

Fingerprint representation and matching using redundant expansions[†]

Cristian Canton-Ferrer
Email: ccanton@gps.tsc.upc.es

1 Introduction

Among all types of images, fingerprints have been always a target for signal processing researchers due to the large existing libraries and the need to store and manage them efficiently. There are plenty of methods devoted to code this type of images ([25], [28]) and this report pretends to present a new approach to fingerprint image compression based on redundant expansions. Furthermore, an analysis technique based on convex optimization methods is also presented.

This report is structured in three sections: first, a mathematical introduction to redundant representations of data and Matching Pursuit algorithms (based on my MS Thesis [4]). Secondly, an image matching algorithm based on convex optimization over several variables is presented. It will be applied to classify simple patterns such as polygons and typographic letters. Finally, the prior algorithm is adapted to deal with redundant representations of fingerprint images. Unfortunately, due to the large scope of this work few points are discussed just theoretically without any practical result leaving an open door for future researchers.

In addition, we must say that this report is an application of the knowledge obtained at the Mathematical Methods for Communications (UPC PhD program subject) in combination with the techniques described in my MS Thesis [4].

1.1 Image Transform Coding

In the field of image coding¹, transform strategies have been widely used. That means not to code the information in the spatial domain (the raw image) but to transform it into another domain (i.e frequency,...) where the coding process can be performed better [24],[5]. Let us consider an image in its most elemental representation

$$\mathbf{I} = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} c(x, y) \delta(x, y), \quad (1)$$

where $c(x, y)$ is the intensity of the pixel (x, y) . If we want to store this image we need $N_x \cdot N_y$ bytes². There are plenty of strategies to reduce the amount of data necessary to

[†]Nevertheless, there would be many other titles for this document such as: **A bundle of ideas regarding convex optimization.**

¹In all the scope of this document, we are referring to black and white images.

²Let us consider a quantification of 256 levels of gray, that is 8 bits per pixel.

reconstruct the original image with a controlled degree of distortion and one of these is the transform coding.

Basically, the transform coding of information lies in the election of a linear transform that should achieve a property very suitable to compress the data: *sparsity* of the transformed coefficients. That is to concentrate the most of the energy in very few transformed coefficients in order to dismiss the ones under a treshold and be able to reconstruct the image just with these most powerful (allowing a controlled distortion). Among all the linear transforms, the one that achieves the optimum energy concentration is the Karhunen-Loïve Transform (KLT) that takes as basis functions the eigenvectors of the covariance matrix of the input signal. The main disadvantages of this transform are its dependence on the input signal statistics and unavailability to be separated into blocks [23]. In order to solve these drawbacks, its widely known the use of the Discrete Cosine Transform (DCT) used by the JPEG algorithm [25]. The characteristic of this transform is that for stationary image statistics, the energy concentration properties converge against those of the KLT for large block sizes.

Hence, there is a lower bound for the compression achieved by transform coding stated by the KLT. But we have to recall that transform coding is based on linear transformations and these ones regard on basis of vectors. Our goal is to introduce decompositions over sets of vectors that go beyond a basis; we define it as a "transform" but it lacks of the injection property because the decompositions of a vector over this set are infinite. Even though, we will keep the main idea of sparsity and introduce a theory of decomposition over a redundant set of vectors.

2 Mathematical background

2.1 Adaptive Greedy Approximations

For data compression applications and fast numerical methods it is important to accurately approximate functions from a Hilbert space \mathcal{H} using a small number of vectors from a given family $\{g_\gamma\}_{\gamma \in \Gamma}$. The standard problem in this regard is the problem of M -term approximation where one fixes a basis and looks to approximate a target function f by a linear combination of M terms of the basis. For any $M > 0$, we want to minimize the approximation error

$$\epsilon(M) = \|f - \tilde{f}\| = \left\| f - \sum_{\gamma \in I_M} c_\gamma g_\gamma \right\|, \quad (2)$$

where $I_M \in \Gamma$ is the subspace formed by the M vectors that approximate our function f , c_γ are the ponderation coefficients and $\|\cdot\|$ is a general norm.

When the basis is orthogonal (a wavelet basis for instance), then, this type of approximation is the starting point for compression algorithms. In this special case, when an orthogonal basis $\{\psi_k\}_{k=1}^N \in \mathcal{H}$ ($\dim \mathcal{H} = N$) is taken to perform our M -term decomposi-

tion, this decomposition will be unique

$$f = \sum_{k=1}^M c_k(f) \psi_k + \epsilon(M) \quad M < N, \quad (3)$$

where the coefficients $\{c_k(f)\}_{k=1}^M$ are the set of the Fourier coefficients of f , that is, the set of M vectors which have the largest inner products within $\langle f, \psi_k \rangle_{k=1}^N$. The problem of M -term approximations with regard to a basis has been studied thoroughly in [11],[10],[9].

One way to greatly improve these approximations consists in enlarging the collection $\{g_\gamma\}_{\gamma \in \Gamma}$ beyond a basis. This enlarged, redundant family of vectors will be called dictionary. To be more precise, we define a dictionary as a family $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ of vectors in a N -dimensional Hilbert space \mathcal{H} , where the cardinality of \mathcal{D} is P and $P > N$. All the vectors of \mathcal{D} accomplish that $\|g_\gamma\| = 1$ and the finite linear expansions of \mathcal{D} are dense in \mathcal{H} ($\overline{\text{span}}\mathcal{D} = \mathcal{H}$) [34]. For our purposes, the application of adaptive greedy approximations to image processing, we take $\mathcal{H} = \mathbf{L}^2(\mathbb{R}^2)$ ³ and we will call the vectors belonging to \mathcal{D} as *atoms*.

Under an overcomplete basis (dictionary) the decomposition of a signal is not unique and this redundancy can offer some advantages (and also few drawbacks). One is that there is greater flexibility in capturing structure in the data. For example, if a signal is largely sinusoidal, it will have a compact representation in a Fourier basis. Similarly, a signal composed of chirps is naturally represented in a chirp basis. Combining both of these bases into a single overcomplete basis would allow compact representations for both types of signals [6],[33],[12]. It is also possible to obtain compact representations when the overcomplete basis contains a single class of basis functions, for instance: an overcomplete Fourier basis, with more than the minimum number of sinusoids, can compactly represent signals composed of small number of frequencies.

When the dictionary is redundant, finding a family of M vectors that approximates f with an error close to the minimum is clearly not achieved by selecting the vectors that have maximal inner products with f [8]. It is proven in [8] that for general dictionaries the problem of finding an M -element optimal approximations belongs to a class of computationally intractable problems: the set of NP-hard problems. That means that there is no

³Notation:

The space $\mathbf{L}^2(\mathbb{R}^2)$ is the Hilbert space of complex valued functions such that

$$\|f\| = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |f(x, y)|^2 dx dy$$

The inner product of $(f, g) \in \mathbf{L}^2(\mathbb{R}^2)$ is defined by

$$\langle f, g \rangle = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \bar{g}(x, y) dx dy$$

And the norm is defined as

$$\|f\| = \langle f, f \rangle^{1/2}$$

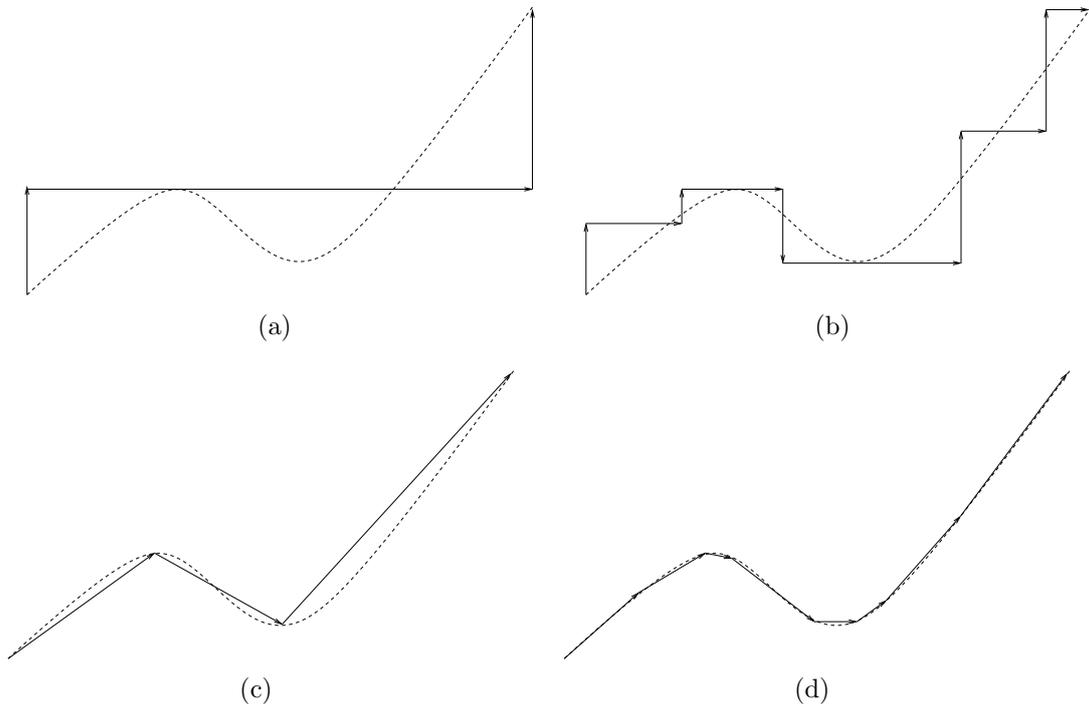


Figure 1: Example of representations. In the figures (a) and (b) there are the decompositions over an orthogonal basis of a curve with 3 and 10 terms respectively; in the figures (c) and (d) there are the decompositions of the same curve over an overcomplete dictionary with 3 and 8 terms. Using an overcomplete dictionary we capture the structure of the curve with much more or less terms than with a decomposition using an orthogonal basis.

known polynomial time algorithm that can compute the approximation \tilde{f} that minimizes $\|f - \tilde{f}\|$ [7],[21].

Because of the difficulty of computing optimal expansions, we turn to suboptimal algorithms: pursuit algorithms or adaptive greedy algorithms. These algorithms reduce the computational complexity by searching for efficient but non-optimal approximations. Within this family of algorithms we can enumerate Matching Pursuit (with its variants) and Basis Pursuit, among others [34],[12].

Under certain circumstances the approximation given by the Matching Pursuit algorithms can achieve *sparse* characteristics due to the fact that M (the number of terms to make this approximation) is much smaller than the dimension. The sparseness constraint refers to the requirement that to represent the approximation function \tilde{f} we must have as few c_k coefficients as possible [37]. Furthermore, it is proved that Matching Pursuit produces a (ϵ, M) -Sparse⁴ approximation with exponential decay of the error [34],[39]. For

⁴A (ϵ, M) -Sparse problem or a (ϵ, M) -Approximation refers to an approximation that achieves $\|f - \tilde{f}\| < \epsilon$ with M terms.

any M -term approximation obtained with Matching Pursuit we have

$$\|f - \tilde{f}_{\text{MP}}\| \leq \sqrt{1 + \frac{2\mu M^2}{(1 - 2\mu M)^2}} \|f - \tilde{f}_{\text{Opt}}\|, \quad (4)$$

being μ the coherence of the dictionary \mathcal{D} (see Appendix 1), $\|f - \tilde{f}_{\text{MP}}\|$ the approximation obtained by Matching Pursuit and $\|f - \tilde{f}_{\text{Opt}}\|$ the *optimal* approximation. That means that the error is bounded, hence the assumption of Matching Pursuit as a (ϵ, M) -Sparse problem is demonstrated.

2.2 Matching Pursuit Algorithm

2.2.1 Introduction

Matching Pursuit algorithm was introduced by Mallat and Zhang [33] giving examples for the application on unidimensional time-frequency signals (but it can be applied to any type of signal). This method produces a suboptimal function expansion by iteratively choosing the waveforms from a general dictionary (typically a rich collection of potential atoms in a Hilbert space) that best match the function's structures. The choice of the functions is performed through a progressive refinement of the signal approximation with an iterative procedure [8]. This method is closely related to the algorithms used in statistics [19].

The Matching Pursuit algorithms have already found applications in medicine [18] and image [2] and video coding [1], [32] (though in video coding it is usually used to code the motion estimation errors). Other flavors of Matching Pursuit can also be found in [33] and [34] like the Orthogonalised Matching Pursuit that is able to achieve a zero estimation error by orthogonalizing the directions of projection, with a Gram-Schmidt procedure proposed by [36]. The resulting orthogonal pursuit converges with a finite number of iterations, which is not the case for a non-orthogonal pursuit. The price to be paid is the important computational cost of the Gram-Schmidt orthogonalization, though this is not used due to practical reasons (fast algorithms to perform this Orthogonal Matching Pursuit have been already proposed in [22]).

In this section we want to show that Matching Pursuit is much more efficient to do an image approximation than the usual methods used nowadays in the standard formats (DCT for JPEG [25] and wavelets for JPEG2000 [26]), so it is possible to transmit an image at lower bit-rate [14]. Matching Pursuit, though results strongly depended on the choice of the dictionary(ies) used. In many applications, Gabor functions or symmetric dictionaries are used; we can greatly improve the results by using two or more dictionaries that catch more efficiently different features of the image (like edges or textures) as done in [4].

2.2.2 Formulation

Matching Pursuit is a greedy algorithm that decomposes any signal belonging to a Hilbert space \mathcal{H} into a linear expansion of waveforms that are selected from a redundant dictionary

(or set of dictionaries) \mathcal{D} of functions. These waveforms are iteratively chosen to best match the signal structures, producing a sub-optimal expansion. Vectors are selected one by one from the dictionary, while optimizing the signal approximation at each step k (this is the minimization $\|f - \tilde{f}\|_k$ with reference to $\|f - \tilde{f}\|_{k-1}$).

Let $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ be a dictionary of $P > N \times M$ vectors, with the properties cited above. This dictionary includes $N \times M$ linearly independent vectors that define a basis of the space $\mathbb{R}^{N \times M}$ of signals with size $N \times M$. The Matching Pursuit algorithm begins by projecting the target function f on a vector $g_{\gamma_0} \in \mathcal{D}$ and computing the residue Rf (see [33] and [34]):

$$f = \langle f, g_{\gamma_0} \rangle g_{\gamma_0} + Rf, \quad (5)$$

where Rf is the residual vector after approximating f in the direction of g_{γ_0} . Since we impose Rf to be orthogonal to g_{γ_0} :

$$\|f\|^2 = |\langle f, g_{\gamma_0} \rangle|^2 + \|Rf\|^2. \quad (6)$$

As we want to minimize $\|Rf\|^2 = \|f\|^2 - |\langle f, g_{\gamma_0} \rangle|^2$ we must choose $g_{\gamma_0} \in \mathcal{D}$ such that $|\langle f, g_{\gamma_0} \rangle|$ is maximum. In some cases, it is not computationally efficient to find the solution given by the Matching Pursuit algorithms, and a Matching Pursuit-suboptimal solution is computed instead:

$$|\langle f, g_{\gamma_0} \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle f, g_\gamma \rangle|, \quad (7)$$

where $\alpha \in (0, 1]$ is an optimality factor ($\alpha = 1$ means that we choose the optimal solution given by the Matching Pursuit method).

Into the next step, Matching Pursuit subdecomposes iteratively the residue Rf by projecting it on a vector of \mathcal{D} that matches Rf at best. If we consider $R^0 f = f$ and we suppose the n -th order residue $R^n f$ ($n \geq 0$) has been computed, the next iteration will choose $g_{\gamma_n} \in \mathcal{D}$ such that:

$$|\langle R^n f, g_{\gamma_n} \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle R^n f, g_\gamma \rangle|. \quad (8)$$

With this choice $R^n f$ is projected onto g_{γ_n} and decomposed as follows:

$$R^n f = \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n} + R^{n+1} f, \quad (9)$$

where $R^{n+1} f$ and g_{γ_n} are orthogonal, so the quadratic module of the previous equation is:

$$\|R^n f\|^2 = |\langle R^n f, g_{\gamma_n} \rangle|^2 + \|R^{n+1} f\|^2. \quad (10)$$

From Eq.9, we can see that the decomposition of f is given by:

$$f = \sum_{n=0}^{N-1} \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n} + R^N f, \quad (11)$$

and with the same principle we can also deduce from Eq.10 that the module of the signal f is:

$$\|f\|^2 = \sum_{n=0}^{N-1} |\langle R^n f, g_{\gamma_n} \rangle|^2 + \|R^N f\|^2, \quad (12)$$

where $\|R^N f\|$ converges exponentially to 0 when n tends to infinity⁵:

$$\lim_{n \rightarrow \infty} \|R^n f\| = 0. \quad (13)$$

Hence

$$f = \sum_{k=0}^{\infty} \langle R^k f, g_{\gamma_k} \rangle g_{\gamma_k}, \quad (14)$$

and

$$\|f\|^2 = \sum_{k=0}^{\infty} |\langle R^k f, g_{\gamma_k} \rangle|^2. \quad (15)$$

Only with Orthogonalised Matching Pursuit⁶ [13], [34], [8] the residue is reduced to 0 in a finite number of iterations but, in most signal processing applications, the fact of having a non-zero residual is not relevant, due to the fact that the image distortion is under the visible threshold.

Despite Matching Pursuit is a sub-optimal M -term approximation of a function and it avoids the NP -hard problem of finding the optimal approximation, the computational load is still very high. Optimizations to perform the Matching Pursuit via the FFT have been proposed in [4] and a short review is done in the Appendix 3.

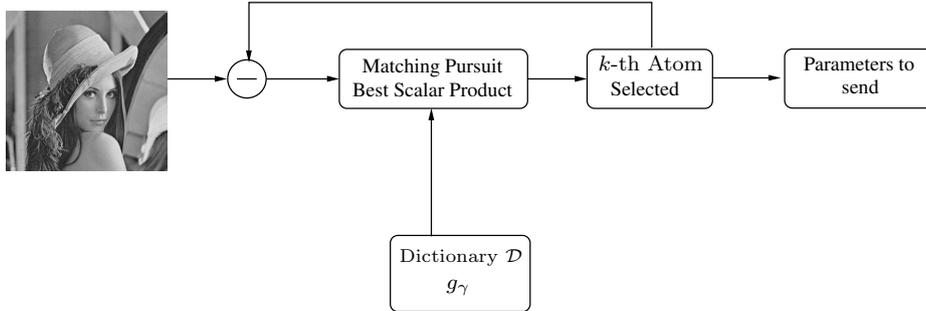


Figure 2: Matching Pursuit coding process.

⁵See Appendix 2 for a detailed demonstration

⁶It is also true that certain types of signals (for example, signals composed by a linear combination of atoms) can achieve zero error without this method.



Figure 3: Results obtained with by the Matching Pursuit when coding the original image Lenna, (a). Subfigure (b), (c) and (d) have 50, 100 and 300 atoms each one with and average of 0.3 bpp.

2.2.3 Properties of Matching Pursuit

Translation invariance

A dictionary \mathcal{D} is translation invariant if for any $g_\gamma[\vec{n}] \in \mathcal{D}$ and any $\vec{p} = [p_x, p_y] \in [0..N_x - 1, 0..N_y - 1]$ then $g_\gamma[\vec{n} - \vec{p}] \in \mathcal{D}$. If we compute the Matching Pursuit in a translation invariant dictionary, then it will be translation invariant. Given the matching of f in \mathcal{D} ,

$$f[\vec{n}] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}[\vec{n}] + R^M f[\vec{n}], \quad (16)$$

it is easy to verify [8] that the Matching Pursuit of $f_{\vec{p}}[\vec{n}] = f[\vec{n} - \vec{p}]$ selects a translation by \vec{p} of the same vectors g_{γ_m} with the same decomposition coefficients

$$f_{\vec{p}}[\vec{n}] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}[\vec{n} - \vec{p}] + R^M f_{\vec{p}}. \quad (17)$$

Rotation invariance

By analogy, we will obtain a rotation invariant Matching Pursuit when we use a rotation invariant dictionary \mathcal{D} . A dictionary is rotation invariant if for any $g_\gamma[\vec{n}] \in \mathcal{D}$ and any $\theta \in [0, 2\pi)$ then $g_\gamma[r_\theta \vec{n}] \in \mathcal{D}$ where r_θ is the rotation operator given by the matrix:

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (18)$$

Given the decomposition of f in \mathcal{D} ,

$$f[\vec{n}] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}[\vec{n}] + R^M f[\vec{n}], \quad (19)$$

one can verify that the Matching Pursuit of $f_\theta[\vec{n}] = f[r_\theta \cdot \vec{n}]$ selects a rotation by θ of the same vectors g_{γ_m} with the same decomposition coefficients:

$$f_\theta[\vec{n}] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}[r_\theta \cdot \vec{n}] + R^M f_\theta[\vec{n}] \quad (20)$$

This makes Matching Pursuit a useful technique to rotate images, because the only extra calculation we have to do is to modify the index of the reconstruction atoms when computing the coded image, instead of applying the rotation matrix to every pixel of the image.

Dilation invariance

As in the previous two cases, Matching Pursuit is dilation invariant if the dictionary of functions used by the pursuit is dilation invariant. A dictionary \mathcal{D} is dilation invariant when for any $g_\gamma[\vec{n}] \in \mathcal{D}$ and any $s \in [0, s_{\max}]$ then $g_\gamma[\frac{\vec{n}}{s}] \in \mathcal{D}$. In this case, the matching pursuit of $f_s[\vec{n}] = f[\frac{\vec{n}}{s}]$ will select a dilation by s of the same vectors g_{γ_m} with the same decomposition coefficients:

$$f_s[\vec{n}] = \sum_{m=0}^{M-1} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m} \left[\frac{\vec{n}}{s} \right] + R^M f_s[\vec{n}]. \quad (21)$$

One thing to take in account is that MP is dilation invariant only when the dilation is applied to the whole \vec{n} . If one applied a different scaling for every component of the vector, the property of dilation invariance will be lost.

Dilation invariance gives an easy way of scaling an image: the only thing that has to be done is to modify the scaling parameter (the same for x and y) when reconstructing the image and we will have a larger or smaller image.

Also if we join rotation, translation and dilation invariance, we get a good tool for pattern recognition because the coefficients do not depend on the position, the orientation of the size of the object. Certain coefficients and certain relation of the atom parameters would mean the presence of a concrete pattern in the analyzed image.

Energy conservation

When we have an infinite decomposition, the energy in the transformed domain and the energy in the space domain is the same, as we can deduce from Eq.12. As

$$\lim_{M \rightarrow \infty} R^M f = 0 \quad (22)$$

due to the exponential decreasing of the coefficients, when $M \rightarrow \infty$, Eq.12 turns to:

$$\|f\|^2 = \sum_{m=0}^{\infty} |\langle R^m f, g_{\gamma_m} \rangle|^2, \quad (23)$$

which mimics Parseval's equality for Fourier series.

Invertible

A complete Pursuit recovers a perfect version of the image:

$$f = \sum_{m=0}^{\infty} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}. \quad (24)$$

Thus the image f may be reconstructed from its Matching Pursuit coefficients, but if the decomposition is not complete we will not be able to get a perfect reconstruction, we will have a reconstruction error given by $R^M f$, where M is the number of coefficients used by the decomposition.

3 An Image Matching Algorithm

Prior to analyze such complex problem as a fingerprint matching algorithm we will try to formulate a generic image matching algorithm based on convex optimization. After a theoretical analysis we will perform some simulations to show the efficiency of our algorithm and its limitations. Due to the wide scope of this work, some results will be shown only theoretically and will be pending for a deeper study.

Let us have a library of images called \mathbf{I}_k with $0 \leq k \leq N - 1$ (for example a typographic set of the alphabet letters or a set of geometrical figures). Let us define $\tilde{\mathbf{I}}$ as an input image that we know, *a priori*, belongs to the set \mathbf{I}_k . This input image $\tilde{\mathbf{I}}$ presents an unknown spatial offset (x_0, y_0) , a scaling factor s_0 , a rotation angle θ_0 and even an additive noise. Hence, deciding which image from the set \mathbf{I}_k is being represented by $\tilde{\mathbf{I}}$ can be seen as an optimization problem based on the minimization of the following equation:

$$\min_{\underline{\mathfrak{N}}, k} \|\mathbf{I}_k - \tilde{\mathbf{I}}(\underline{\mathfrak{N}})\|_2 = \min_{\underline{\mathfrak{N}}, k} \xi(\underline{\mathfrak{N}}, k), \quad (25)$$

where $\underline{\mathfrak{N}} = (x_0, y_0, s_0, \theta_0)$ under the constraints:

$$0 \leq x_0 \leq N_x/2, \quad (26)$$

$$0 \leq y_0 \leq N_y/2, \quad (27)$$

$$-1 \leq s_0 \leq 1, \quad (28)$$

$$0 \leq \theta_0 \leq \pi, \quad (29)$$

$$0 \leq k \leq N - 1. \quad (30)$$

So, we want to estimate the values of the parameters that minimize the cost function for each element of the reference set \mathbf{I}_k and then select the element that presents the global minimum as depicted in the Figure 4.

There are several methods to recognize shapes or forms based, for instance, on neural networks [31] but the method presented here outperforms by its simplicity when dealing with simple shapes. Although, this method is limited to very simple and particular geometrical forms.

The first step to solve this problem is to demonstrate that it holds the convexity conditions. That is to show that

$$\left\| \mathbf{I}_k - \tilde{\mathbf{I}}\left(x - x_0, y - y_0, \theta - \theta_0, \frac{s}{2s_0}\right) \right\|_2, \quad (31)$$

is convex with the constraints cited above. The convexity of this equation strongly depends on the shape of the images \mathbf{I}_k and $\tilde{\mathbf{I}}$ and it must be studied for each case. Then, let us study a simple case: \mathbf{I}_k being the set of regular polygons with the same area \mathcal{A} (with k being the number of sides, i.e. $3 \leq k \leq 10$) centered at $(\frac{N_x}{2}, \frac{N_y}{2})$. Hence, we must include another constrain for this concrete problem:

$$\text{Area}_{n\text{-gon}} = \frac{1}{4}ns^2 \cot\left(\frac{\pi}{n}\right) = \mathcal{A}, \quad (32)$$

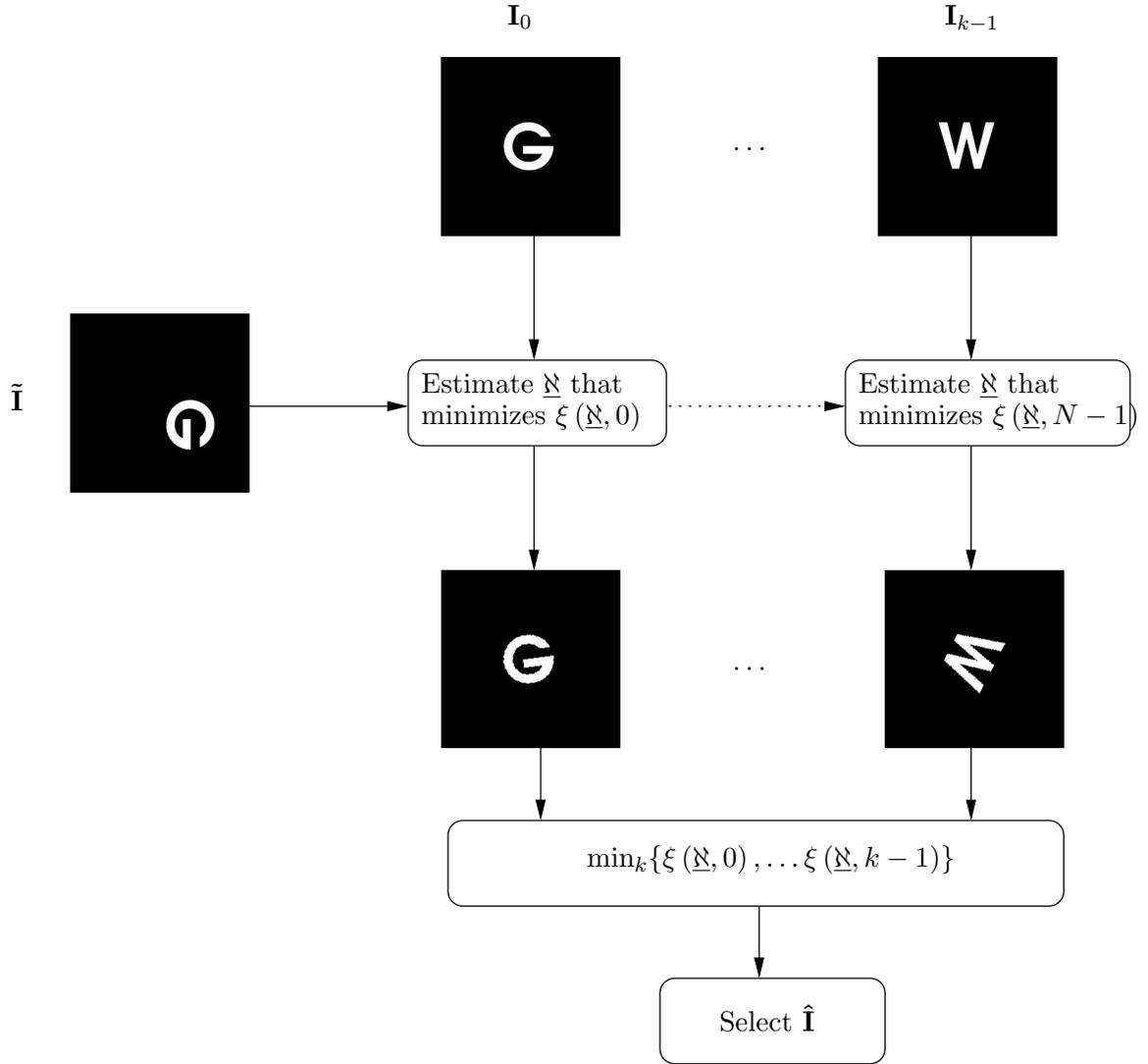


Figure 4: General scheme of an image matching algorithm based first on the search of the parameters that minimize the cost function between the input image $\tilde{\mathbf{I}}$ and the set of reference images \mathbf{I}_k . After, the decided image $\hat{\mathbf{I}}$ is that one that reaches the minimum cost function among all the partial cost functions. The aim of this report is to design a reliable algorithm to compute these minimizations.

with n being the number of sides and s the length of each side.

As noted before, the behavior of the cost function $\xi(\underline{\mathfrak{N}}, k)$ can not be studied analitically and must be analyzed though taking into account few premises. To show that the cost function is convex we should study the behavior of this function for each parameter and, only if it is convex in all the variables, the function will be convex or at list quasi-convex [38], [3].

- **Convexity on the offset** (x_0, y_0)

To perform a first analysis on the convexity of $\xi(x - x_0, y - y_0)$ let us take the assumption that the other parameters are set in the following conditions: $\theta_0 = 0$, $s_0 = 1$ and the *a priori* knowledge of which polygon is being represented by the input image ($k = z$). Hence, the cost function would be

$$\|\mathbf{I}_z - \tilde{\mathbf{I}}(x - x_0, y - y_0, 0, 0)\|_2. \quad (33)$$

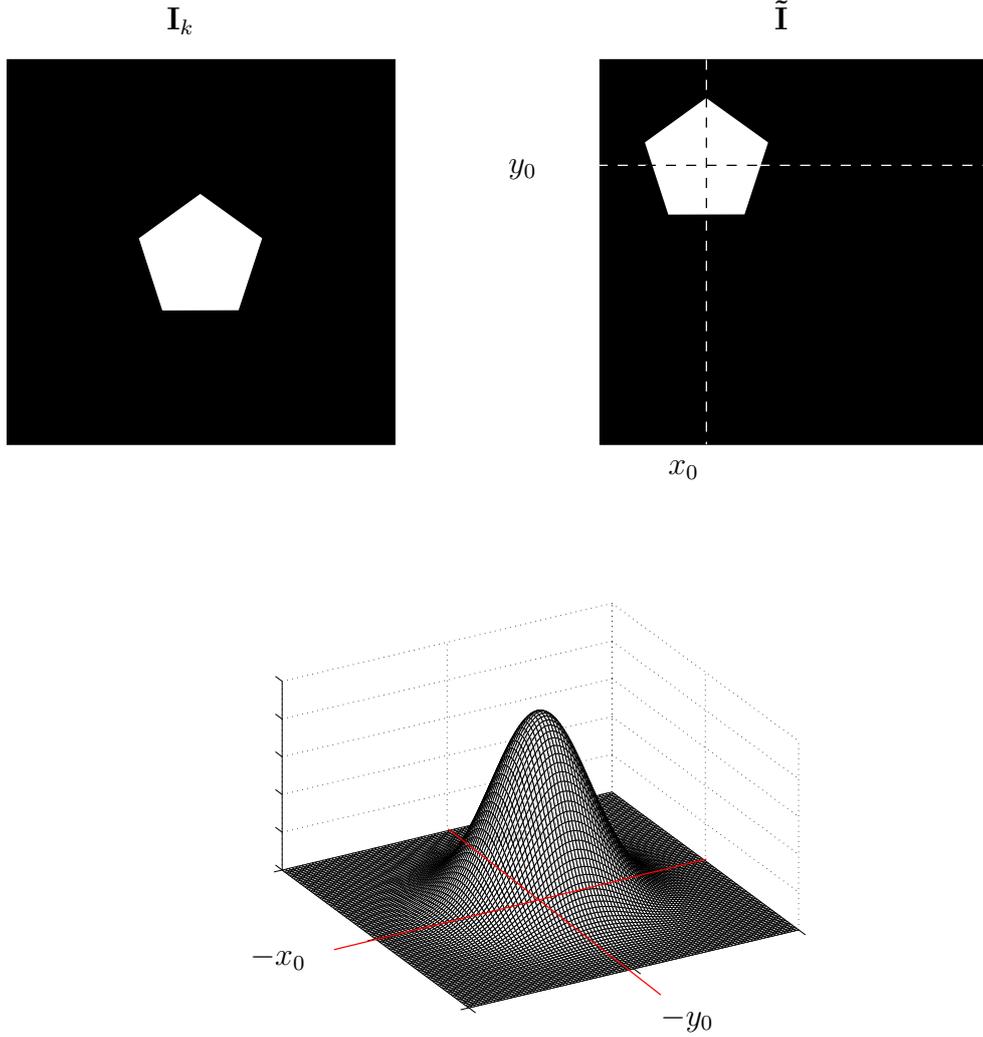


Figure 5: On the top, there are depicted the reference and the input image. On the bottom, the likelihood function between the two images in function of an offset applied to the input image. The maximum is reached at the point $(-x_0, -y_0)$ and therefore this is a minimum of the cost function.

As depicted in Figure 5, the input image resembles more the reference image when displaced to the point $(-x_0, -y_0)$. Hence, the cost function is convex in relation with

the offset parameter when we know which polygon is being represented and there is neither rotation nor scaling.

To perform a more accurate analysis, let us imagine that the represented polygon is different from the reference polygon (i.e. the input image represents a pentagon and the reference image is a triangle). As all the polygons have the same area (hence none can be completely overlaped by another), the convexity of the cost function depending on k is granted and, moreover, there is just one minimum per each k . Moreover, the estimation of the offset (x_0, y_0) is valid even if the compared polygons are different.

Even in the case when there is a rotation or a scaling on the input image, the cost function is still convex and presents a minimum. Finally, we can conclude that the cost function is convex independently from the rotation, scaling and the reference image.

- **Convexity on the rotation θ_0**

Let us consider a rotation angle θ_0 on the input image. If we study the Figure 6 we will notice that the similarity function (that is the complement of the cost function), is non-convex. But this is not a drawback because, despite there are several maximums, all them are valid points due to the simmetry of the polygons. Hence, in this case, maybe we should talk about local convexity instead of global convexity.

In the case of rotation, there is the same consideration as in the offset: for different polygons, the estimation of the angle is still valid because when the polygons are aligned the overlapping reaches its maximum.

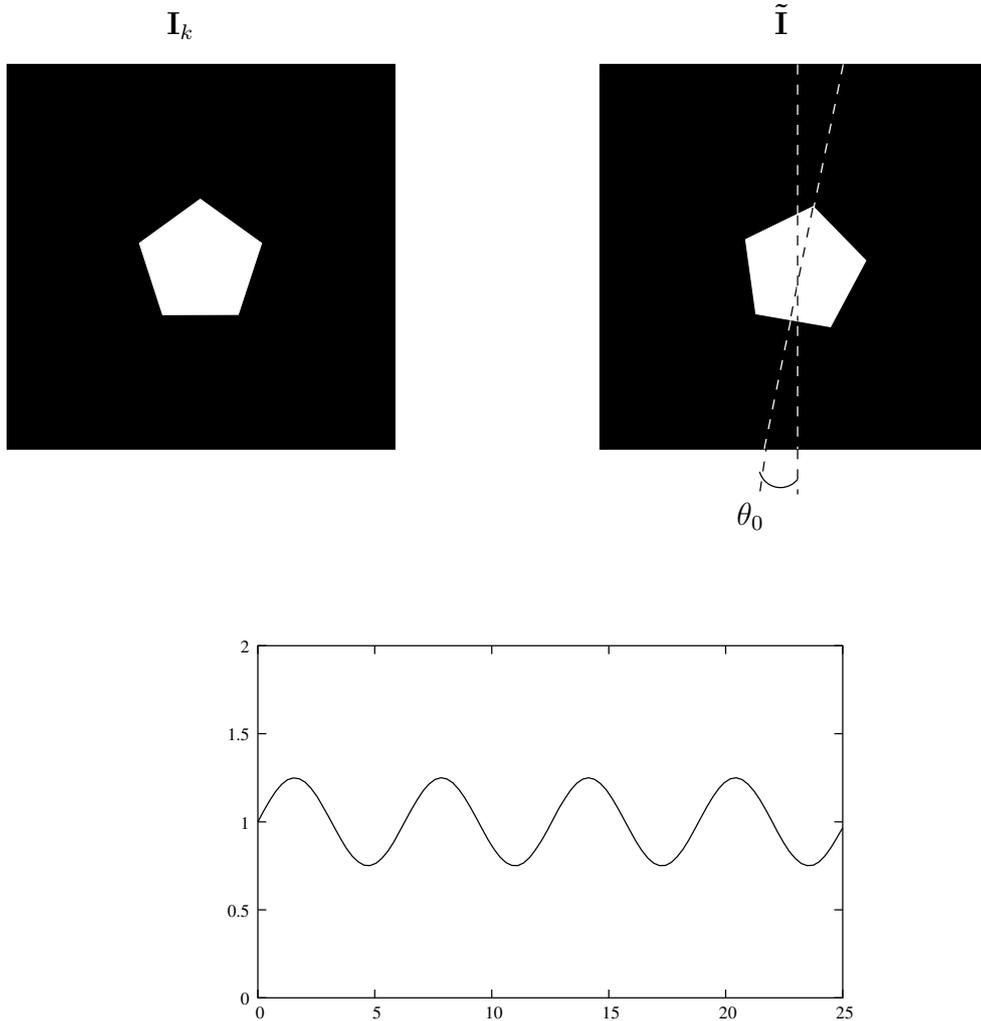


Figure 6: On the top, there are depicted the reference and the input image. On the bottom, the likelihood function between the two images in function of a rotation angle applied to the input image. There are several maximums due to the simmetry of the polygons, hence the interest interval should be bounded between $[0, \frac{2\pi}{k})$.

- **Convexity on the scaling s_0**

As we stated before: $-1 \leq s_0 \leq 1$ that means that the input image can be enlarged or shrunk. As depicted in Figure 7, the cost function ξ presents a minimum when the two polygons are correctly scaled. As in the previous cases, the cost function is convex independently from the other variables.

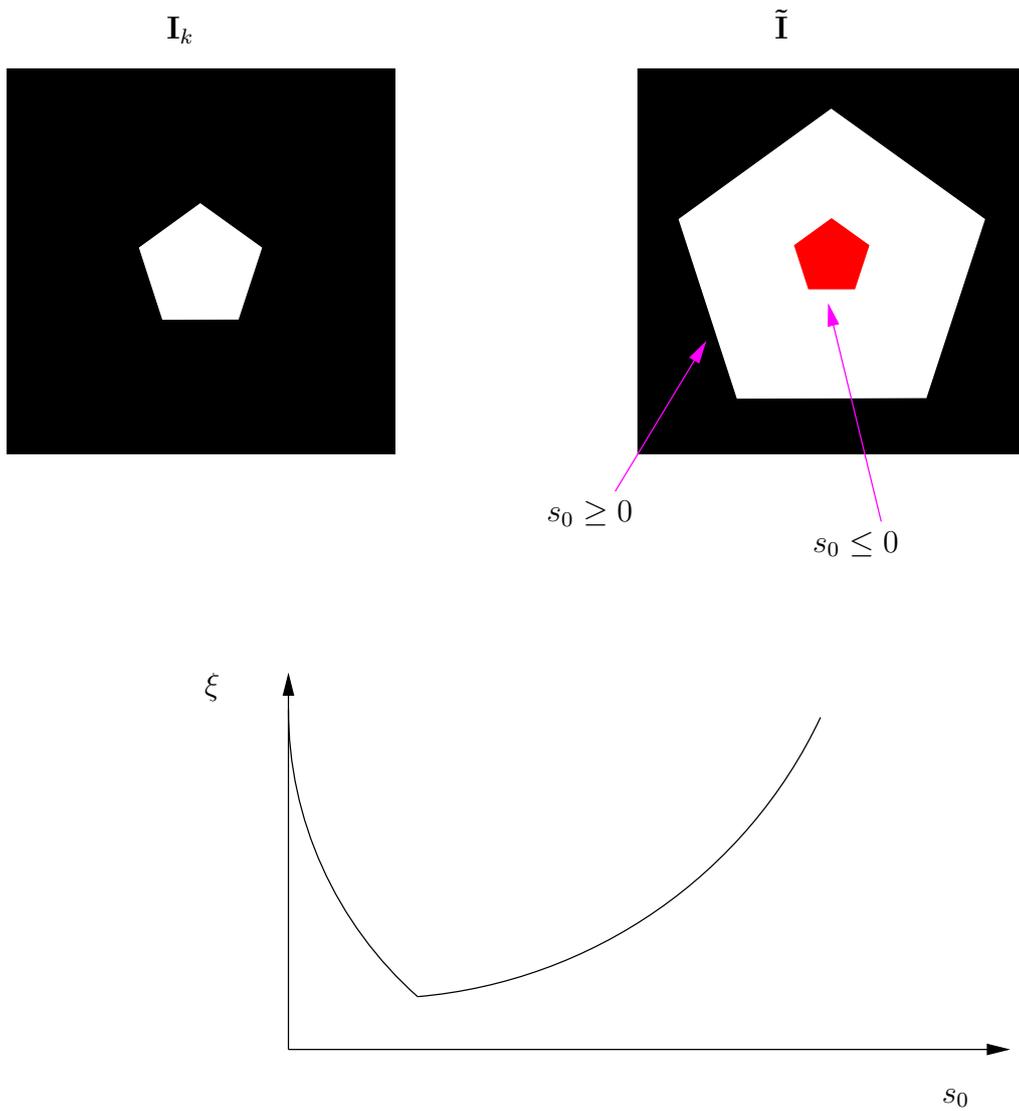


Figure 7: The cost function reaches a minimum when the input image and the reference image are fully overlapped. Even if the the reference image is not the same polygon, or it is rotated or has an offset, the cost function is still convex ensuring the feasibility of the problem.

Once the convexity of $\xi(x_0, y_0, \theta_0, s_0)$ has been proved, we can define a method to reach this minimum. Among all the existing adaptive methods to reach this minimum we focus on two: the gradient descent method and the Differential Steepest Descent (DSD) method. There are other adaptive algorithms such as de RLS or LMS methods but, despite they are widely used into communications signal processing, its application to image processing is rather complex in a direct form [30].

The search of the minimum of $\xi(\underline{\mathbf{x}})$ based on a gradient descent algorithm basically updates the set of parameters $\underline{\mathbf{x}}$ in order to decrease the value of the cost function. To perform this operation the parameters are updated by the formula

$$\underline{\mathbf{x}}_{n+1} = \underline{\mathbf{x}}_n - \mu \nabla \xi(\underline{\mathbf{x}}_n). \quad (34)$$

Also, a graphic representation can be seen in Figure 8 and more technical references can be found in [29]

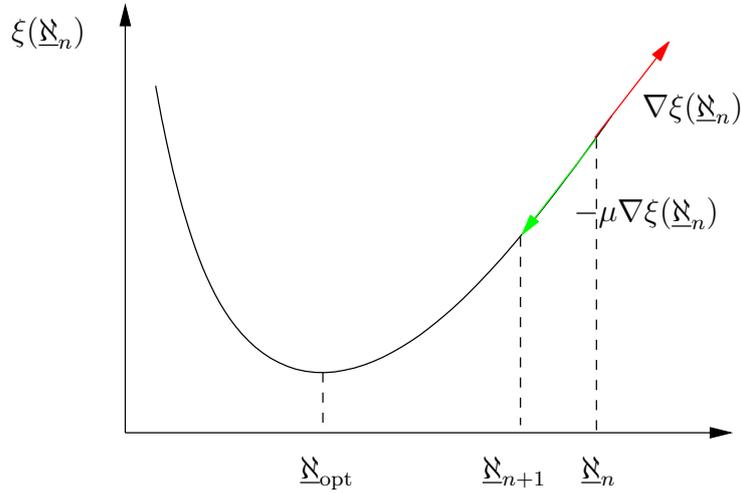


Figure 8: Gradient descent scheme.

The parameters we must take in account to fully describe our algorithm will be the estimation of the gradient and the value of the step-size μ . The gradient for each variable, according to [29] would be estimated as

$$\nabla \xi(\underline{\mathbf{x}}_n) = \frac{\xi(\underline{\mathbf{x}}_n + \delta) - \xi(\underline{\mathbf{x}}_n - \delta)}{2\delta} \quad (35)$$

with δ a small increment (with different values depending on the variable we are calculating the gradient).

The performance of this method is shown in the Figure 9 for various values of the parameter μ . The reader can see that the convergence and the final error of the algorithm depends on the μ value. To choose the optimal value, there exists methods like the *backtracking line search* algorithm. For our simulations we estimate this value to be $\mu = 10$.

- **Advantages**

- This basic algorithm to recognize regular polygons based on convex optimization gives accurate results into the classification. One of the direct applications would be to mimic a OCR⁷ system when dealing with information coded with a system based on this type of polygons. Also, it performs very well (10% error) when classifying shapes drawn by hand more or less accurately. Despite its a simple application its main advantage is the low error ratio when performing a classification.
- It has been proved its performance for convex regular polygons but it also copes with concave regular polygons (i.e. stars) due its nice simmetry properties. Moreover, we expect to have the same performance ratio when dealing with non-regular convex polygons (to be proved).

- **Disadvantages**

- We could not state a method to deal with concave non-regular polygons. As a future research, it might be possible to split a concave non-regular polygon into a set of convex polygons and try to perform a convex optimization (taking into account few constraints on the relative position of this sub-polygons). Perhaps to model this relative position into the minimization process, graph theory could be very useful.
- The use of DSD algorithm does not perform better in comparison with the gradient descent one. At list, in the best of the cases, they perform the same.
- Registration algorithms based on the maximization of mutual information have been proved to be a more efficient when trying to solve problem like this. Also, those algorithms are based on convex optimization of functions.

⁷Optical Character Recognition.

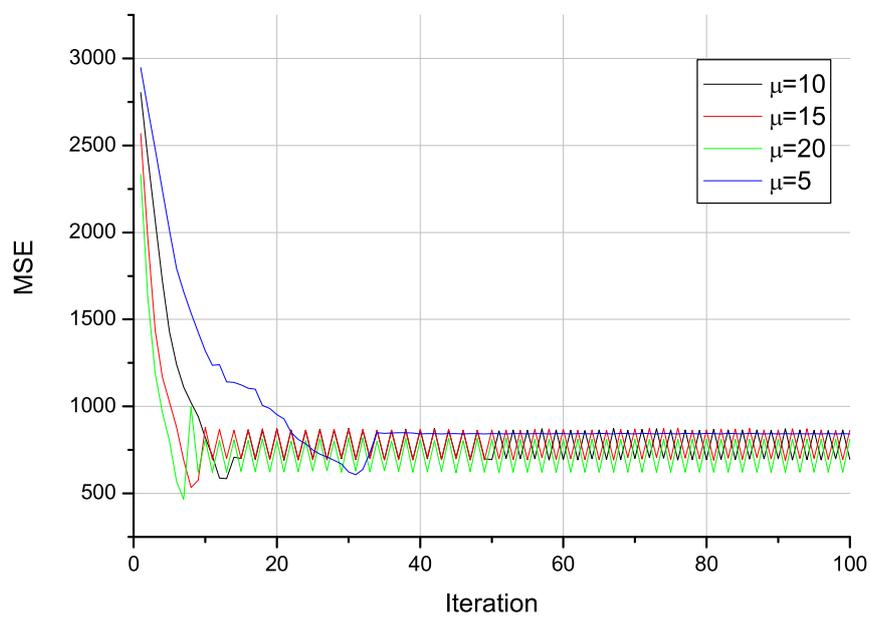


Figure 9: Dependence of the MSE depending on the number of iterations and the step-size μ for the gradient descent method. As usual in these cases, the larger the μ the fastest the decay but the greater the error.

4 Fingerprint representation[†]

Our first goal will be to code fingerprint images by the Matching Pursuit method described before.

4.1 Dictionary design

Fingerprint are highly textured images and this feature can give advantages in order to define the dictionary used by Matching Pursuit. When coding natural images [14], the use of atoms based on anisotropic refined functions have shown a good performance but it has been demonstrated though in [4] that atoms based on Gabor functions lead to better results in case of images with patterns or textures. Hence, in the case of fingerprints the election of Gabor atoms is justified but the parameters that defines the dictionary must be chosen accordingly to the statistics of the signal. In fact, the most relevant parameter is the frequency and orientation of the Gabor atoms. If we take a look to the spectrum average of a fingerprint (obtained by averaging 150 fingerprint images) we get Figure 10. With this information we can define a tailor-made dictionary for fingerprint images setting the range of frequencies in the interval $[\omega_{\min}, \omega_{\max}]$.

4.2 Results

The use of redundant approximations for such type of images leads to good results achieving high compression ratios [14],[4]. We will not enter in details due to the extense existing bibliography and let the Figures 11 and 12 show an example of its performance.

[†]The techniques described here are the basis of an article that is being written to be submitted to a conference.

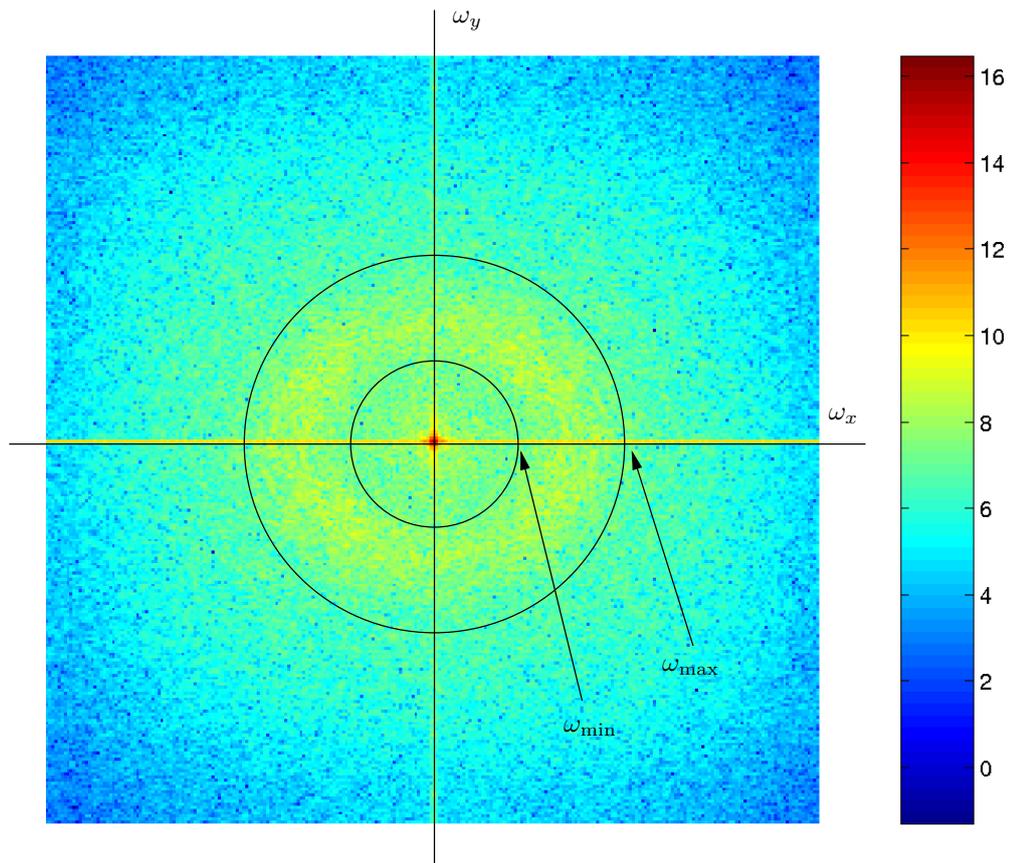


Figure 10: Average spectrum of a fingerprint set of images. The circular corone defined by $[\omega_{\min}, \omega_{\max}]$ contains all the relevant information of a fingerprint image (the stripes). For our coding and analysis purposes, this area will be the our coding target. The diccionary used to code the fingerprints will be taylored according to this extreme frequencies in order to reduce the computational load.



Figure 11: Example of a fingerprint image coded by the Matching Pursuit algorithm. In (a), the original image and in (b), (c) and (d) the coded image with 100, 200 and 300 atoms. As the reader can see, with few atoms we can reproduce the foremost regions of the image. Moreover, the difference between (c) and (d) is very small due to the sparsity of the coding process: in the few first coefficients is concentrated the most of the information.

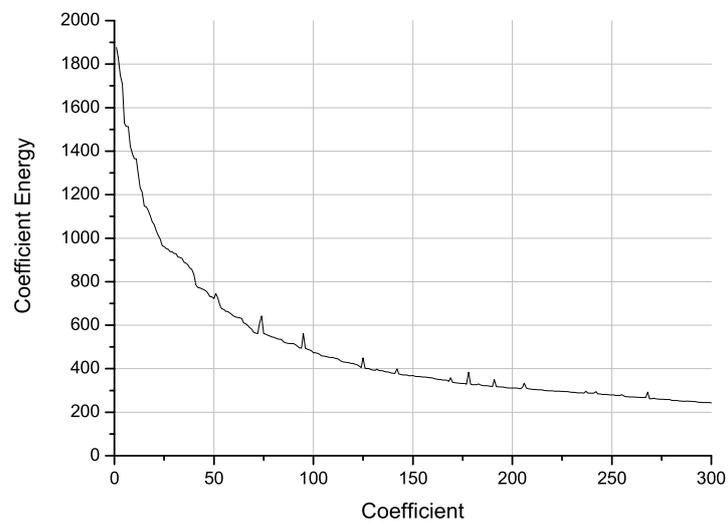


Figure 12: Decay of the energy of the Matching Pursuit coefficients. The most of the energy is regarded by the few first ones, allowing sparse representations. The decay of this coefficients always follows an exponential rule that have been explored to design an adaptive quantizer [16] leading to very high compression ratios.

5 Convex Optimization Fingerprint Matching

There are many methods to perform an accurate fingerprint recognition based on very diverse techniques [35]. Taking advantage of our redundant representation of the fingerprint we will try to formulate two methods to perform a matching of the input image over a library of fingerprints.

Despite, this redundant representation is very useful for its sparseness and its capability to catch the most of the information necessary to perform a matching in few coefficients, the viability of the methods exposed here is not yet fully proved. Few theoretical methods are shown here but the practical results have been impossible to carry out due to the lack of time. Hence, this is a collection of nice ideas about convex optimization, fingerprint representation that would be used for a future research.

5.1 Notation

First of all, let us review few aspects on the notation to solve this problem:

- \mathbf{I}_k where $0 \leq k \leq N - 1$: The set of N images containing the reference fingerprints. We assume that they do not present neither rotation, deformation nor any unpleasant artifact.
- $\tilde{\mathbf{I}}$: The input image we want to classify and identify its origin, that is from which reference image \mathbf{I}_k they present a maximum likelihood.
- $\Gamma_{(k,p)} = (c_p, x_p, y_p, s_p, \theta_p, f_p)_{(k)}$ where $0 \leq k \leq N - 1$ and $0 \leq p \leq M - 1$: The set of M vectors that define the Gabor atoms obtained by the Matching Pursuit decomposition for each of the N images of the reference set \mathbf{I}_k . The components of each vector are the necessary fully to define each atom: *normalized* scalar product c , position (x, y) , scaling factor s , rotation angle θ and frequency f .
- $\underline{\beta} = (\varphi, x_0, y_0, s_0)$: The factors that can distort the input image $\tilde{\mathbf{I}}$. Mainly, they are: a rotation angle from the original position φ , a spatial offset (x_0, y_0) or an uniform scaling over the fingerprint s_0 as depicted in the Figure 13. Although this last assumption about the scaling is not completely true because a scaling or a deformation could be applied only locally (our fingerprint does not deform globally) it will deal with our objectives.

There would be at list two ways to carry out the fingerprint match using an adaptive algorithm based on convex optimization. Here we present them theoretically commenting their strong and weak features.

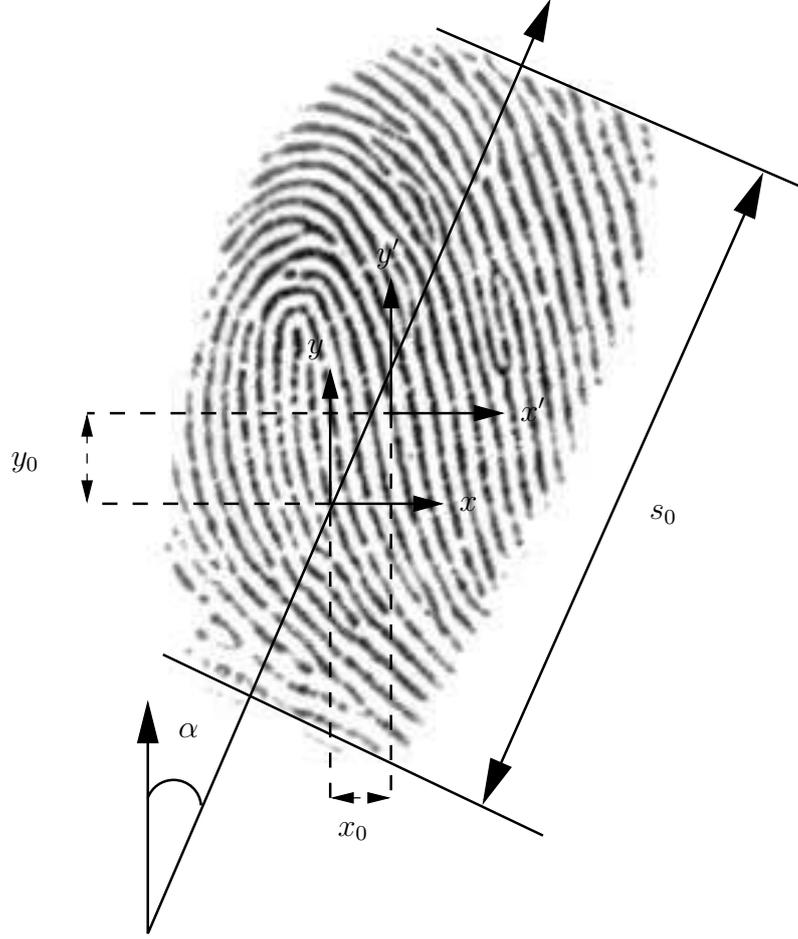


Figure 13: Illustration of the parameters that would distort a fingerprint in comparison with its reference: the spatial offset (x_0, y_0) , the scaling factor s_0 and the rotation angle α .

5.2 Method 1

A first method would be defined in the same way as a gradient descent algorithm:

1. Estimate the a set of initial parameters $\underline{\beta}_0$.
2. Apply the parameters $\underline{\beta}_n$ to the set $\Gamma_{(k,p)}$ as to obtain the set of atoms for the reconstruction $\hat{\Gamma}_{(k,p)}$:

$$\hat{\Gamma}_{(k,p)} = \left(c_p, (x_p - x_0) \cdot r_\alpha, (y_p - y_0) \cdot r_\alpha, \frac{s_p}{s_0}, \theta_p + \alpha, f_p \right)_{(k)}, \quad (36)$$

where r_α is the rotation operator.

3. Reconstruct the image using the atoms $\hat{\Gamma}_{(k,p)}$.
4. Compute the cost function $\xi(\cdot)$ (basically, the MSE between the reconstructed image and the reference but it could be any other convex measure).
5. Update $\underline{\beta}_{n+1} = g(\underline{\beta}_n)$ ($g(\cdot)$ is usually defined as a small increment of each parameter towards the inverse direction of the gradient of its parameter).
6. Check if the stop condition is fulfilled and end the process or return to the step 2 in the other case.

5.3 Method 2

1. Generate a set of *feature maps*:

$$\text{Frequency and Orientation Map } \mathcal{M}_f^{\tilde{I}}(x, y) = \sum_{p=0}^{M-1} A_p^f(x, y), \quad (37)$$

$$\text{Normalized Energy Map } \mathcal{M}_e^{\tilde{I}}(x, y) = \sum_{p=0}^{M-1} A_p^e(x, y), \quad (38)$$

with

$$A_p^f = \begin{cases} f_p \cdot (\cos \theta_p + j \sin \theta_p) & \text{if } \sqrt{x^2 + y^2} \leq s_p \\ 0 & \text{otherwise} \end{cases}, \quad (39)$$

and

$$A_p^e = \begin{cases} \frac{c_p}{c_{\max}} & \text{if } \sqrt{x^2 + y^2} \leq s_p \\ 0 & \text{otherwise} \end{cases}. \quad (40)$$

The complex valued map $\mathcal{M}_f(x, y)$ collects the information regarding with the distribution of the frequencies in the image. For every atom there is assigned a disk of radius s_p with a complex valued height describing the horizontal and vertical components of the frequency. It is proved that the most of the information of a fingerprint is regarded by the stripes represented by the frequencies of the atoms. Hence, by allocating the frequencies and orientations represented in each region of the image we can define a technique to compare two fingerprints and its coherence.

In the other hand, the real valued map $\mathcal{M}_e(x, y)$ models the intensity of the stripes within the finger. In this case, each atom has assigned a disk with a height proportional to its relevance.

The minimization problem can be seen thus as a vectorial minimization:

$$\min_{\underline{\beta}, k} \left\| \left(\mathcal{M}_f^{\mathbf{I}_k} \left(\frac{x-x_0}{s_0}, \frac{y-y_0}{s_0} \right) \cdot r_\alpha, \mathcal{M}_e^{\mathbf{I}_k} \left(\frac{x-x_0}{s_0}, \frac{y-y_0}{s_0} \right) \cdot r_\alpha \right) - \left(\mathcal{M}_f^{\tilde{\mathbf{I}}} \left(\frac{x-x_0}{s_0}, \frac{y-y_0}{s_0} \right) \cdot r_\alpha, \mathcal{M}_e^{\tilde{\mathbf{I}}} \left(\frac{x-x_0}{s_0}, \frac{y-y_0}{s_0} \right) \cdot r_\alpha \right) \right\|_{\xi(\cdot)} \quad (41)$$

2. Define a first estimation of the parameters $\underline{\beta}_0$.
3. Apply the parameters $\underline{\beta}_0$ to each map. By using these feature maps we can perform the scaling, rotation and offset addition directly to the A_p^f and A_p^e functions. In this way, the algorithm is computationally more efficient than the first one. These operations are defined by:

$$A_p^f = \begin{cases} f_p \cdot (\cos(\theta_p + \alpha) + j \sin(\theta_p + \alpha)) & \text{if } \sqrt{(x-x_0)^2 + (y-y_0)^2} \leq \frac{s_p}{s_0} \\ 0 & \text{otherwise} \end{cases}, \quad (42)$$

and

$$A_p^e = \begin{cases} \frac{c_p}{c_{\max}} & \text{if } \sqrt{(x-x_0)^2 + (y-y_0)^2} \leq \frac{s_p}{s_0} \\ 0 & \text{otherwise} \end{cases}. \quad (43)$$

4. Compute the cost function $\xi(\cdot)$ between the estimated feature maps and the reference feature maps (the fingerprints database). Here we find a point not solved yet. The function $\xi(\cdot)$ can not be defined as the MSE function because it is non-convex. Hence, we should try to define a measure that would be convex over the space we are working on. Unfortunately, this is a topic we are still working on and our algorithm is staked here.
5. Update the coefficients and return to the point 2 until a stop criteria is reached.

5.4 Discussion

Two methods to match fingerprint images have been proposed in this report. When we tried to study a matching algorithm based on convex optimization we had to face the problem of non-convexity and even the suitability of this methods for this problem. Nevertheless, the use of convex optimization seems to be a feasible approximation to the solution but it must be studied more thoroughly.

Despite the solution of this problem has not been found, it has given some light to some other aspects of this problem, for example, the method 2 is a new approach to this problem and the author will study the existence of a norm $\xi(\cdot)$ convex over this space of functions. Indeed, the use of convex theory is very enrichful and promising in this field!

Appendix 1: Dictionary coherence

Matching Pursuit greedy approximation of functions depends exclusively on the election on the dictionary/ies the decomposition will be done over. Hence, a careful study over the design and performance of the dictionary/ies must be done. Many parameters are involved in the design of a dictionary: shape of the atoms, orientation, scales, frequency (i.e. the case of Gabor atoms),... The values those parameters take will define the size, properties and performance of the dictionary. For instance, a careful design of the dictionary would lead us to reduce its size by eliminating atoms that would never be used (for example, analyzing the spectrum of the input images as done here). Furthermore, the redundancy of the dictionary can be controlled.

Once the dictionaries are already chosen, a quality parameter might be defined to evaluate how good our choice is. The most fundamental quality parameter associated with a dictionary is the *coherence* μ [39], defined as

$$\mu = \max_{j \neq k} |\langle g_{\gamma_j}, g_{\gamma_k} \rangle| \quad g_{\gamma_k}, g_{\gamma_j} \in \mathcal{D}. \quad (44)$$

Roughly speaking, this number measures how much two vectors in the dictionary look alike. This parameter is not a definitive way to evaluate the performance of the dictionary since it only reflects the most extreme correlations in the dictionary. Nevertheless, it is easy to calculate, and it captures well the behavior of uniform dictionaries. This is our case of a dictionary formed by a set of Gabor atoms. In other cases, where the structure of the dictionary does not present this property, more effective techniques are presented as *the Babel function* [39].

Appendix 2: Exponential convergence to 0 of the residual in MP

Let be $R^n f$ the approximation error of f after choosing n vectors in the dictionary and the energy of this error is given by

$$\|R^n f\|^2 = \|f\|^2 - \sum_{k=0}^{n-1} |\langle R^k f, g_{\gamma_k} \rangle|^2. \quad (45)$$

Let be \mathcal{H} a Hilbert Space, then, for any $f \in \mathcal{H}$, the convergente of the error to zero is shown in [40] to be a consequence of a theorem proved by [27]. Here is a detailed demonstration of the exponential convergence to 0 of the residual in MP.

Theorem

There exists $\lambda > 0$ such that for all $m \geq 0$ and $\forall f \in \mathbb{C}^N$:

$$\|R^m f\| \leq 2^{-\lambda m} \|f\|. \quad (46)$$

As a consequence

$$f = \sum_{m=0}^{+\infty} \langle R^m f, g_{\gamma_m} \rangle g_{\gamma_m}, \quad (47)$$

and

$$\|f\|^2 = \sum_{m=0}^{+\infty} |\langle R^m f, g_{\gamma_m} \rangle|^2, \quad (48)$$

where the convergence of 2 is intended in the strong sense.

Proof

Let us verify that exists $\beta > 0$ such that for any $f \in \mathbb{C}^N$

$$\sup_{\gamma \in \Gamma} |\langle f_m, g_\gamma \rangle| \geq \beta \|f\|. \quad (49)$$

Suppose that it is not possible to find such a β . This means that we can construct $\{f_m\}_{m \in \mathbb{N}}$ with $\|f_m\| = 1$ and

$$\lim_{m \rightarrow \infty} \sup_{\gamma \in \Gamma} |\langle f_m, g_\gamma \rangle| = 0. \quad (50)$$

Since the the unit sphere \mathbb{C}^N is compact, there exists a subspace $\{f_{m_k}\}_{k \in \mathbb{N}}$ that converges to a unit vector $f \in \mathbb{C}^N$. It follows that

$$\sup_{\gamma \in \Gamma} |\langle f, g_\gamma \rangle| = 0, \quad (51)$$

so $\langle f, g_\gamma \rangle = 0$ for all $g_\gamma \in \mathcal{D}$. Since \mathcal{D} contains a basis of \mathbb{C}^N , necessarily $f = 0$ which is not possible because $\|f\| = 1$. This proves that our initial assumption is wrong and, hence, there exists β such that Eq.49 holds.

The decay condition Eq.46 is derived from the energy conservation:

$$\|R^{m+1}f\|^2 = \|R^m f\|^2 - |\langle R^m f, g_{\gamma_m} \rangle|. \quad (52)$$

The Matching Pursuit chooses g_{γ_m} that satisfies

$$\langle R^m f, g_{\gamma_m} \rangle \geq \alpha \sup_{\gamma \in \Gamma} |\langle R^m f, g_\gamma \rangle|, \quad (53)$$

and Eq.49 implies that $\langle R^m f, g_{\gamma_m} \rangle \geq \alpha\beta\|R^m f\|$. So

$$\|R^{m+1}\| \leq \|R^m f\| \sqrt{1 - \alpha^2\beta^2}, \quad (54)$$

which verifies Eq.46 for

$$2^{-\lambda} = \sqrt{1 - \alpha^2\beta^2} < 1. \quad (55)$$

This also proves that

$$\lim_{m \rightarrow \infty} \|R^m f\| = 0. \quad (56)$$

Appendix 3: Full search Matching Pursuit via FFT

Matching Pursuit is a greedy algorithm that decomposes a signal over a redundant set of functions, the dictionary. As we have seen in the former section, the algorithm computes for each iteration all the scalar products $\langle R^k f, g_{\gamma_k} \rangle$ (in the k -th iteration) and then chooses the one with the largest absolute value. If we analyze in detail the computational load this calculations generate, for example for a dictionary formed by anisotropic refined atoms [14] we will find that it is huge. For a square image of size $N_x \times N_x$, the number of scalar products between images to be done in the computation of one decomposition term is

$$N_x^2 \times \frac{1}{2} (3 \times (\log_2(N_x) - 2) + 1)^2 \times 18 \sim O(N_x^2 \times \log_2^2(N_x)), \quad (57)$$

equivalent to have this amount of operations per coefficient

$$O(N_x^4 \times \log_2^2(N_x)). \quad (58)$$

Then, optimizations of this algorithms are required in order to reduce the computational load. But here arises another problem: when performing Matching Pursuit, it can be chosen between generating the atoms at each iteration to compute the scalar products or storing them in memory in order to save computational time. The option to generate and store the atoms in the memory is the optimal one in terms of speed but on the other hand there is the problem of memory capacity. Hence, there is a compromise between memory and speed. Moreover, to perform the Matching Pursuit decomposition without any optimization implies to store in memory all the functions of the dictionary (a prohibitive amount of memory indeed, for further details on the size of the dictionaries see [4]). In the pursue of optimizations, suboptimal approximations of Matching Pursuit have been proposed by using genetic algorithms ([14], [15]) that reduce the computational load but the drawback of this type of techniques are the non repeatability of the process, that is: if you apply twice the algorithm over the same image you obtain different results.

An optimization, proposed by [17], is to use the properties of the Discrete Fourier Transform to reduce the computational load (using also the Fast Fourier Transform) and reduce though the memory required to store the dictionary. This optimization is based on the property of the duality product-convolution of the DFT already introduced in the first chapter.

Here we do a detailed explanation. Let \mathcal{D} be a dictionary defined by a set of parameters $\gamma = (\aleph, \mathbf{p})$ where \aleph are the set of parameters concerning the shape of the atom (orientation, scaling, ...) and $\mathbf{p} = (p_x, p_y)$ the point into the image where the atom will be centered. Let us define $\mathcal{V} \in \mathcal{D}$ the sub-dictionary generated by $\gamma = (\aleph, \mathbf{0})$, the set of atoms centered in the middle of the image⁸. Strictly applying the Matching Pursuit algorithm, to find the most powerful atom at the n -th iteration of the process we should compute all the scalar products $|\langle R^n f, g_{\gamma_n} \rangle|$ with $g_{\gamma_n} \in \mathcal{D}$ and then choose the largest one. The search of the

⁸For this expansion, let us consider an image bounded into $[-\frac{N_x}{2}, \frac{N_x}{2}] \times [-\frac{N_y}{2}, \frac{N_y}{2}]$.

most powerful can be rewritten as the search of $z_{\gamma_n}(x, y) \in \mathcal{V}$ that maximizes

$$\max_{\gamma_n} |\langle R^n f, z_{\gamma_n}(x - p_x, y - p_y) \rangle|, \quad (59)$$

with $p_x \in [-\frac{N_x}{2}, \frac{N_x}{2}]$ and $p_y \in [-\frac{N_y}{2}, \frac{N_y}{2}]$. With a simple manipulation, Eq.59 can be formulated in terms of a convolution operation

$$\max_{\gamma_n} \left\| R^n f * z_{\gamma_n}(x, y) \right\|, \quad (60)$$

bounded into the frame $[-\frac{N_x}{2}, \frac{N_x}{2}] \times [-\frac{N_y}{2}, \frac{N_y}{2}]$. At this point, by applying the duality product-convolution of the DFT it leads to

$$R^n f * z_{\gamma_n}(x, y) \xrightarrow{\mathcal{F}} \widehat{R^n f} \cdot \hat{Z}_{\gamma_n}(x, y), \quad (61)$$

where $\widehat{R^n f}$ and $\hat{Z}_{\gamma_n}(x, y)$ are the Fourier transforms of $R^n f$ and $z_{\gamma_n}(x, y)$ respectively. Finally, to search the most powerful atom by using this DFT based method can be written as

$$\max_{\gamma_n} \left\| R^n f * z_{\gamma_n}(x, y) \right\| = \max_{\gamma_n} \left\| \mathcal{F}^{-1} \left\{ \widehat{R^n f} \cdot \hat{Z}_{\gamma_n}(x, y) \right\} \right\|. \quad (62)$$

This full-search method proposed below takes advantage of the FFT usage, saving a lot of computational load. Exactly, the Matching Pursuit complexity for one atom is reduced to

$$O(N_x^2 \times \log_2^3(N_x)). \quad (63)$$

Appendix 4: Dictionary based on Gabor functions

In most of the previous research done on Matching Pursuit over images, most of the decompositions have been done over dictionaries based on Gabor [20] atoms due to its good time-frequency localization. For our purposes, the use of Gabor atoms is justified due to the fact that they are appropriate functions to code patterns and textures.

A Gabor atom is defined as a modulated Gaussian function:

$$g_\gamma(x, y) = \sqrt{2}K e^{-(x^2+y^2)} e^{i(\omega_x x + \omega_y y)}, \quad (64)$$

where K is a normalization constant to have unitary norm. But, for our purposes, we will take just the real part of $g_\gamma(x, y)$. Then we get:

$$g_\gamma(x, y) = \sqrt{2}K e^{-(x^2+y^2)} \cos(\omega_x x). \quad (65)$$

Assuming that our image has a size of $N_x \times N_y$ pixels, we can define the set of parameters $\gamma = (\mathbf{p}, \mathbf{s}, \theta)$ necessary to generate the whole dictionary:

$$\begin{array}{ll} \mathbf{p} = [p_x, p_y] \text{ where } p_x \in [0, N_x), p_y \in [0, N_y) & \text{translations} \\ \mathbf{s} = [s_x, s_y] & \text{scaling factors} \\ \theta \in [0, \pi) & \text{rotation,} \end{array}$$

with \mathbf{p} the translation vector that will set the center of the atom into the image, \mathbf{s} where s_x is the dilation in the x axis and s_y is the dilation in the y axis and θ the rotation angle. For simplicity, we will take isotropic Gaussians instead of anisotropic Gaussians, that is $s_x = s_y$.

At this point, we have to choose our parameters carefully to be sure that this set of atoms have an associated family that is a frame of $\mathbf{L}^2(\mathbb{R}^2)$ [34]. If $\Delta\theta$ and Δs are small enough finite linear expansions of space-frequency atoms are dense in $\mathbf{L}^2(\mathbb{R}^2)$, hence this dictionary is also complete and, then, valid for image coding. To satisfy these conditions we have chosen our parameters in this way:

- $\Delta\theta = 10^0$.
- $s_x \in [0, NN \cdot (\lfloor \log_2(N) \rfloor - 3)] \in \mathbb{Z}$ where $NN \in [1, \log_2(N)] \in \mathbb{Z}$, $N = \min(N_x, N_y)$ and $s_x > s_y$.
- ω_x covering the frequencies where there is presence of a signal.

Once we have defined the parameters to generate the atom's family, we can define the procedure to create the atom. This procedure is non-commutative, so, the order is fixed and its application is:

1. Apply the translation by $[p_x, p_y] \in \mathbb{Z}^2$.
2. Rotate by θ the translated atom.

3. Scale the translated and rotated atom by $\sigma_x = 2^{\frac{s_x}{NN}}$ in the axis x .

This leads to compute:

$$x_\gamma = \frac{(x - p_x) \cos(\theta) + (y - p_y) \sin(\theta)}{2^{\frac{s_x}{NN}}} \quad (66)$$

$$y_\gamma = \frac{(x - p_x) \sin(\theta) - (y - p_y) \cos(\theta)}{2^{\frac{s_y}{NN}}}, \quad (67)$$

and $g_\gamma = g(x_\gamma, y_\gamma)$.

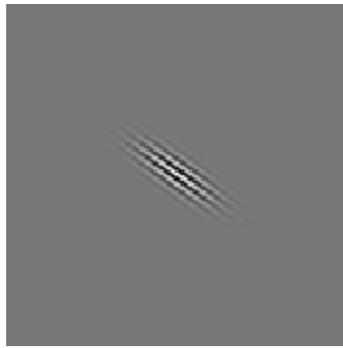


Figure 14: Gabor atom.

References

- [1] O. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, A. Zakhor, *Video Compression Using Matching Pursuits*, IEEE Transactions on Circuits and Systems for Video Technology, 17(2):123–143, Feb. 1999.
- [2] C.C.Bisell, S.Mallat, *Matching Pursuit of images*, IEEE Proc. Int. Conf. Image Proc., volume I, 53-56, Washington DC, Oct. 1995
- [3] S.Boyd, L.Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [4] C.Cantó Ferrer, *Edge and Texture Adaptive Weak Matching Pursuit*, MS Thesis, EPFL, Aug. 2003.
- [5] J.R.Casas, F.Marqués, P.Salembier, *Processament d'imatge (PIM), transparències del curs*, Departament de Teoria del Senyal i Comunicacions, UPC.
- [6] R.R.Coifman, M.V.Wickerhauser *Entropy-based algorithms for best basis selection* IEEE Transactions on Information Theory, vol. 38(2), 713-718, 1992.
- [7] I.Daubechies, *Ten lectures on Wavelets*, CBMS-NSF Series in Appl. Math., SIAM, 1991.
- [8] G.Davis, S.Mallat, M.Avellaneda *Adaptative Greedy Approximations*, Constructive Approximation, Springer-Verlag, New York INC.1997.
- [9] R.A.DeVore, B.Jawerth V.Popov, *Compression of wavelet decompositions* American Journal of Mathematics vol. 114, 737-785, 1992.
- [10] R.A.DeVore, *Nonlinear Approximation*, Acta Numerica 51-150, 1998.
- [11] D.L.Donoho, *Unconditional bases are optimal bases for data compression and for statistical estimation*, Appl. Comput. Harmon. Anal., vol. 1, 100-115, 1993.
- [12] D.L.Donoho, S.Chen, M.A.Saunders *Atom decomposition by basis pursuit*, Technical report, Dept. Stat., Stanford Univ., Stanford (CA), 1996.
- [13] H.G.Feichtinger, A.Tirk, T.Strohmer, *Hierarchical Parallel Matching Pursuit*, NUHAG-Numerical Harmonic Analysis Group, University of Vienna, Department of Mathematics.
- [14] R.M.Figueras i Ventura, *Image Coding with Matching Pursuit*, MS Thesis, EPFL, Aug. 2001.
- [15] R.M.Figueras i Ventura, P.Vandergheynst, *Matching Pursuit through Genetic Algorithms*, LTS Technical Report 01.02, July 2001.

- [16] R.M.Figueras i Ventura, P.Vandergheynst, P.Frossard, M.Kunt, *A Posteriori Quantization of Progressive Matching Pursuit Streams*, to appear in IEEE Transactions of Signal Processing.
- [17] R.M.Figueras i Ventura, O.Divorra Escoda, P.Vandergheynst, *Full Search Matching Pursuit*, LTS Technical Report, publication pending.
- [18] P.J.Franaszczuk, G.K.Bergey, H.Eisenberg, *Matching Pursuit analysis of Mesial Temporal Lobe Complex Partial Seizures suggests increased nonlinear complexity at seizure conclusion*, Annual Meeting of the American Epilepsy Society, San Diego, California, Dec. 6-9, 1998.
- [19] J.H.Friedman, W.Stuetzle, *Projection pursuit regression*, Journal of American Statistics Society, vol. 76,817-823, 1981.
- [20] D.Gabor, *Theory of communication*, J.IEE, 93:429-457, 1946.
- [21] M.R.Garey, D.S.Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H.Freeman and Co., New York, 1979.
- [22] M.Gharavi-Alkhansari, T.S.Huang, *A fast orthogonal matching pursuit algorithm*, Proc. ICASSP, p.1389-1392, Seattle (WA), May 1998.
- [23] R.C.Gonzalez, R.E.Woods, *Digital Image Processing*, Addison-Wesley Publishing Company,1992.
- [24] Jain, Anil K., *Fundamentals of digital image processing*, Englewood-Cliff, NJ: Prentice-Hall, 1989.
- [25] Joint Picture Expert Group, *JPEG Standard*, <http://www.jpeg.org>.
- [26] Joint Picture Expert Group, *JPEG 2000 Standard*, <http://www.jpeg.org/JPEG2000.html>.
- [27] L.K.Jones, *On a conjecture of Huber concerning the convergence of projection pursuit regression*, The Annals of Statistics, vol.15, No.2, p.880-882, 1987.
- [28] G.A.Khuwaja, G.A. and A.S.Tolba, *Fingerprint image compression*, Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop, Vol.2, pp. 517-526, 2000.
- [29] M.A.Lagunas, J.Vidal and A.I.Perez-Neira, *Procesado del Señal: apuntes de la asignatura*, UPC.
- [30] J.N. Lin, X.Nie and R.Unbehauen, *Two-dimensional LMS adaptive filter incorporating a local-mean estimator for image processing*, IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing, vol.40, pp.417-428, 1993.

- [31] S.Loncaric, *A survey of shape analysis techniques*, Pattern Recognition, vol.31:8, pp.983-1001, 1998.
- [32] E.Maggio, *Matching Pursuit in Video Coding*, MS Thesis, EPFL, Sept. 2003.
- [33] S.Mallat, Z.Zhang *Matching Pursuits with time-frequency dictionaries*, IEEE Transactions on Signal Processing, vol. 41(12), 3397-3415, 1993.
- [34] S.Mallat, *A Wavelet Tour of signal processing*, Academic Press (1998), USA.
- [35] D.Maltoni et al., *Handbook of Fingerprint Recognition*, Springer (2003), USA.
- [36] Y.C.Patri, R.Rezaifar, P.S.Krishnaprasad, *Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition*, 27th Asilomar Conf. on Signal, Systems and Comput., Nov. 1993.
- [37] B.D.Rao, *Signal Processing with the sparseness constraint*, Proc. ICASSP, p.1861-1864, Seattle (WA), May 1998.
- [38] J.E.Marsden, A.J.Tromba, *Vector Calculus*, W.H. Freeman and Company, 1996.
- [39] J.A.Tropp, *Greed is good: Algorithmic results for sparse approximation*, Texas Institute of Computational Engineering and Sciences, Technical Report, Feb. 2003.
- [40] Z.Zhang, *Matching Pursuit*, Ph.D. dissertation, New York University, 1993.