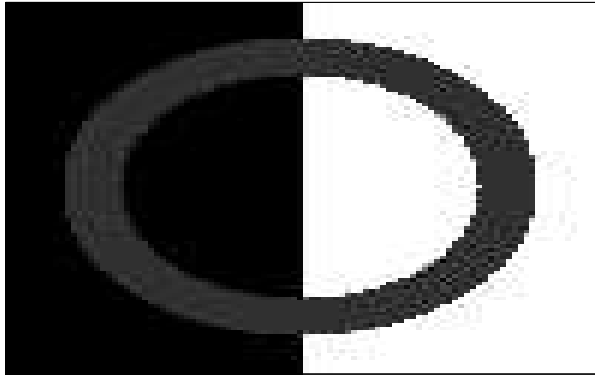
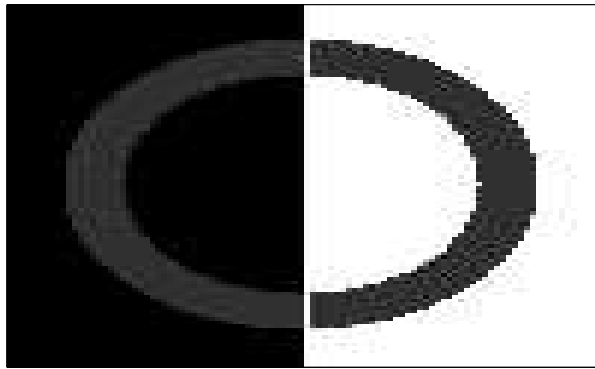


Figure 4: Cornsweet edge. The left and right halves of the image have identical intensities (see (b)). Nevertheless, the brightness of the left side looks brighter than the right.



(a)



(b)

Figure 5: (a) - Benussi ring with mid-line. (b) - Benussi ring without mid-line. The thin mid-line affects the brightness. When the mid-line is present, this is a version of simultaneous contrast display. When the mid-line is not present, the brightness of the ring tends to be uniform.

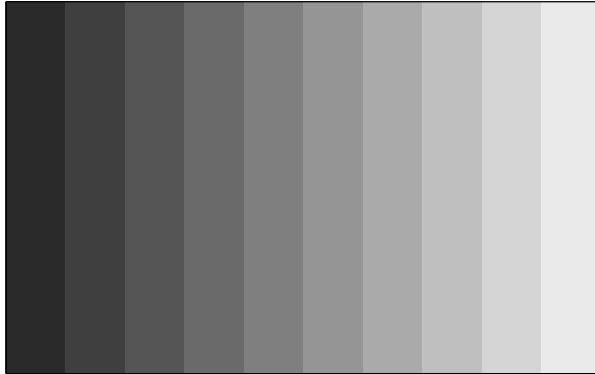


Figure 6: Staircase display. Each strip has uniform intensity, and non uniform brightness.

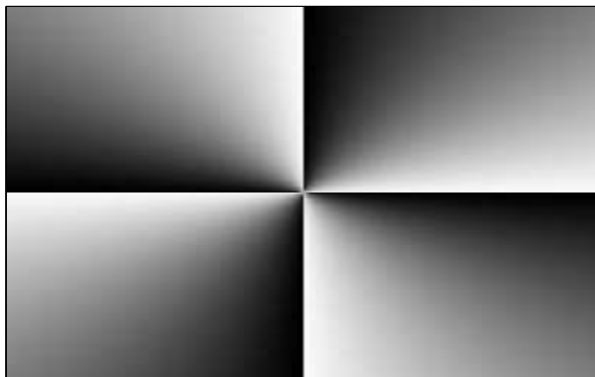


Figure 7: Windmill. The small gradients are perceived by the visual system in contrary to Cornsweet edge.

3 Algorithms for reflectance and brightness computations

We will consider two major families of reflectance and brightness algorithms. One includes the retinex-like algorithms and the other includes the linear algorithms. Retinex-like algorithms give good brightness constancy for images from the mini world of mondrians where the following assumptions hold: The surface is planar and lambertian and the illumination function contains only small gradients. On the other hand, these algorithms fail to detect simultaneous contrast which is important for brightness computations. For example, the two identical rectangles in the simultaneous contrast display will have identical brightness. The linear algorithms can give brightness constancy and respond to simultaneous contrast. However, in most cases the result is not satisfactory. For example, in the simultaneous contrast display, the two rectangles differ in their brightness but each one of them is not uniform as desired. In this chapter we introduce the basic ideas behind these two families.

3.1 Retinex algorithms

The first retinex algorithm for Lightness computations was proposed by Land and McCann in [2] and it was essentially a one-dimensional computation. A two-dimensional retinex algorithm, which is an extension of the algorithm by Land and McCann, was proposed by Horn in [7].

The main idea behind retinex-like algorithms is to separate the effects of reflectance from the effects of the illumination falling on the surfaces. This separation is based on the assumption that illumination varies smoothly across the image, whereas reflectance changes abruptly at the border between different objects. This assumption, together with the Lambertian model that approximates intensity image by the product of the illumination and the reflectance map, can help explaining the basic structure of these algorithms.

Let us first consider the retinex algorithm of Land and McCann. Take a one-dimensional image and denote it by $p(x)$. By the assumption, $p(x) = r(x)i(x)$, where $r(x)$ presents the reflectance and $i(x)$ presents the illumination. Logarithm of this model gives us: $P(x) = R(x) + I(x)$ where $P(x) = \log(p(x))$, $R(x) = \log(r(x))$, and $I(x) = \log(i(x))$. The first derivative of

$P(x)$, denoted by $P_x(x)$, describes local changes in intensities. Using the hypothesis that sharp local changes in intensities are contributed only by the reflectance map, one can apply a threshold operator that keeps only values of $P_x(x)$ that are contributed by reflectance transitions, and can obtain a new function $P'_x(x)$ that has only $R(x)$ contribution in it (See for example the case of Cornsweet edge). The reflectance $R(x)$ is computed by solving the first order differential equation (see also figure 8):

$$\frac{d}{dx}R(x) = P'_x(x) \tag{1}$$

E.Land extended this idea to the two-dimensional case by the following procedure. First, choose a reference point, P_r , and give it zero reflectance. Then, for each image point, P_i , sample a few random curves that join P_r and P_i . On each curve, the reflectance of P_i relative to P_r can be computed using the previous one-dimensional computation. The final reflectance of P_i relative to P_r will be then the average of the relative reflectances computed on the randomly sampled curves.

The windmill example shown in figure 7 demonstrates why this extension does not give reasonable results. To be consistent, the computation along closed curves should be close to zero since in these cases we have $P_r = P_i$. Therefore, the reflectance computed by going from P_r to P_i

clockwise along a closed curve containing them should approximately equal to the reflectance computed by going from P_r to P_i counterclockwise along this curve. Now, take any two points P_r and P_i in the windmill image. If the two points are in the same quarter, they will have the same reflectance since small gradients are ignored and therefore each quarter should have uniform reflectance. If the two points are in opposite quarters, take any two simple curves from P_r to P_i , one passing through the upper quarter, not containing P_i and P_r , and the other through the lower quarter, not containing P_i and P_r . The relative reflectance computed along one curve equals to minus the relative reflectance computed along the other curve and therefore, the average of relative reflectance computed by randomly sampled curves will eventually converge to zero. This means that opposite quarters will have the same reflectance. In addition, each quarter has one neighbor that should have higher reflectance and one that should have smaller reflectance but, since this is true for all the quarters, the only possibility is that they all have the same reflectance.

Horn suggested to extend the one-dimensional retinex algorithm by using the Laplacian, which is a second order differential isotropic operator. First, the Laplacian operator is applied to the image, and then a threshold operator is applied to the result. The threshold operator zeros all values below some threshold and leaves all values above it unaltered. Horn claimed, using the same arguments as in the one dimensional case, that such operation will retain only the reflectance information. It is important to notice that any operator that discards information related to illumination and retains the information related to reflectance can be used at this stage. Finally, an operator that approximates the inverse of the Laplacian is used to find an image having the particular Laplacian given by the threshold operator. The full process is shown in figure 9.

3.1.1 The Laplacian operator

The last part in Horn's algorithm requires estimation of the inverse of the Laplacian operator. Horn suggested to obtain this inverse as a solver for the non-homogeneous Dirichlet problem:

$$\Delta u = f(x, y) \text{ in } \Omega \tag{2}$$

and boundary condition:

$$u|_{\partial\Omega} = g(x, y)$$

When $\Omega \subseteq R^2$ and $g(x, y) \equiv 0$. In our case, $u(x, y)$ is the image we are looking for and $f(x, y)$ are the values of the Laplacian given by the threshold operator. Note that when the boundary condition is changed to the form:

$$\frac{\partial u}{\partial \vec{n}}|_{\partial\Omega} = g(x, y)$$

Where \vec{n} is a vector field normal to the boundary $\partial\Omega$, the problem is called non-homogeneous Neumann problem. The solution to this problem is unique up to an additive constant: if u solves the problem then also $u + c$ solves the problem.

Horn used known analysis of Dirichlet problems for this specific case. Solving this problem is relatively easy when $\Omega = R^2$ and the solution to the problem is given by Poisson's formula (for details see [13]):

$$u(x, y) = \int \int_{\Omega} G(\xi, \eta, x, y) f(\xi, \eta) d\xi d\eta \tag{3}$$

Where $G(\xi, \eta, x, y)$ is called **Green's** function and equals:

$$G(\xi, \eta, x, y) = \frac{1}{2\Pi} \log\left(\frac{1}{2r}\right) \quad (4)$$

and

$$r^2 = (\xi - x)^2 + (\eta - y)^2$$

Horn wanted to use the last formula, therefore he assumed an infinite retina (i.e. $\Omega = R^2$). He justified this assumption by the following argument. Consider a scene on an infinite uniform background and assume that the image of that scene is totally contained within the retina. Then, since the Laplacian operator gives zeros in uniform areas, anything outside the image will give zero contribution to equation 3. There are two problems with this assumption. First, different uniform backgrounds affect our brightness perception as illustrated for example by the simultaneous contrast display. Second, Poisson's formula works in the continuous case while in this case the problem is discrete. Horn's alternative approach was to solve the set of difference equations that approximate the Laplacian operator and solve them by iterative methods such as relaxation.

3.2 Linear algorithms

In this section we discuss briefly the linear approach to reflectance and brightness computations, where a linear operator is used for the computation. The linear approach typically uses a convolution operator with a center-surround kernel, having positive weights in the center and negative weights in the surround. This operator will therefore measure everywhere in the image the difference between some average intensity of the center and surround regions. A center-surrounded linear operator not only suppresses illumination gradients, but also tends to naturally produce simultaneous contrast. When simultaneous contrast occurs and two pixels with the same intensity have different brightness, there are usually many center surround kernels that will give the brighter pixel a higher measure than the other pixel. This observation is used to justify the linear strategy. Unfortunately, two pixels with a different measure will not always have different brightness. For example, consider the 1-d step function shown in figure 10a. Any center surround 1-d kernel such as the one shown in figure 10b will give something similar

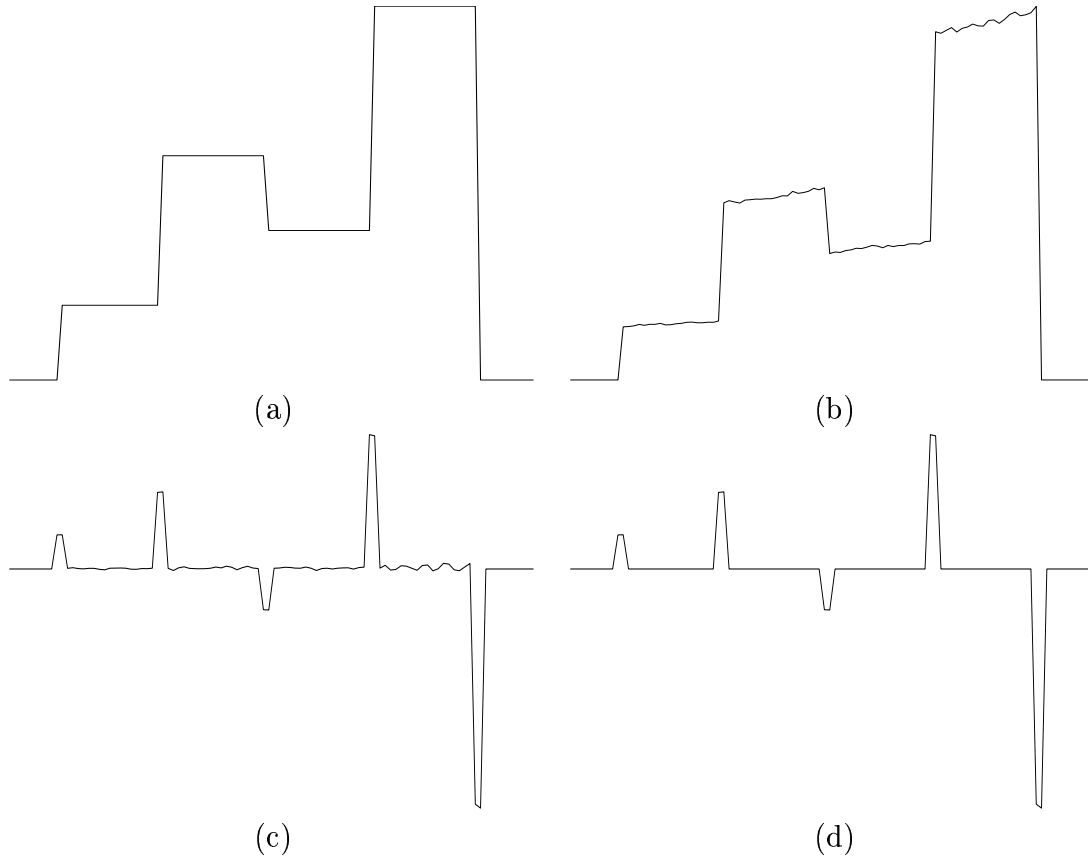


Figure 8: (a) 1-D image of reflectance. (b) 1-D image of (a) in gradient illumination with noise. (c) 1-D profile of the first spatial derivative of (b). (d) (c) after threshold. Note that (d) represents a function which is very similar to the spatial derivative of (a), the profile to reconstruct.

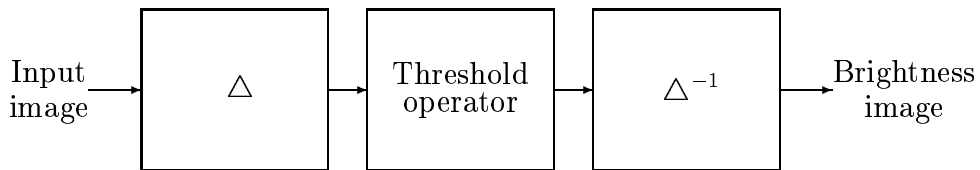


Figure 9: The structure of Horn's retinex algorithm

to the signal shown in figure 10c. Points which are closer to the edge will have a different measure from points which are farther away. This gives a wrong prediction, since in our perception each side of the step edge appears uniform. Nevertheless, linear algorithms are still important in the study of brightness because it seems that a measure of difference between a region and its background is an important parameter in brightness or reflectance calculations. In addition, it is interesting to note that different approaches for linear models usually end with some center surround kernel (see [12] 5.1,5.2 , [5]).

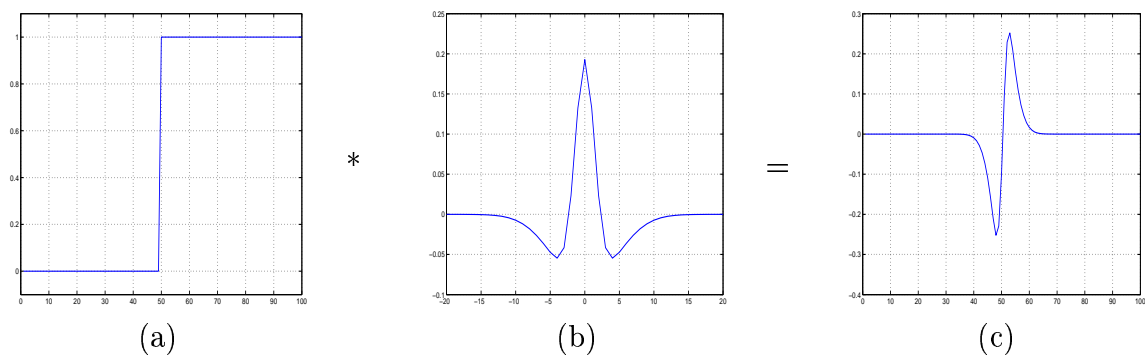


Figure 10: The result of applying a center-surround filter (b) to a 1-d step function (a) will always be similar to the one presented in (c).

4 Analysis of the Retinex computation

In this section, the Laplacian operator for the discrete case is defined. This definition allows us to present this operator in a matrix form. The remaining section of this chapter are devoted to analysis of retinex computations. As will become apparent, the matrix presentation is very helpful for this analysis.

4.1 The Laplacian operator in finite discrete space

Let $p(x, y)$ be an image defined over a grid. The discrete approximation for the Laplacian operator on an image $p(x, y)$ can be given in the form of difference equations. Let $d(x, y)$ be the value of the Laplacian on the grid point (x, y) . $d(x, y)$ is given by:

$$d(x, y) = 4p(x, y) - p(x - 1, y) - p(x + 1, y) - p(x, y - 1) - p(x, y + 1) \quad (5)$$

We refer to the last four pixels on the right-hand side of this equation as the nearest neighbors of pixel (x, y) . When the grid is finite, this equation cannot be used for pixels (x, y) located at the image boundary since they do not have all the four nearest neighbors. We therefore change equation 5 to:

$$d(x, y) = Np(x, y) - \sum_{(x', y') \sim (x, y)} p(x', y') \quad (6)$$

Where $(x', y') \sim (x, y)$ if (x', y') is a nearest neighbor to (x, y) and $N =$ number nearest neighbors of (x, y) . The last equation allows us to estimate Laplacian values for pixels on the boundary of the image. Note that this change is analogous to the Neumann problem in the continuous case with $g(x, y) \equiv 0$ for the boundary condition. As will be seen later, the solution in the discrete case is unique up to an additive constant just as it is for the continuous Neumann problem.

From now on, we consider only cases where the image is defined over a rectangular finite grid $(0..N - 1, 0..M - 1)$. In these images we have three types of pixels.

1. Interior pixels: These pixels have all 4 nearest neighbors. The Laplacian operator in these pixels looks in the following form:
$$\begin{array}{ccc} & & -1 \\ -1 & 4 & -1 \\ & & -1 \end{array}$$

2. Boundary pixels: These pixels have 3 nearest neighbors. The Laplacian operator in these pixels looks in one of the following forms, depending on its location:

$$\begin{matrix} -1 & 0 & 0 & -1 \\ 3 & -1 & -1 & 3 \\ -1 & 0 & 0 & -1 \end{matrix}, \quad \begin{matrix} -1 & 3 & -1 \\ 0 & -1 & 0 \end{matrix}, \quad \begin{matrix} 0 & -1 & 0 \\ -1 & 3 & -1 \end{matrix}$$

3. Corner pixels: These pixels have 2 nearest neighbors. The Laplacian operator in these pixels looks in one of the following forms, depending on its location:

$$\begin{matrix} 2 & -1 & -1 & 0 & -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 & -1 & -1 & 2 \end{matrix}$$

Every pixel gives one equation in the form of equation 6 and therefore if the image contains n pixels, we have a system of n equations. When the pixels are ordered as a one dimensional vector, these equations can be presented in a matrix form:

$$L\underline{p} = \underline{d} \tag{7}$$

Where the vector \underline{p} is constructed such that:

$$p(x, y) = \underline{p}(y + xM) \tag{8}$$

In other words, \underline{p} is constructed from a given image $p(x, y)$, by putting all the columns of the image one under the other, starting with the left most column. The vector \underline{d} is constructed in a similar way. We refer to L of equation 7 as the **Discrete Laplacian operator**.

An example of L for an image in size 3×3 is shown in figure 11. Note that row 5 is for an interior pixel, rows 2,4,6,8 are for boundary pixels and rows 1,3,7,9 are for the corner pixels.

The general form of L for arbitrary grid $(0..N - 1, 0..M - 1)$ can be seen in figure 12.

4.2 The inverse of L

Recall that in the last step of Horn's algorithm, given \underline{d} we want to compute \underline{p} such that equation 7 is satisfied. To do this, we need to find L^{-1} . The next claim shows that L does not have an inverse and therefore, the problem

$$\begin{array}{cccccccc}
2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 3 & -1 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\
0 & 0 & -1 & 0 & -1 & 3 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2
\end{array}$$

Figure 11: The matrix L for an image of size 3×3 .

of equation 7 is ill posed since given \underline{d} there are two options. One is when $\underline{d} \in \text{range}(L)$, in this case there are infinite number of solutions. The other is when there are no solutions at all.

Claim 1 :

$$\text{rank}(L) = n - 1, \text{Null}(L) = \text{span}(\underline{1}) \text{ and } \underline{x} \in \text{Range}(L) \text{ iff } \underline{1}^T \underline{x} = 0.$$

Proof: $L = L_{n \times n}$ since there are n equations for n pixels. According to equation 6, the Null space of L includes all the constant vectors and therefore, $\text{rank}(L) \leq n - 1$. On the other hand, zero on the left-hand side of equation 6 means that $p(x, y)$ has the average of all its nearest neighbors. Therefore, $p(x, y)$ cannot be either greater than the maximal neighbor nor smaller than the minimal neighbor. The system $L\underline{p} = \underline{0}$ implies that the above is true for all the pixels and therefore no one of them can solely attain neither a maximum nor a minimum. The only possible solutions for this case are constant images. We therefore get $\text{rank}(L) \geq n - 1$. Conclusion:

$$\text{rank}(L) = n - 1 \tag{9}$$

and:

$$\text{Null}(L) = \text{span}(\underline{1}) \tag{10}$$

L is also symmetric since:

$$L(i, j) = \begin{cases} -1 & \mathbf{j} \text{th pixel is a nearest neighbor of } \mathbf{i} \text{th pixel} \\ N & \mathbf{j} = \mathbf{i} \text{ and } N \text{ is the number of nearest neighbors to } \mathbf{i} \\ 0 & \text{Otherwise} \end{cases}$$

Then, if \mathbf{j} is a neighbor of $\mathbf{i} \Rightarrow \mathbf{i}$ is a neighbor of $\mathbf{j} \Rightarrow L(i, j) = L(j, i)$.

Now, denote by $\underline{\underline{A}}$ a $n \times n$ matrix with all its elements equal to A . L being symmetric implies that:

$$\underline{\underline{0}} = L\underline{\underline{1}} = \underline{\underline{1}}^T L^T = \underline{\underline{1}}L \quad (11)$$

This gives us the following:

$$\underline{\underline{0}} = (\underline{\underline{1}}L)\underline{\underline{p}} = \underline{\underline{1}}(L\underline{\underline{p}}) = \underline{\underline{1}}\underline{\underline{d}} \quad (12)$$

□.

This claim proves that all solutions for the system of equation 7 are unique up to an additive constant. This is what we had expected when we examined some properties of the continuous Neumann problem.

4.3 How to make the problem well posed

The previous section showed that the problem defined by the system of equation 7 is ill posed. It could either have infinite number of solutions or no solution at all. We will next define two new equivalent problems that are closely related to the original problem, but that are guaranteed to always have a unique solution.

A reasonable way to make the problem of equation 7 well posed is to add a constraint such as average of $\underline{\underline{p}}$ equals m . This addition, gives the following set of equations:

$$\hat{L}\underline{\underline{p}} = \begin{bmatrix} L \\ \left[\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \right] \end{bmatrix} \underline{\underline{p}} = \begin{bmatrix} \underline{\underline{d}} \\ m \end{bmatrix} = \hat{d} \quad (13)$$

The additional constraint actually picks up a unique solution by constraining it to have a specific mean value m . This new system can be solved only when $\underline{\underline{d}} \in Range(L)$ and therefore it is still ill posed since it cannot be solved when $\underline{\underline{d}} \notin Range(L)$. There is a general approach which turns the problem into a minimization problem, and makes it well posed in the sense that there is always a unique solution giving the minimum. Consider the following system of equations:

$$H\underline{\underline{p}} = \hat{d} \quad (14)$$

Where:

$$H_{[(n+k) \times n]} = \begin{bmatrix} L_{[n \times n]} \\ C_{[k \times n]} \end{bmatrix}_{[(n+k) \times n]} \quad \text{and} \quad \hat{d} = \begin{bmatrix} d_{[n \times 1]} \\ c_{[k \times 1]} \end{bmatrix}_{[(n+k) \times 1]} \quad (15)$$

$C_{[k \times n]} \underline{p} = \underline{c}$ represents an additional set of k equations to the original system generated by the Laplacian operator L . In fact C plays a regularization role (for a discussion about regularization methods refer to [6]). Now, there are more equations ($n + k$) than variables (n) and, in general, not all the equations can be satisfied simultaneously. We call \underline{p}_2 a solution for equation 14 if it minimizes the following Euclidean norm:

$$\min_{\underline{p}} \|H\underline{p}(x, y) - \hat{d}(x, y)\|^2 \quad (16)$$

When $\text{rank}(H) = n$, there exists a unique \underline{p}_2 minimizing the last expression and it can be found by the pseudo inverse H^\dagger of H as follows:

$$\underline{p}_2 = H^\dagger \hat{d} \quad (17)$$

Where:

$$H^\dagger = (H^T H)^{-1} H^T \quad (18)$$

In our specific case we set $H = \hat{L}$ and if we want to find \underline{p}_2 using equations 17 and 18 then it remains to show that $\text{rank}(\hat{L}) = n$:

$$\text{Null}(\hat{L}) \subseteq \text{Null}(L) = \text{span}\{\underline{1}\} \quad (19)$$

But,

$$\underline{1} \notin \text{Null}(\hat{L}) \quad (20)$$

Therefore, $\text{Null}(\hat{L}) = \{0\}$, $\dim(\text{Null}(\hat{L})) = 0$ and by conservation of dimension we get:

$$\dim(\text{Null}(\hat{L})) + \text{rank}(\hat{L}) = n \quad (21)$$

□.

We next define a second well posed problem by adding the matrix $\alpha \underline{1}$ to L as follows: given a Laplacian \underline{d} we look for an image \underline{p}_1 such that

$$(L + \alpha \underline{1}) \underline{p}_1 = \underline{d} \quad (22)$$

Where α is any non zero constant. We denote the matrix $(L + \alpha \underline{1})$ by L_1 .

Claim 2 :

$$\text{rank}(L_1) = n$$

Proof: By claim 1 we have:

$$\text{Null}(L) = \text{span}\{\underline{\mathbf{1}}\}, \quad \text{Range}(L) = \{x : \underline{\mathbf{1}}^T x = 0, x \in R^n\}$$

We also have:

$$\text{Null}(\underline{\mathbf{1}}) = \{x : \underline{\mathbf{1}}^T x = 0, x \in R^n\}, \quad \text{Range}(\underline{\mathbf{1}}) = \text{span}\{\underline{\mathbf{1}}\}$$

$$\Rightarrow \text{Null}(L) \cap \text{Null}(\underline{\mathbf{1}}) = \{0\} \text{ and } \text{Range}(L) \cap \text{Range}(\underline{\mathbf{1}}) = \{0\}$$

$$\text{Null}(L_1) = \{x : Lx + \alpha \underline{\mathbf{1}}x = 0, x \in R^n\} = \{x : Lx = -\alpha \underline{\mathbf{1}}x, x \in R^n\}$$

But, since $\text{Range}(L) \cap \text{Range}(\alpha \underline{\mathbf{1}}) = \{0\}$,
equality can hold only for $x \in \text{Null}(L) \cap \text{Null}(\alpha \underline{\mathbf{1}}) = \{0\}$.

$$\Rightarrow \text{Null}(L_1) = \{0\}$$

By conservation of dimension we therefore get:

$$\dim(\text{Null}(L_1)) + \text{rank}(L_1) = n = \text{rank}(L_1) \quad (23)$$

□.

The last claim shows us that $L_1 = (L + \alpha \underline{\mathbf{1}})$ has an inverse and that the problem of equation 22 is well posed. The solution \underline{p}_1 for this equation is given by:

$$\underline{p}_1 = L_1^{-1} \underline{d} \quad (24)$$

Let us now check the Laplacian of this solution:

$$L \underline{p}_1 = L(L + \alpha \underline{\mathbf{1}})^{-1} \underline{d} = \quad (25)$$

$$((L + \alpha \underline{\mathbf{1}}) - \alpha \underline{\mathbf{1}})(L + \alpha \underline{\mathbf{1}})^{-1} \underline{d} =$$

$$\underline{d} - \alpha \underline{\mathbf{1}}(L + \alpha \underline{\mathbf{1}})^{-1} \underline{d}$$

Notice that:

$$\underline{\mathbf{1}}(L + \alpha \underline{\mathbf{1}}) = \underline{\mathbf{1}}L + \alpha \underline{\mathbf{1}} \underline{\mathbf{1}} = 0 + \alpha n \underline{\mathbf{1}} \quad (26)$$

Multiplying from the right the last equation by $\frac{1}{n}(L + \alpha\underline{1})^{-1}$ we get:

$$\frac{1}{n}\underline{1} = \alpha\underline{1}(L + \alpha\underline{1})^{-1} \quad (27)$$

And if we substitute it in equation 25 we have:

$$L\underline{p}_1 = \underline{d} - \frac{1}{n}\underline{1} \underline{d} = \underline{d} - \frac{\underline{1}^T \underline{d}}{n} \underline{1} \quad (28)$$

The last equation shows that when $\underline{d} \in \text{Range}(L)$ ($\underline{1}^T \underline{d} = 0$) the solution \underline{p}_1 has Laplacian \underline{d} and is therefore also a solution for equation 7. When $\underline{d} \notin \text{Range}(L)$ equation 7 has no solution, but the new problem gives a solution with Laplacian that differs from \underline{d} in $\frac{\underline{1}^T \underline{d}}{n} \underline{1}$.

The two problems presented by \hat{L} and L_1 are equivalent. Appendix A shows that for a given \underline{d} , the solutions for these two problems, \underline{p}_1 and \underline{p}_2 , differ only in a constant vector. Therefore, we will focus next on the solution of the problem presented by L_1 . We show an efficient method to compute it and show the form of L_1^{-1} . Given any Laplacian \underline{d} , the retinex-like computation is obtained directly by applying the pre-computed L_1^{-1} to this Laplacian.

4.4 Horn's iterative method converges

Horn suggested to solve equation 7 by using either Gauss-Seidel or Jacobi iterative methods. Among all possible solutions for this equation, he suggested to choose the one that gives the pixel with the highest intensity a fixed value of one. We denote this pixel by p_x where x is its index in the \underline{p} vector of pixels. Holding the value of this pixel fixed, allows us to work with a smaller system of $n - 1$ variables. This system is presented by the following system:

$$L^{Horn} \underline{p}_{(n-1)} = \underline{d}_{(n-1)} \quad (29)$$

Where the vectors $\underline{p}_{(n-1)}$ and $\underline{d}_{(n-1)}$ are the same as \underline{p} and \underline{d} except that their x_{th} element is excluded. The matrix L^{Horn} which is in size $(n - 1) \times (n - 1)$ is constructed from the original matrix L as follows. First, the x_{th} row corresponding to the fixed pixel p_x is excluded from L and we have:

$$L_{(n-1) \times n} \underline{p} = \underline{d}_{(n-1)} \quad (30)$$

Then, the x_{th} column in $L_{(n-1) \times n}$ corresponding to the fixed pixel p_x , call it \underline{l}_x , is excluded from $L_{(n-1) \times n}$ to give L^{Horn} and the last equation can be rewritten as:

$$L^{Horn} \underline{p}_{(n-1)} + p_x \underline{l}_x = \underline{d}_{(n-1)} \quad (31)$$

But , since p_x is fixed (and actually in this case equals one) we have:

$$L^{Horn} \underline{p}_{(n-1)} = \underline{d}_{(n-1)} - \underline{l}_x \quad (32)$$

Previous analysis did not show that this computation will converge. The new matrix L^{Horn} , representing the problem of equation 1 with the additional constraint that $p_x = 1$, is irreducible diagonally dominant and we can use the following theorem:

Theorem 1 *If A is irreducible diagonally dominant matrix, then the associated Jacobi and Gauss-Seidel iterations converge for any initial value $x^{(0)}$.*

The proof for this theorem can be found in [17].

A $n \times n$ matrix $A = (a_{i,j})$ is irreducibly diagonally dominant if it is irreducible and:

$$|a_{i,i}| \geq \sum_{\substack{j \\ j \neq i}} |a_{i,j}| \quad (33)$$

for all $1 \leq i \leq n$ with strict inequality for at least one i . The Laplacian matrix L satisfies the last equation with equality for all i . However in L^{Horn} at least two rows will give us strict inequality since we removed the x_{th} column of the L matrix and this column contains $N_x + 1$ non zero entries where N_x is the number of nearest neighbors to the pixel p_x . A $n \times n$ matrix $A = (a_{i,j})$ is irreducible if and only if its directed graph $G(A)$ is strongly connected. The directed graph $G(A)$ is a graph with n nodes where node P_i is connected to node P_j whenever $a_{i,j} \neq 0$. A directed graph is strongly connected if, for any ordered pair of nodes P_i and P_j , there exists a directed path connecting P_i to P_j . The directed graph of the Laplacian matrix is strongly connected since from each node we can go to all of his nearest neighbors. The graph represented by L^{Horn} is a subgraph of the Laplacian matrix with the x_{th} node excluded and therefore it is also strongly connected. We conclude that the matrix L^{Horn} is indeed irreducible diagonally dominant and the theorem above can be used.

5 Computing the inverse Laplacian efficiently

The last chapter introduced two new equivalent problems which are well posed and at the same time have a solution that solves the problem of equation 7 when such exists. We also know that \underline{p}_1 can be found once we know L_1^{-1} so we will give an explicit form for the structure of this inverse along with an efficient computation for \underline{p}_1 .

5.1 Computing the solution efficiently

In this section we show how to compute \underline{p}_1 efficiently and proceed as follows: First, we introduce useful notations from Fourier analysis including the discrete Fourier transform and the cyclic convolution operator. The linear operator $(L + \alpha\underline{1})$ operates on an $N \times M$ image. We will show that we can represent it instead by a cyclic convolution of an invertible kernel with an image that is four times as large, $2N \times 2M$. We will obtain the inverse to the convolution operator, and by a “folding” operation recover the inverse to our original problem $L_1\underline{p}_1 = \underline{d}$.

The 2D Discrete Fourier Transform (DFT) \mathcal{F}_d operating on functions defined over the discrete grid $(0..N - 1, 0..M - 1)$ is given by:

$$\mathcal{F}_d\{f\}(k, l) = \frac{1}{2\pi} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) W_N^{kn} W_M^{lm} = F(k, l) \quad (34)$$

Where

$$W_N = \exp \frac{-i2\pi}{N}$$

The inverse transform \mathcal{F}_d^{-1} is defined as:

$$\mathcal{F}_d^{-1}\{F\}(n, m) = \frac{1}{2\pi} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} F(k, l) W_N^{-kn} W_M^{-lm} = f(n, m) \quad (35)$$

The 2D cyclic convolution with kernel g operating on all functions defined over the discrete grid $(0..N - 1, 0..M - 1)$ is given by:

$$\{g \otimes f\}(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(n, m) g(k - n \bmod N, l - m \bmod M) \quad (36)$$

It is well known that (the convolution theorem):

$$\mathcal{F}_d\{(g \otimes f)\}(k, l) = \mathcal{F}_d\{g\}(k, l)\mathcal{F}_d\{f\}(k, l) = G(k, l)F(k, l) \quad (37)$$

Suppose that we wish to recover f given $g \otimes f$ and g . When this is possible, there exists g^{-1} such that:

$$g^{-1} \otimes (g \otimes f) = f \quad (38)$$

By Applying a Fourier transform to equation 38 we have:

$$G^{-1}(GF) = F \quad (39)$$

In case that $G(k, l) \neq 0$ for all (k, l) we can divide both sides by (GF) and have:

$$G^{-1} = \frac{1}{G}$$

Now, by using the inverse Fourier transform:

$$g^{-1} = \mathcal{F}_d^{-1}\left\{\frac{1}{G}\right\} \quad (40)$$

In other words, when a linear operator P can be represented by cyclic convolution with kernel g , then the inverse operator P^{-1} (if exists) is also a convolution with the kernel g^{-1} given by equation 40. Since convolution is a linear operator, it can be represented by a matrix. Our motivation now is to show that the result of applying the L operator to an image p is equivalent to the cyclic convolution of an extended image $p^*(n, m)$ with the following kernel:

$$g = \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \quad (41)$$

The extended image p^* is constructed from the original image p in the following manner. Let $p(n, m)$ be our image with size of $N \times M$. Our coordinate system origin is chosen, without loss of generality, to be the lower left corner of the image, that is, $p(0, 0)$ represents the grey level value of the lower left corner in the image p . Now, let p_x , be the symmetric expansion of the image p with respect to the \hat{x} axis. That is, if p_x was a continuous function then

$p_x(x, y) = p_x(-x, y)$. In our case, since p_x is a function on a discrete grid we define: $p_x(n, m) = p(n, m)$ for $n, m \geq 0$ and $p_x(n, m) = p(-1 - n, m)$ for $m \geq 0, n < 0$. Then, we let p^* be the symmetric expansion of the image p_x with respect to the \hat{y} axis. That is, $p^*(n, m) = p_x(n, m)$ for $m \geq 0$ and $p^*(n, m) = p_x(n, -1 - m)$ for $m < 0$. This expansion, p^* , of p is defined on the grid $(-N..N - 1, -M..M - 1)$. Fig. 13 demonstrates the construction of p^* given p in size 3×3 .

Recall that $d(n, m)$ is the Laplacian of the image $p(n, m)$, that is $\underline{d} = L\underline{p}$ so d is also in size $N \times M$. We expand d in the same way we expanded p , to get d^* which is also defined on the grid $(-N..N - 1, -M..M - 1)$.

Claim 3 :

If g is the kernel defined in equation 41, then: $\mathbf{d}^ = \mathbf{g} \otimes \mathbf{p}^*$*

Proof: This follows from the construction of p^* and from the fact that the mask g is invariant under reflections.

Define d_1 to be:

$$\underline{d}_1 = (L + \alpha \underline{\mathbf{1}})\underline{p} = \underline{d} + \alpha \underline{\mathbf{1}}^T \underline{p} \underline{\mathbf{1}} \quad (42)$$

By expanding d_1 to d_1^* we have:

$$d_1^* = d^* + (\alpha \underline{\mathbf{1}}^T \underline{p}) \underline{\mathbf{1}}_{2N \times 2M} = g \otimes p^* + \frac{\alpha}{4} \underline{\mathbf{1}}_{2N \times 2M} \otimes p^* = (g + \frac{\alpha}{4} \underline{\mathbf{1}}_{2N \times 2M}) \otimes p^* \quad (43)$$

We divide α by 4 since every element of \underline{p} appears four times in p^* . The structure of d_1^* allows us to find d_1 defined in equation 42 simply by:

$$d_1 = d_1^*(0..N - 1, 0..M - 1) \quad (44)$$

Therefore:

$$(L + \alpha \underline{\mathbf{1}})\underline{p} \stackrel{\text{equation 42}}{=} \underline{d}_1 \stackrel{\text{equation 43,44}}{=} ((g + \frac{\alpha}{4} \underline{\mathbf{1}}_{2N \times 2M}) \otimes p^*)(0..N - 1, 0..M - 1) \quad (45)$$

The last equation shows that the operator $(L + \alpha \underline{\mathbf{1}})$ on the image \underline{p} can be represented as a cyclic convolution of the extended image p^* with the kernel $\tilde{g} = (g + \frac{\alpha}{4} \underline{\mathbf{1}})$. Its inverse then, which exists when $\alpha \neq 0$, can be also represented as a cyclic convolution. The kernel \tilde{g}^{-1} of this convolution is

given by equation 40. Now, by using this kernel, we can compute the solution p_1^* for equation 43 as follows:

$$p_1^* = \tilde{g}^{-1} \otimes d_1^* \quad OR \quad p_1^* = \mathcal{F}_d^{-1}\left(\frac{D_1^*}{\tilde{G}}\right) \quad (46)$$

p_1^* enables us to find p_1 since:

$$p_1 = p_1^*(0..N-1, 0..M-1)$$

This gives us an efficient computation for p_1 solving equation 42. The computation as can be seen in equation 46 involves $2N \cdot 2M$ multiplications and Fourier transform of a matrix in size $2N \times 2M$ which has complexity of $O(NM \log(NM))$. Along with efficiency we can have an explicit form for the inverse operator $(L + \alpha \underline{1})^{-1}$. The following section shows how to find this form and gives examples of its structure.

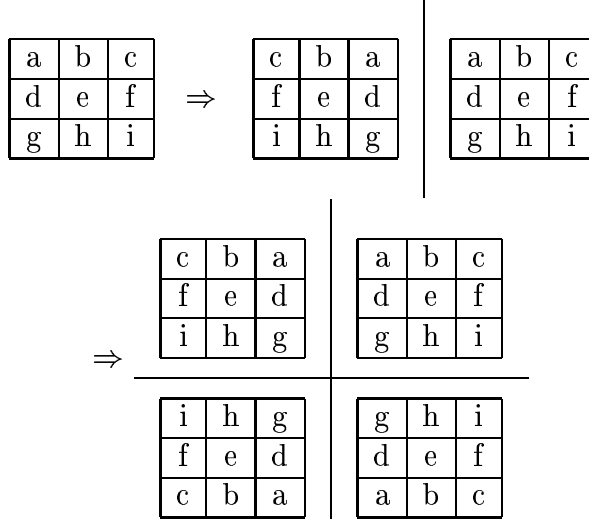


Figure 13: The 3 steps of expanding an image p into p^* .

5.2 The structure of the inverse operator

The previous section showed that the operation of $(L + \alpha \underline{1})^{-1}$ on \underline{d} can be computed by the cyclic convolution of d_1^* with \tilde{g}^{-1} . This convolution is a

linear operator and therefore can be presented by a matrix multiplying the vector \underline{d}_1^* . However, each component of \underline{d}_1 appears in \underline{d}_1^* exactly four times so such presentation is redundant and we can represent it by a smaller matrix which multiplies the vector \underline{d}_1 . This is actually the matrix $(L + \alpha \underline{\underline{1}})^{-1}$, the inverse of the invertible Laplacian operator that we were looking for, since it operates on \underline{d}_1 and gives \underline{p}_1 . The following paragraph describes a procedure for the construction of this matrix.

Recall that the value of each pixel $p^*(n, m)$ in the extended image is found by the weighted sum of d_1^* with weights given by the mask \tilde{g}^{-1} as its center is placed on $d^*(n, m)$. According to the construction of d^* , it is actually 4 copies of d in different reflections with respect to the \hat{x} and \hat{y} axes so If we fold d^* lying on the $(-N..N - 1, -M..M - 1)$ grid together with the weights associated with it back into the $(0..N - 1, 0..M - 1)$ grid we get d together with four weights associated with each of its components. At this point, each element of d , d_i , is associated with a new weight $w(n, m)_i$ which is the sum of these four weights. This $\underline{w}(n, m)$ is actually a row in the matrix $(L + \alpha \underline{\underline{1}})^{-1}$, the row associated with the grid point (n, m) . After repeating this row construction for all grid points, we end with the matrix $(L + \alpha \underline{\underline{1}})^{-1}$. Figures 14 and 15 show the weights computed for the center pixel, a corner pixel, an internal pixel and a pixel on the edge. The weights for the center pixel which also represent the kernel of the convolution are very similar to the kernel $\frac{1}{2\pi} \log(\frac{1}{r})$ that was proposed by Horn (figure 17). Appendix B examines the Fourier transform \tilde{G} of the kernel \tilde{g} and shows that this operator is also scalable.

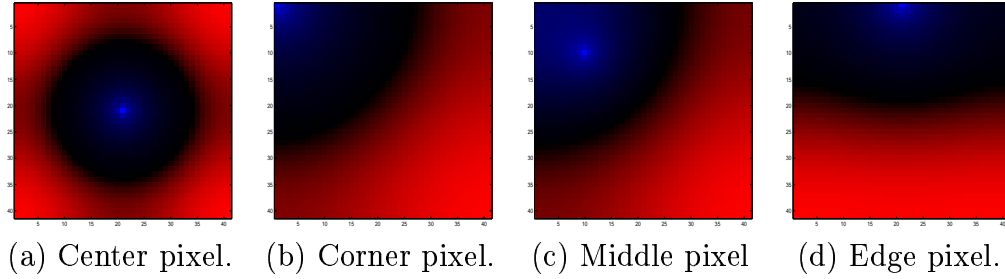


Figure 14: Weights for the inverse Laplacian computation at four locations. Positive weights have blue colors, negative weights have red colors. The weights for the center pixel also gives the kernel of the convolution.

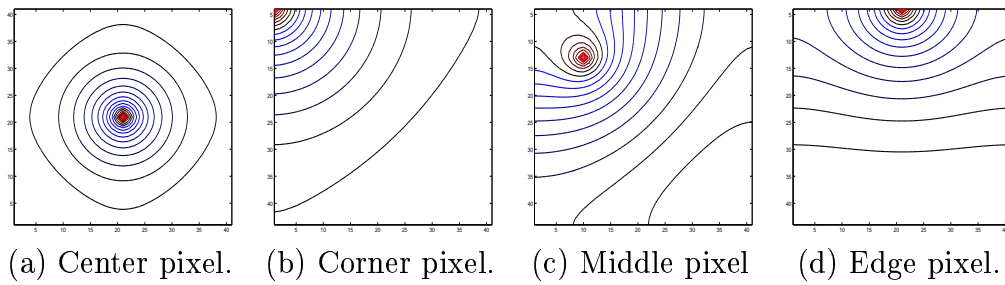


Figure 15: Contour plot of the weights in last figure.

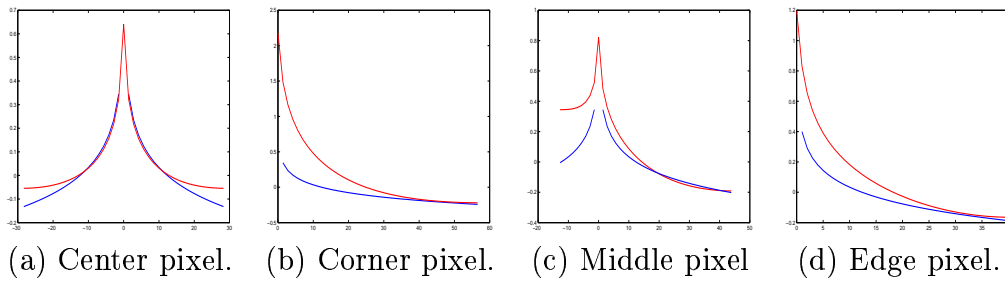


Figure 16: Diagonal profile of the inverse weights (red) vs. the Green kernel $\frac{1}{2\pi} \log \frac{1}{r} + 0.4$ (blue).

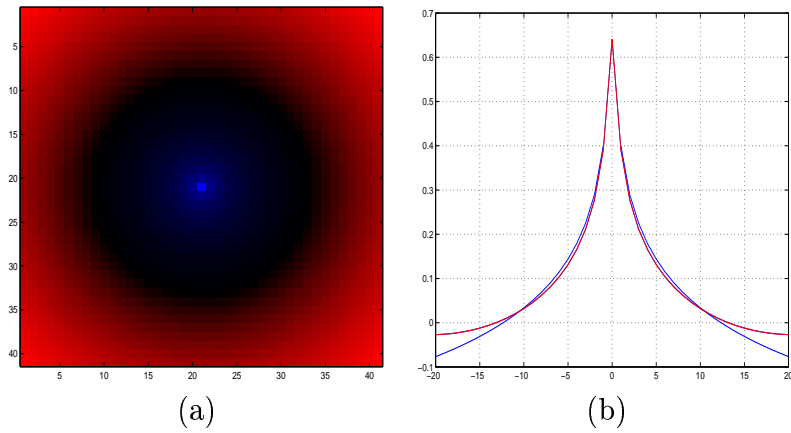


Figure 17: (a)- 2D presentation for $\frac{1}{2\pi} \log \frac{1}{r} + 0.4$. (b)- The kernel $\frac{1}{2\pi} \log \frac{1}{r} + 0.4$ (blue) vs. discrete inverse of Laplacian kernel (red).

5.3 Additional application

The last chapter gives a procedure for solving the set of difference equations which is analog to the following Neumann problem:

$$\Delta u = f(x, y) \text{ in } \Omega$$

and boundary condition:

$$\frac{\partial u}{\partial \vec{n}} \Big|_{\partial\Omega} = 0$$

Where Ω is a rectangular domain in R^2 and the vector field \vec{n} which is a field normal to the boundary $\partial\Omega$ is not defined on its corners.

The same procedure can be used for solving the following diffusion equation in equilibrium with constant coefficient a :

$$\Delta u(x, y) + au(x, y) = f(x, y) \text{ in } \Omega$$

and boundary condition:

$$\frac{\partial u}{\partial \vec{n}} \Big|_{\partial\Omega} = 0$$

In this case we have a new operator $T = \Delta + a$ instead of the Laplacian operator. This operator can also be represented by a convolution operator on an expanded image so if it is invertible, we can solve the problem by the same technique. Indeed, the kernel T of this operator is the following:

$$T = g^{equation 41} + a\delta = \begin{matrix} & 0 & -1 & 0 \\ -1 & 4 + a & -1 & \\ & 0 & -1 & 0 \end{matrix} \quad (47)$$

As long as a is a positive constant, the convolution operator T has an inverse since the matrix that presents it is strictly positive definite and therefore non singular. It can be shown by using the Fourier transform of the Laplacian operator (see equation 56) and of the delta function that when $a < -4$ the Fourier transform of T will not vanish in all possible frequencies and therefore the operator T is also invertible in this case.

6 Limitations of retinex-like algorithms

In this section we show that, in general, retinex-like algorithms have severe limitations in the task of brightness as well as lightness computations.

6.1 Using retinex-like scheme for brightness computations

Let us first define formally what we refer here as retinex-like algorithms. A retinex-like algorithm is any scheme that first computes $\underline{z} = L\underline{p}$ where \underline{p} is a given image, then applies to the result \underline{z} any operator D such that $\underline{d} = D(\underline{z})$ and finally finds a solution \underline{p}_1 such that $(L + \alpha\underline{1})\underline{p}_1 = \underline{d}$.

$$(i) \quad L\underline{p} = \underline{z}$$

$$(ii) \quad \underline{d} = D(\underline{z})$$

$$(iii) \quad (L + \alpha\underline{1})\underline{p}_1 = \underline{d}$$

A specific example for a retinex-like algorithm is the one that uses a threshold operator. The aim of the threshold operator is to remove illumination effects and therefore it is supposed to be useful for reflectance computations where we wish to remove illumination gradients and maintain changes due to material differences. When we consider brightness computation, we wish to reproduce also the phenomena of simultaneous contrast. Horn's algorithm does not produce simultaneous contrast. An interesting problem therefore is to look for an operator that, on the one hand, can remove illumination gradients but, on the other hand, can produce simultaneous contrast.

We suggested that simultaneous contrast may result from the attempt of the visual system to increase the dynamic range of brightness images. In the physical world, reflectance values range from approximately 0.1 to 1.0. One may therefore assume that the highest ratio between neighboring pixels caused by reflectance changes is about 10, and higher ratios result from changes in illumination. This assumption, along with the usual retinex assumption, can motivate an operator that divides the range of the Laplacian values into three domains. The first contains small values, the third contains large values and the second contains all the intermediate values. Suppose that we use an operator that leaves the intermediate values unchanged and

reduces all the values in the first and third intervals. When we reduce values in the first domain, small illumination gradients will be removed or reduced according to the usual retinex assumption. When we reduce the values in the third domain, the dynamic range is increased, but according to the assumption above, there will be no loss of visual information. The following example demonstrates how such an operator can produce simultaneous contrast. Consider the standard simultaneous contrast display. Its Laplacian cross section (figure 19a) shows that when all values above $\log(10)$ are reduced, and all the other values stay unchanged (figure 19b), only the Laplacian values in the middle of the image are reduced, giving the desired brightness effect as shown in figure 19c. While such an approach can reproduce the standard simultaneous contrast effect, in many other cases such as the Benussi ring with mid-line (see figure 5a) the scheme fails because along the mid-line only Laplacian values which are outside the ring are effected. This makes the resulting ring non-uniform near the mid-line. We also applied this operator to real images, and as will be seen in the next section, this operator gave undesired results, The reason for this failure will be explained later.

6.2 The threshold operator

A basic problem with the threshold operator, already mentioned by Horn, is how to determine the threshold value. The following examples demonstrate that the threshold must change from one input to the other. The first example is the Mach-band. Each row in the Laplacian of the Mach-band, contains a pair of negative and positive pulses (see figure 20) that depend on the slope of the Mach-band profile. A flat slope will give small pulses so the threshold operator might remove them, but they carry all the image information. When they are removed, the resulting brightness will be constant, which is not the desired brightness in the case of Mach-bands.

The second example shows again that there is no specific threshold value for all inputs if one wishes to handle a broad range of inputs. Consider the image of a white square placed on a uniform black background as in figure 21a. The Laplacian of this image is shown in figure 21b. Its values at the four corners of the square are twice as large as the values anywhere else on the square boundary. Using this observation, for any threshold value, we can find a square having intensity such that only the four Laplacian values on the corners will exceed the threshold value. The solution for such a Laplacian

is shown in figure 21e. This does not match our perception since we either see a uniform square, or, at low contrast, fail to see the figure.

One possible way to overcome this particular problem is to use Laplacian decomposition into x and y components. The threshold operator is applied to each component separately, and then all components are summed back to form the new Laplacian. In this particular example, this decomposition solves the problem since now, on the corners the two components, one in the x direction and the other in the y direction are prevented from being added together before the threshold operator. So now the threshold operator either removes all the components and gives a uniform output or it does not remove anything and the square is visible.

The last problem is also the most serious one and it concerns the requirement that the given Laplacian must be valid. Claim 1 showed that a valid Laplacian \underline{d} must satisfy $\underline{1}^T \underline{d} = 0$ (must have zero mean). We also saw that a solution \underline{p}_1 of invalid \underline{d} has the following Laplacian:

$$L\underline{p}_1 = \underline{d} - \frac{\underline{1}^T \underline{d}}{n} \underline{1}$$

This means that areas where we wished the Laplacian \underline{d} to be zero cannot be zero whenever $\underline{d} \notin \text{Range}(L)$. The threshold operator does not guarantee to give a valid Laplacian. This problem is general for all non-linear operators and will be discussed in the next section.

6.3 General operators in retinex like algorithms

In the last section we focused on a particular form of a non-linear operator, the threshold operator, and discussed its limitations. In this section we consider more general non-linear operators and conclude that they also have inherent limitations in brightness computations.

Consider all non linear point operators which are defined by non linear functions, whose argument is the value of the Laplacian, and every Laplacian value is mapped into a new value. Formally,

$$\underline{z} \xrightarrow{D} \underline{d}$$

such that

$$d_i = f(x_i)$$

Where $f : R^1 \rightarrow R^1$ is any non linear function. The following example shows why all such operators cannot give the correct brightness in all cases, and at the same time produce simultaneous contrast. Consider the 1-D image shown in figure 22a. From psychophysical experiments (for example [14]), we know that the brightness of the the B-bar is higher than the brightness of the A-bar, since the B-bar is surrounded by a darker background than the background surrounding the A-bar. However, since the surrounding steps of the left structure equals to the inner steps of the right structure and vice versa, any operator that simply maps these values into new values gives a valid Laplacian \underline{d} ¹ whose corresponding image \underline{p}_1 such that $(L + \alpha \underline{1})\underline{p}_1 = \underline{d}$ is shown in figure 22b. The brightness of bar-A still equals to the brightness of bar-B and simultaneous contrast is not produced.

The second problem as stated in the previous section is that these operators will not always give a valid Laplacian, that is $\underline{1}^T \underline{d} \neq 0$. It turns out that in solving $(L + \alpha \underline{1})\underline{p}_1 = \underline{d}$, small changes in \underline{d} can have a large effect on the final output \underline{p}_1 . To demonstrate this, consider the 1-d signal having the following Laplacian:

$$d(x) = \begin{cases} -1 & x = 199 \\ 0.99 & x = 200 \\ 0.01 & x = 201 \\ 0 & otherwise \end{cases} \quad (48)$$

Now, consider the same $d(x)$ except that $d(201) = 0$. This Laplacian is invalid since $\sum_x d(x) = -0.01$. Figure 23 shows the two signals solving the problem for each of the two Laplacians. The blue signal is for the valid $d(x)$ and the red for the invalid. Indeed, the two outputs are significantly different although the two Laplacians are almost identical.

6.4 Towards a better Non linear operator

All the non linear operators, discussed up to this point, did not guarantee to give a valid Laplacian \underline{d} . The inverse Laplacian operator, as we mentioned, validates \underline{d} by $\underline{d} \rightarrow \underline{d} - \frac{\underline{1}^T \underline{d}}{n} \underline{1}$. This validation operation gives undesired results and we would like to discuss some other options to validate \underline{d} .

¹In this specific example, for every Laplacian value z_i there is a corresponding value $-z_i$ and if we assume that $f(x) = -f(-x)$ we therefore have that the new Laplacian is valid.

The first option is to multiply all the positive values by a constant c :

$$d(i) = \begin{cases} c \cdot d(i) & d(i) \geq 0 \\ d(i) & d(i) < 0 \end{cases}$$

Where c is chosen such that:

$$c \sum_{d_i \geq 0} d_i + \sum_{d_i < 0} d_i = 0$$

This method will leave all the zero values unchanged and no artificial gradients will be added in regions where the Laplacian is zero.

The idea behind the second option raises from an observation concerning mach-bands and the simple sharp edge examples. These examples might give a clue for a local operator that validates the Laplacian \underline{d} . The Laplacian of a one-dimensional sharp edge is composed by two adjacent opposite pulses that locally balance to zero. The Laplacian of a one-dimensional mach-band is composed by two opposite pulses that are farther away from each other and locally, there is no balance to zero in their regions. An operator that checks for local balance and when necessary adds balancing pulses will add such pulses to the Laplacian of the mach band and leave the Laplacian of the sharp edge unchanged. This operation will give the desired brightness for both the mach band and the sharp edge (See figure 18).

6.5 conclusions

The observations and phenomena discussed in this section lead to the conclusion that if a brightness or lightness algorithm uses a retinex scheme, then its non-linear operator must be context dependent, and not depend solely on the pointwise value of the Laplacian. That is, the non-linear operator, must depend in a more complex manner on the entire input, or at least a neighborhood of each point. The problem of determining how such an operator depends on the context raises complex new problems for the brightness computations. These conclusions support the assumption (for example, see [16]) that Lightness perception involves all levels of processing: low-level, mid-level and high level.

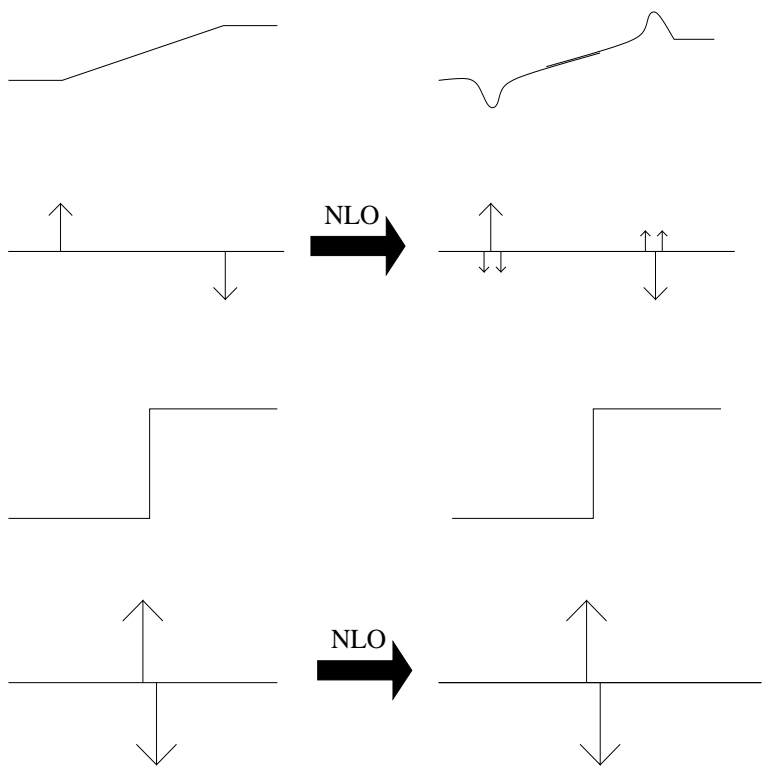


Figure 18: Non linear operator that balances the Laplacian locally towards zero will give the desired brightness for both mach-band and sharp edge.

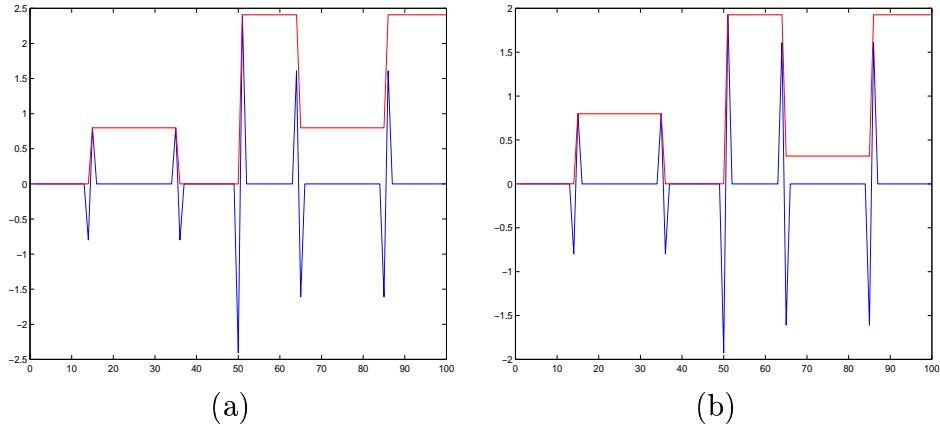


Figure 19: (a) - The logarithm of the simultaneous contrast display (red) along with its Laplacian values (blue). (b) The resulting profile retrieved (red) when we reduce all Laplacian values above $\log(10)$ (blue).

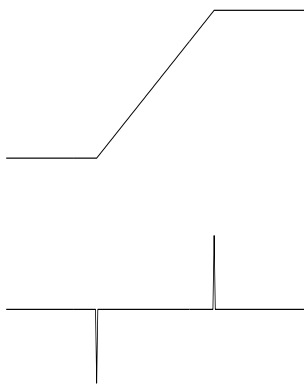


Figure 20: Cross section of the mach-band image and a cross section of its corresponding Laplacian.

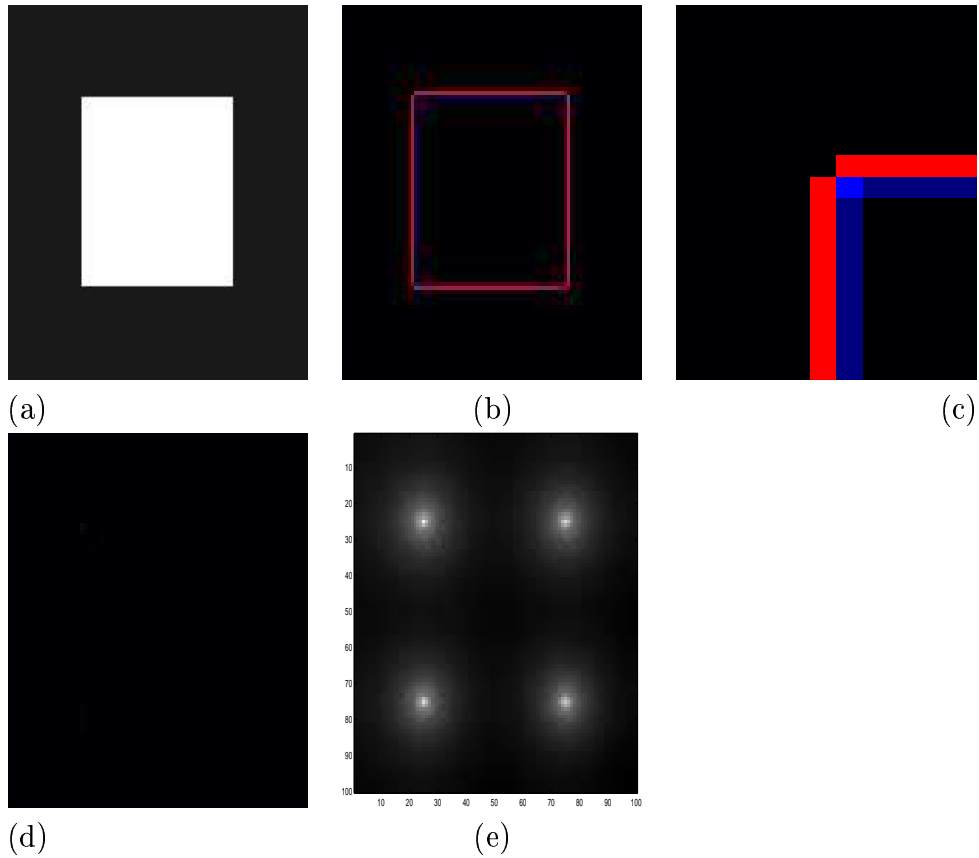


Figure 21: (a) - White box over uniform background. (b) Its Laplacian values. Red is for negative values and blue for positive. (c) Zoom in on the upper-left corner of (b). (d) The new Laplacian image after the threshold operator. (e) The image which has Laplacian image as in (d).

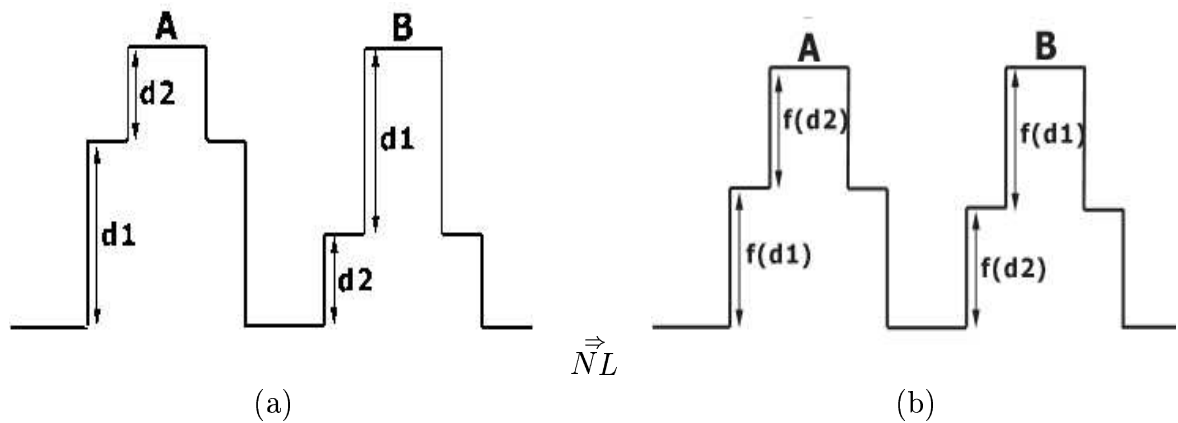


Figure 22: (a) - Two bars A,B in the same intensity are placed over a different background and therefore should have different brightness. (b) - After using a simple non-linear operator, the two bars still have the same brightness.

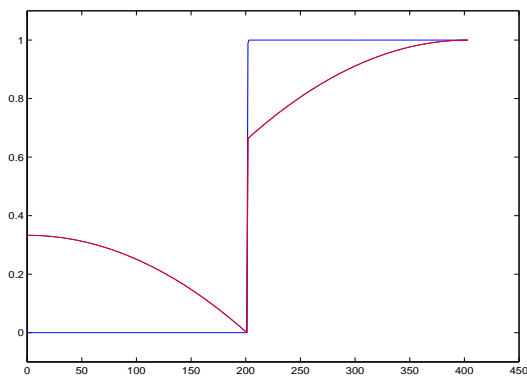


Figure 23: The blue signal has the valid Laplacian $d(x)$ defined in equation 40. The red signal is the output given by the minimization problem for the invalid Laplacian $d^i(x)$ that is almost identical to $d(x)$ except that $d(201) = 0.01$ and $d^i(201) = 0$.

7 Results and Conclusions

7.1 Results

In previous sections we examined primarily simple artificial images. In this section the retinex-like computation is applied to several natural images. These results include outputs produced by the standard threshold operator and also by the three intervals operator suggested in the previous chapter. The three intervals operator is defined by the following non linear function:

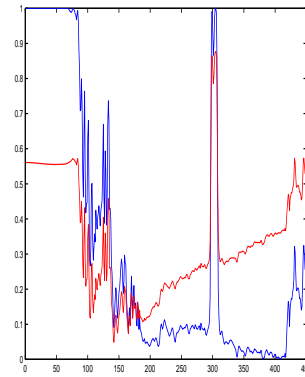
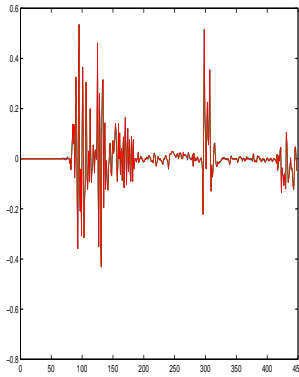
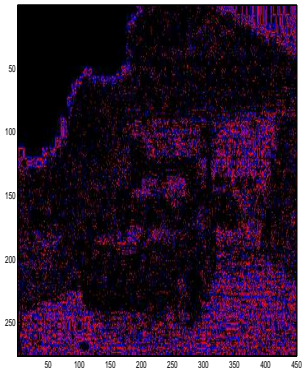
$$f(x) = \begin{cases} 0 & |x| < T1 \\ x & T1 \leq |x| < T2 \\ \frac{x}{2} \left(1 + \frac{T2}{|x|}\right) & T2 \leq |x| \end{cases}$$

When $T2 = \infty$, this is the regular threshold operator. For RGB images, the calculations were performed in each color panel independently. At the head of each figure, we present the input image. The second row shows the output image and two additional graphs. The first graph is a cross section of two images of Laplacian in the red channel. The blue plot is for $L\underline{p}$ where \underline{p} is the input image. The red plot is for $L\underline{p}_1$ where \underline{p}_1 is the output image. Note that these two Laplacians are very similar and therefore the red plot overrides most of the blue plot. The second graph is a cross section of two intensity levels in the red channel. The blue plot is a cross section of the input image \underline{p} and the red plot is a cross section of the output image \underline{p}_1 . The two numbers below the output image are the pair T1,T2 used by the three intervals operator. The main conclusion from all these examples is that when it comes to real images, the retinex algorithm does not give satisfying results. In the car figure, for example, the sky on the left side looks greenish. This part of the sky is surrounded by regions of green leaves and grass and the boundary between them is very vague. As can be seen in the figure, most of the Laplacian along this boundary are smaller than the threshold. After the threshold operator takes affect, there is almost no boundary to prevent the colors in the neighboring regions from diffusing into the region of the sky. The dominant color in these regions is green and therefore the sky gets greenish. This shows again that small values in the Laplacian can hold reflectance information. In the House and Dawn figures, we see that invalid Laplacians can make the output look very different from the input. Notice that the outputs are even worse when we also reduce values which are higher than 0.7. In these cases the distance of the Laplacian average from zero is

increasing and this appears to be responsible for the bad results.

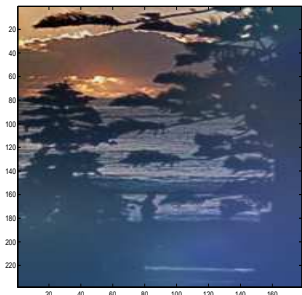


$(T1, T2) = (0.005, \infty)$

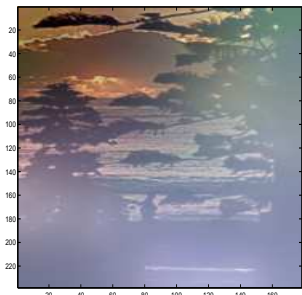


The last image is image of all Laplacian values which have absolute value less than threshold. The color marks the sign, blue for positive and red for negative.

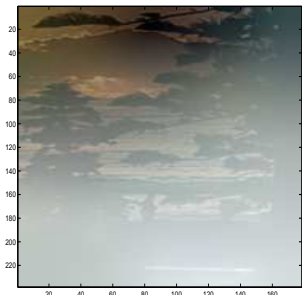
Figure 24: Car



$$(T_1, T_2) = (0.005, \infty)$$



$$(T_1, T_2) = (0.01, \infty)$$



$$(T_1, T_2) = (0.01, 0.7)$$

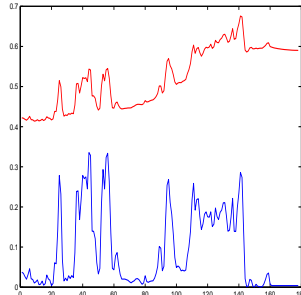
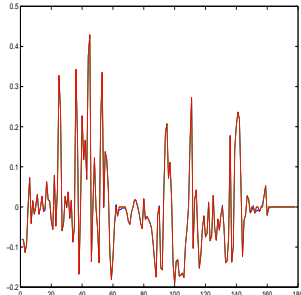
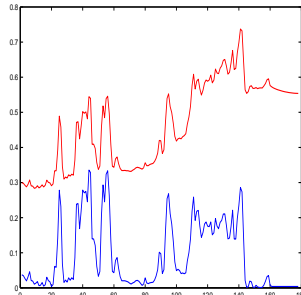
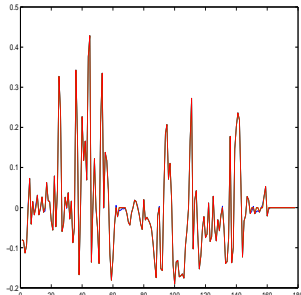
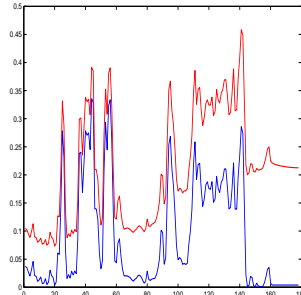
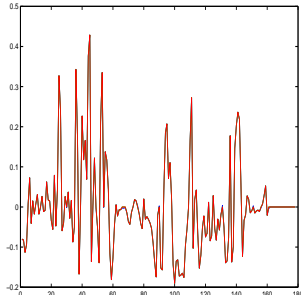


Figure 25: Dawn



$$(T1, T2) = (0.01, \infty)$$



$$(T1, T2) = (0.01, 0.7)$$

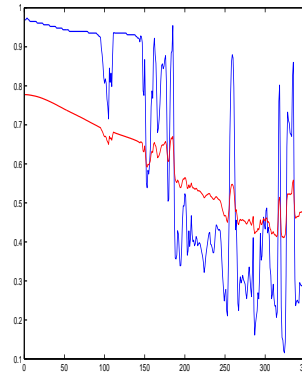
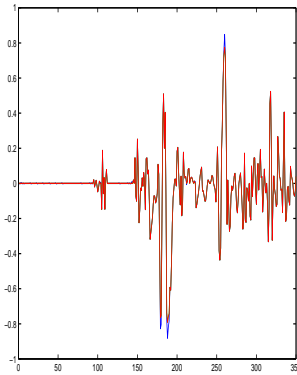
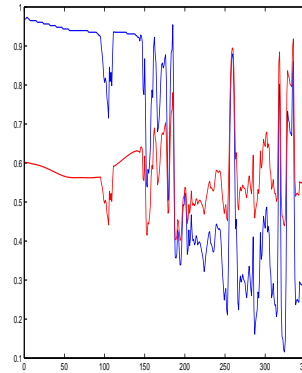
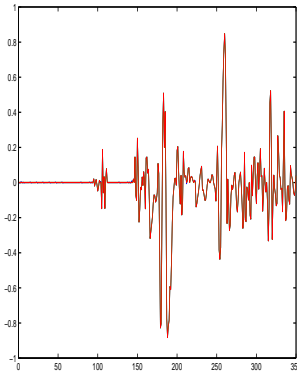


Figure 26: House

7.2 Conclusions

In this thesis we extended and analyzed the retinex-like algorithms for brightness and reflectance computations. We showed that an appropriate Non linear operator may exhibit simultaneous contrast which is important for brightness computations. The original retinex algorithm was essentially a one-dimensional computation. Previous extensions to two-dimensions had two major shortcomings. First, the problem was not fully analyzed in terms of the existence uniqueness and properties of the solution, and the convergence of the proposed method. Second, the solution was inefficient and consequently it was not applied to large-scale images. In this work we developed a well-defined and efficient solution, and examined its behavior on test images. The method appeared to have, both experimentaly and theoretically, severe limitations. Future beneficial use of the method will therefore require to replace the threshold operator with a much more complicated Non-linear operator.

To summarize:

- The retinex-like computation requires a solution to a set of n equations in n unknowns. We have shown that the degree of this system is exactly $n - 1$ and, that an additional constraint can be added to increase the degree of the system to n . We developed an efficient method to solve this well-defined problem with complexity of $O(n \log n)$ where $n =$ number of image pixels. Using this computation we explicitly solved for the inverse Laplacian for large images. The inverse Laplacian was then used to apply the retinex-like computation to artificial and natural images. We have shown that an iterative solution to the equation is guaranteed to converge. We also showed that two-dimensional extension proposed originally by E.Land is ill-defined and will not produce reasonable results.
- The explicit solution for the inverse Laplacian enables a neural network representation for the retinex-like algorithm, shown in figure 27. Each neuron in the output layer is fed with a weighted sum, with weights given by the inverse Laplacian L_1^{-1} , of the \underline{d} layer. The \underline{d} layer is computed by a suitable non-linear operator depending on the \underline{z} layer which is the Laplacian of the input layer, the image \underline{p} .
- The efficient computation of the solution \underline{p}_1 , given a Laplacian \underline{d} can

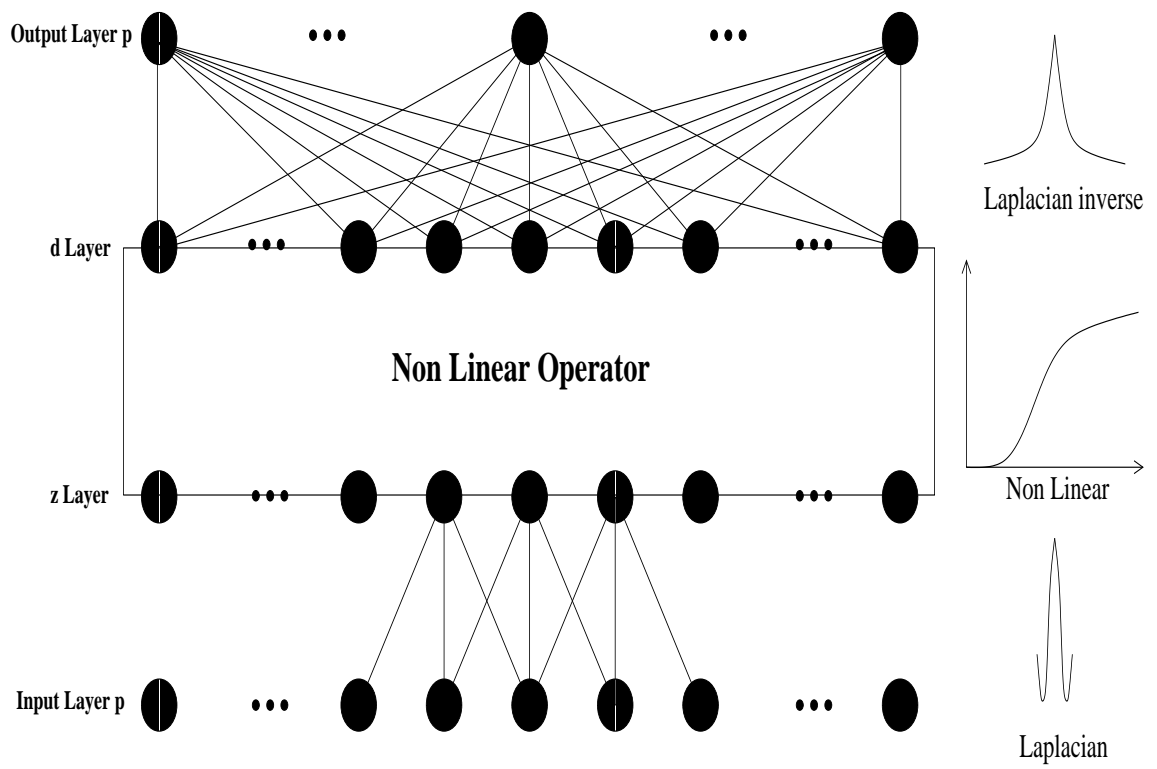


Figure 27: Neural network for retinex-like algorithms.

be summarized in the following scheme.

$$\underline{d} \xrightarrow{\text{Extend}} d^* \xrightarrow{FT} D^* \xrightarrow{\tilde{G}^{-1}} D^* \tilde{G}^{-1} \xrightarrow{FT^{-1}} p_1^* \xrightarrow{\text{Crop}} \underline{p}_1$$

- Retinex-like algorithms can give good model for reflectance and brightness computations in the restricted world of mondrians, and can remove illumination gradients and produce simultaneous contrast in many cases. However, when the assumptions of the mondrian world do not hold, these algorithms usually fail, and the model they present is no longer valid. The nonlinear operator in the general case must be context dependent and dynamic which means that it cannot be global for all inputs.
- Linear algorithms do not give a good model for reflectance or brightness computations. However, in special cases they can be used to explain simultaneous contrast and brightness constancy phenomena.
- We saw that small changes in the Laplacian of an image p can cause significant changes in the output p_1 given by the inverse Laplacian operator. When the inverse Laplacian operator is given an invalid Laplacian \underline{d} , it behaves as it was given $\underline{d} - \frac{1^T \underline{d}}{n} \underline{1}$ instead. This means that regions where we wanted the Laplacian to be zero will have Laplacian that equals to $-\frac{1^T \underline{d}}{n}$ and they will not have uniform brightness or reflectance. This property should be therefore considered when looking for non-linear operators that can give reasonable reflectance and brightness computations.

7.3 Future work

Retinex-like algorithms were developed in the past to remove illumination effects, and recover the underlying reflectance properties in the scene. We have seen that the standard versions of these algorithms have problems in both reflectance and brightness computations. Interesting future work in this area can include the following two problems:

- How to construct a non-linear operator that satisfies all our requirements, namely, gives a valid Laplacian, removes illumination gradients and preserves simultaneous contrast.

- Find more applications to the Laplacian operator and its inverse in image processing and computer graphics. One possible application is in computer graphics where we want to paint regions bounded by possibly incomplete contours. This might be possible using the inverse of the Laplacian since it actually helps solving a diffusion problem.

A Proof for equivalence

We wish to show that the two problems which were defined in chapter 4.3 are equivalent in the sense that if \underline{p}_1 solves the first problem and if \underline{p}_2 solves the second problem then $\underline{p}_1 - \underline{p}_2 = c\underline{1}$ for some constant c . A solution \underline{p}_2 for the minimization problem in the form of equation 16 can be found by the pseudo inverse \hat{L}^\dagger using equation 17 and equation 18. The following notations will be used:

$$\underline{\underline{1}} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \vdots & & & \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{n \times n} \quad \underline{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{n \times 1}$$

Claim 4 Let $\hat{L} = \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix}$, $\hat{\underline{d}} = \begin{pmatrix} \underline{d} \\ - - - \\ m \end{pmatrix}$, $\underline{d} \in R^n$ and $\underline{p}_1, \underline{p}_2$ the solutions defined in chapter 4.3. Then,

$$\underline{p}_2 - \underline{p}_1 = \hat{L}^\dagger \hat{\underline{d}} - (L + \alpha \underline{1})^{-1} \underline{d} = c\underline{1}$$

for some constant c .

Proof:

$$\begin{aligned} \hat{L}^\dagger &= \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix}^\dagger = \left(\left(\begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix}^T \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix} \right) \right)^{-1} \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix}^T \\ &= \left(\left(\begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix}^T \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix} \right) \right)^{-1} = \left(\left(L^T \mid \frac{1}{n}\underline{1} \right) \begin{pmatrix} L \\ - - - \\ \frac{1}{n}\underline{1}^T \end{pmatrix} \right)^{-1} \\ &= \left(L^T L + \frac{1}{n^2} \underline{1} \underline{1}^T \right)^{-1} = \left(L^T L + \frac{1}{n^2} \underline{\underline{1}} \right)^{-1} \end{aligned}$$

By using that $L^T = L$ and that $L\underline{\underline{1}} = \underline{\underline{1}}L = \underline{\underline{0}}$ we have:

$$(L^T L + \frac{1}{n^2}\underline{\underline{1}}) = (L^T + \frac{1}{n}\underline{\underline{1}})(L + \frac{1}{n}\underline{\underline{1}}) = (L + \frac{1}{n}\underline{\underline{1}})^2$$

so,

$$\left(\left(\begin{array}{c} L \\ -\frac{1}{n}\underline{\underline{1}}^T \end{array} \right)^T \left(\begin{array}{c} L \\ -\frac{1}{n}\underline{\underline{1}}^T \end{array} \right) \right)^{-1} = ((L + \frac{1}{n}\underline{\underline{1}})^2)^{-1} = (L + \frac{1}{n}\underline{\underline{1}})^{-2}$$

Therefore,

$$\begin{aligned} \hat{L}^\dagger &= (L + \frac{1}{n}\underline{\underline{1}})^{-2} \left(L^T \mid \frac{1}{n}\underline{\underline{1}} \right) = (L + \frac{1}{n}\underline{\underline{1}})^{-2} \left((L + \frac{1}{n}\underline{\underline{1}}) - \frac{1}{n}\underline{\underline{1}} \mid \frac{1}{n}\underline{\underline{1}} \right) \\ &= \left((L + \frac{1}{n}\underline{\underline{1}})^{-1} - \frac{1}{n}(L + \frac{1}{n}\underline{\underline{1}})^{-2}\underline{\underline{1}} \mid \frac{1}{n}(L + \frac{1}{n}\underline{\underline{1}})^{-2}\underline{\underline{1}} \right) \end{aligned} \quad (49)$$

Again, by using that $L\underline{\underline{1}} = 0$ we have:

$$(L + \frac{1}{n}\underline{\underline{1}})\underline{\underline{1}} = L\underline{\underline{1}} + \frac{n}{n}\underline{\underline{1}} = \underline{\underline{1}}$$

Multiplying both sides by $(L + \frac{1}{n}\underline{\underline{1}})^{-1}$ gives:

$$\begin{aligned} \underline{\underline{1}} &= (L + \frac{1}{n}\underline{\underline{1}})^{-1}\underline{\underline{1}} \\ \Rightarrow (L + \frac{1}{n}\underline{\underline{1}})^{-2}\underline{\underline{1}} &= \underline{\underline{1}} \end{aligned} \quad (50)$$

And if we substitute it in equation 49 we receive:

$$\hat{L}^\dagger = \left((L + \frac{1}{n}\underline{\underline{1}})^{-1} - \frac{1}{n}\underline{\underline{1}} \mid \frac{1}{n}\underline{\underline{1}} \right) \quad (51)$$

The solution \underline{p}_2 was defined in chapter 4.3 as follows:

$$\underline{p}_2 = \hat{L}^\dagger \hat{\underline{d}} \quad (52)$$

so by equation 51 we finally have:

$$\begin{aligned} \underline{p}_2 &= \left((L + \frac{1}{n}\underline{\mathbf{1}})^{-1} - \frac{1}{n}\underline{\mathbf{1}} \mid \frac{1}{n}\underline{\mathbf{1}} \right) \begin{pmatrix} \underline{d} \\ - \\ - \\ m \end{pmatrix} \\ &= (L + \frac{1}{n}\underline{\mathbf{1}})^{-1}\underline{d} - \frac{1^T \underline{d}}{n}\underline{\mathbf{1}} + \frac{m}{n}\underline{\mathbf{1}} \end{aligned} \quad (53)$$

The solution \underline{p}_1 was defined as follows:

$$\underline{p}_1 = (L + \alpha\underline{\mathbf{1}})^{-1}\underline{d}$$

for any $\alpha \neq 0$.

Equation 28 shows that the Laplacian of $(L + \alpha\underline{\mathbf{1}})^{-1}\underline{d}$ does not depend on α . We also know that constant vectors contribute zero to the Laplacian. Therefore we have:

$$L\underline{p}_1 = L\underline{p}_2 \quad (54)$$

We also know that if two images have the same Laplacian, they can differ only by a constant c and therefore

$$\underline{p}_2 - \underline{p}_1 = c\underline{\mathbf{1}} \quad (55)$$

□.

B The Fourier transform of the Laplacian kernel

This section explores the behavior of $\tilde{G}(k, l) = \mathcal{F}_d\{\tilde{g}\}$. In the previous section, \tilde{g} was defined as: $\tilde{g} = g + \frac{\alpha}{4}\underline{\mathbf{1}}$ and therefore we have:

$$\tilde{G}(k, l) = \mathcal{F}_d\{g + \frac{\alpha}{4}\underline{\mathbf{1}}\}(k, l) = G(k, l) + \frac{\alpha}{8\pi}\delta(k, l)$$

$$\text{and } \delta(k, l) = \begin{cases} 1 & (k, l) = (0, 0) \\ 0 & \text{Otherwise} \end{cases}$$

It will be suffice then, to explore the behavior of $G(k, l)$:

$$g = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\begin{aligned} \Rightarrow G(k, l) &= \frac{1}{2\pi} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} g(n, m) W_N^{kn} W_M^{lm} \\ &= \frac{1}{2\pi} (g(0, 0) + g(-1, 0)W_N^{-k} + g(1, 0)W_N^k + g(0, -1)W_M^{-l} + g(0, 1)W_M^l) \\ &= \frac{1}{2\pi} (4 - W_N^{-k} - W_N^k - W_M^{-l} - W_M^l) \\ &= \frac{1}{\pi} \left(2 - \cos\left(\frac{k}{2\pi N}\right) - \cos\left(\frac{l}{2\pi M}\right) \right) \end{aligned} \tag{56}$$

$$\begin{aligned} &= \frac{2}{\pi} - \frac{1}{\pi} \left(1 - \left(\frac{k}{2\pi N}\right)^2 + O\left(\left(\frac{k}{2\pi N}\right)^4\right) \right) - \frac{1}{\pi} \left(1 - \left(\frac{l}{2\pi M}\right)^2 + O\left(\left(\frac{l}{2\pi M}\right)^4\right) \right) \\ &= \frac{1}{\pi} \left(\frac{k^2 M^2 + l^2 N^2}{(2\pi N M)^2} \right) + O\left(\left(\frac{k}{2\pi N}\right)^4, \left(\frac{l}{2\pi M}\right)^4\right) \end{aligned} \tag{57}$$

Conclusions:

1. $\| G(k, l) \| = 0 \xLeftrightarrow[\text{equation 56}] (k \bmod N, l \bmod M) = (0, 0)$
 $\Rightarrow \tilde{G}(k, l) \neq 0 \quad \forall (k, l), \alpha \neq 0$

This is an alternative proof that the operator \tilde{g} has an inverse or in other words that the matrix $(L + \alpha\underline{\mathbf{1}})$ is invertible whenever $\alpha \neq 0$.

2. The operator \tilde{g}^{-1} is scaleable since increasing N or M (the size of the image) only increases the sampling rate of the function in equation 56. For example, if we double the image size, at every even coordinate, the new $\tilde{G}(2k, 2l) = \text{old } \tilde{G}(k, l)$ and all the other values can be approximated by interpolation. This is a general property of Fourier transforms and therefore any convolution operator has a scale-able inverse.

3.
$$\| G(k, l) \| = O\left(\frac{k^2 M^2 + l^2 N^2}{(2\pi NM)^2}\right)$$

$$\Rightarrow \| \frac{1}{G}(k, l) \| \stackrel{\text{equation 57}}{=} O\left(\frac{(2\pi NM)^2}{k^2 M^2 + l^2 N^2}\right)$$

References

- [1] Edwin H. Land, *Color vision and the natural image Part I*, Proc. Nat. Acad. Sci, USA 45:115-129, 1959.
- [2] E.H.Land and J.J.McCann Lightness theory. J.Opt.Soc 62,1971.
- [3] E.H.Land and J.J.McCann Lightness and retinex theory. J. Opt soc. am. 61 1-11.
- [4] Edwin H. Land, *Recent Advances in Retinex Theory*, Vision Res. 26:7-21, 1986.
- [5] Edwin H. Land, *An alternative technique for the computation of the designator in the retinex theory of color vision*, Proc. Nat. Acad. Sci. USA. Vol 83. pp.3078-3080. May 1986
- [6] T.Poggio, V.Torre, C.Koch. Computational vision and regularization theory, Nature 317 1985 315-319
- [7] B. K. P. Horn, Determining lightness from an image, Computer graphics and image processing (1974) 3, 277-299.
- [8] Berthold K. P. Horn and Robert W. Sjoberg, *Calculating the reflectance map*, Applied Optics Vol. 18,No.11 June 1979.
- [9] A. C. Hurlbert and T. A. Poggio, *Synthesizing a color algorithm from examples*, Science, 239:482-485, 1988.
- [10] B. A. Wandell. 1995 *Foundations of vision*. Sunderland , Massachusetts.
- [11] Marr, D. 1982 *Vision*. W.H.Freeman and company.
- [12] A. C. Hurlbert, *The computation of color*, Technical report 1154, MIT Artificial intelligence laboratory.
- [13] Michael E. Taylor, *Partial Differential Equations I - Basic theory* Springer chapter 5 sections 1,2,3,7
- [14] E. G. Heinemann, *Simultaneous brightness induction as a function of inducing and test field luminances* J. Experimental Psychology Vol 50, No 2. 1955.

- [15] T. N. Cornsweet. 1970. *Visual perception*. Academic press.
- [16] E.H. Adelson To appear in M.Gazzaniga, ed., *The cognitive Neurosciences*, 2nd ed. Cambridge, MA: MIT press.
- [17] Y. Saad. 1996. *Iterative methods for Sparse linear systems*. PWS Publishing company.
- [18] G. Strang. 1976. *Linear Algebra and Its Applications*. Academic press.