IDIAP COMMUNICATION

# Lighting Normalization Algorithms for Face Verification

Guillaume Heusch [1] Fabien Cardinaux [2]
Sébastien Marcel [2]

IDIAP–Com 05-03

March 2005

[1] EPFL student, MSc. internship performed at IDIAP
[2] IDIAP Research Institute, Martigny, Switzerland

**IDIAP Research Institute** **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

# Table des matières

# Table des figures

# 1 Introduction

In the field of image analysis, face recognition has been one of the most active area of research in the last few years. According to a recent survey [49], there are two main reasons for this trend : first, there exists a wide range of real-life applications, ranging from user-authentication (access control, ATM, secure online banking) to video surveillance and law enforcements. Second is the availability of feasible technologies after 30 years of research. Although reliable biometrics authentication techniques are available and widely used nowadays (such as iris scan or fingerprint analysis), they rely on an active participation of the user. One of the major advantages of face recognition is its user-friendly capability, since it demands very little cooperation from the user. Thus it is said to be *non-intrusive*.

Face recognition actually deals with two tasks : identification and identity verification. Based on the face image information, the aim of identification is to find out the identity of a given individual out of a set people, through the use of pre-recorded faces images. On the other hand, face verification has to discriminate between two kinds of events : either the person claiming a given identity is the true claimant, either the person is an impostor [12].

Many techniques have been proposed for face recognition : early ones were mainly based on facial features extraction (nose, mouth, eyes) and on the distance between those features points. Since the seminal paper by Turk and Pentland [42] in 1991, where they introduce the Principal Component Analysis (PCA) applied to face images (eigenfaces), more sophisticated techniques for face recognition arose. Among them, one can mention Fisherfaces [7], based on linear discriminant analysis (LDA), Elastic Bunch Graph Matching [49], as well as approaches borrowed from neural networks and machine learning theory, like Multi-Layer Perceptron (MLP) [13] , Support Vector Machines (SVM), or Hidden Markov Models (HMM) [33] to quote a few.

Nevertheless, face recognition remains an active research area : one evidence is the Face Recognition Vendor Test (FRVT) [28] carried out in 2002. It showed interesting and promising results but also provided new research directions. Actually, this test reveals improvements in the last few years concerning face recognition applied to indoor images under controlled lighting conditions : the best systems are comparable to results obtained with fingerprint analysis. Pose handling has also been examined and the use of 3D morphable models showed substantial improvement in performance for non-frontal images. On the other hand, FRVT also concluded that research had to be made in order to cope with recognition across time : the performance of the tested systems decreases as the time elapsed between the gallery image and the probe image grows. Another major challenge is the recognition performed on outdoor images. Moreover, Zhao *et al.* [49] also identifies pose and illumination as major factors affecting the performance of face recognition algorithms.

## 1.1 Illumination Variation in Face Recognition

It is easy to imagine that illumination variations affects the performance of face recognition systems, since different illumination conditions can produce very different images of the same object, as illustrated in the figure below. Adini, Moses and Ullman [2] showed that image variation due to lighting changes is more significant than that due to different personal identities. In other words, the difference between two face images of the same individual taken under varying lighting conditions is larger than the difference between any two face images taken under the same lighting conditions.

In a face recognition framework, it is possible to deal with illumination at three different stages : during the pre-processing, the feature extraction and the classification [33]. However, the feature extraction stage does not seem to cope with illumination variation. Actually, Adini et al. [2] empirically showed that classical image representation such as edge maps, derivatives of the gray level as well as the image filtered with 2D Gabor-like functions are not sufficient for recognition purpose under a wide

**Fig. 1:** Illustration of the variation in face images of the same individual due to different lighting conditions.

variety of lighting conditions. This observation was later formally proved in [14], where the authors showed that there are no functions of an image that are discriminative illuminant invariants. They yield this result by showing that for any two images (not necessarily of the same object), there is always a family of surfaces, albedos and light sources that could have produced them. Consequently, face recognition based on feature extraction only had been given up by the community (although more recent work ([37], [20]) reveals that certain features are less sensitive to lighting conditions).

In the recent years, two main different approaches have been studied to circumvent the problems induced by illumination variations : the first one tries to modify the input image, to turn it into in a more suitable representation for recognition purpose. Consequently, this approach deals with pre-processing only, no matter which kind of classifier is used. A well-known example of such pre-processing algorithm is histogram equalization. On the other hand, the second approach is generative : it tries to model the object of interest under all possible lighting conditions and thus cannot be dissociated from the classification procedure.

## 1.2 Outline of the report

The remainder of this report is organized as follows : in section 2, common solutions to illumination invariant face recognition are briefly outlined, including state-of-the-art generative methods, as well as simple, but widely used pre-processing algorithms. Section 3 then presents selected lighting normalization algorithms relying on properties of the human visual system. Namely, the study is performed with the Multiscale Retinex algorithm, as well as two up-to-date algorithms : the Self-Quotient Image and anisotropic diffusion-based normalization, which involves elliptic partial differential equations. A specific section is thus devoted to the numerical resolution of boundary value problems, since there exists numerous techniques allowing the resolution of large sparse systems induced by the finite difference representation of partial differential equations. Section 5 then focus on the face verification system, namely Gaussian Mixture Models, used to assess the performance of the developed pre-processing algorithms. Experiments settings and implementation issues are addressed in section 6, as section 7 presents the obtained results. Finally, section 8 draws a conclusion of our work and suggests further possible developments.

# 2 Related Work

In this section, related work to face recognition across illumination is described. As mentioned in the introduction, the literature proposes two distinct ways to tackle illumination variation. Roughly speaking, the aim of the first approach is to *model*, based on training examples, the set containing all images of an object viewed under arbitrary lighting conditions. Some of the developed algorithms belonging to this family are outlined in the next subsection.

The second way to cope with illumination variation is to apply a pre-processing step to the image, so as to yield a representation that is better suited for recognition purposes. This approach can be further divided into two categories : operations used to enhance the original image, and operations that compensates illumination conditions, based on properties of the human visual system. Image enhancement algorithms are briefly described in section 2.2, as illumination normalization techniques will be more deeply investigated later in the report.

## 2.1 Model-based approaches

This section outlines some of the various model-based algorithms developed to perform face recognition under variable lighting conditions. Contrary to pre-processing algorithms, model-based approaches require a training set containing several images of the same individual under varying illumination conditions. A training phase is then performed so as to derive a *model* for every identities, which will be used for recognition purpose. Those algorithms are called *generative*, since they generally have the ability, based on the derived model, to synthesize images of an object under a wide variety of lighting conditions.

Such approaches use the lambertian model for describing images, which states that an image is defined by the surface normals of the object $n^T$, its albedo[1] $\rho$ and the light source $s$ falling on it. Hence, a point located at position $(x, y)$ in the image can be expressed as :

$$I(x,y) = \rho(x,y) \cdot n^T(x,y) \cdot s \qquad \text{without shadows} \qquad (1)$$

$$I(x,y) = \max\left(\rho(x,y) n^T(x,y) s,\, 0\right) \qquad \text{with shadows} \qquad (2)$$

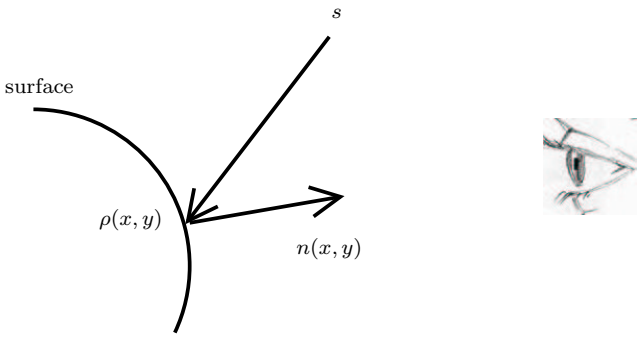



**Fig. 2:** Illustration of the lambertian model

Based on one of these models and on the assumption that the object shape is convex, the outlined algorithms essentially tries to recover information about the albedo and the surface normals $\rho(x,y)n^T(x,y)$ which is supposed to be unique for each individual. To do so, they assume that the set of images of an object under varying lighting conditions forms a linear subspace of the image space.

[1] the albedo is defined as the amount of the light reflected by the surface to the amount incident upon it.

### 2.1.1 Illumination Cone

Belhumeur and Kriegman showed in [5] that the set of images of an object seen under arbitrary lighting conditions is a convex cone which lies in a low-dimensional subspace of the image space. Using the lambertian model with shadows $I = \max(Bs, 0)$ (2), where $B$ is a matrix containing both the shape and the albedo information, they constructed the set of all possible images of a convex lambertian surface created by varying the light source :

$$\mathcal{C} = \left\{ x : x = \sum_{i=1}^{k} \max(Bs_i, 0) \right\} \tag{3}$$

Then, they showed that $\mathcal{C}$ forms a convex cone, defined by its extreme rays (images that are produced by a light source lying at the boundary of two different viewing scenarios, where a viewing scenario defines the illuminated and shadowed pixels in the image, regardless of the light source). Moreover, they proved that the dimension of the cone is equal to the number of distinct normals of the surface, and that in practice, the illumination cone lies near a low-dimensional subspace of the image (by analyzing the eigenvalues of the extreme rays matrix).

In order to perform face recognition, an illumination cone is built for each identity in the database [6]. The illumination cone is derived using a variant of photometric stereo algorithm which does not assume known light source(s). Actually, the authors try to find the matrix $B$ associated with each individual by recursively estimating and refining albedo and surface normals, through the minimization of $\|X - B^*S\|$, where $X$ is the training data matrix, and $S$ is a matrix representing different lighting conditions. Once the matrix $B$, representing a basis for the illumination subspace, is found, a dimensional reduction through SVD is performed so as to represent illumination cones in a more compact way. Recognition is finally performed by measuring the distance between image representations (i.e. approximated illumination cones). The authors reported no recognition errors on the Yale B face database.

### 2.1.2 Spherical Harmonics

In their work [4], Basri and Jacobs also proposed an analysis of the set of images with arbitrary lighting conditions. They first modeled the reflectance function of a sphere of unit albedo under isotropic lighting conditions using spherical harmonics representation. They found that 99% of the reflectance function energy resides in the first few harmonics, confirming the idea that images of an object under different illumination conditions can be well approximated by a low-dimensional subspace. Real images are assumed to be formed using the previously defined reflectance function : each point of the object inherits its intensity from the point on the sphere whose normal is the same. Given a reflectance function $r(x, y, z)$, the image of the point $p$ with albedo $\rho$ and surface normal $(n_x, n_y, n_z)$ is thus expressed as :

$$I(p) = \rho \cdot r(n_x, n_y, n_z) \tag{4}$$

Using the first 9 spherical harmonics as a basis for the reflectance function, the authors approximate the set of images of an object by a linear subspace using so-called *harmonic images* (images obtained by using the basis vectors of the reflectance function). Recognition is then achieved by computing the distance between the probe image and the nearest image that a model can produce.

### 2.1.3 Others related approaches

Several others approaches to cope with illumination variation in face recognition use the result derived by the two preceding works, that is : illumination variation could be modeled as a low-dimensional subspace in the image space. For instance, Wang et al. [45] also showed that the set of images of an object under varying lighting conditions is convex. Based on this observation, they showed that the linear combination of two different illuminated face images of the same person is still a face image of

the same person. So modeling face is done by constructing the subspace from sampled face images under different lighting conditions. In order to perform face recognition, the lighting condition of the probe image is found by interpolation, and every image in the gallery set is relighted, prior to comparison, according to the lighting conditions of the probe image.

Another approach by Zhou and Chellapa [50] deals with both pose and illumination by using an *Illuminating Light-field*. Illumination subspace representation is embedded into a light-field which is used to model pose, so as to derive an identity that is invariant to pose and illumination. However, here we focus on the model they derived for an illumination-free identity representation. Based on the lambertian model : $I = T \cdot s$, where $s$ is the light source and $T$ is a matrix containing albedo and surface normals information, they assumed that any $T$ matrix can be represented as a linear combination of some basis matrix : $T = \sum_i f_i T_i$. Hence, the illumination invariant representation of a face is given by the coefficients vector $f$.

Ho *et al.* [23], extends the classification procedure of [5], by proposing a metric to efficiently cluster illumination cones. Actually, the authors proposed an affinity measure, between illumination cones, which expresses the likelihood that two points are generated by the same cone. Their algorithm acts as follow : first they approximate, by least-square minimization, each input sample $x_i$ as a linear combination of the others input samples : $x_i = \sum_{j,j\neq i} b_{ij} x_j$. A matrix $B$ is filled with the coefficients $b_{ij}$ and symmetrized : $A = (B+B^T)/2$, the clusters are then obtained by applying a spectral clustering method on the constructed matrix $A$.

## 2.2 Pre-processing algorithms

In this section, we briefly describe some simple pre-processing algorithms for image enhancement, used for comparison purpose in [35]. Operations presented in this section generally act directly on the dynamic range of the image and regardless of the image content. They were first studied for display purposes, but may be helpful in various computer vision tasks.

**Gain/offset correction** consists in stretching the image dynamic range so that it fits the dynamic range of the display medium (or any given interval). Considering the dynamic range of the original scene $[r_{min}; r_{max}]$, the operation applied to each pixel value so as to cover the entire dynamic range of the display device, which maximum value is $d_{max}$, is simply :

$$I'(x,y) = \frac{d_{max}}{r_{max} - r_{min}} \cdot \Big( I(x,y) - r_{min} \Big) \tag{5}$$

**Histogram Equalization** tries to transform the distribution of pixels in the image in order to obtain a uniform-distributed histogram. For this purpose, the probability of each gray levels $k$ in the image is computed : $P(k) = \frac{n_k}{N}$ where $n_k$ is the number of pixels with value $k$ and $N$ is the total number of pixels. Then, the distribution of Gray level value $k$ is transformed using the following formula :

$$k' = \sum_{i=0}^{k} \frac{n_i}{N} = \sum_{i=0}^{k} P(i) \tag{6}$$

**Non-linear transforms** is another widely used technique for providing dynamic range compression. It consists in applying a non-linear function, such as logarithm or power functions, on the image so as to enhance dark regions of the scene :

$$I'(x,y) = T(I(x,y)) \qquad \text{with} \qquad T(x) = \log(x) \quad \text{or} \quad T(x) = x^{\gamma} \tag{7}$$

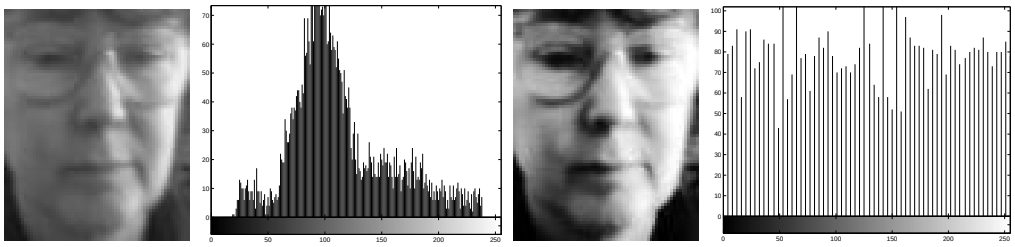


**Fig. 3:** A face image and its histogram before and after equalization

**Homomorphic Filtering** is a technique acting on the frequency domain of the image : after a logarithmic operation, a Fourier transform $\mathcal{F}$ is applied to the image. It is then convolved in the frequency domain with a filter $\mathcal{H}$ that reduces low-frequencies and amplifies high-frequencies [41]. The final image is obtained by taking the inverse Fourier transform $\mathcal{F}^{-1}$ of the filtered image and by applying an exponential, as shown on the diagram below :

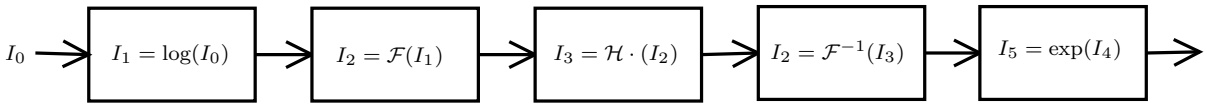



**Fig. 4:** Block-diagram of the homomorphic filtering process

# 3 Selected approaches

## 3.1 Motivation

Considering the framework for our experiments, it turns out that the more suitable choice to deal with illumination variation in face verification is to apply a pre-processing step prior to verification. This choice is motivated by the fact that several face verification systems already exists at IDIAP (and thus the pre-processing could be applied regardless of the following classifier) and the fact that in a realistic application several images under different and slightly varying lighting conditions of the same subject are not always available for training.

As a consequence, we propose to study the effect of several pre-processing algorithms, including state-of-the-art approaches. Roughly speaking, all of those algorithms are based on properties of the human visual system, evidenced by Land & McCann in [27]. This section thus aims at describing the selected algorithms, as well as to present the underlying theory.

## 3.2 Land's Retinex Theory

The Retinex theory (from the words retina and cortex) developed by Land & McCann [27] aims to describe how the human visual system perceives the color/lightness of a natural scene. Actually, our eye is able to build a visual representation with vivid details and colors across a wide range of lighting variations [35]. Moreover, the human visual system has the property of color consistancy. In [10], it is defined as "the maintenance of color appearance despite variation in the color of nearby objects *and* despite variation in the spectral power distribution of the ambient light". In their experiments, Land & McCann found that the color sensation has arbitrary relation with the composition of lights in term of wavelengths and luminance, and that the color sensation is dependent on the surface reflectance [27]. Hence, the goal of Land's retinex theory was thus to understand and to find a computational model of how the eye process the physical stimuli in such a way that color consistancy is preserved. For this purpose, two major assumptions have been stated :

1. The human visual system performs the same computation independently in each of the color channels[2].
2. In each channel, the intensity signal is proportional to the product of the illumination and the surface reflectance (i.e. $I(x, y) = L(x, y) \cdot R(x, y)$)

The general idea of Land's Retinex algorithm assumes that the perceived sensation is dependent on a relative measure of brightness : the so-called *lightness* value, which is the biologic correlate of reflectance (and will be referred as reflectance hereafter for consistancy). So the authors proposed that the reflectance value of a pixel can be computed by a series of comparison of the pixel's intensity with that of many other pixels in the scene [19]. A pixel reflectance value is computed by taking the ratio of the pixel intensity with others pixels in the image. Several practical implementation have been proposed in the literature, and we will focus on one of the latest in the next subsection.

## 3.3 Center/Surround Retinex Algorithm

In the center/surround version of the retinex theory, the reflectance value is found by computing, for each pixel in the image, the ratio of the pixel with a weighed average of the pixels contained in a surrounding area. This implementation makes sense since it keeps the main idea of the original retinex algorithm : as stated in [27] : "Processing the image in terms of the ratios of luminance at closely adjacent points generates dimensionless numbers that are independent of illumination". Moreover and according to [24], center/surround version of retinex is related to the neurophysiological functions of individual neurons in the primate retina, lateral geniculate nucleus and cerebral cortex.

[2] In our case, the processing is performed on the luminance channel only, since we process gray-scale images.

In [25], Jobson *et al.* propose a single scale retinex (SSR) algorithm where the reflectance value is given by the ratio of the treated pixel with a weighed average of the intensity in the surrounding area. The weighed average of the surrounding pixels is obtained by performing a convolution of the image with a gaussian filter. Mathematically, the derived reflectance value could be expressed as :

$$R(x,y) = \log I(x,y) - \log[I(x,y) * G_s(x,y)] \tag{8}$$

where $R(x,y)$ denotes the reflectance value, $I(x,y)$ stands for the original image and $G_s(x,y)$ denotes the surround function of size (or scale) $s$, which is a free parameter, usually dependent of the application. An important result evidenced by equation (8) is that the reflectance value can be derived from a blurred version of the image, since the illuminance field $L(x,y)$ is obtained by convolving the initial image with a gaussian filter.

This model was then extended to the multiscale case, which consists basically in combining several single-scale retinex so as to achieve better results [24]. Hence, the MSR output is simply the sum, with equal weighting, of the outputs of several SSR at different scales :

$$R(x,y) = \sum_{s=1}^{S} \Big( \log[I(x,y)] - \log[I(x,y) * G_s(x,y)] \Big) \tag{9}$$

According to the authors, the Multiscale Retinex algorithm defined by equation (9) can be applied without prior assumptions on any image. Unlike other simple image preprocessing algorithms, MSR has more stable performance : both high and low intensity levels are correctly processed.

## 3.4 Self-Quotient Image

### 3.4.1 Theoretical Basis : The Quotient Image

Riklin-Raviv and Shashua introduced the concept of Quotient Image ([36]) in order to derive an illuminant invariant signature image of an object of a particular class (in our case, the class of objects is face). For this purpose, they started by defining the ideal class of objects as the collection of 3D lambertian objects that have the same shape, but differ in their albedo patterns. Thus objects belonging to the ideal class all share the same surface normals. They also based their work on the result of their previous work, stating that the image space generated by varying the light source vector $s$ lives in a 3D linear subspace : consider three images $I_1$, $I_2$ and $I_3$ of the same object, defined by its surface normal $n^T$ and its albedo patterns $\rho$, and viewed under light source $s_1$, $s_2$ and $s_3$ respectively. The linear combination of the three images is an image $I = \rho n^T s$, where $s = \sum_{i=1}^{3} x_j s_j$. Thus, the authors showed that without considering shadows, three images are sufficient for generating the image space of an object.

Given two objects **a** and **b** belonging to the same class, the quotient image is defined as the ratio of their albedo functions : $Q = \frac{\rho_a}{\rho_b}$. It is obvious that the quotient image is indeed illumination free, but both of the albedo functions are unknown. Nevertheless, the authors claim that it is possible to recover $Q$ using a bootstrap set of images of objects belonging to the same ideal class. Let us consider the case where the bootstrap set is made of three images $a_1$, $a_2$ and $a_3$ of the same object **a** viewed under different lighting conditions. Following the lambertian model, each image is expressed as : $a_i = \rho_a n^T s_i$, where $n^T$ represents the surface normals and $s_i$ represents the light source. Let **y** be another object of the same class and let $y_s = \rho_y n^T s$ be an image of the object, viewed under light $s$. The quotient image of object **y** against **a** is expressed as :

$$Q_y = \frac{y_s}{\sum_{j=1}^{3} x_j a_j} = \frac{\rho_y n^T s}{\sum_{j=1}^{3} \rho_a n^T s_j} = \frac{\rho_y \sum_{j=1}^{3} x_j s_j}{\rho_a \sum_{j=1}^{3} x_j s_j} = \frac{\rho_y}{\rho_a} \quad \text{for some } x_1, x_2, x_3 \tag{10}$$

In order to recover $x = (x_1, x_2, x_3)$, the authors first defined the normalized albedo function of the training set : consider now that the bootstrap set contains 3 images, taken under three different lighting conditions (same for all objects), of $N$ objects. The normalized albedo function is defined as : $\rho(x,y) = \sum_{i=1}^{N} \rho_i^2(x,y)$. Under the hypothesis that $\frac{\rho(x,y)}{\rho_y(x,y)} = \alpha_1\rho_1(x,y) + \ldots + \alpha_N\rho_N(x,y)$, the authors showed that $x$ is the minimum of an appropriate energy function, and can be found by resolving a linear least-square problem.

### 3.4.2 Definition & Analysis

The Self Quotient Image (SQI) approach was proposed by Wang, Li and Wang in [44]. This method is basically an extension of the quotient image : instead of finding a illumination free representation of an object using images from a training set, the denominator is derived directly from the input image :

$$Q(x,y) = \frac{I(x,y)}{F * I(x,y)} \tag{11}$$

where $I(x,y)$ is the input image and $F$ is a smoothing kernel. With respect to the quotient image, the SQI has several advantages : the first, and main important one, is that no training images are required, which implies that it can be used as a pure pre-processing algorithm prior to classification. It also get rid of the assumption that all faces share the same shape, and of the problem induced by objects alignment.

Using the lambertian model with shadows : $I = \min(\rho n^T s, 0)$, the authors propose an analysis of the illumination free representation for three cases of different shapes and shadow conditions.

**Case 1 : no shadows on almost constant surface** This is the simplest case : there are no shadows and the variations of $n(x,y)^T$ are small (the surface is more or less flat), so the term $n(x,y)^T s$ is assumed to be constant. The Self Quotient Image is illumination free since it is given by :

$$Q(x,y) = \frac{\rho(x,y)}{\rho(x,y) * F} \tag{12}$$

**Case 2 : no shadows on varying surface** In this case, the term $n(x,y)^T s$ is not constant, so it cannot be removed from the ratio. In the regions where $n(x,y)^T$ vary significantly (i.e. edges), the SQI is not illumination free :

$$Q(x,y) = \frac{\rho(x,y)n(x,y)^T s}{(\rho(x,y)n(x,y)^T s) * F} \tag{13}$$

**Case 3 : shadowed regions** In these regions, two more assumptions has been stated : first, the gray value is low and less variable, hence $n(x,y)^T$ remains constant. Second, the light is uniformly distributed from all directions and thus $s = \sum_{j=1}^{\infty} s_j(x,y)$. Consequently, the dot product $n(x,y)^T s$ is constant and $Q(x,y)$ may be expressed as in equation (12).

### 3.4.3 The Smoothing Filter

The core component of the Self Quotient Image is the smoothing filter $F$ in equation (11). It actually consists in a modified gaussian kernel, so as to perform anisotropic smoothing. Anisotropic smoothing allows to smooth an image without discarding edges, as shown in figure 5.

**Fig. 5:** Illustration of anisotropic smoothing : the original and the filtered image, where edges are preserved.

For this purpose, the kernel is weighed, for each convolution region, according to the following policy : first the mean intensity $\tau$ of the convolution region is computed. Then, two non-overlapping sub-regions, say $M_1$ and $M_2$, are built. Each pixel in the window is assigned to a sub-region : if $I(x,y) \leq \tau$ then $I(x,y) \in M_1$, otherwise $I(x,y) \in M_2$ for all $x, y$ in the convolution window. Then assuming than $|M_1| > |M_2|$, the weighed gaussian filter is defined by :

$$\frac{1}{N}\sum_{\Omega} WG = 1 \qquad \text{with} \qquad W(i,j) = \begin{cases} 1 & \text{if} \quad I(i,j) \in M_1 \\ 0 & \text{if} \quad I(i,j) \in M_2 \end{cases}$$

where $N$ is a normalization factor and $G$ is the original gaussian kernel, as $\Omega$ denotes the convolution window domain. Note that the $WG$ multiplication is performed element-wise. This filter indeed preserves edges since the convolution will only be performed with one side of the edge, as depicted in the figure below.

A natural question raised by this implementation is how to choose the filter size. Actually, if the scale is too small, the filtered image will be close to the original image, and hence $Q$ will be close to 1, resulting in a loss of information. On the other hand, if the filter is too large, the filtered image will be

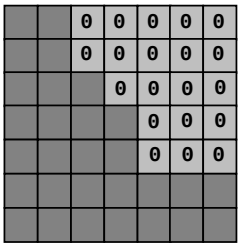



**Fig. 6:** The weighed gaussian filter. Zero coefficients prevents the average to be computed for the whole window region. Only the biggest region is taken into account, such that edges do not tend to be discarded.

more or less constant, so the input image will just be normalized with its mean value. So, the authors decided to use several scale, as in the Multiscale Retinex, in order to achieve more robust results. Moreover, they introduced a non-linear transform as a post-processing step to perform dynamic range compression on $Q$. If this non-linear operation is a logarithm, the only difference between SQI and MSR resides in the use of the weighed kernel instead of a pure gaussian one, as shown below.

$$\begin{aligned} Q(x,y) &= \sum_{k=1}^{n} \log(Q_k(x,y)) \\ &= \sum_{k=1}^{n} \log\left(\frac{I(x,y)}{F_k * I(x,y)}\right) \\ &= \sum_{k=1}^{n} \Big( \log[I(x,y)] - \log[WG_k(x,y) * I(x,y)] \Big) \end{aligned} \tag{14}$$

## 3.5 Diffusion-based methods

The diffusion processes in image processing were first applied for image restoration and rely on partial differential equations. Actually, partial differential equations were widely use as a mean to reduce noise, but also for deblurring purpose, so as to transform the original image into a more pleasant one, according to our visual system. Recently, partial differential equations were applied to more sophisticated problems, such as the inpainting problem, which consists in finding the original image from a corrupted one (the image can be corrupted by adding text on it for instance). Applications of partial differential equations methods may be useful in a wide range of practical applications, such as medical image analysis or in satellite imagery for instance ([15, 18]).

In our framework, partial differential equations are used to derive a blurred version of the original image, since it has been remarked in the previous sections that a blurred version of the image can be seen as an approximation of its illuminance field $L(x,y)$.

### 3.5.1 Isotropic Diffusion

The blurring of an image can be achieved through the use of the linear diffusion equation (also known as the *heat conduction* equation) with the original image as initial conditions [18]. Mathematically, the linear diffusion process could be expressed as :

$$\frac{\partial I(x,y,t)}{\partial t} = \operatorname{div}\big[\nabla I(x,y,t)\big] \tag{15}$$

where $I(x,y,t)$ is the image at time $t$, and $\nabla I$ is the image gradient. It has been shown that the solution of this linear diffusion equation is indeed the original image $I(x,y,0)$ convolved with a gaussian

kernel. Consequently, filtering an image with a gaussian filter is similar to apply a given number of diffusion steps : the number of diffusion steps is referred as the total diffusion time and is related to the scale parameter $\sigma$ of the gaussian filter.

In an image processing framework dealing with pixels and thus not continuous, this equation has to be discretized. Using the same discretization scheme as in [32] and [1], the diffusion process could be written as a function of the time index $t$ :

$$I_{t+1}(x,y) = \left[ I(x,y) + \frac{1}{\Omega} \sum_{d=1}^{\Omega} \nabla I_d(x,y) \right]_t \tag{16}$$

where $\Omega$ is the number of directions in which diffusion is computed and $\nabla I_d(x,y)$ is the directional derivative in direction $d$ at location $(x,y)$. As it can easily be seen in equation (16), a diffusion step basically updates the treated pixel using an average of its neighboring pixels, regardless of the image content in the considered surrounding region.

### 3.5.2 Anisotropic Diffusion

Anisotropic diffusion was first introduced by Perona and Malik in 1990 in [32]. Until then, scale-space description of images[3] was derived using gaussian filters, and this approach has a major drawback : semantic meaningful informations, represented by edges, are discarded at coarse scales. To circumvent this problem, Perona and Malik improved the previous isotropic diffusion model by adding spatially-varying diffusion coefficients :

$$\frac{\partial I(x,y,t)}{\partial t} = \operatorname{div}\big[\mathbf{c}(x,y,t)\nabla I(x,y,t)\big] \tag{17}$$

They chose $\mathbf{c}(x,y,t)$ to be a function of the gradient $g(\nabla I)$, which makes sense since the gradient measures the discontinuities in an image, often representing boundaries between two distinct regions. In their paper, they suggested two different functions for the conduction coefficients :

$$g(\nabla I) = \exp\left( -\left( \frac{\|\nabla I\|}{K} \right)^2 \right) \qquad \text{and} \qquad g(\nabla I) = \frac{1}{1 + \left( \frac{\|\nabla I\|}{K} \right)^2}$$

where the constant $K$ has to be empirically chosen and is application-Dependant. As can be seen in figure 7, if $\nabla I$ is small, then the diffusion process, in the corresponding direction, is more pronounced and, on the other hand, when $\nabla I$ is small, the smoothing is attenuated and thus edges are preserved. However, analysis conducted in [9] and [48] showed that both of the functions $g(\nabla I)$ may lead to unstable results.

### 3.5.3 Variational Formulation

Variational models have received many attention in the recent years from the image processing community. Actually, it is a powerful framework to deal with classical image processing problems and has the advantage to provide well-established results derived through decades of research by mathematicians. Examples of application includes shape from photometric stereo, smart image binarization [26], deconvolution (deblurring), texture modeling [16] or even segmentation [15] [43]. The study of variational, or energy-based models in image processing is by far beyond the scope of this thesis, but since it has been used in Gross & Brajovic's algorithm [21], its use to perform anisotropic diffusion is briefly introduced.

[3]The scale-space description is obtained by generating coarser resolution images of the original image.

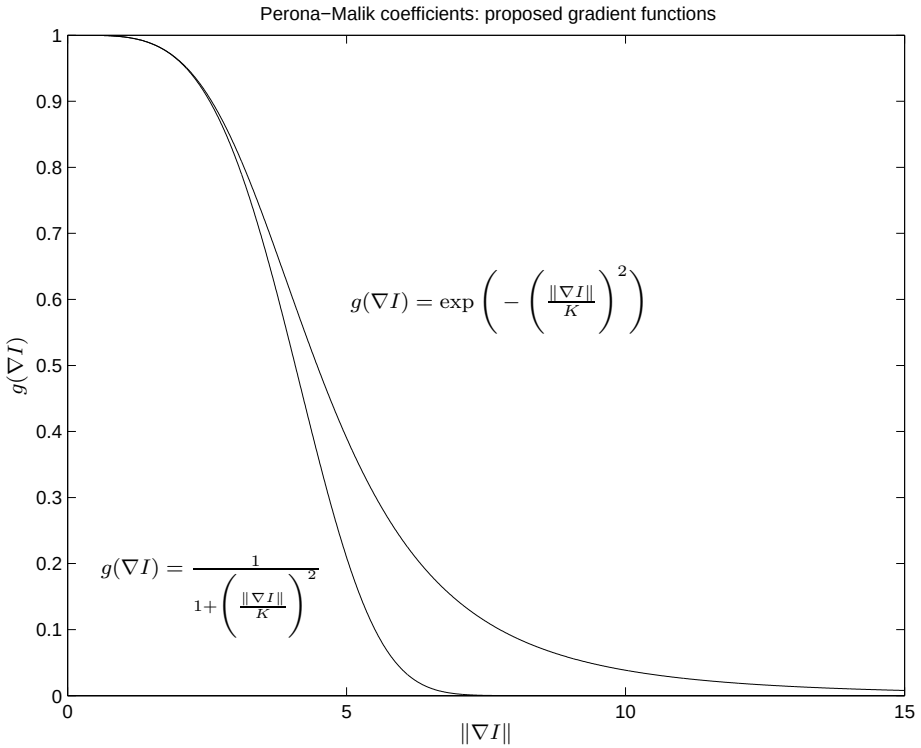



**Fig. 7:** Functions proposed in [32] for the anisotropic coefficients.

In their analysis [48], You *et al.* showed the relation between Perona-Malik anisotropic diffusion and energy-based methods. They started by defining the following energy functional :

$$E(u) = \int_{\Omega} f(|\nabla u|) d\Omega \tag{18}$$

where $u$ denotes the image and $f(|\nabla u|) \geq 0$ is an increasing function of $|\nabla u|$. Hence, $E(u)$ can be viewed as a measure of the image smoothness and minimizing it, is similar to smoothing. After some mathematical computations, the authors found the gradient of $E$ to be given by :

$$\nabla E(u) = -div\left[f'(|\nabla u|)\frac{\nabla u}{|\nabla u|}\right] \tag{19}$$

As in classical optimization approaches, the minimization of $E(u)$ is performed by moving in the negative direction of the gradient (steepest descent). It then becomes a time process and so we reach the parabolic partial differential equation :

$$\frac{\partial u}{\partial t} = -\nabla E(u) = div\left[f'(|\nabla u|)\frac{\nabla u}{|\nabla u|}\right] \tag{20}$$

Equation (19) is obviously the same as equation (17), provided that the coefficients satisfy $\mathbf{c}(x, y, t) = \frac{f'(|\nabla u|)}{|\nabla u|}$. The authors then used this framework to study the behavior of the anisotropic diffusion process with respect to the shape of the energy surface.

### 3.5.4 Gross and Brajovic's algorithm

In their paper [21], Gross and Brajovic addressed the problem of lighting normalization by using calculus of variations. Recovering the reflectance from the illuminance-reflectance model, where an

image is regarded as the product between the illuminance and the reflectance : $I(x,y) = R(x,y)I(x,y)$, is an ill-posed problem. Based on the common assumption that the luminance $L(x,y)$ can be estimated as a blurred version of the original image, they derive the following energy-based model to recover $L(x,y)$ :

$$E(L) = \iint_{\Omega} \rho(x,y)(L(x,y) - I(x,y))^2 dxdy + \lambda \iint_{\Omega} (L_x^2 + L_y^2) dxdy \tag{21}$$

where the first term forces the luminance function to be close to the image and the second term adds a smoothness constraint on $L(x,y)$. $\Omega$ denotes the image domain, $\rho(x,y)$ are the anisotropic diffusion coefficients and the parameter $\lambda$ controls the relative importance of the smoothness constraint. Note also that if $\rho(x,y) = 1 \,\forall\, x, y$, then isotropic diffusion is performed. The minimum of this functional is found using the corresponding Euler-Lagrange equation :

$$L(x,y) + \frac{\lambda}{\rho(x,y)} \left( \frac{\partial^2 L(x,y)}{\partial x^2} + \frac{\partial^2 L(x,y)}{\partial y^2} \right) = I(x,y) \tag{22}$$

Equation (22) is an elliptic partial differential equation representing a boundary value problem : the values of $L(x,y)$ whenever $x$ or $y$ is on the boundary have to be specified. Since we are in a discrete case, the backward finite difference approximation is used to estimate the second derivative value at $(x,y)$ :

$$\frac{\partial^2 L(x,y)}{\partial x^2} = \frac{-L_{x-h,y} + 2L_{x,y} - L_{x+h,y}}{h} \quad \text{and} \quad \frac{\partial^2 L(x,y)}{\partial x^2} = \frac{-L_{x,y-h} + 2L_{x,y} - L_{x,y+h}}{h}$$

where $h$ is the grid interval : in practice it is equal to 1 and thus can be removed. Consequently the discretized version of the Euler-Lagrange equation (22) is :

$$\begin{aligned} L_{i,j} \quad &+\lambda \Big[ \tfrac{1}{\rho_{i,j-}}(L_{i,j} - L_{i,j-1}) + \tfrac{1}{\rho_{i,j+}}(L_{i,j} - L_{i,j+1}) \\ &+ \tfrac{1}{\rho_{i-,j}}(L_{i,j} - L_{i-1,j}) + \tfrac{1}{\rho_{i+,j}}(L_{i,j} - L_{i+1,j}) \Big] \quad = I_{i,j} \end{aligned} \tag{23}$$

where $\frac{1}{\rho_{a,b}}$ denotes the anisotropic coefficients and are defined as the reciprocal of Weber's contrast. The authors also noted that equally good results were obtained using Michelson's contrast. Those measures give the contrast between pixel $a$ and $b$ by :

$$\begin{aligned} \rho_{a,b} &= \frac{|I_a - I_b|}{\min(I_a, I_b)} \qquad \text{(Weber)} \\ \rho_{a,b} &= \frac{|I_a - I_b|}{I_a + I_b} \qquad \text{(Michelson)} \end{aligned} \tag{24}$$

The use of those coefficients makes sense, since they yield to a well-posed anisotropic diffusion. Actually, You *et al.* derived a criterion to determine whether the diffusion process is well-posed : With the notation $\mathbf{c}(|\nabla u|) = \frac{f'(|\nabla u|)}{|\nabla u|}$, they showed that the diffusion is well-posed if $f''(x) \geq 0$. If we consider the gradient value $|\nabla u|$ in direction $d$ of a single pixel to be approximated as the difference between the two pixels (which is the case), then we have : $c(x) = 1/x$, thus $f'(x) = 1$ and $f''(x) = 0 \,\forall x$.

To summarize, Gross and Brajovic's algorithm derives an illumination free representation of the image : $R(x,y) = \frac{I(x,y)}{L(x,y)}$, where the illuminance field $L(x,y)$ is found thanks to equation (23). Actually, equation (23) represents a linear system of equations : one equation for each pixel in the original image. The next section is thus dedicated to the resolution of large sparse linear systems, and will present several approaches to solve such problems.

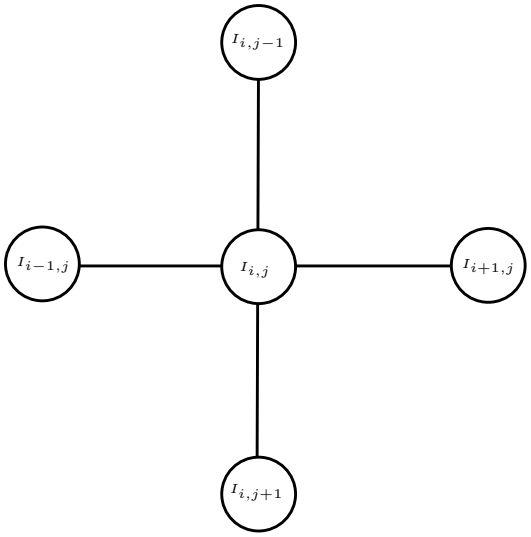



**Fig. 8:** Practical computation of the contrast coefficients with respect to the centered pixel.

# 4 Resolution of Boundary Value Problems

As mentioned in the previous section, Gross and Brajovic's algorithm essentially consists in the resolution of a 2D elliptic partial differential equation. The numerical resolution of such equations usually involves its finite difference representation, which finally leads to a large and sparse linear system of equations (23). Hence, the aim of this section is to present some common iterative approaches used to resolve such systems.

Consider the image $I(x, y)$ to be arranged line-wise in a column vector. Then, equation (23) can be written as : $\mathbf{A} \cdot L = I$, where $\mathbf{A}$ is the so-called *differential operator*. This matrix is actually tri-diagonal with two fringes, hence its structure looks like :

$$\mathbf{A} = \begin{pmatrix} \mathbf{c_D^1} & c_E^1 & & c_S^1 & & & \mathbf{0} \\ c_W^2 & \mathbf{c_D^2} & c_E^2 & & \ddots & & \\ & \ddots & \ddots & \ddots & & \ddots & \\ c_N^w & & \ddots & \ddots & \ddots & & \ddots \\ & \ddots & & \ddots & \ddots & \ddots & & c_S^{n-w} \\ & & \ddots & & \ddots & \ddots & \ddots & \\ & & & \ddots & & \ddots & \ddots & c_E \\ \mathbf{0} & & & & c_N^n & & c_W^n & \mathbf{c_D^n} \end{pmatrix} \tag{25}$$

where $c_N^i, c_S^i, c_E^i$ and $c_W^i$ are the anisotropic diffusion coefficients associated to pixel $i$ in the northern, southern, eastern and western direction respectively (see figure 3.5.4), $n$ is the total number of pixels in the image and $w$ denotes the image width. The diagonal coefficient $\mathbf{c_D^i}$ is given by : $\mathbf{c_D^i} = 1 + (c_N^i + c_S^i + c_E^i + c_W^i)$. Since this square matrix is of size $n$x$n$, $n$ being the number of pixel in the image, the linear system could not be solved by direct matrix inversion, since it requires $n^3$ operations and $n^2 + n$ storage requirements [22]. We will now see alternate approaches to solve large, sparse systems of equation induced by the discretization of partial differential equations.

## 4.1 Relaxation

Relaxation is an iterative method that takes advantage of the sparse structure of the matrix (25). Consider the linear problem : $Ax = b$ and suppose that we have an estimate of the solution $x^0$. The aim of this linear iterative method is to recursively correct the initial estimate according to :

$$x^{k+1} = x^k + B(b - Ax^k) \tag{26}$$

where $B$ will be an approximate of $A^{-1}$. Since we do not take boundary values into account[4], The element-wise formulation of equation (26) can be easily derived for each element to update in $x^{k+1}$. Consider the matrix multiplication $Ax$ in equation (26), we denotes the element of $A$ by $a_{i,j}$ and the element of $x$ by $x_{i,j}$ in order to respect the image geometry. Each line of the matrix, except those corresponding to boundaries, contains only five non-zero coefficients, so we have :

$$(Ax^k)_{i,j} = a_{i-1,i}x^k_{i-1,i} + a_{i,i-1}x^k_{i,i-1} + a_{i,i}x^k_{i,i} + a_{i,i+1}x^k_{i,i+1} + a_{i,i+1}x^k_{i,i+1}$$

Now, an appropriate choice of the matrix $B$ has to be made. Note that if $B = A^{-1}$ only one iteration is sufficient to resolve the problem, thus $B$ is an approximate of $A^{-1}$. Two popular choices are Gauss-Seidel and Jacobi relaxation, and their difference resides in the choice of the matrix $B$. Let us write $A = D - L - U$ where $D$ is the diagonal of $A$, $-L$ and $-U$ are the lower and the upper part of $A$ respectively.

**Jacobi** uses $B = D^{-1}$, thus $d_{i,i} = \frac{1}{a_{i,i}}$ and $d_{i,j} = 0 \quad \forall (i,j) : i \neq j$. We can then write the equation (26) element-wise :

$$\begin{aligned} x^{k+1}_{i,i} &= x^k_{i,i} + b_{i,i} - \frac{1}{a_{i,i}}\Big(a_{i-1,i}x^k_{i-1,i} + a_{i,i-1}x^k_{i,i-1} + a_{i,i}x^k_{i,i} + a_{i,i+1}x^k_{i,i+1} + a_{i,i+1}x^k_{i,i+1}\Big) \\ &= b_{i,i} - \frac{1}{a_{i,i}}\Big(a_{i-1,i}x^k_{i-1,i} + a_{i,i-1}x^k_{i,i-1} + a_{i,i+1}x^k_{i,i+1} + a_{i+1,i}x^k_{i+1,i}\Big) \end{aligned}$$

**Gauss-Seidel** uses $B = (D - L)^{-1}$, thus and differently from Jacobi, the exact formula depends on the ordering of the indices [22]. However, following the formula given in [34], and assuming that elements are processed sequentially, we can write the equation (26) element-wise :

$$x^{k+1}_{i,i} = \frac{1}{a_{i,i}}\Big(b_{i,i} - \Big(a_{i-1,i}\mathbf{x^{k+1}_{i-1,i}} + a_{i,i-1}\mathbf{x^{k+1}_{i,i-1}} + a_{i,i+1}x^k_{i,i+1} + a_{i+1,i}x^k_{i+1,i}\Big)\Big)$$

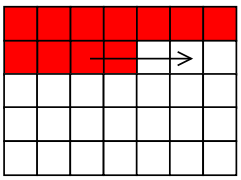


**Fig. 9:** sequential ordering of the Gauss-Seidel relaxation

It has been shown in [47] and [22] that Gauss-Seidel relaxation always converge, provided that $A$ is a positive definite matrix, but that the convergence rate is relatively slow. Actually, relaxation has the property to rapidly reduces high-frequency components[5] in the error, but on the other hand,

[4] although the solution to be found has the form of a column-vector, one must not forget the underlying geometry of the grid defined by the image

[5] frequencies of the error are defined with respect to the underlying grid

low-frequency components of the error will be reduced more slowly [11]. For this reason, relaxation operations are often called *smoothing operators*. As suggested above, Gauss-Seidel relaxation is dependent on the ordering of the mesh points and different schemes can be used, yielding different results [22]. The one used in our work is the so-called *red-black* ordering, also referred as *chequer-board*, since it has been suggested in [34] and [1] as being well-suited for typical second order elliptic equations. It consits in processing the red elements first and then the black ones, as shown in figure 4.1.

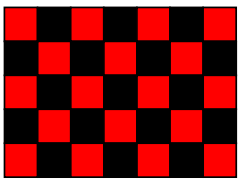


**Fig. 10:** red-black ordering of the Gauss-Seidel relaxation

## 4.2 Coarse Grid Correction

The general principle of coarse grid correction consists in improving an (arbitrary) estimate of the solution thanks to the use of a coarser grid. For this purpose, we will use two grids of different resolution : the first grid is the finer, its size being the same as the original image. The second grid is a coarser one, and is usually obtained by sub-sampling the fine grid by two in each directions.

Let us now assume that we have an estimate $\hat{x}_{l_0}$ of the solution $x_{l_0}$ at the fine grid denoted by $l_0$. The idea of the procedure is to use the coarser grid $l_1$ to derive a correction for the initial estimate $\hat{x}_{l_0}$. The coarse grid correction algorithm is summarized below :

1. Compute the residual on the fine grid : $r_{l_0} = A_{l_0}\hat{x}_{l_0} - b_{l_0}$
2. Project (restriction) the residual down to the coarse grid : $r_{l_1} = \mathcal{R}r_{l_0}$
3. Solve exactly on the coarse grid $A_{l_1}c_{l_1} = -r_{l_1}$
4. Project (interpolation) the correction $c_{l_1}$ to the fine grid : $c_{l_0} = \mathcal{P}c_{l_1}$
5. Correct the initial estimate : $x_{l_0}^{new} = \hat{x}_{l_0} + c_{l_0}$

Let us now look more carefully at the described algorithm, and how it practically works. The first operation is to compute the residual on the finer grid : this will give an indication on how close the initial estimate is to the solution, and can be viewed as an error measure. Then, the residual is projected ($\mathcal{R}$) onto the coarse grid in order to derive a *correction*, in the sense that on the coarse grid the exact difference between the estimate and the solution is found, as described below. Let us consider that we have an (arbitrary) estimate on the coarser grid $\hat{x}_{l_1}$. We have :

$$\begin{aligned} A_{l_1}c_{l_1} &= -r_{l_1} \\ &= -(A_{l_1}\hat{x}_{l_1} - b_{l_1}) \end{aligned}$$

multiplying both sides with $A_{l_1}^{-1}$ :

$$\begin{aligned} c_{l_1} &= -\hat{x}_{l_1} + A_{l_1}^{-1}b_{l_1} \\ \hat{x}_{l_1} - c_{l_1} &= A_{l_1}^{-1}b_{l_1} \end{aligned}$$

hence adding $c_{l_1}$ to $\hat{x}_{l_1}$ will give the exact solution on the coarse grid. Thus, projecting back ($\mathcal{P}$) the correction $c_{l_1}$ on the fine grid and adding it to the initial estimate will drive $\hat{x}_{l_0}$ toward the exact solution. Moreover, this procedure will speed up the process, since on the coarse grid, the dimension

of the original system is divided by a factor 4.

Now the inter-grids transfer functions $\mathcal{R}$ and $\mathcal{P}$ have to be defined ; for simplicity and as suggested above, we assume that coarser grids are obtained by successively dividing the original grid size by two in each direction. According to [22], an obvious proposal for the prolongation (coarse-to-fine) operation $\mathcal{P}$ is the two-dimensional linear piecewise interpolation (bilinear interpolation). Once prolongation has been defined, on has to define the fine-to-coarse operation. The most simple restriction scheme $\mathcal{R}$ is certainly the *injection* : copy fine-grid pixels to the corresponding position on the coarser grid. A more precise, and also popular choice ([34], [11], [47]) for fine to coarse projection is the *full-weighting* scheme, which consists in assigning a weighed combination of a nine-points window to each coarse grid points. Moreover, *full-weighting* restriction is the adjoint to bilinear interpolation : $\mathcal{R} = \frac{1}{4}\mathcal{P}$.

$$\mathcal{P} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \qquad \mathcal{R} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

## 4.3 Two Grids Iteration

Two grids iteration is just an extension of the coarse grid correction scheme described above. Actually, it uses both relaxation and coarse grid correction so as to improve the estimate $\hat{x}_{l_0}$. The idea is to use the relaxation main advantage : the error term decreases rapidly with the first relaxation sweeps (then it decreases more slowly as the relaxation proceed), since this method is good at reducing high-frequency components of the error. Moreover and as stated in [34] and [11], high-frequency components of the error, whose wavelenghts are less than $2H$, $H$ being the fine grid-spacing, could not be represented on the coarse grid and so cannot be removed on this grid. The idea is hence to reduce high-frequency components of the error on the fine grid, before proceeding with the coarse grid correction :

1. Compute $\bar{x}_{l_1}$ by applying $\nu_1 > 0$ relaxation steps to $\hat{x}_{l_1}$
2. Perform coarse grid correction (4.2) to derive $\bar{x}_{l_1}^{new}$ from $\bar{x}_{l_1}$
3. Compute $\hat{x}_{l_1}^{new}$ by applying $\nu_2 > 0$ relaxation steps to $\bar{x}_{l_1}^{new}$

## 4.4 Multigrid V-cycle

The Multigrid V-cycle algorithm is, once again, an extension of the method described above. It is based on the idea that on a sufficiently coarse grid, even very low frequencies are oscillatory with respect to the mesh size, so the use if multiple grids to solve the linear system of equations through relaxation makes sense. The V-cycle multigrid algorithm actually consists in nested iterations of the two grids correction procedure.

Consider the two-grid operation described above : on the coarse grid, the equation $A_{l_1} c_{l_1} = -r_{l_1}$ is solved by direct resolution. The idea of the V-cycle is then to use another two grid iteration to solve the residual equation instead of a direct resolution, and so on, until the residual equation can be solved rapidly by direct matrix inversion (i.e. until the size of the differential operator $\mathbf{A}$ is small enough to invert it without large computational expense). The complete V-cycle multigrid algorithm is outlined below :

1. Apply $\nu_1 > 0$ relaxation steps to initial estimate $\hat{x}_{l_0}$
2. Compute the residual on the fine grid : $r_{l_0} = A_{l_0}\hat{x}_{l_0} - b_{l_0}$
3. Restrict the residual to the next coarser grid : $r_{l_1} = \mathcal{R} r_{l_0}$
4. Solve $A_{l_1} c_{l_1} = -r_{l_1}$ by a **recursive call** to the V-cycle algorithm
5. Interpolate the coarse grid solution $c_{l_1}$ to the fine grid $c_{l_0} = \mathcal{P} c_{l_1}$

6. Add the correction to the estimate $\hat{x}_{l_0} \leftarrow \hat{x}_{l_0} + c_{l_0}$
7. Apply $\nu_2 > 0$ relaxation steps to the corrected estimate $\hat{x}_{l_0}$

So, the algorithm begins on the finest grid $l_0$ by applying few relaxation sweeps, hence removing oscillatory components of the residual $r_{l_0}$. Then, the residual is projected onto a coarser grid, where smooth components becomes oscillatory, and we have another sparse linear system to solve (the residual equation $A_{l_1} c_{l_1} = -r_{l_1}$). Instead of solving this problem directly, the same procedure as in the first grid is applied, until we reach the coarsest grid $l_n$. On the coarsest grid, the correction $c_n$ for the problem at level $l_{n-1}$ is obtained by direct resolution. It is then interpolated back to the next to the grid $l_{n-1}$ and added to the estimate of the solution. Before being interpolated, the estimate of the solution at level $l_{n-1}$ is again improved by applying few relaxation sweeps and interpolated to the next finer grid $l_{n-2}$, and so on, until the finer grid $l_0$ is reached. The complete procedure is represented on figure 11

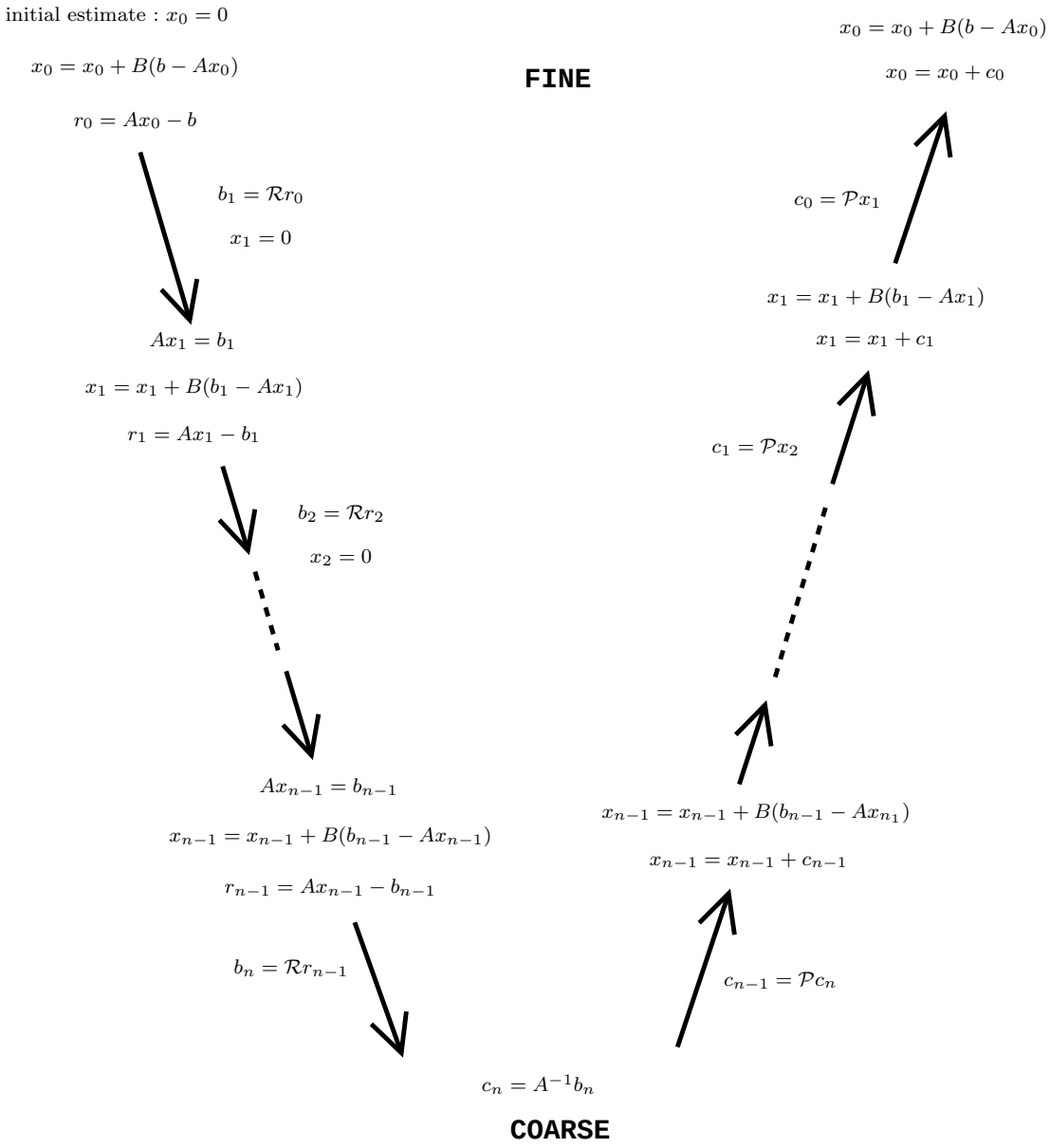



**Fig. 11:** Schema of the V-cycle algorithm

# 5 Face Verification using GMM

Consider the scenario where a client $C$ must be authenticated through face verification. The task of a face verification system is then to make a decision, based on the face image of the client : either the claimed identity corresponds to the identity derived from the face image (true claimant), or the client is an impostor. This section presents an algorithm that gives an opinion on whether the client is the true claimant or not.

## 5.1 Feature Extraction

The feature extraction technique applied prior to classification is known as DCT$mod2$ and is described in [39]. Each face image is analyzed on a block by block basis (with overlap), a block being of size $N$x$N$ with $N = 8$.

**2D DCT** Each block, denoted as $b(x, y)$ is decomposed using 2D DCT basis functions, the results is a $N$x$N$ matrix containing DCT coefficients given by :

$$C(u, v) = \alpha(v)\alpha(u) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} b(x, y)\beta(x, y, u, v) \tag{27}$$

where $\alpha(x)$ is given by :

$$\alpha(x) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for} \quad x = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases} \tag{28}$$

and

$$\beta(x, y, u, v) = \cos\left[\frac{(2x + 1)u\pi}{2N}\right] \cos\left[\frac{(2y + 1)v\pi}{2N}\right] \tag{29}$$

**DCT delta** Following [39], DCT deltas, derived from the coefficients of the 2D DCT transform, are usually used in speech processing. However, here they are used to derive additional information to embed into the feature vector. Actually, the delta coefficients of a block will give informations of the relation of the block with its neighbors. The delta coefficients may be computed in the horizontal (neighboring blocks on the same line) and in the vertical (neighboring blocks on the same column) directions.

**DCT*mod2*** is a way to build a features vector that are less sensitive to illumination changes. Actually, it has been noted in [39] that the first three coefficients are affected by lighting conditions. So, the authors proposed to replace those coefficients by their horizontal and vertical deltas. The final feature vector for the block located at $(a, b)$ is then given by :

$$\vec{x}_{(a,b)} = [\Delta^h c_0 \quad \Delta^v c_0 \quad \Delta^h c_1 \quad \Delta^v c_1 \quad \Delta^h c_2 \quad \Delta^v c_2 \quad c_3 \quad c_4 \dots c_{M-1}]^T \tag{30}$$

## 5.2 Gaussian Mixture Models

Gaussian Mixture Models (GMM) are used compute the likelihood that a set of observed data $X = \{\vec{x}_t\}_{t=1}^{N_v}$ were generated according to a previously learned model of parameter set $\theta$. In a face recognition framework, GMM are responsible to model the set of feature vectors derived from a face image of a given client $C$. The probability that the set of observed feature vectors was generated according to the parameter set $\theta$ is computed with :

$$P(X|\theta) = \prod_{t=1}^{N_v} P(\vec{x}_t|\theta) \tag{31}$$

where the probability distribution $P(\vec{x}_t|\theta)$ is modeled as a sum of gaussian distributions :

$$P(\vec{x}_t|\theta) = \sum_{i=1}^{N_G} m_i \mathcal{N}(\vec{x}|\vec{\mu}_i, \Sigma_i). \tag{32}$$

Thus the parameter set denoted by $\theta$, for each distribution, consists in the number of gaussian $N_G$, the weight associated to each gaussian $m_i$, as well as the mean $\vec{\mu}_i$ and the covariance matrix $\Sigma_i$.

**Training** In order to compute the probability given by equation (31). One has to know the parameter set $\theta$. For this purpose, the approach followed in [12] is to first derive the parameters of the world model $\overline{C}$ using the EM algorithm with training data of different clients. Hence, the world model describes the parameters of a *generic* face. Each client model parameter set is then *adapted*, starting from the world model parameters $\theta_{\overline{C}}$, through an iterative procedure acting on the means :

$$\mu_k^{new} = (1-\alpha)\mu_k^w + \alpha \frac{\sum_{t=1}^{T} P(k|\vec{x}_t)\vec{x}_t}{\sum_{t=1}^{T} P(k|\vec{x}_t)} \tag{33}$$

## 5.3 Classification

Let us consider that we have a client $C$, described by its parameters $\theta_C$. We also have the non-client specific parameters $\theta_{\overline{C}}$ describing the world model. Suppose now that the client claims its identity and feed the system with a set of feature vectors $X = \{\vec{x}_t\}_{t=1}^{N_v}$ derived from its face image (5.1), an opinion on whether the client is an impostor or not is given by :

$$\Theta(X) = \log P(X|\theta_C) - \log P(X|\theta_{\overline{C}}) \tag{34}$$

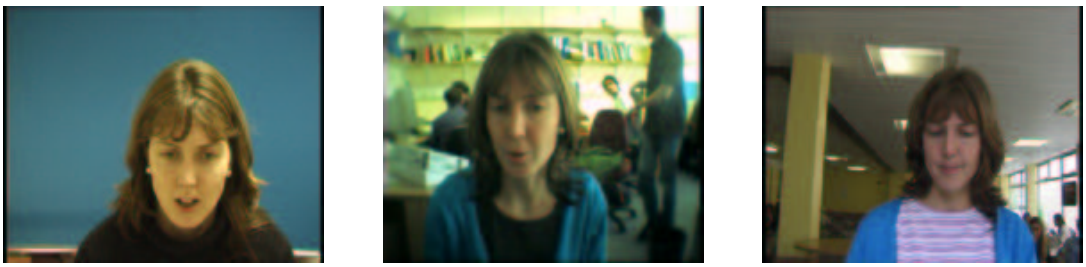
where $P(X|\theta_C)$ denotes the likelihood that the features vectors comes from the true claimant and $P(X|\theta_{\overline{C}})$ is the likelihood that the feature vectors comes from an impostor. The decision is then based on a threshold $\tau$ : if $\Theta(X) \geq \tau$, then the claim is accepted, and if $\Theta(X) < \tau$ the claim is rejected. The threshold $\tau$ is derived thanks to a validation set.

# 6 Experiments

## 6.1 The BANCA Database

The BANCA database [3] was especially meant for multi-modal biometric authentication and includes both speech and image recordings. Image acquisition was performed with two different cameras : a cheap analogue webcam, and a high-quality digital camera, under several realistic scenarios : controlled (high-quality camera, uniform background, controlled lighting), degraded (webcam, non-uniform background) and adverse (high-quality camera, arbitrary conditions). The recordings have been made for 4 distinct corpus (English, French, Spanish and Italian) containing 52 individuals each (26 females and 26 males) during 12 sessions reaching a total of 624 images. Each gender-specific subset of each corpus is again divided into two subsets $g1$ and $g2$, thus containing 13 images of different individuals of the same gender. Each corpus is extended with an additional set of 30 other subjects (15 males and 15 females) and is referred as the *world model*. Since the development of a face authentication system usually involves the use of configurable parameters (i.e. thresholds), one needs to define a development set and an evaluation set : this is achieved by considering the two sets defined by $g1$ and $g2$.

In the BANCA protocol, seven distinct configurations for the training and testing policy has been defined. In our experiments, the configurations referred as Match Controlled (Mc), Unmatched Adverse (Ua), Unmatch Degraded (Ud) and Pooled Test (P) are used. All of the of listed configurations use the same training conditions : each client is trained using images from the first recording session, which corresponds to the controlled scenario. Testing is then performed on images taken from the controlled scenario (Mc), adverse scenario (Ua), degraded scenario (Ud), while (P) does the test for each of the previously described configurations. In other words, all configurations consist in training the system in good conditions, and then to test it using all available conditions. This procedure makes sense since it corresponds to the realistic and challenging scenario of capturing images for the gallery set in good conditions, whereas probe images are more likely to be acquired in more arbitrary conditions.



(a) (b) (c)

**Fig. 12:** Example of the different scenarios : controlled (a), degraded (b) and adverse (c)

**Performance Measures** In every biometric authentication systems, there are two kind of errors, namely the false acceptance, which consists in accepting the claim of an impostor, and the false rejection, which rejects the true client. Those two measures are related, since reducing the number of false rejection will increase the number of false acceptance and *vice versa*.

$$FAR = \frac{\#\text{ of false acceptance}}{\#\text{ of impostor accesses}} \tag{35}$$

$$FRR = \frac{\#\text{ of false rejection}}{\#\text{ of client accesses}} \tag{36}$$

Verification results are then expressed using a DET-curve, which gives the false acceptance rate versus the false rejection rate, as a function of the threshold $\tau$. Two important measures associated to DET-curves are the Equal Error Rate (EER) and the Half Total Error Rate (HTER). Equal Error Rate is actually the point on the DET-curve where the false acceptance rate equals to false rejection rate, whereas HTER is defined as the mean of the false acceptance rate and false rejection rate, and is thus evolving along the curve.

$$HTER = \frac{FAR + FRR}{2} \tag{37}$$

## 6.2 Verification Settings

Face windows are manually extracted from the original images of the English corpus. They are then resized to 64x80 pixels, converted to grayscale and finally pre-processed by the lighting normalization algorithm. As suggested in [41], histogram equalization was used with each method; the complete procedure is illustrated in the figure 13. Training and testing are then performed directly on the normalized images.

**Fig. 13:** Block diagram of the preprocessing algorithm

Since only five images per client are available for training, the size of the set is artificially doubled by mirroring each face image. The world model was trained with faces belonging to the Spanish corpus, and is thus completely independent of the client in the database.

## 6.3 Implementation Issues & Parameters Selection

Optimization of the different parameters related to the lighting normalization algorithms is made using the validation set $g1$ defined above. For each algorithms, GMM were trained on $g1$, using different parameters related to the normalization algorithm (such as the size of the filters or $\lambda$ for the diffusion-based methods). Selected optimal parameters for the lighting normalization algorithms are those for which the HTER *a posteriori* on the set $g1$ is minimum.

**Multiscale Retinex & Self-Quotient Image** It has been previously shown that MSR and SQI are very similar to each other, the only difference relying on the used filter. Thus, the same parameters selection policy has been applied to both algorithms. First, and as suggested in [35] in the Multiscale Retinex implementation, three scales were used. This choice also makes sense for the Self Quotient Image, since it has been shown in [36] that three images were sufficient to derive the quotient image. Moreover, we decided to grant the same weight to each size (as suggested in [44] and [24]), this has the advantage to reduce the set of parameters to tune, which are the scales of the filters.

Since there are three scales to choose, the policy applied to select those scales is the following : one size could vary while the two others remain fixed (fixed size are 5x5 for the minimal size, 15x15 for the intermediate size and 25x25 for the maximum size). Thus, a minimum is reached for each varying size : the final parameters are the three scales that respectively minimize the error (provided that they yield better results, otherwise, the best investigated combination is kept).

**Diffusion** Diffusion experiments were performed with two different methods for resolving the partial differential equation (22), they are simple Gauss-Seidel Relaxation and Multigrid V-cycle. In both case, the main parameter to tune is $\lambda$, which controls the smoothness of the resulting image. Nevertheless, there are others implementations choices for both algorithms. Concerning the relaxation, red-black Gauss-Seidel scheme was preferred to Jacobi, since it ensures convergence, and zero was used as an initial guess. Concerning multigrid V-cycle algorithm, we also use Gauss-Seidel relaxation, since it was shown in [47] that it has better smoothing properties than Jacobi, and smoothing is a crucial operation in multigrid solvers [22]. Moreover, we decided to apply only one pre- and post-smoothing step so as to reduce computational load, and it has also been note in [34] that to be sufficient in many cases. Zero was also used as initial guess for the correction at every grid level (see figure 11). The number of grids was dictated by practical concerns : starting from 64x80 images, we chose to use 5 grids, hence the differential operator $A$ on the coarsest grid is of size 20x20.

Contrary to Multiscale Retinex and Self-Quotient image, diffusion-based normalization is easier to calibrate, since there is only a single parameter to optimize, namely $\lambda$. Figure 14 shows the estimated illuminance field for different values of $\lambda$. So the procedure was to select the value of $\lambda$ for which the verification error rate is minimal, for each investigated type of diffusion and for the two possible resolution schemes (relaxation and multigrid v-cycle). In the appendix, the curves representing the HTER as a function of $\lambda$ are presented. Note that other parameters inherent to the different resolution schemes (such as the number of steps in the relaxation for instance) have been set up according to empirical visual results.

**Fig. 14:** Effect of the smoothing parameter $\lambda$ : the bigger, the smoother the resulting image (images derived using Weber contrast coefficients).

# 7 Results & Discussion

## 7.1 Verification Results

In this section, the verification results are reported for the investigated algorithms[6]. They are expressed in terms of the Half Total Error Rate (HTER), which combines both the false acceptance and the false rejection rate. Final testing is performed on the test set $g2$ (using the parameters optimized on $g1$), with 234 true client access and 312 impostor accesses. The following table presents results for each previously described protocols, namely Mc, Ua, Ud and P. For comparison purpose, we also report some state-of-the-art results on the BANCA database [38] [12] on the last lines of the table 1.

| **Normalization algorithm** | Mc [%] | Ua [%] | Ud [%] | P [%] |
|---|---|---|---|---|
| No preprocessing | 3.686 | 20.192 | 28.205 | 24.733 |
| Histogram equalization | 7.372 | 20.032 | 22.917 | 19.818 |
| Multiscale Retinex | 7.853 | 19.391 | 22.115 | 18.964 |
| Self Quotient Image | 11.378 | 26.763 | 33.654 | 28.098 |
| Isotropic diffusion : relax | 4.808 | 20.192 | **11.538** | 14.850 |
| Isotropic diffusion : v-cycle | **1.923** | 16.186 | 11.699 | 15.224 |
| Weber diffusion : relax | 7.692 | 18.109 | 14.583 | 19.391 |
| Weber diffusion : v-cycle | 4.487 | **15.705** | 12.019 | **12.981** |
| Michelson diffusion : relax | 3.686 | 20.192 | 28.205 | 24.733 |
| Michelson diffusion : v-cycle | 4.006 | 16.346 | 12.340 | 13.782 |
| **Comparison** | | | | |
| P2D-HMM [12] | 3.4 | 12.7 | 15.4 | 16.4 |
| LDA [38] | 4.9 | 16.0 | 20.2 | 14.8 |
| SVM [38] | 5.4 | 25.4 | 30.1 | 20.3 |
| LDA + Weber diffusion [41] | 4.2 | 8.6 | 20.7 | 13.4 |

**Tab. 1:** Verification performances on the BANCA database

## 7.2 Discussion

The first and most important observation is that the diffusion-based normalization improves significantly the performance of the face verification system : this is consistent with the results obtained in [41]. It is also interesting to see that the multigrid implementation yields generally better results than Gauss-Seidel relaxation. The only case which relaxation performs better is in the isotropic case. A possible explanation is stated in [22] : the simpler the problem, the faster the convergence. Since, isotropic diffusion is a simpler problem than the anisotropic one, relaxation is sufficient to approximate the solution efficiently. Note also that isotropic-based normalization improves the verification results in the controlled environment (Mc), which is not the case with the others normalization algorithms.

The performance obtained with Multiscale Retinex is comparable to histogram equalization, whereas Self Quotient Image degrades the baseline performance. This is surprising, since SQI was especially meant to remove illumination variation. Nevertheless, experiments showed that the anisotropic smoothing filter introduced severe artifacts (false contours) in sharp discontinuities as its size increase. Thus it is difficult to find a trade-off between the use of small scales (loss of information) and large scales (artifacts). Moreover, optimal scales seem to be image-dependent (see appendix A). In other words, the anisotropic smoothing implemented in the SQI is not accurate enough to provide stable

[6] Visual results can be found in the appendix A.

results across the whole database.

The best results, obtained by applying the Weber diffusion-based normalization can be compared to state-of-the-art approaches : actually, applying the lighting normalization scheme prior to GMM-based verification yields better results than more sophisticated HMM-based approaches reported in [12]. Moreover, its performances are even better than those of the best systems reported in [38], as can be seen in table 1.

**Computational Load** Since face recognition may be a time critical task, it is interesting to consider the computational load of each normalization approach. It will be dependent on the image size, and on the different parameters used in the normalization algorithm. The most efficient is, as one could have expected, the multigrid V-cycle : it takes 20 ms to process an image using Weber coefficients (independently of $\lambda$) on a P4 running at 3.2 Ghz. Relaxation is more dependent on its intrinsic parameter : the number of relaxation sweeps. However, the computational load of the relaxation-based resolution of the isotropic diffusion remains acceptable for real-time application. On the other hand, relaxation applied with anisotropic coefficients takes almost the same time as the Multiscale Retinex. Note however that the computational load of the MSR, and of the SQI, is strongly dependent on the filter scales : MSR using only small scales is faster than relaxation applied to anisotropic diffusion. The difference between MSR and SQI is explained by the need, in the SQI, to build a specific filter for every convolution window.

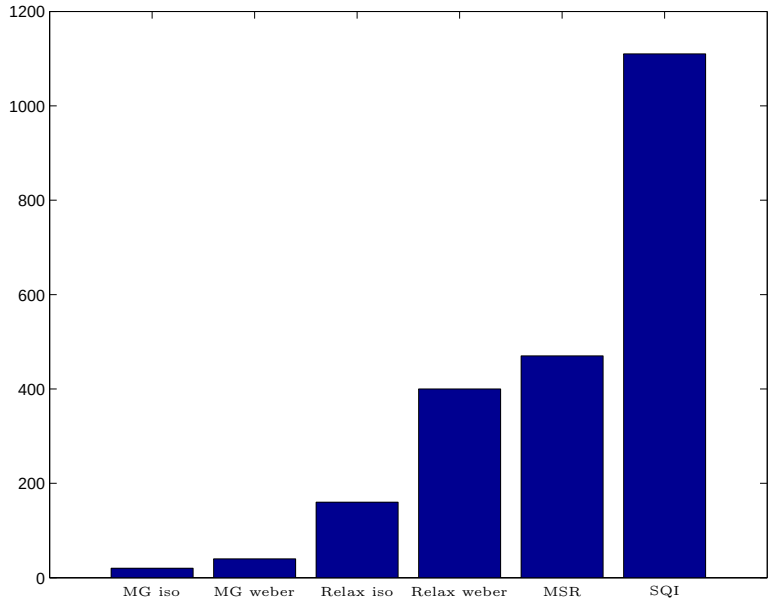



**Fig. 15:** Relative computational load of the investigated algorithms : the most efficient algorithm is defined as the unit time measure, other loads are expressed with respect to this time unit (i.e : as a multiple of the computational load of the most efficient one.)

**Others Considerations** The efficient multigrid implementation allows to process the entire image, instead of the face window, at a resolution of 512x384. So, diffusion-based normalization was also investigated by applying it prior to the face window extraction. This procedure has also the advantage of getting rid of the boundary artifacts induced by the multigrid resolution. However, conducted experiments shows no improvements on the verification performance, but increases the computational

load. Note also that others normalization algorithms cannot be applied to original images due to the huge increase of their respective computational load.

Another interesting point is the feature extraction scheme. A previous work performed at IDIAP [37] investigated the effect of different feature extraction techniques and empirically showed that the DCT*mod*2 features are the best when dealing with illumination variation. However, no pre-processing steps were applied prior to classification, since experiments were conducted on a database containing controlled images only (xm2vts). Thus, we also investigated the use of classical 2D-DCT features to determine whether the normalization algorithm is sufficient to cope with illumination variation, but it was not the case : the combination DCT*mod*2 and lighting normalization performs better.

# 8 Conclusion

In this report, we presented a comparative study of the effect of different pre-processing algorithms on the performance of a face verification system, namely Gaussian Mixtures Models. For this purpose, we investigated three approaches, as well as their practical implementation. Since the most promising approach involved partial differential equations, two different techniques, including the computationally efficient multigrid framework, were studied for the numerical resolution. Experiments results showed that diffusion based lighting normalization clearly outperforms other algorithms and are also easier to calibrate. Moreover, significant improvements on the BANCA database are reported, reducing the verification error rate of the system by 34% on the realistic *pooled test* configuration. Obtained results are comparable with other state-of-the-art approaches for face verification on the same database.

# 9 Acknowledgments

The author would like to thank Fabien Cardinaux and Dr. Sébastien Marcel for their guidance, as well as the AMI project (*www.amiproject.org*) for funding its internship at IDIAP Research Institute, through the student exchange program.

# Références


[1] S.T. Acton : *Multigrid Anisotropic Diffusion*, IEEE Transactions on Image Processing, volume 7, no. 3, 1998.

[2] Y. Adini, Y. Moses and S. Ullman : *Face Recognition : The Problem of Compensating for Changes in Illumination Direction*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 19, Issue 7, 1997.

[3] E. Bailly-Baillire *et al.* : *The BANCA Database and Evaluation Protocol*, International Conference on Audio- and Video-Based Biometric Person Authentication, 2003.

[4] R. Basri and D. Jacobs : *Lambertian Reflectance and Linear Subspaces*, in IEEE International Conference on Computer Vision, 2001.

[5] P. Belhumeur and D. Kriegman : *What Is The Set of Images of an Object under All Possible Illumination Conditions*, International Journal of Computer Vision, volume 28, Issue 3, 1998.

[6] A. Georghiades, P. Belhumeur and D. Kriegman : *From Few to Many : Illumination Cone Models for Face Recognition Under Variable Lighting and Pose*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 23, Issue 6, 2001.

[7] P. Belhumeur, J. Hespanha and D. Kriegman : *Eigenfaces Vs. Fisherfaces : Recognition using Class Specific Linear Projection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 19, Issue 7, 1997.

[8] M. Bichsel : *Illumination Invariant Object Recognition*, Proceedings of the International Conference on Image Processing, 1995.

[9] M. Black, G. Sapiro, D.H. Marimont and D. Hegger : *Robust Anisotropic Diffusion*, IEEE Transactions on Image Processing, Volume 7, number 3, 1998.

[10] D.H. Brainard and B.A. Wandell : *Analysis of the Retinex Theory of Color Vision*, Journal of the Optical Society of America A, 3, 1651-1661, 1992.

[11] W. L. Briggs, V. E. Henson and S.F. McCormick : *A Multigrid Tutorial*, slides derived from the corresponding book, SIAM publications, 2000.

[12] F. Cardinaux, C. Sanderson and S. Bengio : *Face Verification Using Adapted Generative Models*, in IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

[13] F.Cardinaux, C. Sanderson and S. Marcel : *Comparison of MLP and GMM Classifiers for Face Verification on XM2VTS*, International Conference on Audio- and Video-Based Biometric Person Authentication, 2003.

[14] H. Chen, P. Belhumeur and D. Jacobs : *In search of Illumination Invariants*, IEEE International Conference on Computer Vision and Pattern Recognition, 2000.

[15] T. F. Chan, J. Shen and L. Vese : *Variationnal PDE Models in Image Processing*, Notices of the American Mathematical Society, January 2003.

[16] T. Chan, S. Esedoglu, F. Park and A. Yip : *Recent Developments in Total Variation Image Restoration*, Mathematical Models in Computer Vision : The Handbook, Springer, 2005.

[17] C. Cuvelier, J. Descloux, J. Rappaz, C. Stuart and B. Zwahlen : *Eléments d'équations aux dérivées partielles pour ingénieur*, Presses Polytechniques Romandes, 1988.

[18] M. Elayyadi : *Equations aux Dérivées Partielles et Réseaux de Neurones pour le Traitement d'Images*, Thèse de Doctorat, Université Joseph Fourier - Grenoble, 1999.

[19] G.D. Finlayson, S.D. Hordley and M.S. Drew : *Removing Shadows From Images Using Retinex*, Color Imaging Conference, 2002.

[20] B.Fröba and A. Ernst : *Face Detection with the Modified Census Transform*, IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

[21] R. Gross and V. Brajovic : *An Image Preprocessing Algorithm for Illumination Invariant Face Recognition*, International Conference on Audio- and Video-Based Biometric Person Authentication, 2003.

[22] W. Hackbusch : *Iterative Solution of Large Sparse Systems of Equations*, Applied Mathematical Sciences, Volume 95, Springer-Verlag, 1994.

[23] J. Ho, M-H Yang, J. Lim, K-C Lee and D. Kriegman : *Clustering Appearances of Objects Under Varying Illumination Conditions*, IEEE International Conference on Computer Vision and Pattern Recognition, 2003.

[24] D. Jobson, Z. Rahmann and G. Woodell : *A Multiscale Retinex for Bridging the Gap Between Color Images and the Human Observations of Scenes*, IEEE Transanctions on Image Processing, volume 6, Issue 7, 1997.

[25] D. Jobson, Z. Rahmann, and G. Woodell : *Properties and Performance of a Center/Surround Retinex*, IEEE Transactions on Image Processing, March 1997.

[26] R. Kimmel and I. Yavneh : *An Algebraic Multigrid Approach For Image Analysis*, SIAM Journal on Scientific Computing, Volume 24, Number 4, 2003.

[27] E.H. Land and J.J. McCann : *Lightness and Retinex Theory*, Journal of the Optical Society of America , volume 61, n1, 1971.

[28] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone : *FRVT 2002 : Evaluation Report*, available at *www.frvt.org*

[29] K. Messer et al : *Face Competition on the BANCA database*, International Conference on Biometric Authentication, Hong-Kong, 2004.

[30] L. Meylan and S. Süsstrunk, *Bio-inspired Color Image Enhancement*, Proceedings of IS&T/SPIE Electronic Imaging, 2004

[31] A. Le Negrate, A. Beghdadi et K. Boussaid-Belkacem : *Quelques traitements bas-niveau basés sur une analyse du conraste local*, Proceedings of Vision Interfaces, 1997.

[32] P. Perona and J. Malik : *Scale-Space and Edge Detection using Anisotropic Diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 12, Issue 7, 1990.

[33] F. Perronnin and J-L Dugelay : *A Model of Illumination Variation for Robust Face Recognition*, Multimodal User Authentication Workshop, 2003.

[34] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling *Numerical Recipes in C*, Cambridge University Press, 1988.

[35] Z. Rahmann, G. Woodell and D. Jobson : *A Comparison of the Multiscale Retinex with other Image Enhancement Techniques*, Proceedings of IS&T 50th Anniversary Conference, 1997.

[36] A. Shashua and T. Riklin-Raviv : *The quotient image : Class based re-rendering and recognition with varying illuminations*, IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 23, Issue 2, 2001.

[37] M. Saban and C. Sanderson : *On Local Features for Face Verification*, IDIAP-RR 04-36, 2004

[38] M. Sadeghi, J. Kittler, A. Kostin and K. Messer : *A Comparative study of Automatic Face Verification Algorithms on the BANCA database*, International Conference on Audio- and Video-Based Biometric Person Authentication, 2003.

[39] C. Sanderson and K. K. Paliwal : *Polynomial Features for Robust Face Authentication*. IEEE International Conference on Image Processing, 2002.

[40] L. Zhang and D. Samaras : *Pose Invariant Face Recognition under Arbitrary Unknown Lighting using Spherical Harmonics*, Proceedings of Biometric Authentication Workshop 2004.

[41] J. Short, J. Kittler and K. Messer : *A Comparison of Photometric Normalisation Algorithms for Face Verification*, IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

[42] M. Turk and A. Pentland : *Face recognition using eigenfaces*, IEEE International Conference on Computer Vision and Pattern Recognition, 1991.

[43] M. Unser : *Multigrid Adaptive Image Processing*, IEEE International Conference on Image Processing, 1995.

[44] H. Wang, S. Li and Y. Wang : *Face Recognition under Varying Lighting Conditions Using Self Quotient Image*, IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

[45] H. Wang, S. Li and Y. Wang and W. Zhang : *Illumination Modeling and Normalization for Face Recognition*, in International Workshop on Analysis and Modeling of Face and Gesture, 2003.

[46] H. Wang, S. Li and Y. Wang : *Generalized Quotient Image*, IEEE International Conference on Computer Vision and Pattern Recognition, 2004.

[47] J. Xu : *An introduction to multilevel methods*, in *Wavelets, Multilevel Methods and Elliptic PDEs*, Eds. M. Ainsworth, J. Levesley, M. Marletta and W.A. Light, Oxford University Press, 1997.

[48] Y. You, W. Xu, A. Tannenbaum and M. Kaveh : *Behavioral Analysis of Anisotropic Diffusion in Image Processing*, IEEE Transactions on Image Processing, volume 5, number 11, 1996.

[49] W. Zhao, R. Chellapa, P.J. Phillips and A. Rosenfeld : *Face Recognition : A Literature Survey*, ACM Computing Surveys (CSUR), Volume 35 , Issue 4, 2003

[50] S. K. Zhou and R. Chellapa : *Illumination Light Field : Image-based Face Recognition across Illumination and Pose*, IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

# A Visual Results

Here we present some visual results obtained with the different lighting normalization algorithms. Note that visual image quality is not correlated to verification results. The original images are faces window taken from the English corpus of the BANCA database, and belongs to the degraded scenario. For each studied algorithm, the original as well as the processed images are presented.

**Fig. 16:** Effect of the Multiscale Retinex

**Fig. 17:** Effect of the Self Quotient Image

**Fig. 18:** Effect of the Isotropic Diffusion (relaxation)

**Fig. 19:** Effect of the Weber Anisotropic Diffusion (relaxation)

**Fig. 20:** Effect of the Michelson Anisotropic Diffusion (relaxation)

**Fig. 21:** Effect of the Isotropic Diffusion (Multigrid V-cycle)

**Fig. 22:** Effect of the Weber Diffusion (Multigrid V-cycle)

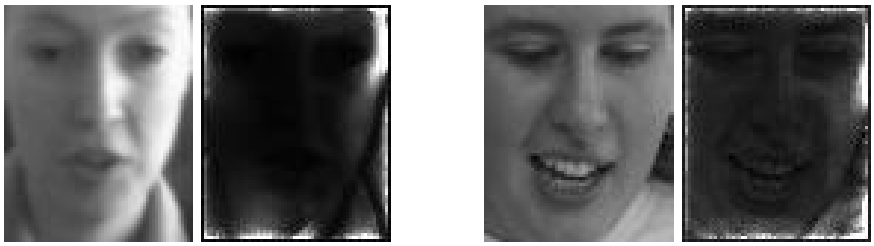


**Fig. 23:** Effect of the Michelson Diffusion (Multigrid V-cycle)

# B Parameters Selection

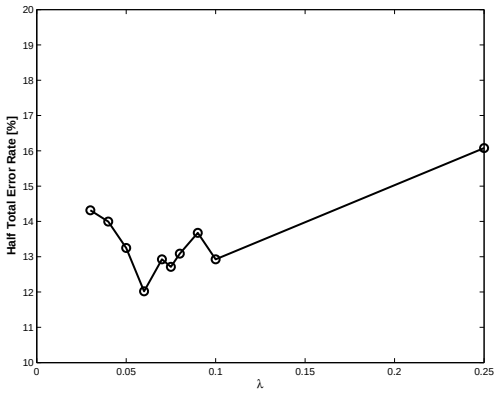



**Fig. 24: Isotropic Diffusion :** Parameter selection for the relaxation procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.

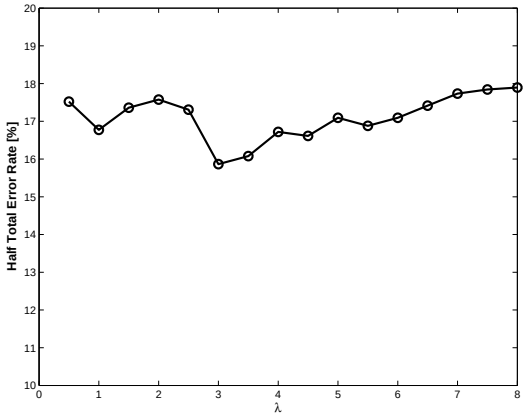



**Fig. 25: Isotropic Diffusion :** Parameter selection for the V-cycle multigrid procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.

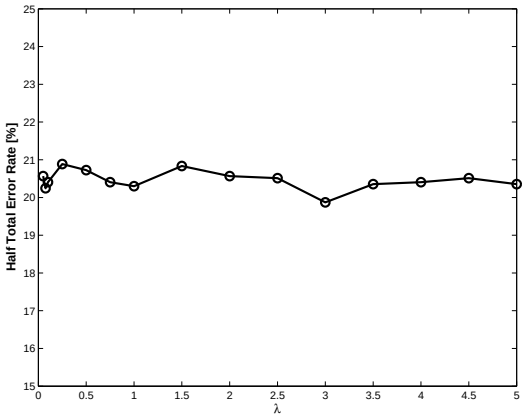



**Fig. 26: Weber Diffusion :**Parameter selection for the relaxation procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.

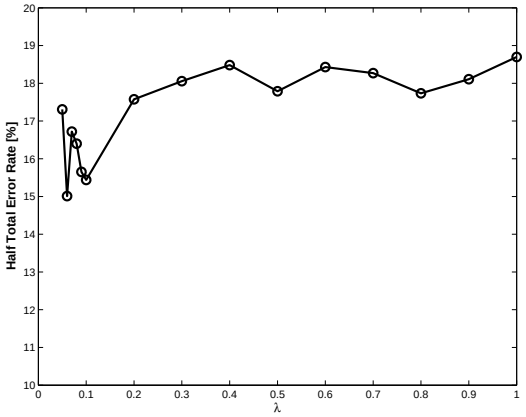



**Fig. 27: Weber Diffusion :** Parameter selection for the V-cycle multigrid procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.

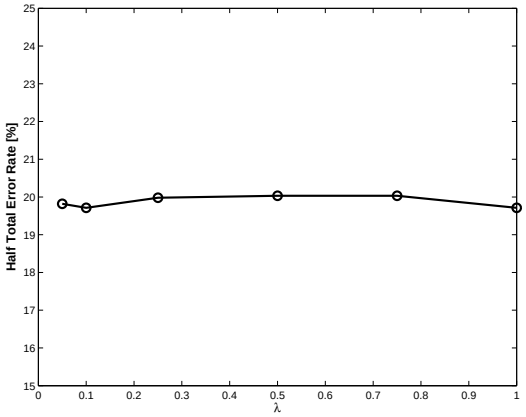



**Fig. 28: Michelson Diffusion :** Parameter selection for the relaxation procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.

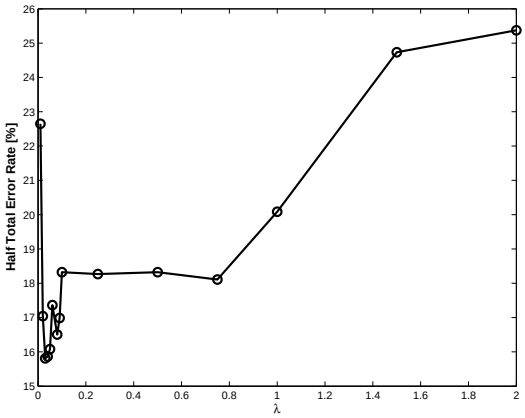



**Fig. 29: Michelson Diffusion :** Parameter selection for the V-cycle multigrid procedure : the figure shows the verification rate as a function of the smoothing parameter $\lambda$.