University of Bergen

Department of Mathematics

Master Thesis

Sub-Riemannian geometry and its applications to Image Processing

Author: Francesco Ballerin

Acknowledgements

The author would like to thank Prof. Erlend Grong (UiB) for supervising this thesis and being the source of inspiration for the topic. His dedication and the countless hours he spent providing insights and explanations over the difficult bits were fundamental in the writing of this work.

He, together with Prof. Irina Markina and the PDE and Analysis group of the University of Bergen, has organized the PDE and Analysis seminar as well as multiple conferences and smaller meetings that were instrumental in deepening my relationship with many branches of mathematics and to receive feedback on my ongoing work.

A special thanks to Adrien Laurent (UiB) and fellow students Andrea Dalla Costa (EIT), Alessandro Foradori (UniTN), Daniele Gottardini (PoliMI), Stefano Huber (EPFL) and Katja Phillips (UiB) for reviewing the original manuscript, listening to the author's explanations and providing constructive criticism.

Abstract

When using our sense of vision, lightrays that are reflected by the objects positioned in the 3D world form a 2D image on the retinas of our eyes. This image is perceived and processed by the human brain through the primary visual cortex V1, a part of the occipital lobe which is highly specialized in processing information about orientation, motion and is of vital importance for pattern recognition. The orientation sensitivity in particular helps us to perceive depth and superposition of objects, detect obstructions in the field of view (corruption), and up to a certain extent, intrinsically fill gaps in the image we perceive. This last process is what in the field of Image Processing is called image inpainting: the process of restoring an image by filling the missing information in a region by using information from the rest of the image. The digital image that is acquired by the sensor of a camera can be regarded (without color information) as a real valued function of 2 variables $f(x, y)$ that maps the position of a pixel in a rectangular domain to the intensity of the light that corresponds to that pixel. In this setting the perceived orientation is just the orientation of the level curves of the image, which can be easily computed as the gradient of the function $f(x, y)$ rotated by an angle of $\frac{\pi}{2}$. In this work we model the neurons of the visual cortex V1 as points of the Lie group $SE(2) \simeq \mathbb{R}^2 \times S^1$, introducing a lifting function $\mathcal{L} : \mathbb{R}^2 \to SE(2)$ together with a projection $\pi : SE(2) \to \mathbb{R}^2$ to link the space of digital images to the space of representations of images in the visual cortex V1. On $SE(2)$ it is possible to define a sub-Riemannian geometry with differential operators analogous to the ones of classical calculus, and therefore model the neural activity as a partial differential equation with $\mathcal{L}(f)(x, y, \theta)$ as its initial condition. The diffusion equation on $SE(2)$ with a sub-Laplacian defined in such a way to propagate neural activity in the direction of the level curves of the image seems to be particularly natural and effective at restoring corrupted images. Ultimately we can ask ourselves the question: *"how do we evaluate the effectiveness of a restoration algorithm?"*. At a first glance, without taking into account the subjectivity of human perception, this might seem like a mathematically trivial task. But after a deeper analysis on what a *good restoration* means we can see how the problem is in fact non that trivial, and does not admit a single solution.

The main new results that are presented within this work are an example-based analysis of the current state of the art in PDE-based image restoration methods, a novel PDE-based algorithm for image inpainting that appears to be more stable while producing sharper results, and the introduction of two new metrics to assess the quality of an image restoration process.

Contents

Chapter 1

Introduction

1.1 Motivation

Image inpainting is a conservation process that aims to restore an image that has been damaged in a way so that in a certain region of the canvas all the information is lost. This process has its roots in physical artworks, such as paintings and film photography, where a professional performs a manual process in order to fix a deteriorated medium.

The modern use of inpainting can be traced back to 18th century's Venice, a city that inherited an immense art collection produced during the Italian Renaissance and needed a more scientific and procedural approach in the restoration process of its art, especially in the form of paintings [16].

With the advent of computers and the diffusion of digital images the need of proper algorithms to reconstruct missing or damaged areas of digital photography and videos started to rise. Watermarks removal, text removal, image compression, super resolution and red-eye removal are just some of the many applications that digital inpainting addresses.

There are different techniques currently used for image inpainting that can be roughly divided in families of methods: patch-based substitutions, PDE methods and machine learning models (more recently in the form of deep neural networks).

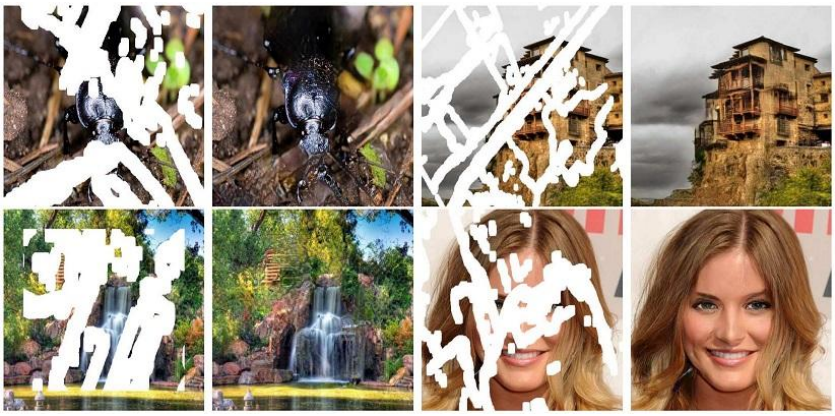


Figure 1.1: Example of deep neural network inpainting, courtesy of Liu et al. [34]

The motivation behind this work is to study a specific PDE-based method that takes inspiration from the primary visual cortex V1 in mammals. This portion of the brain, located in the occipital lobe, is particularly sensitive to orientation, among other things, and is considered responsible for the ability of the brain to fill-in the disruptions in the field of view due to obstructions.

The main new results that are presented within this work are:

- an example-based analysis of the current state of the art in PDE-based image restoration methods
- a novel PDE-based algorithm for image inpainting that appears to be more stable while producing sharper results
- the introduction of two new metrics to assess the quality of an image restoration process

1.2 Overview

The content of this work is divided into chapters according to topic and source of the material.

The mathematical background in geometry that the reader might not be well acquainted with is introduced in Chapter 2. The basics in *Lie groups*, *Lie algebras*, *Riemannian geometry* and *sub-Riemannian geometry* are presented. Afterwards a few examples of sub-Riemannian manifolds are introduced, among which are $PT\mathbb{R}^2$ and $SE(2)$, commonly named *orientation scores*. These examples in particular will be relevant in the following chapters as a good mathematical model for the visual cortex V1.

In Chapter 3 the basics in Image Processing are introduced in order to establish a notation and formally build the foundation on top of which the main results of this work are developed.

The mathematical model of the primary visual cortex V1 and the derived algorithms for image restoration are presented in Chapter 4. In addition, a detailed overview of the work produced so far on the subject and the current state-of-the-art in PDE-based image restoration is discussed.

The author's contribution is presented in Chapters 5 and 6.

In Chapter 5 a new lifting technique based on $PT\mathbb{R}^2$ and a novel diffusion algorithm are introduced. These two combined produce an approach to image restoration that appears extremely versatile and empirically very stable.

In Chapter 6 the problem of quantitatively assessing the quality of the obtained restoration is discussed and new metrics inspired by the neural activation in the cortex V1 are introduced.

1.3 Prerequisites

Although most of the theory required to follow the discussion in this work is presented either as a dedicated chapter, or along the way when needed, the basics in *calculus* (in particular the basics of *partial differential equations*), *linear algebra*, *measure and integration theory*, *topology* and *manifolds* are taken for granted and will not be introduced.

A previous knowledge of differential geometry in the form of *Riemannian manifolds*, *image processing* and *algorithms and data structures* is recommended but not required, as all the tools strictly needed are introduced as to make this work as self-contained as possible.

Chapter 2

Geometric tools

The contents of this section have been heavily inspired by the work of [41], [30] and [33]. It is assumed that the reader has a basic knowledge of algebraic structures and manifolds. A thorough introduction to manifolds can be found at [41]. In the last section of this chapter, a geometric application to optimal control problems is introduced.

2.1 Lie Groups

Lie groups originate between 1874 and 1884 from a series of papers due to Sophus Lie. The original motivation behind the initial study of Lie groups was to apply them to unify the whole area of ordinary differential equations through the study of symmetry. This application didn't quite result in the success that Lie had hoped for, but created a foundation that was inspirational to Riemann and Klein[22]. The theory of Lie groups rapidly spread up to the point that it appeared in the list of problems by David Hilbert at position number 5 [24].

Definition 1 (Lie Group)
A structure G is a Lie Group if it is a smooth manifold and it is a group such that the operation $G \times G \to G$ induced by the group structure is smooth.

To discuss the specific case of matrix Lie groups, we need a notion of differentiability of a matrix. Let I be an open interval of $\mathbb{R}$, and let $A(t)$ be an $n \times m$ matrix for $t \in \mathbb{R}$. If each entry of the matrix $A(t)$ is a differentiable function of t then A is said to be *differentiable* and the *derivative* of A is defined componentwise.

The vector space of $n \times n$ matrices M_n can be endowed with a norm

$$\|X\| = \left(\sum x_{ij}^2\right)^{\frac{1}{2}}$$

Example 2
$(\mathbb{R}^n, +)$

It is trivially a manifold because it is isomorphic to itself.
Summation $+ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth operation componentwise, and therefore smooth as a whole. Hence it is a Lie group.

Example 3
$GL(n)$

Recall that $GL(n) := \{A \in M_n \,|\, \det A \neq 0\} = \det^{-1}(\mathbb{R}\backslash\{0\})$. M_n can also be written as $\mathbb{R}^{n\times n}$ which is clearly isomorphic to $\mathbb{R}^{n^2}$. We also know from calculus that polynomials are continuous functions, and we can characterize the determinant of a matrix as a polynomial. Since the set $\{0\}$ is closed in $\mathbb{R}$ under the euclidean topology, then the inverse is also closed since it is the inverse of a continuous function. Thus the set of matrices that have zero determinant is closed, and therefore the set of matrices with non-zero determinant (which corresponds to $GL(n)$) is open. Hence $GL(n)$ is a manifold by the following theorem:

Theorem 4
If M is a manifold, and $U \in M$ is an open subset of M, then U is also a manifold.

Ref: [41] at *5.12* on pg. 54

Moreover, for $A, B \in GL(n)$ it holds componentwise that

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

which is a polynomial in the coordinates of A and B. Therefore matrix-multiplication is a C^∞ map.

Using Cramer's rule for computing the entry of the inverse matrix A^{-1} one can obtain

$$(A^{-1})_{ij} = \frac{1}{\det A}(-1)^{i+j}((j,i) - \text{ minor of } A)$$

which is well defined for $\det A \neq 0$. Since the inverse is a polynomial division, it is

C^∞ when the denominator is non-zero, which is always the case in $GL(n)$.

Thus $GL(n)$ is a Lie group.

Another fundamental Lie group is $SL(n)$. In the following example the proof of $SL(n)$ being a Lie group is done by considering only results on smooth manifolds, such as the regular value theorem and the theory of regular submanifolds. Later on in this chapter more powerful tools and theorems on Lie groups will be developed, which will produce a simpler proof for $SL(n)$ being a Lie group. More in general closed matrix subgroups of $GL(n)$ will be shown to always be Lie subgroups.

Example 5
$SL(n)$

Recall that $SL(n) := \{A \in GL(n) \mid \det A = 1\} = \det^{-1}(\{1\})$ and that $SL(n) \subset GL(n)$. $SL(n)$ is a manifold by the following theorem:

Theorem 6 (Regular value theorem)
Let $f : M \to \mathbb{R}$ be a smooth function and let M be an $n-$dimensional manifold. Then if $f^{-1}(c)$ is non-empty, it is an $n-1$ regular submanifold of M.

Ref: [41] at *9.8* on pg. 104

Since $SL(n)$ is a regular submanifold of $GL(n)$, we can now consider the inclusion

$$i : SL(n) \times SL(n) \to GL(n) \times GL(n)$$

which is C^∞ by the following theorem:

Theorem 7
If N is a regular submanifold of M, then the inclusion $i : N \to M$, $i(p) = p$ is an embedding.

Ref: [41] at *Thm 11.14* on pg. 124

To see that the inverse map is C^∞ we just compose the inclusion map in $GL(n)$ with the inverse of $GL(n)$ to obtain a C^∞ map $SL(n) \to GL(n)$. We can now conclude the proof by noticing that if A is a matrix with $\det(A) = 1$ then $\det(A^{-1}) = 1$ and therefore the map is actually $SL(n) \to SL(n)$.

It is useful, in the context of Lie groups, to fix in the group operation $\mu : G \times G \to G$ either the left or the right element and introduce the maps $L_a x = \mu(a, x) = ax$ and $R_a x = \mu(x, a) = xa$, respectively the *left multiplication* and *right multiplication*. These

maps are often called *translations*, and since the group operation is smooth by definition of Lie group, these maps are smooth.

Definition 8
Two Lie groups G and H are *isomorphic* if $\exists\ F : G \to H$ which is a group isomorphism and a diffeomorphism.

As in the study of manifolds it is necessary to study some structures as submanifolds, the analogous case happens in Lie groups, with the introduction of *Lie subgroups*.

Definition 9 (Lie subgroup)
Let G be a Lie group. A *Lie subgroup* is an abstract (in the algebraic sense) subgroup H of G, which is also an immersed submanifold via the inclusion map such that the group operations of G are smooth on H.

The definition makes use of the concept of an *immersed* submanifold rather than a *regular* one, because in a regular submanifold the Lie operation map $H \times H \to H$ and the group inverse $H \to H$ would be automatically smooth.

Proposition 10
If H is an abstract subgroup and a regular submanifold of a Lie group G, then H is a Lie subgroup of G.

Ref: [41] at *Prop. 15.11* on pg. 168

Since in the setting of the previous proposition the inclusion map $i : H \to G$ is an embedding, H is called an *embedded Lie subgroup*.

And now we can introduce a powerful theorem that, especially in the case of matrix groups, simplifies the operation of determining if a subgroup of a Lie group is a Lie subgroup.

Theorem 11 (Closed subgroup theroem)
Let G be a Lie group, and A be a closed subgroup in the abstract sense. Then A has a unique manifold structure which makes it into a Lie subgroup of G.

Ref: [43] at *Thm. 3.42* on pg. 110

Example 12
The special linear group $SL(n)$ and the orthogonal group $O(n)$ are the zero sets of polynomial equations on $GL(n)$ (in particular for $Q \in O(n)$ we can write $Q \cdot Q^{\intercal} - \mathbf{I}_n = 0$ and for $K \in SL(n)$ we can write $det(K) - 1 = 0$). As such, they are closed subsets of $GL(n)$ and, by the closed subgroup theorem, they are both embedded Lie subgroups of $GL(n)$.

2.2 Matrix Lie groups and the Exponential map

Many Lie groups of interest, as we have already seen when citing a few examples, can be expressed as matrix groups with the usual matrix multiplication as group operation. In the case of an $n \times n$ matrix Lie group G, both an element $A \in G$ and $X_A \in T_A G$ can be expressed as $n \times n$ matrices.

Let G be a matrix Lie group and $A \in G$ an element. Then the tangent space in A is defined as

$$T_A G = \{\dot{B}(t)|_{t=0} | B : (-\epsilon, \epsilon) \to G,\ B(0) = A\})$$

Example 13
In the case $(\mathbb{R}^2, +)$ we can represent the group as a matrix group endowed with the classic matrix multiplication as operation, by the isomorphism

$$\varphi : (\mathbb{R}^2, +) \to (M_3, \cdot)$$

$$\varphi(\mathbf{x}) = \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}$$

The identity element in $(\mathbb{R}^2), +)$ is mapped to $\mathbf{I}_n$ and in this representation the basis vectors $\partial_x = (1, 0)$ and $\partial_y = (0, 1)$ for $T_e G$ become

$$\partial_x = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \partial_y = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

by taking $\varphi((t, 0))$ and $\varphi((0, t))$ and evaluating the derivative at $t = 0$.

By taking curves on the manifolds we are able to compute the tangent space at every point. But this operation is actually redundant, as in the specific case of a matrix Lie

groups the computation of every tangent space can be brought back to the computation of the tangent space at the identity $T_{\mathbf{I}}G$.

Let G be a matrix Lie group, and consider a curve in G defined as

$$B : (-\epsilon, \epsilon) \to G$$

$$B(0) = A$$

Define now

$$\tilde{B} := A^{-1}B : (-\epsilon, \epsilon) \to G$$

$$\tilde{B}(0) = A^{-1}B(0) = \mathbf{I}$$

Then we can compute the derivative of $B(t)$ and evaluate it at $t = 0$ as

$$\frac{d}{dt}B(t)\big|_{t=0} = \frac{d}{dt}(AA^{-1}B(t))\big|_{t=0} = A\frac{d}{dt}(A^{-1}B(t))\big|_{t=0} = A\frac{d}{dt}\tilde{B}(t)\big|_{t=0}$$

The consequence of this brief computation is that, if we want to compute $X \in T_AG$ there exists $Y \in T_{\mathbf{I}}G$ so that we can write $X = AY$ "shifting" the computation back to the tangent space at the identity by using a left multiplication by A.

As it will be shown later in this chapter this is possible in every Lie group by computing the differential of the group product. However, in the case of a subgroup of $GL(n)$ a curve $A(t)$ of non-singular matrices is needed. The *exponential map* is well suited for this purpose, as it happens to be intrinsically non-singular.

Definition 14 (Matrix exponential and logarithm)
The matrix exponential of a matrix $X \in M_n$ is defined as

$$e^X = \mathbf{I}_n + X + \frac{1}{2!}X^2 + \frac{1}{3!}X^3 + \cdots + \frac{1}{k!}X^k + \ldots$$

which is essentially how the exponential of a real number is defined using Taylor expansions. This construction is well defined as the series converges (see [41] at *chpt. 15.3*).

Given a matrix Y, another matrix X is said to be the *matrix logarithm* of Y if $e^X = Y$.

A formula for the matrix logarithm can be defined in terms of a power series for

$X \in GL(n)$ if $\|X - \mathbf{I}_n\| < 1$ as

$$\log X = (X - \mathbf{I}_n) - \frac{(X - \mathbf{I}_n)^2}{2} + \cdots + \frac{(-1)^{k-1}}{k}(X - \mathbf{I}_n)^k + \cdots$$

The matrix exponential retains some of the properties of the scalar exponential, but is intrinsically different: although $e^{\mathbf{0}} = \mathbf{I}_n$ and $\frac{d}{dt}e^{tX} = Xe^{tX}$ as we are accustomed to, it is however not true in general that $e^A e^B = e^{A+B}$.

A proof of the second property can be found in [41] (*Prop 15.17*), whereas a counterexample to the third property is $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$.

We can immediately see an application of the matrix exponential in the following exercise:

Exercise 15

Let $G = \mathbb{R}^3$ with the product defined as

$$(x, y, z) \cdot_G (\tilde{x}, \tilde{y}, \tilde{z}) = (x + \tilde{x}, y + \tilde{y}, z + \tilde{z} + \frac{1}{2}(x\tilde{y} - \tilde{x}y))$$

and $H \subset GL(3)$ the Heisenberg group with the classic matrix multiplication $\cdot_M$

$$H = \left\{ \begin{pmatrix} 1 & a & c \\ 0 & 1 & b \\ 0 & 0 & 1 \end{pmatrix} \middle| a, b, c \in \mathbb{R} \right\}$$

Show that G and H are isomorphic.

. . . . . .

Let us re-arrange the elements of $\mathbb{R}^3$ in a matrix form, endowed with the same product rule $\cdot_G$, so that

$$G = \left\{ \begin{bmatrix} 0 & x & z \\ 0 & 0 & y \\ 0 & 0 & 0 \end{bmatrix} \middle| x, y, z \in \mathbb{R} \right\}$$

Applying now the exponential map defined above to an element of G it is clear that

$$\exp\left(\begin{bmatrix} 0 & x & z \\ 0 & 0 & y \\ 0 & 0 & 0 \end{bmatrix} \right) = \begin{bmatrix} 1 & x & z + \frac{1}{2}xy \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix}$$

where $z + \frac{1}{2}xy \in \mathbb{R}$ so that the exponential map is a map $G \to H$.

With the following computation

$$\begin{aligned}
\exp\left(\begin{bmatrix} 0 & x & z \\ 0 & 0 & y \\ 0 & 0 & 0 \end{bmatrix}\right) \cdot_M \exp\left(\begin{bmatrix} 0 & \tilde{x} & \tilde{z} \\ 0 & 0 & \tilde{y} \\ 0 & 0 & 0 \end{bmatrix}\right) &= \begin{bmatrix} 1 & x & z + \frac{1}{2}xy \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \cdot_M \begin{bmatrix} 1 & \tilde{x} & \tilde{z} + \frac{1}{2}\tilde{x}\tilde{y} \\ 0 & 1 & \tilde{y} \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & x + \tilde{x} & z + \tilde{z} + x\tilde{y} + \frac{1}{2}(xy + \tilde{x}\tilde{y}) \\ 0 & 1 & y + \tilde{y} \\ 0 & 0 & 1 \end{bmatrix} \\
&= \exp\left(\begin{bmatrix} 0 & x + \tilde{x} & z + \tilde{z} + \frac{1}{2}(x\tilde{y} - \tilde{x}y) \\ 0 & 0 & y + \tilde{y} \\ 0 & 0 & 0 \end{bmatrix}\right)
\end{aligned}$$

we can see that the exponential map is exactly the isomorphism we were looking for. Hence $(G, \cdot_G)$ and $(H, \cdot_M)$ are isomorphic.

2.3 Lie algebras

If one takes the concept of a Lie group G, and endows the tangent space at the identity T_eG with a *Lie bracket* operation

$$[\cdot, \cdot] : T_eG \times T_eG \to T_eG$$

we obtain what is called the *Lie algebra* of the Lie group, a structure that encodes much information about the Lie group itself. Let us now introduce formally the definition of a Lie algebra, and build up the theory that will allow us to study some of its applications.

Definition 16 (Lie algebra)
A *Lie algebra* is a vector space $\mathfrak{g}$ over some field F ($\mathbb{R}$ or $\mathbb{C}$)together with a bracket operation $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ s.t. the following properties hold:

(*i*) *bilinearity*: $[aA+bB, C] = a[A, C]+b[B, C]$ and $[C, aA+bB] = a[C, A]+b[C, B]$ for $a, b \in F$ and $A, B, C \in \mathfrak{g}$

(*ii*) *skew-symmetry*: $[A, A] = 0$ for $A \in \mathfrak{g}$

(*iii*) *Jacobi identity*: $[A, [B, C]] + [B, [C, A]] + [C, [A, B]] = 0$ for $A, B, C \in \mathfrak{g}$

One can notice that the definition of Lie bracket is not unique. A trivial definition can be to set $[X, Y] = 0$ for all $X, Y \in \mathfrak{g}$. This particular case takes the name of *Abelian Lie algebra*. For vector fields the *Lie bracket of vector fields*, also called the *Jacobi-Lie bracket* or *commutator of vector fields*, is defined pointwise as $[X, Y]_p f = (X_p Y - Y_p X) f$ for a C^∞ germ function f. As p varies over the chart (U, φ), $[X, Y]$ becomes a vector field on U and if both X and Y are smooth it follows that $[X, Y]$ is also smooth. Endowed with such Lie bracket the set of all C^∞ vector fields, $\mathfrak{X}(G)$, is a Lie algebra.

A *Lie subalgebra* of a Lie algebra $\mathfrak{g}$ is a vector subspace $\mathfrak{h} \subset \mathfrak{g}$ that is closed under the bracket operation defined on $\mathfrak{g}$.

Let $F : N \to M$ be a smooth map of manifolds. A vector field X on N is said to be *F-related* to a vector field $\bar{X}$ on M if for all $p \in N$

$$F_{*,p}(X_p) = \bar{X}_{F(p)}$$

which in other words means that two vector fields are F-related if at every point we obtain the same result if we first move to the tangent space on N and then apply the differential map of F or if we apply first F and then we move to the tangent space on M.

We can apply the definition of F-relatedness to study the group operation of a Lie group, in particular the left-translation. We define a vector field that is L_a-related to itself to be *left-invariant*. The formal definition is:

Definition 17 (Left-invariant vector field)
Let G be a Lie group. A vector field $X \in \mathfrak{X}(TG)$ is called left-invariant if for $f \in C^\infty(G, \mathbb{R})$, $p \in G$ then $(Xf)(L_a p) = (X(f \circ L_a))(p)$. Equivalently, if $L_{a*} X = X \; \forall a \in G$.

Proposition 18
Any left-invariant vector field X on a Lie Group is C^∞.

Ref: [41] at *16.8* on pg. 181

We denote the set of all left-invariant vector fields of a Lie group G as $\mathcal{L}(G)$. Since this is a vector subspace that is closed under the bracket operation (Ref: [41] at *16.9* on pg. 182) it is a Lie algebra, which is of great interest as it happens to be isomorphic to $T_e G$.

Proposition 19
There is a one-to-one correspondence $T_e(G) \leftrightarrow \mathcal{L}(G)$

Proof. A left-invariant vector field X is completely defined by its value at the identity X_e. In fact, for every $g \in G$

$$X_g = X_{ge} = L_{g*}(X_e)$$

Conversely, given a tangent vector $A \in T_e(G)$, we can define a left invariant vector field $\tilde{A}$ s.t. $\tilde{A}_g = L_{g*}A$. Now

$$L_{g*}(\tilde{A}_h) = L_{g*}L_{h*}(\tilde{A}_e) = L_{g*}L_{g*}(A) = (L_g \circ L_h)_* A = L_{gh}A = \tilde{A}_{gh}$$

□

This relationship is more profound than a simple one-to-one correspondence, but it is in fact a vector space isomorphism, as we shall now see.

Consider the bijection $\varphi : T_eG \to \mathcal{L}(G)$, and given $A, B \in T_eG$ define the Lie bracket $[A, B] \in T_eG$ as $[A, B] := [\tilde{A}, \tilde{B}]_e$ for $\tilde{A} = \varphi(A)$, $\tilde{B} = \varphi(B)$. The following proposition completes the discussion, showing that φ preserves the group structure as

$$\varphi([A, B]) = [\varphi(A), \varphi(B)]$$

Proposition 20
If $A, B \in T_eG$ and $\tilde{A}, \tilde{B}$ are the left-invariant vector fields they generate through φ then

$$\widetilde{[A, B]} = [\tilde{A}, \tilde{B}]$$

Ref: [41] at *16.10* on pg. 183

For the general linear group $GL(n)$ we can identify the point derivations at the identity with M_n via

$$\sum a_{ij} \frac{\partial}{\partial x_{ij}}\bigg|_{\mathbf{I}} \leftrightarrow [a_{ij}]$$

Let $A, B \in T_{\mathbf{I}}G$ such that

$$A = \sum a_{ij} \frac{\partial}{\partial x_{ij}}\bigg|_{\mathbf{I}} \qquad B = \sum b_{ij} \frac{\partial}{\partial x_{ij}}\bigg|_{\mathbf{I}}$$

and let $\tilde{A}, \tilde{B}$ be the left invariant vector fields generated by A, B respectively:

$$\tilde{A}_g = (L_g)_* A = gA = \sum_{i,j} (gA)_{ij} \frac{\partial}{\partial x_{ij}}\bigg|_g$$

$$\tilde{B}_g = (L_g)_* B = gB = \sum_{i,j} (gB)_{ij} \frac{\partial}{\partial x_{ij}}\bigg|_g$$

therefore $\tilde{A}_{\mathbf{I}} = A$ and $\tilde{B}_{\mathbf{I}} = B$. Applying now the bracket $[\tilde{A}, \tilde{B}]_{\mathbf{I}}$ to elements of the standard basis x_{ij} yields

$$[\tilde{A}, \tilde{B}]_{\mathbf{I}} x_{ij} = \tilde{A}_{\mathbf{I}} \tilde{B} x_{ij} - \tilde{B}_{\mathbf{I}} \tilde{A} x_{ij} = A\tilde{B} x_{ij} - B\tilde{A} x_{ij}$$

$$A\tilde{B} x_{ij} = \sum_{p,q} a_{pq} \frac{\partial}{\partial x_{pq}}\bigg|_{\mathbf{I}} \left(\sum_k b_{kj} x_{ik} \right) = \sum_{p,q,k} a_{pq} b_{kj} \delta_{ip} \delta_{kq} = \sum_k a_{ik} b_{kj} = (AB)_{ij}$$

Therefore

$$[\tilde{A}, \tilde{B}]_{\mathbf{I}} x_{ij} = (AB)_{ij} - (BA)_{ij}$$

Denote $T_{\mathbf{I}} GL(n)$ with its Lie algebra structure by $\mathfrak{gl}(n)$.

Now that we have expanded on the concept of Lie groups endowing the tangent space with a Lie bracket one could ask themselves if there is a canonical mapping between a group and its tangent space at the origin. In matrix Lie groups such mapping is provided by the exponential map, previously defined in Definition 14. Consider the following proposition:

Proposition 21

For $X \in M_n$, the matrix exponential $e : M_n \to GL(n)$ satisfies

$$\frac{d}{dt} e^{tX} = X e^{tX} = e^{tX} X$$

Ref: [41] at *Prop 15.17* on pg. 171

Then for $X \in M_n$, $A \in GL(n)$ and

$$\gamma : (-\epsilon, \epsilon) \to GL(n)$$

$$\gamma(t) = A e^{tX}$$

is a smooth curve in $GL(n)$ starting at $\gamma(0) = A$. From manifold theory (Ref: [41] at *8.6* on pg. 92)) we know that the *velocity vector* of the curve at $t = 0$ is $\frac{d}{dt}\big|_{t=0} \gamma(t) = AXe^{tX}\big|_{t=0} = AX$. Therefore we are able to write the expression of curves starting at

point A and initial velocity AX.

This special class of curves is called *exponential curves* and can be denoted as

$$\gamma_A^X(t) := A\exp(tX)$$

This discussion allows us to give an equivalent definition for the commutator $[A, B]$ which has a more practical geometrical definition:

$$[A, B] = \lim_{t\to 0} \frac{1}{t^2} \ln(\gamma(t))$$

with

$$\gamma(t) = \exp(-tB)\exp(-tA)\exp(tB)\exp(tA)$$

Intuitively the commutator can be seen as *"the infinitesimal displacement obtained by following the path of $\gamma(t)$"*.

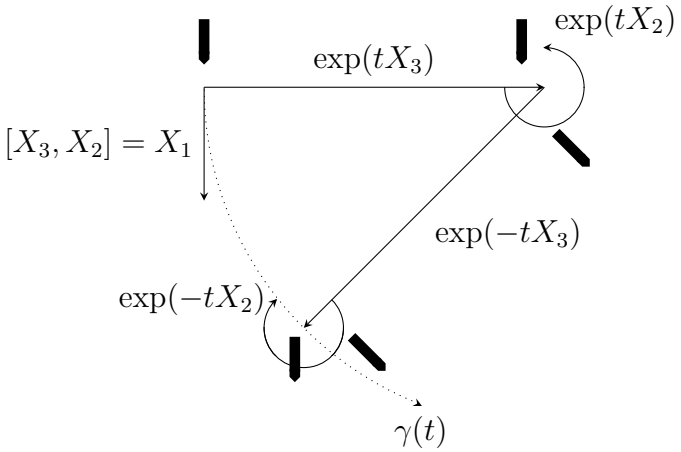


Figure 2.1: Example of displacement in $SE(2)$. The group structure and properties will be formally introduced in section 2.8. For now, assume this is a simplified model of a car with X_1 being forward movement, X_2 being left translation and X_3 being counter-clockwise rotation. By doing a left translation, a counter-clockwise rotation, a right translation of the same magnitude as the left one and a clockwise rotation of the same angle we define a path $\gamma(t)$ such that the resulting infinitesimal displacement is forward movement.

Such equivalent definition is due to the following lemma:

Lemma 22

Let G be a Lie group with Lie algebra $\mathfrak{g}$, and let exp be the exponential mapping of $\mathfrak{g}$ into into G. Then, if $X, Y \in \mathfrak{g}$

1. $\exp(tX)\exp(tY) = \exp\left(t(X+Y) + \frac{t^2}{2}[X,Y] + O(t^3)\right)$
2. $\exp(-tX)\exp(-tY)\exp(tX)\exp(tY) = \exp\left(t^2[X,Y] + O(t^3)\right)$
3. $\exp(tX)\exp(tY)\exp(-tX) = \exp\left(tY + t^2[X,Y] + O(t^3)\right)$

Ref: [23] at *Lemma 1.8* on pg. 106

2.4 Riemannian geometry

The branch of differential geometry that studies smooth manifolds endowed with a metric, i.e. an inner product on the tangent space, takes the name of Riemannian Geometry*. Originated by the work of Bernhard Riemann it is a generalization of geometry of surfaces embedded in* $\mathbb{R}^3$*. This branch of mathematics has been instrumental in the last two centuries to develop several applications, for instance the theory of general relativity.*

Definition 23 (Riemannian manifold)

A Riemannian manifold is a smooth manifold M together with a metric $g \in \mathcal{T}^2(M)$ that is symmetric and positive definite.

Within the scope of this work we assume all the manifolds to be connected. Since a smooth manifold is locally euclidean, it follows that it is also path-connected.

Example 24

The *Euclidean space* $\mathbb{R}^n$ is the simplest example of Riemannian manifold. Let $x^1, \ldots, x^n$ be the standard coordinates on $\mathbb{R}^n$. Then, in standard coordinates, a vector assumes form $\sum_i a_i \frac{\partial}{\partial x^i}$ and the space can be endowed with the metric

$$g\left(\sum_i a_i \frac{\partial}{\partial x^i}, \sum_i b_i \frac{\partial}{\partial x^i}\right) = \sum_i a_i b_i$$

The availability of a metric on the tangent bundle allows the computation of lengths of tangent vectors.

Let $\gamma(t)$ be a curve in M, with $\gamma(0) = p$ for a certain $p \in M$. The length of the tangent

vector $\dot{\gamma}(0) \in T_pM$ is given by

$$|\dot{\gamma}(0)| = \sqrt{g(\dot{\gamma}(0), \dot{\gamma}(0))}$$

With such definition we are able to introduce a distance functional between any two points in the manifold.

Definition 25 (Riemannian distance)
The Riemannian distance between any two points p, q in manifold M is given by

$$d(p,q) = \inf_{\gamma} \int_0^1 \sqrt{g(\dot{\gamma}(\tau), \dot{\gamma}(\tau))} d\tau$$

$$\gamma(0) = p$$

$$\gamma(1) = q$$

This definition is well defined when a manifold is connected, which we have assumed to be the case within the scope of this work. It could be easily extended by setting $d(p,q) = \infty$ when there exists no curve connecting two points $p, q \in M$.

Definition 26 (Geodesic)
The curve minimizing the Riemannian distance as measured by the metric g is called *geodesic*.

2.5 Sub-Riemannian geometry

It is not always possible to model a space in terms of Riemannian geometry. If one tries to model, for example, how a car moves on the 2D plane, one has to allow only a certain type of movements. A car cannot move translating laterally and therefore the tangent bundle does not reflect all the possible directions that a curve follows. This is exactly the type of situation in which the problem of Dubin's car [17] is set. For this reason the concept of a distribution*, a subset of the tangent bundle, is introduced. The model space for sub-Riemannian geometry is the* Heisemberg geometry*, with whom we are already familiar with, having introduced it a few pages ago and that will make a more detailed comeback later in this chapter.*

Definition 27 (Sub-Riemannian manifold)

A sub-Riemannian metric on M is a fiber metric defined on a linear subbundle $\mathcal{H} \subset TM$.

A sub-Riemannian manifold is a triplet $(M, \mathcal{H}, g)$ with M a manifold, $\mathcal{H} \subset TM$ a linear subbundle and $g = \langle \cdot, \cdot \rangle$ a sub-Riemannian metric.

We call $\mathcal{H} \subset TM$ in the previous definition the *horizontal distribution*.

The analogous of a curve in sub-Riemannian geometry is an *horizontal curve*, which differs from a smooth curve allowing $\dot{\gamma}(t)$ to only assume values in $\mathcal{H}$. $\gamma : [a, b] \to M$ is called *horizontal* if $\dot{\gamma}(t) \in \mathcal{H}_{\gamma(t)}$ for any $t \in [a, b]$. Analogously to smooth curves, we define the length of an horizontal curve as

$$L(\gamma) = \int_a^b \|\dot{\gamma}\| dt$$

which induces a distance (between points that can be connected)

$$d(x, y) = \inf_{\substack{\gamma \text{ horizontal} \\ \gamma(a)=x \\ \gamma(b)=y}} L(\gamma)$$

Example 28

Let $M = \mathbb{R}^3$ and $\mathcal{H} = \{\partial x, \partial y\}$ an orthonormal basis of elements of the standard basis.

$$\langle \partial x, \partial x \rangle = \langle \partial y, \partial y \rangle = 1$$

$$\langle \partial x, \partial y \rangle = 0$$

If we consider any curve γ in M, the condition on $\dot{\gamma} \in \mathcal{H}$ means that such curve cannot change $z-$coordinate along its path. The distance between two points becomes therefore:

$$d((x, y, z), (\tilde{x}, \tilde{y}, \tilde{z})) = \begin{cases} \sqrt{(x - \tilde{x})^2 + (y - \tilde{y})^2} & z = \tilde{z} \\ \infty & z \neq \tilde{z} \end{cases}$$

hence $\langle \cdot, \cdot \rangle$ is not induced by a sub-Riemannian metric.

Definition 29 (Bracket-generating distribution)

A distribution $\mathcal{H} \subset TM$ is called *bracket generating* if any local frame X_i for $\mathcal{H}$, together with all its iterated brackets $[X_i, X_j]$, $[X_i, [X_j, X_k]]$, ..., spans the whole tangent bundle TM.

A Bracket-generating distribution, also equivalently said to satisfy the *Hörmander condition*, is a crucial hypothesis in most of the sub-Riemannian spaces, due to the following theorem:

Theorem 30 (Chow–Rashevskii)
If $\mathcal{H} \subset TM$ is bracket-generating then the set of points connected to $p \in M$ by horizontal paths is the same as the connected component of M containing p. If M is path-connected then any pair of points in M can be connected by a horizontal curve and in particular for $p, q \in M$ it holds that $d(p, q) < \infty$ and $d(\cdot, \cdot)$ is a well defined metric distance.

Ref: [33] at *Thm. 1.17* on pg. 10

Recall that within the scope of this work all the manifolds are assumed to be connected, thus path-connected. This means that the Chow-Rashewskii theorem allows us to connect any two points in a bracket-generating sub-Riemannian manifold. This however does not tell us anything on the existence of geodesics, which requires a deeper discussion.

Theorem 31 (Local existence)
If M is a manifold with a bracket-generating distribution then any point $p \in M$ is contained in a neighborhood U such that p is horizontally connected to any $q \in U$ by a minimizing geodesic.

Ref: [33] at *Thm. 1.18* on pg. 10

Theorem 32 (Hopf-Rinow)
If M is a connected manifold with a bracket-generating distribution and M is complete relative to the sub-Riemannian distance function, then any two points $p, q \in M$ can be joined by a minimizing geodesic.

Ref: [33] at *Thm. 1.19* on pg. 10

2.6 Invariant sub-Riemannian structures on Lie groups

A natural question that one could ask themselves is if every Lie group admits a distribution and a metric such that it becomes a sub-Riemannian geometry. In this section we answer this question.

Let G be a Lie group and $\mathfrak{g}$ be its Lie algebra. Let $\mathfrak{h} \subset \mathfrak{g}$ be a Lie bracket generating subspace. Define a positive definite quadratic form $\langle \cdot, \cdot \rangle$ on $\mathfrak{h}$. A natural sub-Riemannian structure on G is given by:

- The distribution $\mathcal{H}$ is the left-invariant distribution defined as

$$\mathcal{H}(g) := g \cdot \mathfrak{h}$$

for $g \in G$.

- The metric on the distribution is given by

$$\mathbf{g}(v_1, v_2) := (g^{-1}v_1, g^{-1}v_2)$$

for $v_1, v_2 \in \mathcal{H}$, $g \in G$.

In this case we say that $(G, \mathcal{H}, \mathbf{g})$ is a *left-invariant sub-Riemannian manifold.*

2.7 The Heisenberg group H

We return now to the Heisenberg group H as a first example of sub-Riemannian geometry.

Recall that the left translation in H is defined as

$$L_{(\tilde{x},\tilde{y},\tilde{y})}(x, y, z) = (\tilde{x} + x, \tilde{y} + y, \tilde{z} + z + \frac{1}{2}(\tilde{x}y - x\tilde{y}))$$

We can compute locally its differential in the form of a Jacobian ([41] *Prop. 8.11*)

$$L_{(x,y,z),*} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{2}y & \frac{1}{2}x & 1 \end{bmatrix}$$

which yields a triplet of left invariant vector fields

$$X = \frac{\partial}{\partial x} - \frac{1}{2}y\frac{\partial}{\partial z}$$

$$Y = \frac{\partial}{\partial y} + \frac{1}{2}x\frac{\partial}{\partial z}$$

$$Z = \frac{\partial}{\partial z}$$

that form an orthonormal frame.

Let $\mathcal{H} = \{X, Y\}$. It is easy to check that $[X, Y] = Z$, $[X, Z] = 0$ and $[Y, Z] = 0$ thus $\mathcal{H}$ is a bracket-generating distribution.

We can write an explicit expression for a curve $\gamma(t) : [0, T] \to H$ as

$$\gamma(t) = \exp\left(\begin{bmatrix} 0 & x(t) & z(t) \\ 0 & 0 & y(t) \\ 0 & 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} 1 & a(t) & c(t) \\ 0 & 1 & b(t) \\ 0 & 0 & 1 \end{bmatrix}$$

for $x(t), y(t), z(t)$ curves in $\mathbb{R}$.

To define a sub-Riemannian distance set $ds^2 = dx^2 + dy^2$, and define the length of an horizontal path in $\mathbb{R}^3$ to be $\int_\gamma ds$, which corresponds to the usual length of its planar projection. The metric ds^2 can be restricted to the distribution $\mathcal{H}$ defining a family of smoothly varying inner products, so that for any $v, w \in \mathcal{H}_{(x,y,z)}$ the scalar product is $\langle v, w \rangle = v_1 w_1 + v_2 w_2$. If we endow the manifold with such family of inner products then the Heisenberg group is a non-trivial example of sub-Riemannian geometry.

In this setting geodesics in the Heisenberg group $\gamma(t) = (x(t), y(t), z(t))$ end up being circular spirals with the projection being an arc of a circle $c(t) = (x(t), y(t))$, and

$$z(t) = \frac{1}{2} \int_c x dy - y dx$$

Intuitively, the height of the curve is proportional to the area of the circle subtended by the circular arc projected.

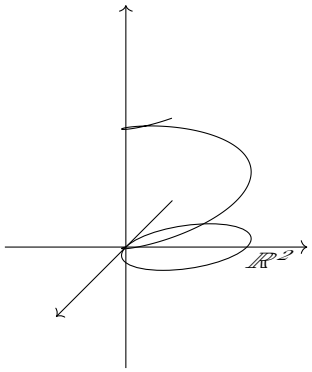


Figure 2.2: A geodesic curve in the Heisenberg group, starting at $(0, 0)$ with initial velocity vector $(1, 0)$.

2.8 The special Euclidean group $SE(2)$

Definition 33 (Special Euclidean group)

The *special Euclidean group* $SE(2)$ is a matrix group defined as

$$SE(2) = \left\{ \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \middle| x, y \in \mathbb{R},\ \theta \in [0, 2\pi) \right\}$$

We can also admit a matrix representation for the general case $SE(n)$

$$SE(n) = \left\{ \begin{bmatrix} A & \mathbf{a} \\ \mathbf{0}_n^\intercal & 1 \end{bmatrix} \middle| A \in SO(n),\ \mathbf{a} \in \mathbb{R}^n \right\}$$

The group $SE(2)$ represents all transformations on $\mathbb{R}$ which preserve distances, orientations and angles (rigid transformations). Any such transformation can be written as a rototranslation

$$T : \mathbb{R}^2 \to \mathbb{R}^2$$

$$T(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$$

for $A \in SO(2)$ a rotation and $\mathbf{b} \in \mathbb{R}^2$ a translation. It has therefore three degrees of freedom: two for translation and one for rotation.

Proposition 34

$SE(2)$ is isomorphic to $\mathbb{R}^2 \times S^1$

Proof. Let $\varphi : SE(2) \to \mathbb{R}^2 \times S^1$ such that

$$\begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix} \mapsto ((x, y), \theta)$$

which admits inverse

$$((x, y), \theta) \mapsto \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}$$

Both φ and its inverse are continuous componentwise (they are actually smooth), so they are continuous. Thus φ is a isomorphism. □

Since $S^1 \simeq \mathbb{R}/(2\pi\mathbb{R})$ we can model $SE(2)$ as $\mathbb{R}^3/(0, 0, 2\pi\mathbb{R})$ and intuitively see it as a torus with a "rectangular" base (figure 2.3).

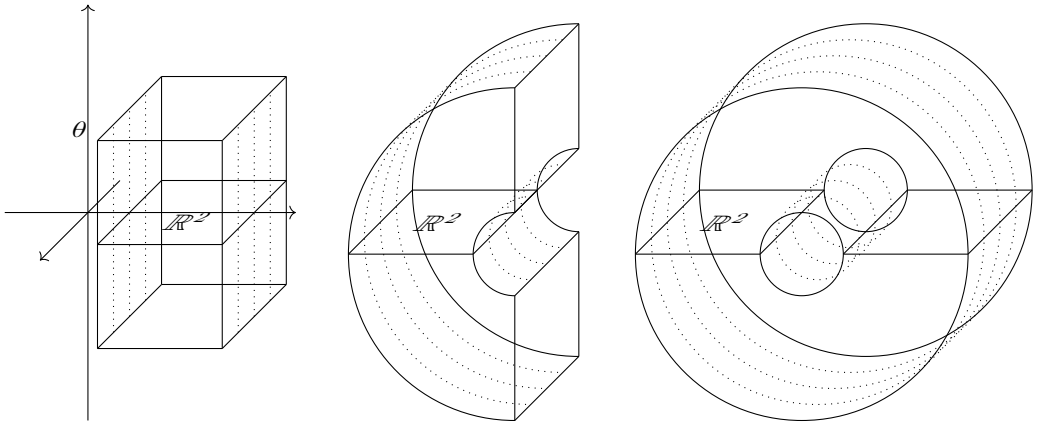


Figure 2.3: Intuitive visualization of the fibers of $SE(2)$ in a closed bounded rectangular region of $\mathbb{R}^2$. This visualization is particularly useful when dealing with the mathematical model of the visual cortex V1 presented in chapter 4.

Proposition 35

$SE(n)$ endowed with the matrix multiplication

$$\begin{bmatrix} A & \mathbf{a} \\ \mathbf{0}_n^\intercal & 1 \end{bmatrix} \cdot \begin{bmatrix} B & \mathbf{b} \\ \mathbf{0}_n^\intercal & 1 \end{bmatrix} = \begin{bmatrix} AB & A\mathbf{b} + \mathbf{a} \\ \mathbf{0}_n^\intercal & 1 \end{bmatrix}$$

where AB is the usual matrix multiplication, is a Lie group.

Proof. We treat the proof for the specific case $n = 2$, as this will be the setting we will be working with. This proof can be easily extended to any dimension n with the appropriate changes.

Let $X \in GL(3)$ so that

$$X = \begin{bmatrix} X_{11} & X_{11} & X_{12} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \end{bmatrix}$$

and consider the following maps $GL(3) \to \mathbb{R}$ and $GL(3) \to GL(2)$.

$$\begin{aligned} \varphi_1 &= X_{31} \\ \varphi_2 &= X_{32} \\ \varphi_3 &= X_{33} - 1 \\ \varphi_4 &= \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}^\intercal - \mathbf{I}_2 \end{aligned}$$

All these maps are C^∞, as matrix multiplication is smooth.

The kernels of these maps are regular submanifolds, according to the *regular value theorem* (Theorem 6) and the regular level set theorem

Theorem 36 (Regular level set theorem)
Let $F : N \to M$ be a C^∞ map of manifolds with $\dim N = n$ and $\dim M = m$. Then a nonempty regular level set $F^{-1}(c)$, where $c \in M$, is a regular submanifold of N of dimension equal to $n - m$.

Ref: [41] at *9.9* on pg. 105

Notice now how

$$\ker \varphi_1 \cap \ker \varphi_2 \cap \ker \varphi_3 \cap \ker \varphi_4 = SE(2)$$

and therefore by constructing the function

$$\begin{aligned} F : GL(n) &\to \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times GL(2) \\ X &\mapsto (\varphi_1(X), \varphi_2(X), \varphi_3(X), \varphi_4(X)) \end{aligned}$$

we can now use the *regular level set theorem* to say that $SE(2)$ is a regular submanifold of $GL(3)$ with dimension $9 - 1 - 1 - 1 - 4 = 3$. □

Proposition 37
The Lie algebra of $SE(n)$ is

$$\mathfrak{se}(n) = \left\{ \begin{bmatrix} X & \mathbf{x} \\ \mathbf{0}_n^\intercal & 0 \end{bmatrix} \middle| X \in \mathfrak{so}(n),\ \mathbf{x} \in \mathbb{R}^2 \right\}$$

where $\mathfrak{so}(n)$ is the subset of skew-symmetric square real matrices.

Proof. We know that there exists an isomorphism between $T_e(G)$ of a Lie group and the Lie algebra of the group. The objective is therefore to characterize such tangent space and prove it is isomorphic to $\mathfrak{so}(n)$.

Let $\tilde{A}(t)$ be a differentiable curve in $SE(2)$ starting at the identity defined on a neighborhood of $t = 0$. Then we can describe it in matrix form as

$$\tilde{A}(t) = \begin{pmatrix} A(t) & \mathbf{a}(t) \\ \mathbf{0}_n^\intercal & 1 \end{pmatrix}$$

for

$$A : (-\epsilon, \epsilon) \to SO(n)$$

$$\mathbf{a} : (-\epsilon, \epsilon) \to \mathbb{R}^n$$

with

$$A(0) = \mathbf{I}_n$$

$$\mathbf{a}(0) = \mathbf{0}_n$$

Differentiate now $\tilde{A}(t)$ to obtain

$$\frac{d}{dt}\tilde{A}(t) = \begin{pmatrix} \dot{A}(t) & \dot{\mathbf{a}}(t) \\ \mathbf{0}_n & 0 \end{pmatrix}$$

Consider now

$$\tilde{A}(t)^\intercal \cdot \tilde{A}(t) = \begin{pmatrix} A(t)^\intercal \cdot A(t) & A(t)^\intercal \cdot \mathbf{a}(t) \\ \mathbf{0}_n^\intercal & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & A(t)^\intercal \cdot \mathbf{a}(t) \\ \mathbf{0}_n^\intercal & 1 \end{pmatrix}$$

where the second equality follows by the fact that $A(t)$ lies in $SO(n)$. Evaluating the derivative of $\tilde{A}(t)^\intercal \cdot \tilde{A}(t)$ at $t = 0$ yields

$$\frac{d}{dt}(\tilde{A}(t)\cdot\tilde{A}(t))\big|_{t=0} = \begin{pmatrix} \frac{d}{dt}(A(t)^\intercal \cdot A(t))\big|_{t=0} & \frac{d}{dt}(A(t)^\intercal \dot{\mathbf{a}}(t))|_{t=0} \\ \mathbf{0}_n^\intercal & 1 \end{pmatrix}\Bigg|_{t=0} = \begin{pmatrix} \mathbf{0}_{n\times n} & \frac{d}{dt}(A(t)^\intercal \dot{\mathbf{a}}(t))|_{t=0} \\ \mathbf{0}_n^\intercal & 0 \end{pmatrix}$$

By looking at the top-left entry we can see that

$$\mathbf{0}_{n\times n} = \frac{d}{dt}(A(t)^\intercal\cdot A(t))\big|_{t=0} = \dot{A}(0)^\intercal A(0)+A(0)^\intercal \dot{A}(0) = \dot{A}(0)^\intercal \mathbf{I}_n+\mathbf{I}_n^\intercal \dot{A}(0) = \dot{A}(0)^\intercal+\dot{A}(0)$$

Hence $\dot{A}(0) = -\dot{A}(0)^\intercal$ and $\dot{A}(0) \in \mathfrak{so}(n)$. □

Proposition 38

A basis for the Lie algebra $\mathfrak{se}(2)$ is

$$p_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad p_2 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad p_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Proof. The proof is trivial knowing the result of proposition 37, as a basis for $SO(n)$ is

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

□

2.8.1 Sub-Riemannian structure on $SE(2)$

Define the following vector fields on $T\,SE(2)$:

$$\begin{aligned} X_1 &= \cos(\theta)\partial_x + \sin(\theta)\partial_y \\ X_2 &= \partial_\theta \\ X_3 &= -\sin(\theta)\partial_x + \cos(\theta)\partial_y \end{aligned}$$

and let $\vec{X}_1$, $\vec{X}_2$ and $\vec{X}_3$ be the sections associated to the vector fields X_1, X_2 and X_3 respectively. Then the Hörmander condition is satisfied, as shown in the next proposition

Proposition 39
$SE(2)$ with $T(SE(2)) = \text{span}\{X_1, X_2, X_3\}$ and $\mathcal{H} = \{X_1, X_2\}$ is bracket-generating.

Proof.

$$\begin{aligned} [X_2, X_1] = X_2X_1 - X_1X_2 &= \partial\theta(\cos\theta\partial x + \sin\theta\partial y) - (\cos\theta\partial x + \sin\theta\partial y)(\partial\theta) \\ &= -\sin\theta\partial x + \cos\theta\partial y = X_3 \end{aligned}$$

□

therefore the Chow-Rashevskii theorem (Theorem 30) holds and it is possible to connect any two points on $SE(2)$ through horizontal curves.

2.8.2 Integral curves and metric of $SE(2)$

The aim of this subsection is to show a way to form integral curves, parametric curves that are solution to an ODE, in order to connect tangent vectors in the case of specific Cauchy problems related to the field of perceptual completion.

Consider the following Cauchy problem

$$\begin{cases} \gamma'(t) = \vec{X}_1(\gamma(t)) + k\vec{X}_2(\gamma(t)) \\ \gamma(0) = (x_0, y_0, \theta_0) \end{cases}$$

where $k \in \mathbb{R}$ is fixed. The coefficient k expresses the curvature of the projection of the curve γ on the xy-plane [36].

Parametrize a solution $\gamma(t) = (x(t), y(t), \theta(t))$ (assuming for now it exists) and then by plugging in the definition of γ' in terms of X_1 and X_2 it is possible to obtain

$$x'(t) = \cos(\theta(t)) \qquad y'(t) = \sin(\theta(t)) \qquad \theta'(t) = k(t)$$

From the first two relations it follows that

$$\theta(t) = \arctan\left(\frac{y'(t)}{x(t)}\right)$$

Differentiating with respect to t (and dropping the parameter to lighten the notation) we obtain

$$k(t) = \theta'(t) = \frac{y''x' - x''y'}{(x')^2 + (y')^2}$$

which in the case of arc-length parametrization corresponds to the usual notion of curvature

$$K_\gamma = \frac{y''x' - x''y'}{((x')^2 + (y')^2)^{\frac{3}{2}}}$$

Up until now we have assumed that a solution for the Cauchy problem exists, but we can easily see it always does by providing a closed formula for any given fixed k

$$\gamma(t) = \exp(t(\vec{X}_1 + k\vec{X}_2))(x_0, y_0, \theta_0)$$

where exp is the exponential map for a Lie group which, in the case of a matrix Lie group, corresponds to the exponential of a matrix.

With the Euclidean metric we have that

$$\|X_1 + kX_2\| = \sqrt{1 + k^2}$$

so that the length of any curve γ can be expressed as

$$L(\gamma) = \int_a^b \|\gamma'(t)\| dt = \int_a^b \sqrt{1 + k(t)^2} dt$$

Chow-Rashevskii's theorem (Theorem 30) ensures that for every couple of points in $SE(2)$ there exists an horizontal curve γ which connects them. Consequently we set

$$d((x, y, \theta), (\bar{x}, \bar{y}, \bar{\theta})) = \inf_{\substack{\gamma \text{ horizontal} \\ \gamma(a)=x \\ \gamma(b)=y}} L(\gamma)$$

and define the ball of center $(\bar{x}, \bar{y}, \bar{\theta})$ and radius r in the classical way as

$$B((\bar{x}, \bar{y}, \bar{\theta}), r) = \{(x, y, \theta) : d((x, y, \theta), (\bar{x}, \bar{y}, \bar{\theta})) < r\}$$

2.8.3 Riemannian approximation of the metric

To extend the Euclidean norm to vectors outside the horizontal distribution we can define a new norm as the projection of the Euclidean one on the horizontal tangent space. For $v \in T_{(x,y,\theta)}(SE(2))$ endowed with standard basis $\partial_x, \partial_y, \partial_\theta$ we define

$$|v|_g^2 = \left\| \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \right\|_E^2 = \|(v_1 \cos\theta + v_2 \sin\theta, v_3)\|_E^2 =$$
$$= (v_1 \cos\theta + v_2 \sin\theta)^2 + v_3^2 = v_1^2 \cos^2\theta + v_2^2 \sin^2\theta + 2v_1 v_2 \cos\theta \sin\theta + v_3^2$$

and therefore

$$g^{ij} = \begin{pmatrix} \cos^2\theta & \sin\theta\cos\theta & 0 \\ \sin\theta\cos\theta & \sin^2\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which has zero determinant and is therefore not invertible to a metric g_{ij}.

If we add a viscosity term however, as suggested by [13], we obtain

$$g_\epsilon^{ij} = \begin{pmatrix} \cos^2\theta + \epsilon\sin^2\theta & (1-\epsilon^2)\sin\theta\cos\theta & 0 \\ (1-\epsilon^2)\sin\theta\cos\theta & \sin^2 + \epsilon^2\cos^2\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

which is now invertible for $\epsilon > 0$ and therefore g_ϵ^{ij} induces a norm on the cotangent space at every point as follows. If $w = (w_1, w_2, w_3) \in T^*_{x,y,\theta}(SE(2))$

$$|(w_1, w_2, w_3)| = (\cos(\theta)w_1 + \sin(\theta)w_3)^2 + \theta^2 + \frac{1}{\epsilon^2}(\sin(\theta)x - (\cos\theta)y)^2$$

Proposition 40

The geodesic distance d_ϵ associated to $g_{ij\epsilon}$ tends to the sub-Riemannian one as $\epsilon \to 0$.

Ref: [13] at *2.6* on pg. 316

2.8.4 Lift of a curve $\mathbb{R} \to SE(2)$

Consider a smooth planar curve $\gamma : [a, b] \to \mathbb{R}^2$ and $x, y : [a, b] \to \mathbb{R}$ s.t. $\gamma(t) = (x(t), y(t))$. Then we can lift the curve to $SE(2)$ setting the coordinates of the lifted curve $\bar{\gamma}$ to $(x(t), y(t), \theta(t))$ where $\theta(t) \in \mathbb{R}/(2\pi\mathbb{R})$ is the direction of the vector $(x(t), y(t))$ measured w.r.t. the euclidean vector $(1, 0)$ on $\mathbb{R}^2$. A closed form for $\theta(t)$ is

$$\theta(t) - \theta(0) = \begin{cases} \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \mod \pi & \dot{y}(t) \geq 0 \\ \left(\arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \mod \pi\right) + \pi & \dot{y}(t) < 0 \end{cases}$$

2.8.5 The projective tangent bundle $PT\mathbb{R}^2$

Another interesting example of sub-Riemannian geometry, which arises naturally as an extension of $SE(2)$ identifying two orientations with same direction as the same orientation, is $PT\mathbb{R}^2 := \mathbb{R}^2 \times P^1$. Since $P^1 = S^1/\mathbb{Z}_2$, $PT\mathbb{R}^2$ can be seen as the quotient of the group of rototranslations of the plane $SE(2) \simeq \mathbb{R}^2 \times S^1$ by $\mathbb{Z}_2$. The geometric properties and sub-Riemannian structure on $PT\mathbb{R}^2$ are analogous to the ones on $SE(2)$.

If one wants to define explicitly the manifold structure, one can do it by using two charts:

- Chart A: $\theta \in (0 + k\pi, \pi + k\pi)$, $k \in \mathbb{Z}$, $x, y \in \mathbb{R}$

$$\dot{q} = u_1^A(t) X_1^A(q) + u_2(t) X_2(q), \quad X_1^A = \begin{pmatrix} \cos(\theta) \\ \sin\theta \\ 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- Chart B: $\theta \in (-\pi/2 + k\pi, \pi/2 + k\pi)$, $k \in \mathbb{Z}$, $x, y \in \mathbb{R}$

$$\dot{q} = u_1^B(t) X_1^B(q) + u_2(t) X_2(q), \quad X_1^B = \begin{pmatrix} \cos(\theta) \\ \sin\theta \\ 0 \end{pmatrix}, \quad X_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

One could argue that the formal expression of X_1^A and X_1^B is the same, but we need to be careful as the definition on different domains means that there is a change of sign when passing from Chart A to Chart B (and vice-versa) in $\mathbb{R}^2 \times \pi/2$ and $\mathbb{R}^2 \times \pi$.

Remark: The lift of a curve from $\mathbb{R}^2$ to $PT\mathbb{R}^2$ is analogous to the case in $SE(2)$ but with a simpler expression for the angle θ, as we do not distinguish anymore by orientation. A

closed form for $\theta(t)$ is

$$\theta(t) - \theta(0) = \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \mod \pi$$

2.9 The problem of completing curves

The problem of completing curves in $\mathbb{R}^2$ that have been partially hidden or corrupted by minimizing a functional depending on length and curvature is one application of the sub-Riemannian spaces $SE(2)$ and $PT\mathbb{R}^2$.

The inspiration for this application is Dubin's car, a car that can only move forward or rotate with a constraint on the curvature. Suppose we are driving this car, with the aim of traveling between two points in the 2D plane given initial and terminal velocities (or directions), trying to follow the shortest path satisfying the constraints.

In our setting the formal definition of the problem assumes this form: let $\gamma_0 : [a, b] \cup [c, d] \to \mathbb{R}^2$ (with $a < b < c < d$) be a smooth curve that is partially hidden in the interval $t \in (b, c)$. We want to find a curve $\gamma : [b, c] \to \mathbb{R}^2$ that completes γ_0 minimizing a cost $J[\gamma]$.

When dealing with the objective of "completing a curve" we generally require that $\gamma(b) = \gamma_0(b)$, $\gamma(c) = \gamma_0(c)$ and, depending on the setting, either $\dot{\gamma}(b) \sim \dot{\gamma}_0(b)$, $\dot{\gamma}(c) \sim \dot{\gamma}_0(c)$ if we only consider directions regardless of orientation or $\dot{\gamma}(b) \approx \dot{\gamma}_0(b)$, $\dot{\gamma}(c) \approx \dot{\gamma}_0(c)$ if we consider also orientation. Recall that $v_1 \sim v_2$ if there exists $\alpha \in \mathbb{R}^+$ such that $v_1 = \alpha v_2$, and that $v_1 \approx v_2$ if there exists $\alpha \in \mathbb{R}\backslash\{0\}$ such that $v_1 = \alpha v_2$.

This problem has been extensively studied for its applications to the fields of image segmentation and reconstruction of spiral lines, with different definitions of the functional to minimize. A few examples can be:

$$E_1[\gamma] = \int_b^c |K_\gamma(s)|^2 ds$$

$$E_2[\gamma] = \int_b^c (1 + |K_\gamma(s)|^2) ds$$

$$E_3[\gamma] = \int_b^c (\eta + |K_\gamma(s)|^2) ds$$

$$E_4[\gamma] = \int_b^c \sqrt{1 + |K_\gamma(s)|^2} ds$$

where $K_\gamma(t) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}$.

Of particular interest for this discussion is however the cost

$$J[\gamma] = \int_b^c \sqrt{\|\dot{\gamma}(t)\|^2 + \|\dot{\gamma}(t)\|^2 K_\gamma^2(t)} dt$$

which is an extension of $E_4[\gamma]$ and arises naturally in problems of geometry of vision. Particularly, it is invariant under rototranslation and reparametrization of the curve. $J[\gamma]$ is well defined on the set

$$\mathcal{D}_1 = \left\{ \gamma \in C^2([b,c],\mathbb{R}^2) \middle| \dot{\gamma}(t) \neq 0 \ \ \forall t \in [b,c], \begin{matrix} \gamma(b) = \gamma_0(b), \gamma(c) = \gamma_0(c) \\ \dot{\gamma}(b) \sim \dot{\gamma}_0(b), \dot{\gamma}(c) \sim \dot{\gamma}_0(c) \end{matrix} \right\}$$

and, by arclength reparametrization, it coincides with $E_4[\gamma]$.

If we extend $\mathcal{D}_1$ to

$$\mathcal{D}_2 = \left\{ \gamma \in C^2([b,c],\mathbb{R}^2) \middle| \|\dot{\gamma}\|^2 + \|\dot{\gamma}\|^2 K_\gamma^2 \in L^1([b,c],\mathbb{R}), \begin{matrix} \gamma(b) = \gamma_0(b), \gamma(c) = \gamma_0(c) \\ \dot{\gamma}(b) \sim \dot{\gamma}_0(b), \dot{\gamma}(c) \sim \dot{\gamma}_0(c) \end{matrix} \right\}$$

then the two functionals do not coincide anymore as γ can't be always reparametrized by arclength over the set $\mathcal{D}_2$.

Remark: $\mathcal{D}_1 \subset \mathcal{D}_2$.

Proposition 41

There exist boundary conditions $\gamma_0(b), \gamma_0(c) \in \mathbb{R}^2$ with $\gamma_0(b) \neq \gamma_0(c)$ and $\dot{\gamma}_0(b), \dot{\gamma}_0(c) \in \mathbb{R}^2 \backslash \{0\}$ such that $J[\gamma]$ does not admit a minimum over neither $\mathcal{D}_1$ nor $\mathcal{D}_2$.

Ref: [7] at *Prop. 2* on pg. 44

The basic and intuitive problem is that we may have a sequence of admissible minimizing curves converging to a non-admissible curve. For an example over $\mathbb{R}^2$ see figure 2.4, while for the actual discussion on how to construct such sequence for $\mathcal{D}_1$ and $\mathcal{D}_2$ see [7].

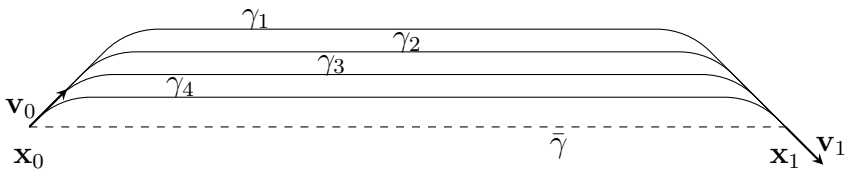


Figure 2.4: Example of absence of minimizers over the smooth curves with fixed initial and ending point and velocity in $\mathbb{R}^2$ endowed with euclidean metric and usual euclidean curve length as cost. There is a sequence of admissible curve with strictly decreasing cost converging to a non-admissible curve.

Changing the boundary conditions to *projective boundary conditions* intruduces a third space of admissible curves

$$\mathcal{D}_3 = \left\{ \gamma \in C^2([b,c],\mathbb{R}^2) \middle| \|\dot\gamma\|^2 + \|\dot\gamma\|^2 K_\gamma^2 \in L^1([b,c],\mathbb{R}),\ \begin{matrix} \gamma(b) = \gamma_0(b), \gamma(c) = \gamma_0(c) \\ \dot\gamma(b) \approx \dot\gamma_0(b), \dot\gamma(c) \approx \dot\gamma_0(c) \end{matrix} \right\}$$

on which $J[\gamma]$ always admits a minimizer.

Proposition 42

For all boundary conditions $\gamma_0(b), \gamma_0(c) \in \mathbb{R}^2$ with $\gamma_0(b) \neq \gamma_0(c)$ and $\dot\gamma_0(b), \dot\gamma_0(c) \in \mathbb{R}^2\backslash\{0\}$ the cost $J[\gamma]$ has a minimizer over the set $\mathcal{D}_3$.

Ref: [7] at *Prop. 3* on pg. 45

Finally, notice that all the results of this section can also be applied to the cost

$$J_\beta[\gamma] = \int_b^c \sqrt{\|\dot\gamma(t)\|^2 + \beta^2 \|\dot\gamma(t)\|^2 K_\gamma^2(t)}dt$$

as considering the homothety $(x,y) \mapsto (\beta x, \beta y)$ which maps γ to γ_β yields $J_\beta[\gamma_\beta] = \beta^2 J[\gamma]$.

Chapter 3

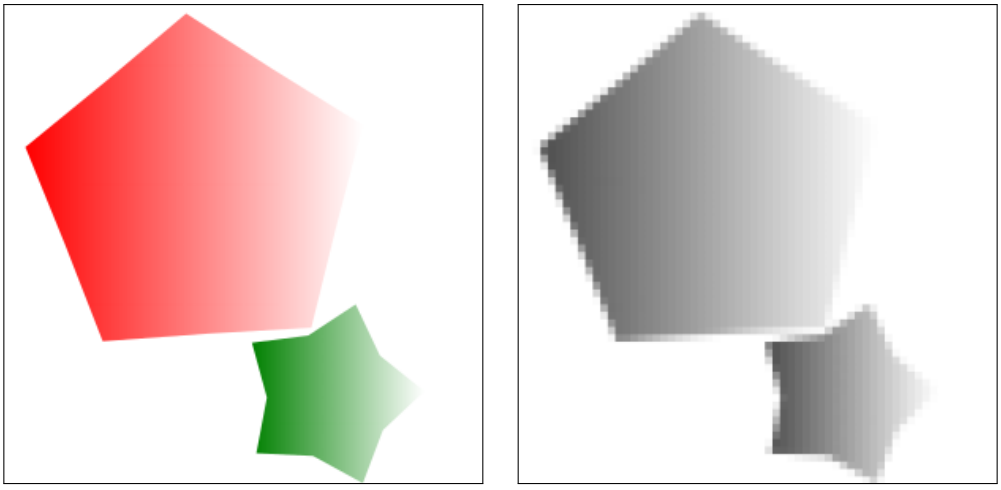
Digital Image Processing

The focus of this chapter is to present an introduction to the field of Digital Image Processing, stating the main concepts and results, which constitute the background material to discuss the applications of the following chapters. A more in-depth discussion can be found in [21].

3.1 Image sampling and digital images

We call *greyscale images* real functions of two variables $f : \mathbb{R}^2 \to \mathbb{R}$, where the scalar value at a specific coordinate (x, y) has the physical meaning of brightness that reaches a certain spatial point on the sensing device. Such device could be the retina inside a human eye, the film of a camera or a sensor array in a digital camera.

In the context of digital computers and digital acquisition devices it makes sense to introduce a bound on the codomain and allow only values ranging from 0 (absence of light, black) to 1 (maximum amount of light, white). Let therefore $f : \mathbb{R}^2 \to [0, 1]$ be a continuous function of two real variables. A *digital image* is obtained from f by *sampling* and *quantization*. Discretizing the coordinate values (domain) is called *sampling*, while discretizing the amplitude values (codomain) is called *quantization*. For such conversion a number of M rows and N columns are chosen.



(a) Original color image, as perceived by the human retina

(b) Sampled and digitized image, with a spatial resolution of 64x64 pixel and a quantization resolution of 256 colors (shades of gray)

Figure 3.1: An example of an image as perceived from the eye retina (left) being digitized by a camera sensor to produce a digital grayscale image (right)

The reason why such representation is necessary is tied to the fact that images are digitally sampled by sensors with a finite resolution and require a finite amount of data storage in a digital mass storage device.

For notational clarity and convenience $x = 0, 1, 2, ..., M-1$ and $y = 0, 1, 2, ..., N-1$ are used to refer to the discrete coordinates, horizontal and vertical respectively. In addition the notation for a digital image $f(x, y)$ is dropped, in favor of the notation $I(x, y)$ which is more common in the field. With such notation we can represent a digital image as a 2D matrix, as follows:

$$I(x,y) = \begin{bmatrix} I(0,0) & I(0,1) & \dots & I(0,N-1) \\ I(1,0) & I(1,1) & \dots & I(1,N-1) \\ \vdots & \vdots & & \vdots \\ I(M-1,0) & I(M-1,1) & \dots & I(M-1,N-1) \end{bmatrix}$$

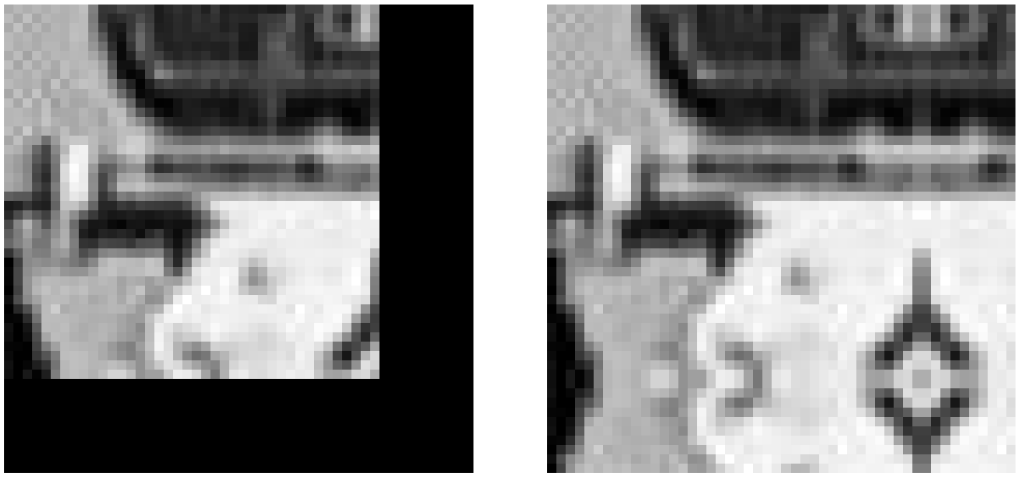
Each element of the matrix takes the name of an *image element*, *picture element*, *pixel* or *pel*. Historically a rectangular mesh of squared pixels has emerged as the canonical way of representing and storing digital images, although more exotic ways to define a different pixel geometry have been proposed and implemented (for example, exagonal pixels [5]).

3.2 Image filtering

Spatial filtering is a technique used in a broad variety of applications in image processing. Spatial filtering acts replacing, for every pixel (x, y), the value of the pixel itself with one determined by a function defined in a neighborhood of (x, y).

In the pixel representation that we have just introduced, a digital image is always spatially bounded due to computational and storage constraints. It has therefore to be finite thus having a boundary ∂R. Let I be a digital image with domain $R \subset \mathbb{R}^2$, then for every pixel $p \in R$ we can define a function on a neighborhood of p, $f_p : R \supset N_p \to [0, 1]$ so that $I_f(p) = f_p(N_p)$ is a *filtered image*, where f_p is the *filter* at p. The function is defined on a finite neighborhood under the assumption that the intensities of the pixels are dependent only locally on the image.

Formally we need to define a filter f_p at every point as the definition of the neighborhood N_p depends on the vicinity of the pixel we are evaluating to the edge of the image. For example we cannot define N_p simply as the ball centered on the pixel p with a radius $r > 0$, as we would incur in a technical issue on the edge pixels. Usually this issue is addressed by extending the domain of the image by a few rows/columns of either constant values pixels or by mirroring the image close to the border using the edge as symmetry axis. Therefore in all following occurrences we will drop the notation f_p in favor of a lighter f assuming one of these techniques is used.



(a) Image preprocessed to have constant border (with value 0)

(b) Image preprocessed to have mirrored border

Figure 3.2: An example of the bottom-right corner of a digital image that is preprocessed before filtering adding a few rows and columns of pixels

If the filter f is a linear function then it is called a *linear spatial filter*, otherwise it takes

the name of *nonlinear spatial filter*.

A *linear spatial filter* performs a weighted sum of the pixels of the image. Usually, given the coordinates of a pixel (x, y), it makes sense for practical reasons to consider only a square neighborhood with odd resolution centered on the pixel as a filter domain.

A kernel w is a square (sometimes rectangular) matrix that is convoluted with the image during the filtering process. Mathematically, in the discrete case of digital image processing, this operation takes the form

$$I_w(x, y) = w * I(x, y) = \sum_{i=-a}^{a} \sum_{j=-b}^{b} w(i, j) I(x + i, y + j)$$

where the matrix has width $2a + 1$ and height $2b + 1$. I_w is the filtered image.

Some linear filters can be seen in action in Figure 3.3. Examples of non-linear filters are the max-filter, the min-filter and the median-filter, where the value of a pixel is substituted respectively with the value of the max, min or median in a neighborhood.

3.2.1 Gabor filters

A special class of linear filters commonly used for texture analysis, which are dependent on orientations, are the Gabor filters. They are defined as

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{\tilde{x}^2 + \gamma^2 \tilde{y}^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{\tilde{x}}{\lambda} + \psi\right)\right)$$

where, with euclidean scalar product

$$\tilde{x} = x \cos\theta + y \sin\theta = \langle x, y \rangle \cdot \langle \cos\theta, \sin\theta \rangle$$

$$\tilde{y} = -x \sin\theta + y \cos\theta = \langle x, y \rangle \cdot \langle -\sin\theta, \cos\theta \rangle$$

The result of applying these filters is an analysis that studies whether in a certain region of a digital image there is a specific frequency content, with a particular orientation. In the spatial domain a 2D-Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave with a certain orientation θ. In the expression of the filter λ represents the wavelength of the sinusoidal factor, θ represents the orientation, ψ is the phase offset, σ is the standard deviation of the Gaussian envelope and γ is the spatial aspect ratio.

These filters are widely used in the field for their versatility in detecting texture and for their similarity to how the visual cortex V1 of the brain detects the edges and the orientations in an image, as we will see in Chapter 4.

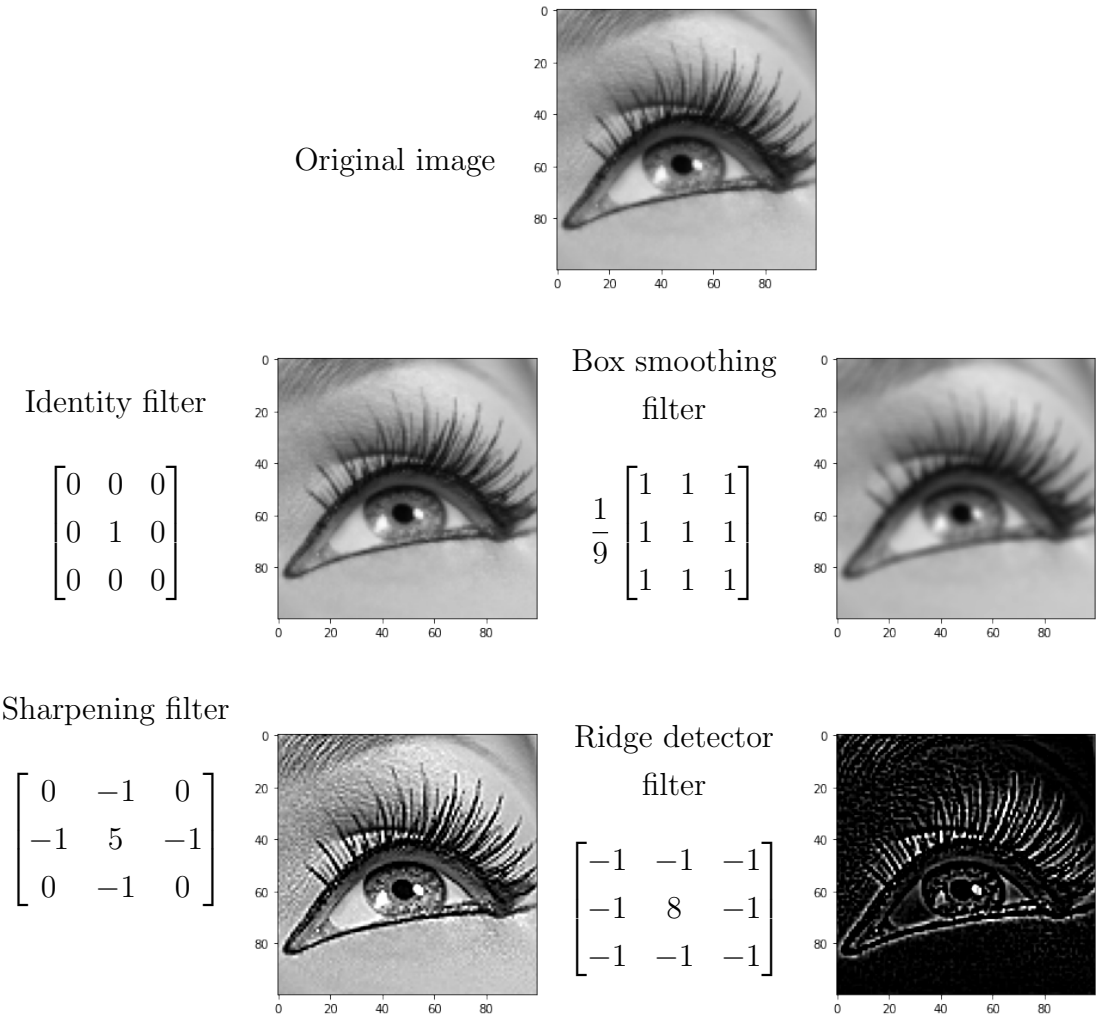


Figure 3.3: Examples of some common convolution kernels used in digital image processing, which produce smoothing, sharpening or ridge detection for image segmentation.

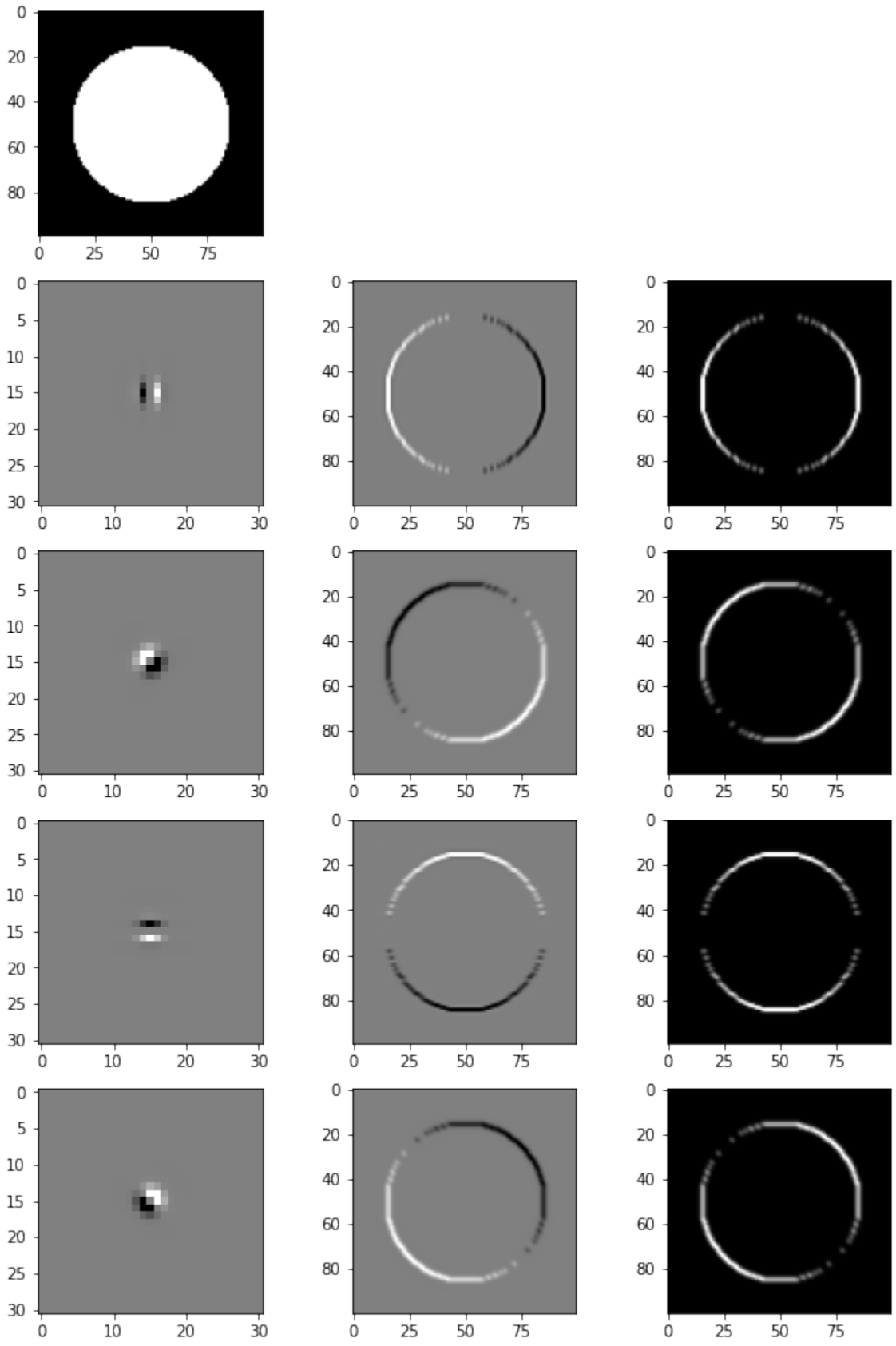


Figure 3.4: Examples of Gabor filters used to detect oriented borders. In the first line a 100x100 input image. In the following lines, in the first column the Gabor kernels visualized, with a kernel size of 30, $\sigma = 1$, $\lambda = \pi/4$, $\gamma = 1$, $\psi = \pi/2$ and respectively $\theta = 0$, $\pi/4$, $\pi/2$, $3\pi/4$. In the second column the result of the convolution (values from -1 to 1) and in the third column the same image taken in absolute value (from 0 to 1).

3.3 Noise

An image may experience a certain corruption that varies the intensity value at some pixels, which can happen for different reasons (discratization, sensor noise, storage problems...) and in different ways. This corruption takes often the name of *noise*, and it is an undesirable by-product of image capture and handling that alters a digital image in some way. In this section a few types of noise are discussed.

3.3.1 Gaussian noise

Noise may be introduced during the acquisition phase, when the image is being sampled by the sensor, and processed by the camera circuitry. Any source of photons, including the radiation effect due to the heat of the camera itself, is a source of Gaussian noise. The electronic circuitry then injects its own share of circuit noise while handling the sensor information.

A typical model of Gaussian noise is additive and independent of position and value of the pixel. It is possible to write it as

$$I_C(x, y) = I(x, y) + G(x, y)$$

where I_C is the corrupted image, I is the original image and $G : \mathbb{R}^2 \to \mathbb{R}_+$ is the additive noise. Since the physical sensor can provide a limited span of values, usually the corrupted image is capped by a value M, so that a more concrete expression takes the form

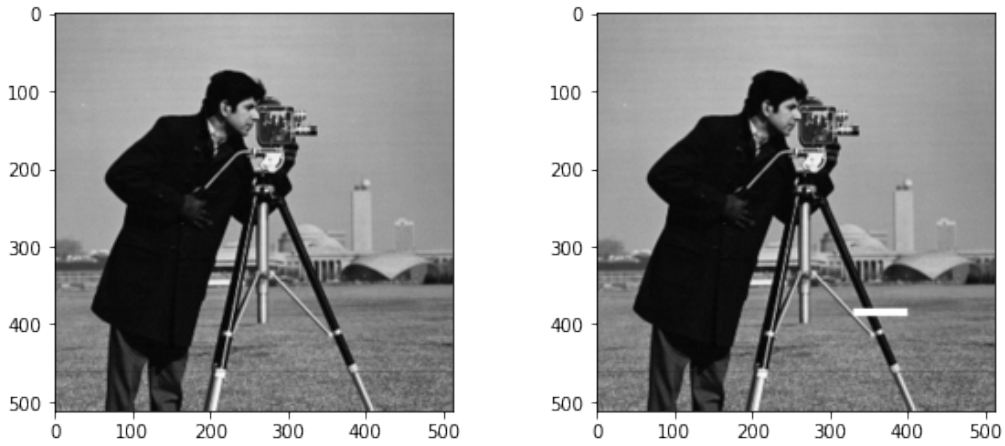
$$I_C(x, y) = \min(I(x, y) + G(x, y), M)$$

3.3.2 Salt and pepper noise

Salt and pepper noise is a form on noise that presents itself as sparsely occurring white and black pixels. The amount of noise can be easily quantified by percentage with respect to the total number of pixels of a digital image, and the probability of a pixel being white or black is a uniform 50-50, but does not depend on the position. An example can be found in Figure 3.6.

3.3.3 Missing information

This form of noise can happen during acquisition in presence of physical obstructions in front of the sensor or when a portion of the sensor is damaged. Also during handling and processing, especially in the case of physical images, it can happen that a portion of the image is ruined. This type of noise is very common, for example, in old film-based cinematography, where film frames can be damaged by the equipment and white vertical lines start to appear in the frames. When the information regarding the position of the noise is present, then it is flagged by setting those pixels at their minimum or maximum value.



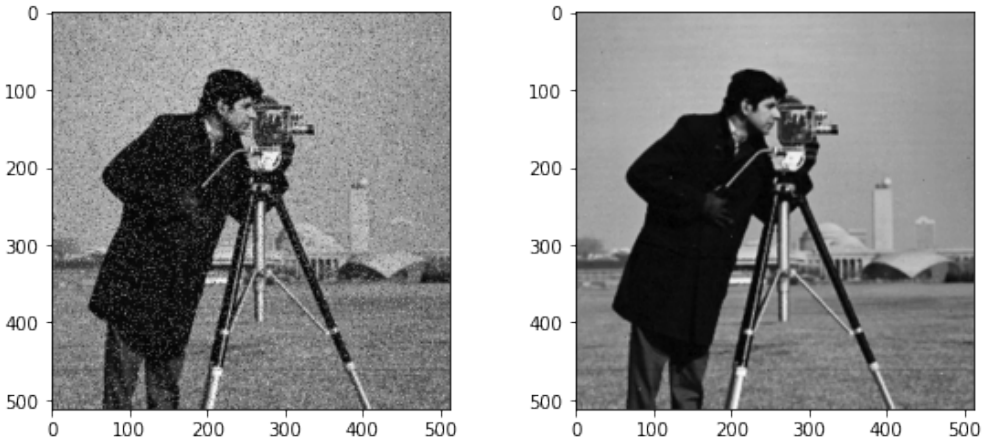
(a) Original image (b) The image after corruption

Figure 3.5: An image has been corrupted in a rectangular region, so that all the information in such region has been lost.

3.4 Image restoration

Image restoration is a process that seeks to recover an image that has been corrupted in some way, according to a certain type of noise.

The most common and effective techniques used to correct noise are linear and non-linear convolutions. For example, Gaussian noise is easily reduced with a Gaussian smoothing kernel. Salt and pepper noise is usually corrected with a median filter. Some types of noise such as motion blurring, Moiré patterns (due to Nyquist sampling theorem [21]) or missing information are more challenging to handle.



(a) Corrupted image with salt and pepper noise

(b) Restored image with a 3x3 median kernel

Figure 3.6: Example of a restoration process using a median filter

More advanced techniques currently used for image inpainting, that seek to recover missing information, can be roughly divided in three families of methods: *patch-based substitution*, *PDE methods* and *machine learning models* (more recently in the form of *deep neural networks*).

Patch-based substitution aims to fill the corrupted areas with connected patches of pixels extracted from the uncorrupted region of the image, in order to produce a restoration that preserves the original texture.

PDE methods aim to solve a partial differential equation with the corrupted image as the initial conditions in order to achieve reconstruction. This is usually done with an equation dependent on time, and iteratively letting the image be altered by the evolution step of the PDE until the optimal result is achieved.

Deep neural networks are machine learning models inspired by the neural connections in the brain, that aim to learn from a dataset of features given the corresponding labels by adjusting how the neurons communicate with each others.

In patch-based inpainting a necessary requirement is to know which parts of the image are corrupted and which are not. PDE and machine learning methods however can be constructed in a way as to not require previous knowledge on the position of the corruption, and therefore can be more suitable for automated applications where the nature or position of the corruption in unknown.

The study of patch-based methods and neural networks, and more broadly the field of machine learning, is however out of the scope of this discussion and will therefore not

be introduced in any more detail. In Chapter 4 we will discuss a particular PDE-based restoration method, while in Chapter 5 a novel algorithm inspired by such method is presented.

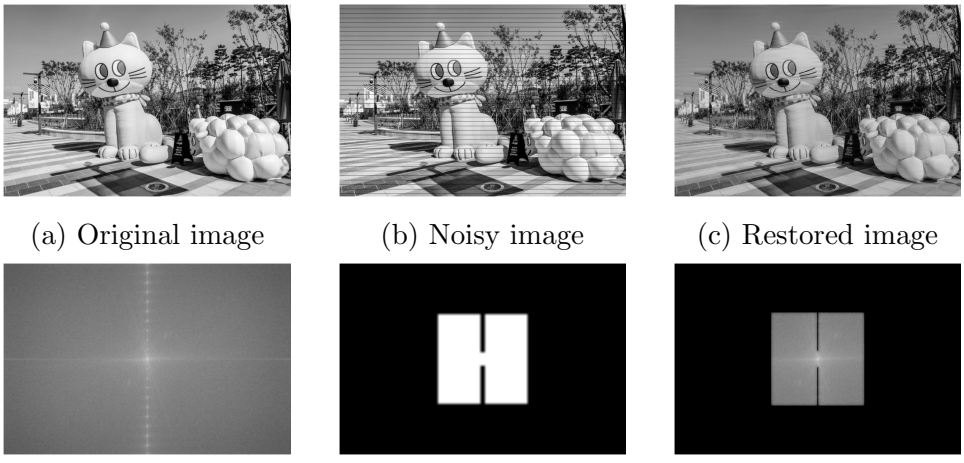
3.5 Frequency domain filtering

If M and N are respectively the number of rows and columns of an image, m and n the number of rows and columns of a filter, it takes $MNmn$ operations to apply kernel convolution to an image. If one uses the Fourier transform to move the operation to frequency domain, where convolution becomes multiplication, the computational cost (including Fourier transform and inverse) becomes $2MN\log_2(MN)$, thanks to the Fast Fourier Transform algorithm.

Theorem 43 (Convolution theorem)
Let $g(\mathbf{x})$ and $h(\mathbf{x})$ be two functions with Fourier transforms $G = \mathcal{F}[g]$, $H = \mathcal{F}[h]$ then

$$g * h = \mathcal{F}^{-1}[G \cdot H]$$

One could also want to directly filter in frequency domain by, for example, excluding a certain class of frequencies. Consider the following example of Moiré noise, in the form of repeated black horizontal lines, filtered with a frequency domain technique to exclude the exact frequencies of the horizontal lines.



(a) Original image (b) Noisy image (c) Restored image

(d) Spectrum of the original image (e) Mask in frequency domain (f) Spectrum of restored image

Figure 3.7: Example of reconstruction in frequency domain. Source image (a) courtesy of [15].

Chapter 4

The current work on PDE-based image restoration

4.1 The primary visual cortex V1

A branch of the field of image restoration seeks to achieve an optimal result drawing inspiration from the biological model that drives human perception. Visual information is processed in the brain by the *visual cortex* in the occipital lobe, which is located in the back of the brain. One of the most studied visual areas in the visual cortex is the *primary visual cortex V1*, which is highly specialized in processing information about pattern recognition. From basic neurology we know that a neuron is either active or inactive. When it gets sufficiently stimulated by an external input (for example a visual input) it *spikes*, which is the action of releasing neurotransmitters from its synapses to connected neurons. If a connected neuron gets sufficiently stimulated, above a certain threshold summing all its inputs, then it spikes itself. The result of this whole operation is the thinking process.

There exist two types of synapses: *excitatory* and *inhibitory*. The excitatory synapses promotes spiking in the connected neurons whereas the inhibitory ones prevent it [37]. Upon studying the visual cortex V1 one finds that the neurons are arranged in cells with elongated receptive fields, which exhibit even or odd symmetric patterns similar to Gabor filters [32]. In a simplified model the neurons inside V1 are grouped into *orientation columns*, each being sensitive to stimuli at a specific point of the retina, corresponding to the spatial coordinate on the field of view, and a specific orientation. Orientation columns are themselves grouped together into *hypercolumns* that are sensitive to stimuli in a certain position of the retina, regardless of the orientation. Orientation columns

are connected between themselves in two different ways: *vertical* (inhibitory) synapses and *horizontal* (excitatory) synapses. The vertical connections happen between columns sensitive to similar directions belonging to the same hypercolumn, whereas horizontal ones happen between columns belonging to different hypercolumns that are however spatially close.

The interpretation is that the human brain tries to "fill-in" the gaps that preserve a similar orientation, while making more pronounced the contrast between different directions. A visual example the reader might relate to can be observed in Figure 4.1 by psychologist Gaetano Kanisza.

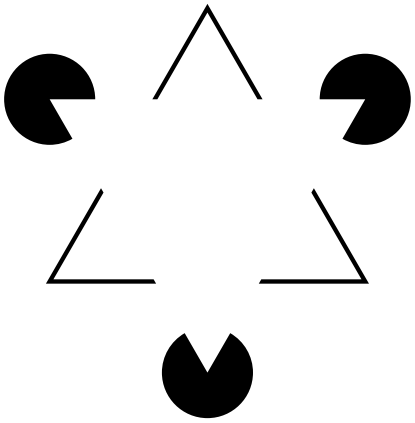


Figure 4.1: Kanisza's triangle, an example of illusory contour that stimulates an high response of the cortex V1

The task of mathematically modeling the cortex behavior aforementioned was introduced by Petitot et al. [36] and [35].

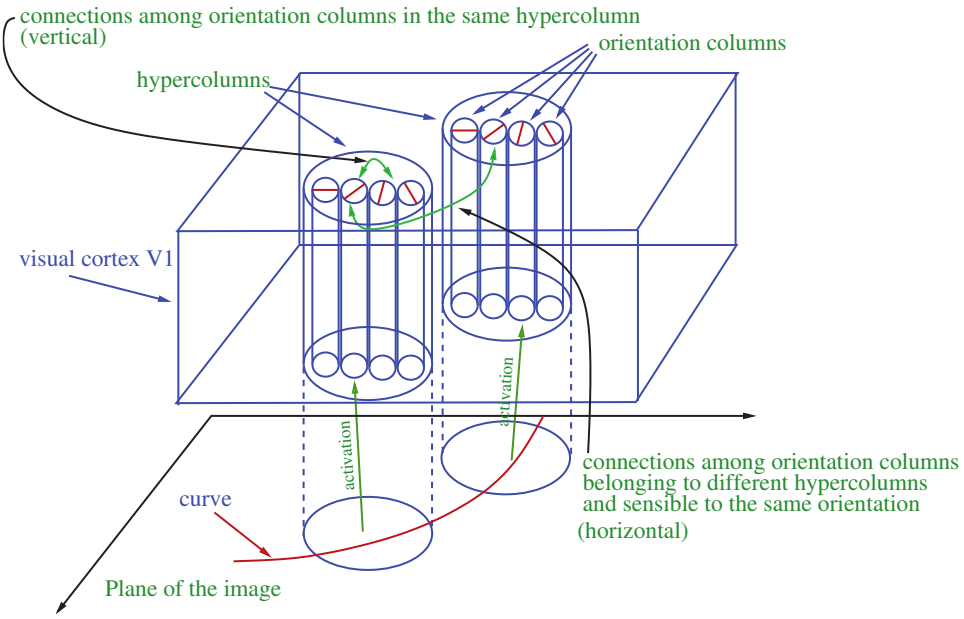


Figure 4.2: A mathematical scheme of the primary visual cortex V1 while "observing" a curve, courtesy of Boscain et al. [8]

Neurologists tell us that in the same hypercolumn are also present neurons that are sensitive to other stimuli properties like colors, displacement directions (motion) and many others. In this simplified model we will only focus on orientations.

4.2 The CPS model

The Citti-Petitot-Sarti restoration model, published in 2006, is a milestone in exploiting the mathematical formulation of the visual cortex V1. It makes no assumption on the position of the corruption, so that it is a "blind algorithm" that can be applied to any visual sample. It assumes however that an image is grayscale, with values between 0 and 1, where 0 is regarded as white and 1 as black. Corruption is represented as an area of constant value 0 (white). The choice of having the intensity values inverted with respect to what it is usually the norm in the field of image processing is due to the fact that commonly the details on a picture, which are the high frequency information, are black rather than white. Additionally it is assumed that the image function $I : \mathbb{R}^2 \to [0, 1]$ is smooth and with no degenerate critical points (which is not a tricky requirement, as we will see as a corollary to a result by Boscain et al. [8] introduced later in this chapter).

In this model the visual cortex V1 is modeled by $SE(2) \simeq \mathbb{R}^2 \times S^1$, where every hypercolumn located on the $\mathbb{R}^2$ euclidean plane corresponds to a fiber S^1. In this representation the retina is $\mathbb{R}^2$. The CPS model takes inspiration from the Gabor filters

(see section 3.2.1) modeling each orientation column as one. It follows the assumption that the visual cortex V1 acts in a similar way to a stack of Gabor filters. This assumption was first formulated by Marcelja [32], and was later empirically confirmed by Jones and Palmer [27].

4.2.1 Lifting procedure

We have discussed in Chapter 2 how to lift a single curve from $\mathbb{R}^2$ to $SE(2)$. The procedure to lift an image is different and, although there appears to be a preferred procedure (preferred both for simplicity and computational complexity), it is not unique nor canonical.

The most common lifting procedure links every point to their counterpart in $SE(2)$ according to the direction of maximum response of the simple cells, and filters out spurious directions. This is done by computing the gradient of the image in order to know the orientation of the level curves of the function $I : \mathbb{R}^2 \to [0, 1]$ at every point $(x, y) \in \mathbb{R}^2$.

Recall that for an image $I : \mathbb{R}^2 \supset R \to [0, 1]$ the directional derivative at an angle θ, $X_3(\theta)I$, takes the form

$$X_3(\theta)I = -\sin(\theta)\partial_x I + \cos(\theta)\partial_y I$$

which gives the projection of the gradient in the direction of the vector $(-\sin\theta, \cos\theta)$. The maximum is achieved when $(-\sin\theta, \cos\theta)$ is the direction of the gradient. Denote by $\bar{\theta}$ such angle

$$|X_3(\bar{\theta})| = \max_{\theta} |X_3(\theta)|$$

Since the gradient is perpendicular to the level curves of the image, it follows that $\bar{\theta}$ is the direction of the level curves, as $(\cos\theta, \sin\theta)$ is perpendicular to $(-\sin\theta, \cos\theta)$.

Citti and Sarti [13] then introduce the quantity

$$O(x, y, \theta) := \begin{cases} I(x, y) & |X_3(\bar{\theta})| = \max_\theta |X_3(\theta)| \\ 0 & \text{otherwise} \end{cases}$$

which will have maximum value for each (x, y) at $\bar{\theta}$. A point (x, y) on the image I gets lifted to the point $(x, y, \bar{\theta})$. The whole image domain is lifted in $SE(2)$ on the domain

$$\Sigma_0 = \{(x, y, \bar{\theta}) : |X_3(\bar{\theta})I| = \max_{\theta} |X_3(\theta)I| > 0\}$$

The lifted set corresponds to the maximum of the activity of the output of simple cells that are stimulated exclusively by external signals which can be modeled mathematically as a Dirac mass concentrated on Σ_0

$$\tilde{I}(x, y, \theta) = O(x, y, \bar{\theta})\delta_{\Sigma_0}$$

After such definition we have arrived to a point where $\tilde{I}(x, y, \theta)$ corresponds to the image on the visual cortex V1, where all the neural activity is concentrated on the Dirac mass δ_{Σ_0}. Integrating over a fiber S^1 that is non-critical at position (x, y) yields exactly the intensity of the original image at position (x, y).

Notice that if a point is critical for the function $I(x, y)$ then the gradient at that point is zero and therefore such point will not be part of Σ_0.

4.2.2 Differential operators and activity propagation on the sub-Riemannian space

Up until now we have built up some geometric foundation on the sub-Riemannian space $SE(2)$ in chapter 2 and defined a lifting procedure that links together 2D images with the neural representation in the visual cortex V1 modeled by $SE(2)$. In such space we are able to define curves, connect points and measure lengths. We lack however a way to introduce propagation of neural activity in the form of partial differential equations. The first step towards this goal is to define the differential operators used in this setting. The most basic differential operator is the concept of derivative of a real function.

Definition 44 (Lie derivative in $SE(2)$)
If $u : SE(2) \to \mathbb{R}$ is a real function, $\xi_0 \in SE(2)$ and $\gamma_i(s) = \exp(sX_i)(\xi_0)$ is an exponential curve, then we define the Lie derivative of the function u at the point ξ_0 as

$$\mathcal{L}_{X_i} u := X_i u(\xi_0) := \frac{d}{ds}(u \circ \gamma)|_{s=0}$$

when the right hand term exists and is finite.

Remark: The Lie derivative is defined for functions, vector fields, tensor fields and differential forms on a manifold. However in this work only the concept of Lie derivative of functions on the manifolds $SE(2)$ and $PT\mathbb{R}^2$ is used and therefore the discussion for the other cases is omitted. An introduction to such definitions can be found at [41] (chpt. 20).

Having endowed the space with a derivation we can now produce a notion of other differential operators in this geometrical setting such as the gradient, the divergence, the Laplacian and the Hessian of a function.

Definition 45 (Differential operators in $SE(2)$)
Let $u : SE(2) \to \mathbb{R}$ be a function for which $X_1 u$ and $X_2 u$ exist and are continuous. Let $\nu = (\nu_1, \nu_2)$ be a vector field in the horizontal distribution. Then we can define differential operators in terms of the Lie derivative, in analogy to the classical differential operators:

$$\nabla_R u = (X_1 u, X_2 u)$$

$$\mathrm{div}_R \nu = X_1 \nu_1 + X_2 \nu_2$$

$$\Delta_R u = \mathrm{div}_R(\nabla_R u)$$

$$\mathrm{Hess}_R = \begin{pmatrix} X_1^2 u & \frac{1}{2}(X_1 X_2 + X_2 X_1)u \\ \frac{1}{2}(X_1 X_2 + X_2 X_1)u & X_2^2 u \end{pmatrix}$$

Note that the mixed coefficients in the Hessian matrix contain the mean of the mixed derivatives in order to make it symmetric, as X_1 and X_2 do not in general commute. The Laplacian can also be seen as the trace of the Hessian matrix.

We denote as C_R^1 the set on which $X_1 u$ and $X_2 u$ exist and are continuous. In general we do not require u to be differentiable w.r.t. X_3 so that $C^1 \subset C_R^1$.

Definition 46 (Regular surface for $SE(2)$)
A subset $\Sigma \subset \mathbb{R} \times S^1$ is called a *regular surface* if it can be locally described as the 0-level set of a function u of class C_R^1 with non vanishing gradient. Precisely there exists a neighborhood U of every point of Σ and a function u defined on U such that

$$\Sigma \cap U = \{(x, y, \theta) \in U : u(x, y, \theta) = 0, \nabla_R u(x, y, \theta) \neq 0\}$$

In particular the lifting procedure defined by Citti-Petitot-Sarti in [13] lifts an image to a regular surface Σ_0.

If Σ is a regular surface we can define the R-normal vector to such surface as

$$\nu_R = \frac{(X_1 u, X_2 u)}{\sqrt{(X_1 u)^2 + (X_2 u)^2}}$$

and define the curvature of the surface as

$$H_R(\Sigma) = \mathrm{div}_R(\nu_R)$$

At this point we have all the tools to model the neural propagation in the visual cortex V1. The diffusion equation taken into consideration in the CPS model is a simple linear diffusion in the geometric structure, which takes the form

$$\partial_t u = \Delta_R u$$

Recall that in this setting

$$\Delta_R = \mathrm{div}_R(\nabla_R u) = \mathrm{div}_R(X_1, X_2) = X_1^2 + X_2^2$$

which takes the name of *sub-Laplacian*.

Since the associated matrix to Δ_R has zero determinant at every point the sub-Labplacian is a strongly degenerate operator. However the following result makes it suitable for our application nonetheless:

Theorem 47

Let P be an operator written in the form

$$P = \sum_{j=1}^{r} X_j^2 + X_0 + c$$

with X_i being the first order homogeneous differential operators with $C^\infty(\Omega)$ coefficients and $c \in C^\infty(\Omega)$ where Ω is an open set of $\mathbb{R}^n$. If among the operators X_{j_1}, $[X_{j_1}, X_{j_2}]$, $[X_{j_1}, [X_{j_2}, X_{j_3}]]$,... there exist n which are linearly independent at any given point in Ω then P is hypoelliptic.

Ref: [26] at *1.1* on pg. 149

In our specific setting Hörmander's theroem (thm. 47) takes the form

Theorem 48

Since the Lie algebra generated by X_1 and X_2 is of maximum rank at every point, then the sub-Laplacian operator is hypoelliptic. This simply means that for every initial condition, the solution of the evolution equation is C^∞.

Ref: [13] at *Thm. 2.1* on pg. 317

4.2.3 The resulting completion model

The completion model developed by Citti and Sarti joins the work of sub-Riemannian diffusion ($\partial_t u = \Delta_R u$) with non maxima suppression, which allows first to propagate the existing information simulating the physical effect of the visual cortex V1 and then complete boundaries and surfaces.

The starting point is to consider the lifted function $\tilde{I}(x, y, \theta)$ and introduce a function with a time variable $u : \mathbb{R}_+ \times SE(2) \to \mathbb{R}$ with initial condition

$$u(0, x, y, \theta) = u_0(x, y, \theta) := \tilde{I}(x, y, \theta)$$

At this point we proceed by discretizing the time variable. Let h be the length of a time interval, n be the step and therefore $t = nh$ be the time at step n. The idea of Citti and Sarti is to fix two natural numbers $N_1 \leq N_2$ and first diffuse the image in a finite time interval for N_1 steps followed by non-maxima suppression for $N_2 - N_1$ steps. In the ideal case $h \to 0$.

The process starts with initial surface $\Sigma_0 = \{(x, y, \theta) : \partial_\theta u_0 = 0, \partial^2_\theta u_0 < C\}$ for some constant C. Such surface corresponds to all the points that are critical with respect to θ and for which u_0 restricted to the fiber is either a concave function or has low convexity.

At every step n the resulting diffusion system is

$$\begin{cases} \partial_t u = \Delta_R u & \text{on } SE(2) \backslash \Sigma_n \text{ for } t \in [nh, (n+1)h] \\ u(\cdot, nh) = u_n(\cdot, nh)\delta_{\Sigma_n} \end{cases}$$

At every step n we get to define a new function u_{n+1} such that

$$u_{n+1}(\cdot, (n+1)h) = u(\cdot, (n+1)h)$$

and a new surface Σ_{n+1} such that

$$\Sigma_{n+1} = \{(x, y, \theta) : \partial_{\nu_{\Sigma_n}} u_{n+1} = 0, \partial^2_{\nu_{\Sigma_n}} u_{n+1} < 0\}$$

which is choosing the concave critical points according to the derivative along the direction of the normal vector field to Σ_n. Intuitively this can be seen as suppressing all the non-maxima when computing

$$u(\cdot,(n+1)h) = u_{n+1}\delta_{\Sigma_{n+1}}$$

In the original work of Citti and Sarti [13] it is presented a proof of convergence to a minimal surface Σ in the rototranslation space for $T \to \infty$ when $h \to 0$. This means that it is satisfied

$$\Delta_R u - \langle (\mathrm{Hess}_R u)\nu_\Sigma, \nu_\Sigma \rangle = 0$$

$$H_R(\Sigma) = \mathrm{div}_R(\nu_\Sigma) = 0$$

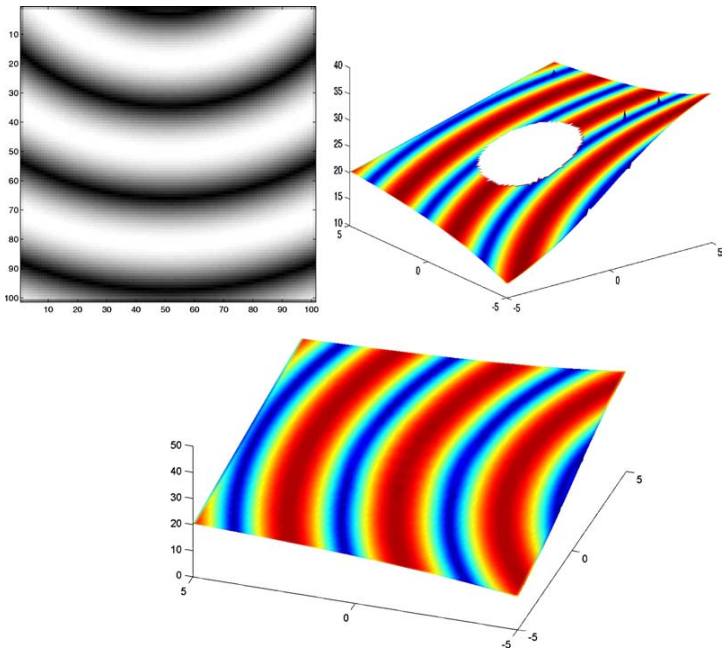


Figure 4.3: The original image (top left) is lifted to $SE(2)$ with missing information in the center (top right). The algorithm restores the information on $SE(2)$ (bottom). Courtesy of Citti and Sarti [13].

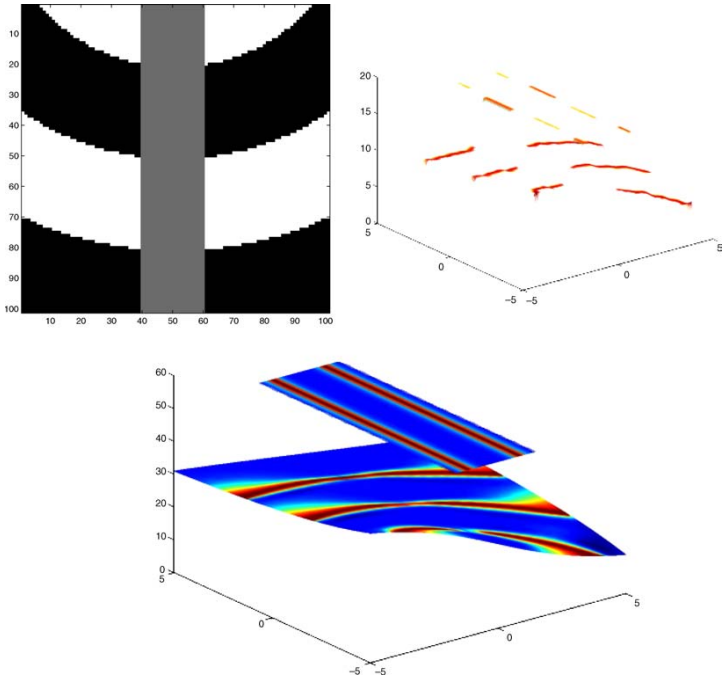


Figure 4.4: The original image (top left) is lifted to $SE(2)$ (top right). Both modal and amodal completion are performed by the algorithm at the same time (bottom). Courtesy of Citti and Sarti [13].

4.2.4 Numerical scheme

Citti and Sarti used finite differences to approximate the model equations, with a grid

$$(t_n, x_l, y_m, \theta_q) = (n\Delta t, l\Delta x, m\Delta y, q\Delta\theta)$$

with

$$\Delta x = \Delta y, \quad \Delta\theta = \Delta x^2, \quad \Delta t = \Delta x^2$$

Denote by u^n_{lmq} the value of the function u at the grid point t_n, x_l, y_m, θ_q.

Citti and Sarti chose to approximate the time derivative with first order forward differences and the space derivatives with second ordered centered differences.

Denote by D_x, D_y and D_θ the finite difference approximations of ∂_x, ∂_y and ∂_θ. Analogously denote by D_1 and D_2 the finite differences approximations for vector fields X_1 and X_2. Denote by D_{ij} the second order derivative corresponding to $D_j D_i$.

$$
\begin{aligned}
D_1 u^n_{lmq} &= \cos(\theta_q) D_x u^n_{lmq} + \sin(\theta_q) D_y u^n_{lmq} \\
D_2 u^n_{lmq} &= D_\theta u^n_{lmq} \\
D_{11} u^n_{lmq} &= \cos(\theta_q)^2 D_{xx} u^n_{lmq} + 2\cos(\theta_q)\sin(\theta_q) D_{xy} u^n_{lmq} + \sin(\theta_q)^2 D_{yy} u^n_{lmq} \\
D_{22} u^n_{lmq} &= D_{\theta\theta} u^n_{lmq} \\
D_{12} u^n_{lmq} &= \sin(\theta_q)(D_{y\theta} u^n_{lmq} - D_x u^n_{lmq}) + \cos(\theta_q)(D_y u^n_{lmq} + D_{x\theta} u^n_{lmq}) \\
D_{21} u^n_{lmq} &= \cos(\theta_q) D_{\theta x} u^n_{lmq} + \sin(\theta_q) D_{\theta y} u^n_{lmq}
\end{aligned}
$$

Therefore the evolution solving the diffusion equation on $SE(2)$ takes the form

$$
\begin{cases}
u^{n+1}_{lmq} = u^n_{lmq} + \Delta_t (D_{11} u^n_{lmq} + D_{22} u^n_{lmq}) & \text{for } 0 \le n < N_1 \\
\hline
v^{N_1}_{lmq} = D_2 u^{N_1}_{lmq} & \text{at } n = N_1 \\
\hline
u^{n+1}_{lmq} = u^n_{lmq} + \Delta_t \Big(\frac{(D_2 v^n_{lmq})^2 D_{11} u^n_{lmq} + (D_1 v^n_{lmq})^2 D_{22} u^n_{lmq}}{D_{11} v^n_{lmq} + D_{22} v^n_{lmq}} & \\
\qquad - \frac{(D_{12} u^n_{lmq} + D_{21} u^n_{lmq}) D_1 v^n_{lmq} D_2 v^n_{lmq}}{D_{11} v^n_{lmq} + D_{22} v^n_{lmq}} \Big) & \text{for } N_1 \le n < N_2 \\
v^{n+1}_{lmq} = D_2 u^n_{lmq} & \text{for } N_1 \le n < N_2
\end{cases}
$$

with Neumann boundary conditions on x and y and periodic boundary condition on θ.

4.3 The Boscain model

The Boscain model is constructed starting from the CPS model, but built over the sub-Riemannian structure $PT\mathbb{R}^2$ rather than $SE(2)$, thus considering direction regardless of orientations, and specifically taking into account the fact that when an image is not just a curve then the sub-Riemannian problem with fixed boundary conditions is insufficient.

The authors came to the diffusion equation

$$\partial_t \phi(x, y, \theta, t) = \Delta_H \phi(x, y, \theta, t)$$

where

$$\Delta_H = (X_1)^2 + \beta^2 (X_2)^2 = (\cos(\theta)\partial_x + \sin(\theta)\partial_y)^2 + \beta^2 \partial_\theta^2$$

is called the *hypoelliptic heat kernel* or *sub-Laplacian*.

The Boscain restoration algorithm is performed in four steps. Let I_c be the corrupted image.

1. **Smoothing of I_c**: The corrupted input image I_c is smoothed by convolution with a Gaussian of standard deviation $\sigma_x = \sigma_y > 0$. Equal standard deviation on both axes guarantees invariance by rototranslation. The resulting image $I_s = I_c * G(\sigma_x, \sigma_y)$ is generically a Morse function.

> **Definition 49** (Morse function)
> A smooth function $f : \mathbb{R}^2 \to \mathbb{R}$ is said to be a *Morse function* if it has only isolated critical points with nondegerate Hessian.
>
> Ref: [8] at *2.9* on pg. 1326

> **Proposition 50**
> Generically, the convolution of an L^2 function over a bounded domain $\mathcal{D} \in \mathbb{R}$ with a Gaussian G is a Morse function.
>
> Ref: [8] at *Appendix* on pg. 1331

2. **Lift of I_s to $PT\mathbb{R}^2$**: Every point (x, y) is associated with an orientation $\theta \in \mathbb{R}/(\pi\mathbb{Z})$ corresponding to the direction of the level set of I_s at the point, which is well defined when $\nabla I_s \neq 0$. Where $\nabla I_s = 0$ we associate every possible direction. A *lifted support* S_I is introduced as

$$S_I = \{(x, y, \theta) \in PT\mathbb{R}^2 : \nabla I_s(x, y) \cdot (\cos\theta, \sin\theta) = 0\}$$

which corresponds to the domain where either $\nabla I_s(x, y) = 0$ or θ is the direction parallel to the level set.

> **Proposition 51**
> If I_s is a Morse function, then S_I is an embedded 2D submanifold of $PT\mathbb{R}^2$.
>
> Ref: [8] at *prop19* on pg. 1327

3. **Lift of I_s to a distribution**:

$$\bar{I}_s(x, y, \theta) = I_s(x, y)\delta(S_I)$$

This step is formally necessary because the support S_I of I_s has Lebesgue measure zero, and therefore I_s is vanishing almost everywhere on $PT\mathbb{R}^2$.

4. **Hypoelliptic evolution**: Fix $T > 0$ and compute the solution at time T to the Cauchy problem

$$\begin{cases} \partial_t \phi(x, y, \theta, t) = \Delta_H \phi(x, y, \theta, t) \\ \phi(x, y, \theta, 0) = \bar{I}_s(x, y, \theta) \end{cases}$$

with

$$\Delta_H = (X_1)^2 + \beta^2 (X_2)^2$$

$$X_1 = \cos\theta \partial_x + \sin\theta \partial_y$$

$$X_2 = \partial_\theta$$

with a fixed parameter β.

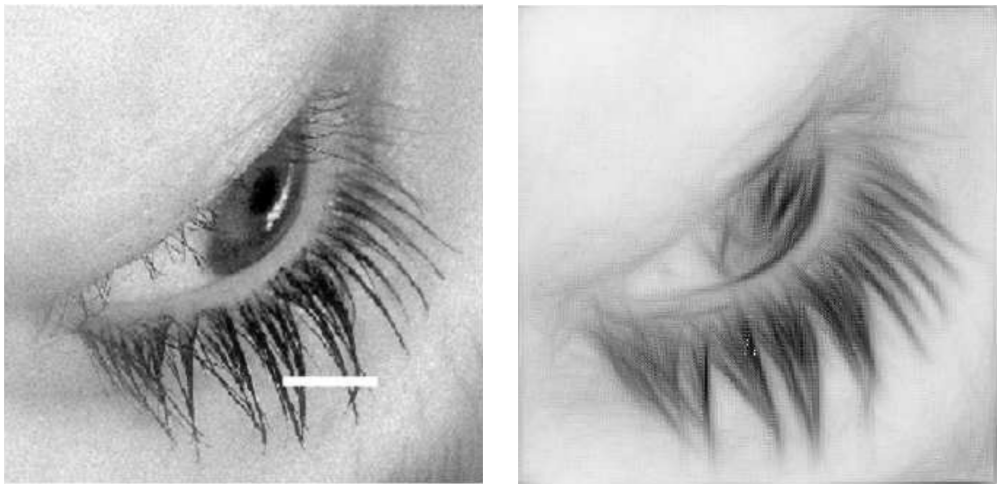
5. **Projecting to** $\mathbb{R}^2$: The reconstructed image is computed as

$$I_T(x, y) = \max_{\theta \in P^1} \phi(x, y, \theta, T)$$

From a computational standpoint Boscain [8] presents also a method to disintegrate the regular representation of $SE(2)$ using the non-commutative Fourier transform with respect to the space variable $\mathbf{x} = (x, y)$. In Fourier space the Cauchy problem becomes

$$\begin{cases} \partial_t \tilde{u}(t, \mathbf{x}, \theta) = \beta^2 \partial_\theta^2 \tilde{u}(t, \mathbf{x}, \theta) - 4\pi^2 (x \cos\theta + y \sin\theta)^2 \tilde{u}(t, \mathbf{x}, \theta) \\ \tilde{u}(0, \mathbf{x}, \theta) = \tilde{\tilde{I}}_c(\mathbf{x}, \theta) \end{cases}$$

which is highly parallelizable as it can be solved simultaneously at each point $\mathbf{x}$.



(a) Corrupted image (b) Restored image

Figure 4.5: Example of restoration on an antropomorphic image, where the details resemble for the most part curves. Courtesy of Boscain et al. [8].

4.4 Advanced techniques

In a more recent article by Boscain et al. [10] from 2018 a few improvements over the aforementioned restoration process have been presented. These improvements aim to exploit the knowledge of the position of the corruption to yield a multi-step algorithm for highly corrupted image inpainting.

4.4.1 Hypoelliptic diffusion with varying coefficients

The key idea is to exploit the knowledge of the position of the corruption to modify the restoration procedure, either by applying the hypoelliptic diffusion only to corrupted regions of the image, or to consider a different final time T for every region.

By modifying the operator Δ_H to be

$$\Delta_H = a(X_1)^2 + b(X_2)^2$$

$$a, b : \mathbb{R}^2 \to \mathbb{R}$$

we can control the intensity of the diffusion as a function of the position (x, y). The key idea is to choose small values of a, b at non-corrupted points and larger values at corrupted points.

Since $I_c = 0$ at corrupted points the coefficients $a(x, y)$ and $b(x, y)$ can be chosen to be a continuous approximation of the indicator function of $\delta_{\mathcal{C}}$ of the corrupted region $\mathcal{C}$.

$$a(x, y) = a_0 + a_1 \exp\left(-\frac{I_c^2(x, y)}{\sigma}\right)$$

$$b(x, y) = b_0 + b_1 \exp\left(-\frac{I_c^2(x, y)}{\sigma}\right)$$

4.4.2 AHE algorithm

Another algorithm presented in [10] is *Averaging and Hypoelliptic Evolution* (AHE) which is a 4-steps algorithm divided as follows

1. **Preprocessing**: The original image $f(x, y)$ is first divided in two domains: B (Bad) is the corrupted domain and G (Good) the non-corrupted domain. Then $f(x, y)$ is preprocessed by filling the corrupted areas with a rough approximation

obtained by convolution in a neighbor. The fill-in algorithm takes inspiration from the Breadth-First Search algorithm (BFS, [11]) by iteratively computing the discrete boundary of the corrupted area, and substituting every pixel with an average on a non-corrupted neighborhood.

More precisely, let $f^0 = f$, $G^0 = G$ and $B^0 = B$. If we consider $K(x_k, y_l)$ a square neighborhood of the point (x_k, y_l) then define G^i_{kl} as $K(x_k, y_l) \cap G^i$.

Define now for $(x_k, y_l) \in \partial B^i$:

$$f^{i+1}(x_k, y_l) = \frac{1}{|G^i_{kl}|} \sum_{(x,y)\in G^i_{kl}} f^i(x, y)$$

and for $(x_k, y_l) \notin \partial B^i$:

$$f^{i+1}(x_k, y_l) = f^i(x_k, y_l)$$

Then

$$G^{i+1} = G^i \cup \partial B^i$$

$$B^{i+1} = B^i \backslash \partial B^i$$

and proceed until $B = \emptyset$. Denote by $g(x, y)$ the image obtained after this step.

2. **Main diffusion**: Following the preprocessing step those that were previously gaps in the image are now filled, leaving us with a partially restored image $g(x, y)$ affected by a strong *mosaic effect*.

   A quantitative way to estimate the intensity of such mosaic effect is to compute the *magnitude of the gradient* (MoG) of the image, $MoG(x, y) = |\nabla g(x, y)|$. Such function will have value 0 in regions of constant intensity and high value in regions where the rate of change of the intensity of the image is high, usually corresponding to the edges of the mosaic effect.

   Once we have established which are the pixels that are to be smoothened we can lift the image $g(x, y)$ to $\bar{g}(x, y, \theta)$ and apply a round of hypoelliptic diffusion with varying coefficients, with coefficients $a(x, y)$ and $b(x, y)$ chosen so that the diffusion is more intense at the points where the MoG is higher. Boscain et al. [10] suggest to use coefficients

$$a_0 = 0.05, \quad a_1 = 0.2, \quad b_0 = 0.55, \quad b_1 = 5, \quad \sigma = 0.4$$

$$a(x, y) = a_0 + a_1 \exp\left(-\frac{\varphi^2(x, y)}{\sigma}\right)$$

$$b(x, y) = b_0 + b_1 \exp\left(-\frac{\varphi^2(x, y)}{\sigma}\right)$$

$$\varphi(x,y) = 1 - \frac{|\nabla g(x,y)|}{\max|\nabla g(x,y)|}$$

Denote by $h(x,y)$ the image obtained by projecting the result of the hypoelliptic diffusion with varying coefficients.

3. **Synthesis** Even by fine-tuning the coefficients $a(x,y)$, $b(x,y)$ a blurring is introduced as a result of the hypoelliptic diffusion since the pixels with high MoG are not only the edges of the mosaic effect, but high-frequency details as well.

   To restore some of the high frequency details we repeat the BFS procedure in step 1, considering however values in $h(x,y)$.

$$f^{i+1}(x_k,y_k) = h(x_k,y_k)\frac{\sum_{(x,y)\in G^i_{kl}} f^i(x,y)^{-1}h(x,y)^{-1}}{\sum_{(x,y)\in G^i_{kl}} f^i(x,y)^{-2}}$$

4. **Weak smoothing** The last step is a final smoothing to reduce the mosaic effect partially reintroduced in step 3. To do so we apply the same procedure as in step 2, with smaller coefficients:

$$a_0 = 0.015, \quad a_1 = 0.1, \quad b_0 = 0.15, \quad b_1 = 1.5, \quad \sigma = 0.3$$

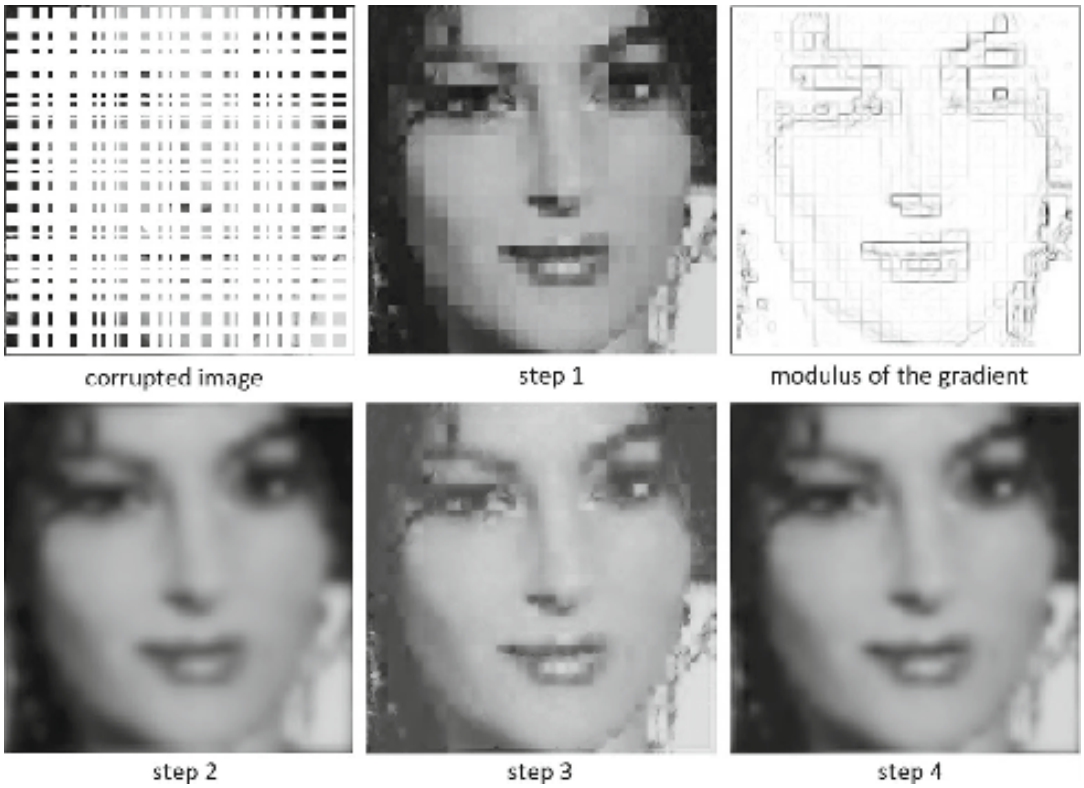


Figure 4.6: Example of AHE algorithm, courtesy of Boscain et al. [10]

Chapter 5

Own work and further developments

In the previous chapter the current state of PDE-based restoration algorithms exploiting the $SE(2)$ and $PT\mathbb{R}^2$ geometric structures was presented. All the literature on the subject, which the reader can find listed in the bibliography, has been developed in the last two decades. As it is a relatively new approach to the problem of image restoration there is still much work that can be done to improve the results one can obtain with such methods (especially compared to the state of the art based on Neural Networks). In this chapter a new lifting method based on $PT\mathbb{R}^2$ is presented, together with a novel algorithm for image restoration.

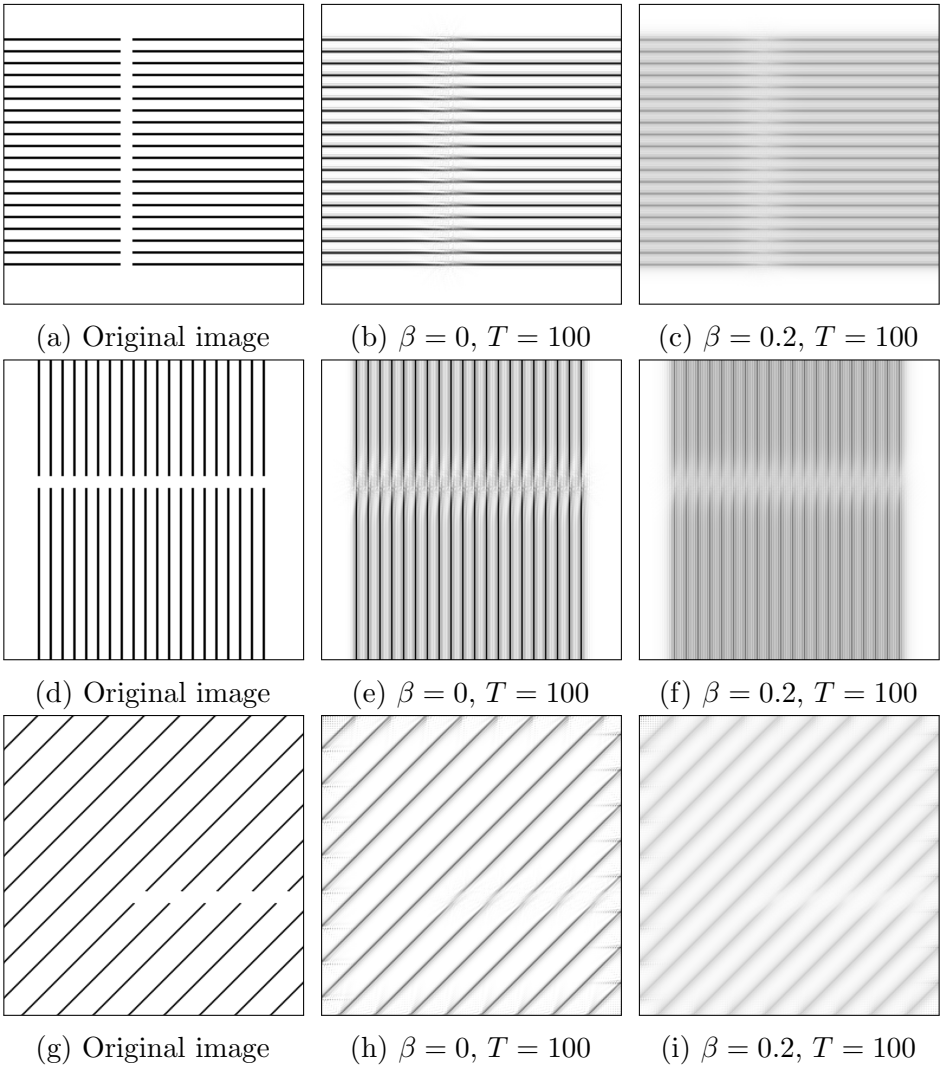
In the writing of this chapter, as well as the next one, the methods and algorithms presented by Citti and Sarti [13] and Boscain et al. [8], [9], [10] have been re-implemented as a Python library based on `opencv`. The choice of using Python as the programming language was made for its versatility, availability of libraries and license-wise compliance with the GNU General Public License (GPL) [39]. All the figures presented in these two chapters are generated using this new library.

5.1 Examples and tests

While reading the literature on the subject it is noticeable a general lack of examples and results regarding the simple cases of neural completion: simple straight lines, circles or other basic shapes. This section aims to cover such lack by producing intuitive examples of perception completion based on the work presented in the previous chapter. This step is of fundamental importance as it provides a deep insight in how the restoration algorithm behaves in the discrete case of digital image processing, providing inspiration for new techniques and resulting in a way to address possible theoretical or

implementation bugs as well as numerical oversights in the code.

The simplest possible example is the case of straight lines, by allowing the diffusion to take place up to a large value T (convergence) with different values of β. We expect that in the case of clean straight lines a diffusion with $\beta = 0$ will produce almost perfect perception completion while non-zero values of β will diffuse the lines also perpendicularly.



(a) Original image (b) $\beta = 0$, $T = 100$ (c) $\beta = 0.2$, $T = 100$

(d) Original image (e) $\beta = 0$, $T = 100$ (f) $\beta = 0.2$, $T = 100$

(g) Original image (h) $\beta = 0$, $T = 100$ (i) $\beta = 0.2$, $T = 100$

Figure 5.1: In the first column the original images. Second and third columns are the results of diffusion with $\beta = 0$ and $\beta = 0.2$ respectively.

We can notice how our assumption was partially wrong as there is in fact a diffusion that takes place also perpendicularly to the direction of the lines even when $\beta = 0$. Although the role of the parameter β modulating the angle of diffusion is what we have foreseen.

If we take one of the lines and we zoom in to try to analyze what might be the origin of the perpendicular component of the diffusion we soon notice that at the edge of the

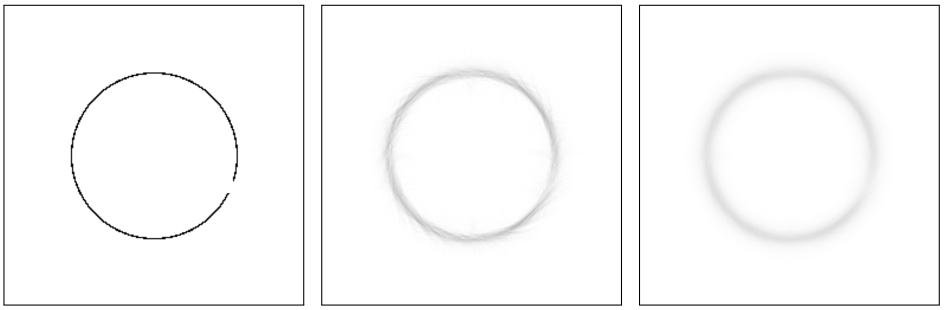
corruption for a small portion of pixels the level line of the image function is in fact perpendicular to line itself, and therefore the model is diffusing the image also in that direction.

(a) Original image (b) T=1

Figure 5.2: One step of diffusion with $\beta = 0$ and $dt = 1$. dt is chosen to be large in order to make the effect more visible. The original image (a) is the same as the one in figure 5.1 but only a cropped portion is displayed.

.

As a last example an interrupted circumference is processed (figure 5.3). Choosing $\beta = 0$ every point on the circumference is diffused towards the tangent direction. This diffusion produces a visible distortion as the discrete nature of digital images makes some tangent vectors predominant with respect to the others. With $\beta = 0.2$ the restoration algorithm produces less distortion as the angle of diffusion is wider, but the smoothing effect is more extensive and the resulting image appears less sharp.



(a) Original image (b) $\beta = 0$, $T = 100$ (c) $\beta = 0.2$, $T = 100$

Figure 5.3: A corrupted circumference (a) is blurred using a Gaussian filter with $\sigma = 1.8$ and afterwards diffusion is applied (b and c).

5.2 Gaussian Lift

In this section a different and yet unexplored approach to a lift over $PT\mathbb{R}^2$ is introduced. The idea is based on "spreading" the input signal around the orientation of maximum

response $\bar{\theta}$ of the simple cells, and doing so following a Gaussian distribution centered on such orientation.

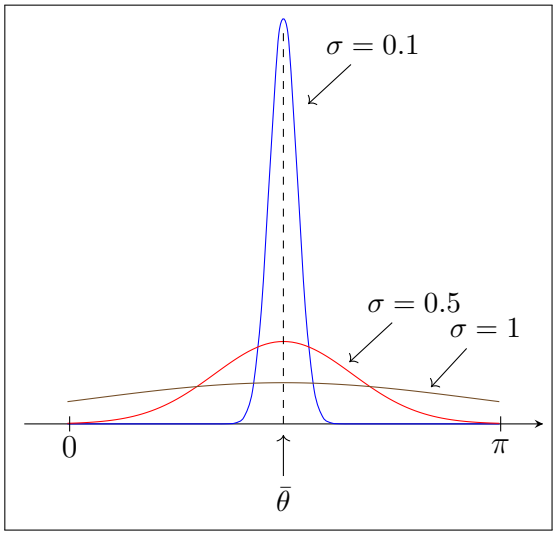


Figure 5.4: Illustration of the distribution of the neural activation on a specific fiber P^1 with different values for σ, for $\bar{\theta} = \frac{\pi}{2}$. If σ is small enough the neural activation is negligible outside a ball centered around $\bar{\theta}$. If on the other hand σ is too large then the neural activation happens at every point of the fiber resulting in a very blurry image.

Let ∇I be the gradient of the image and θ be the angle of the level curve (perpendicular to the gradient). The lifting formula takes the form

$$\mathcal{L}(I)(x,y,\theta) = I(x,y) \cdot \exp\left(-\frac{\left\langle \frac{\nabla I(x,y)}{|\nabla I(x,y)|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2} \right)$$

and we can denote by $\tilde{I} = \mathcal{L}(I)$ the lifted image on $PT\mathbb{R}^2$.

To obtain a viable lifting procedure an appropriate corresponding inverse has to be defined. Introduce the following quantities:

$$Q(x,y,\theta) := -\int_0^\theta \frac{\partial}{\partial\tilde{\theta}} \ln(\tilde{I}(x,y,\tilde{\theta}))d\tilde{\theta}$$

$$Q_{\min}(x,y) := \min_{\theta\in[0,\pi]} Q(x,y,\theta)$$

$$\mu(x,y,\theta) := Q_{\min}(x,y) - Q(x,y,\theta)$$

Then the following theorem holds:

Theorem 52

$\pi : PT\mathbb{R}^2 \to \mathbb{R}^2$ defined as

$$\pi(\tilde{I})(x,y) = \frac{1}{\pi}\int_0^{\pi} \tilde{I}(x,y,\theta)\cdot e^{-\mu(x,y,\theta)}d\theta$$

is an inverse for

$$\mathcal{L}(I)(x,y,\theta) = I(x,y)\cdot \exp\left(-\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta,\sin\theta)\right\rangle^2}{2\sigma^2}\right)$$

Proof. Compute explicitly $Q(x,y,\theta)$ by applying the fundamental theorem of calculus

$$\begin{aligned}
Q(x,y,\theta) &= -\int_0^{\theta} \frac{\partial}{\partial\tilde{\theta}} \ln(\tilde{I}(x,y,\tilde{\theta}))d\tilde{\theta} \\
&= -\int_0^{\pi} \frac{\partial}{\partial\tilde{\theta}} \left[\ln(x,y) - \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\tilde{\theta},\sin\tilde{\theta})\right\rangle^2}{2\sigma^2}\right] d\theta \\
&= \int_0^{\pi} \frac{\partial}{\partial\tilde{\theta}} \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\tilde{\theta},\sin\tilde{\theta})\right\rangle^2}{2\sigma^2} d\tilde{\theta} \\
&= \left[\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\tilde{\theta},\sin\tilde{\theta})\right\rangle^2}{2\sigma^2}\right]_0^{\theta} \\
&= \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta,\sin\theta)\right\rangle^2}{2\sigma^2} - \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (1,0)\right\rangle^2}{2\sigma^2}
\end{aligned}$$

and compute explicitly its minimum w.r.t. the angle θ knowing that the normalized gradient of the image is a vector of norm 1 oriented at an angle in $[0,\pi)$, and that there will be at least one angle such that $(\cos\theta,\sin\theta)$ is parallel to the gradient of

the image.

$$\begin{aligned}
Q_{\min}(x,y) &= \min_{\theta\in[0,\pi]} \left(\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2} - \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (1,0) \right\rangle^2}{2\sigma} \right) \\
&= \min_{\theta\in[0,\pi]} \left(\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2} \right) - \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (1,0) \right\rangle^2}{2\sigma^2} \\
&= 0 - \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (1,0) \right\rangle^2}{2\sigma^2} \\
&= -\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (1,0) \right\rangle^2}{2\sigma^2}
\end{aligned}$$

Now define μ as follows

$$\mu(x,y,\theta) = Q_{\min}(x,y) - Q(x,y,\theta) = \frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2}$$

and by plugging in these terms the conclusion follows.

$$\begin{aligned}
\pi(\tilde{I})(x,y) &= \frac{1}{\pi}\int_0^\pi \tilde{I}(x,y,\theta)\cdot e^{-\mu(x,y,\theta)} d\theta \\
&= \frac{1}{\pi}\int_0^\pi I(x,y) e^{-\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2}} e^{\frac{\left\langle \frac{\nabla I}{|\nabla I|}, (\cos\theta, \sin\theta) \right\rangle^2}{2\sigma^2}} d\theta \\
&= \frac{1}{\pi}\int_0^\pi I(x,y) d\theta \\
&= I(x,y)
\end{aligned}$$

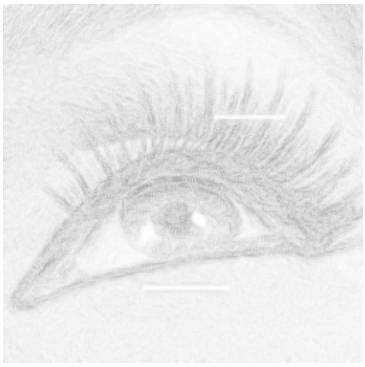
□

Although it seems that this whole computation has been done as a mathematical exercise for its own sake there is a deeper meaning in computing Q, $Q_{\min}$ and μ. If one takes an image, lifts it to $PT\mathbb{R}^2$ and applies a diffusion for $T > 0$ or actually any operation that alters it, then the information about the gradient of the original image in $\mathbb{R}^2$ loses its meaning, as it is no longer linked to the information in $PT\mathbb{R}^2$. Thus it is fallacious to project $\tilde{I}$ using the "old" gradient map. However μ can be directly computed in $PT\mathbb{R}^2$ without knowing the gradient of the image in $\mathbb{R}^2$, hence providing an actual well-defined way to project the neural image back to a digital image.

(a) Original image

(b) Blurred image

(c) CPS lift, $T = 0.5$



(d) CPS lift, $T = 3$

(e) Gaussian lift, $\sigma = 10$, $T = 3$

(f) Gaussian lift, $\sigma = 1$, $T = 3$

(g) Gaussian lift, $\sigma = 0.5$, $T = 3$

(h) Gaussian lift, $\sigma = 0.1$, $T = 3$

Figure 5.5: A corrupted image (a) is blurred with a Gaussian kernel (b) and lifted using the lift presented by Citti and Sarti (b, c and d) or by Gaussian lift (e, f, g and h). On the same time interval diffusion performed on a Gaussian lift with higher σ is more stable than the same diffusion applied on the lift proposed by Citti and Sarti [13].

(a) Gaussian lift, $T = 21$ and $\beta = 0.1$

(b) Gaussian lift, $T = 21$ and $\beta = 1$

(c) Gaussian lift, $T = 21$ and $\beta = 10$

Figure 5.6: The same original image as in 5.5 is blurred and lifted through Gaussian lift with $\sigma = 100$ and then diffused with $\beta = 0.1$ (a), $\beta = 1$ (b) and $\beta = 10$ (c). Smaller values of β produce crisper results.

Remark: The biological justification behind the introduction of this new lifting procedure takes inspiration yet again from the Gabor filters studied in the work by Marcelja [32] and Jones and Palmer [27]. In a Gabor filter the signal decays exponentially away from its orientation θ. If an hypercolumn is modeled as a stack of Gabor filters with different orientations what this newly introduced lifting procedure actually does is suppressing the signal of all the non-maxima filters.

5.3 WaxOn-WaxOff technique

> *"Wax on, left hand. Wax off, right hand. Wax on, wax off. [...] Don't forget to breath....very important!"* (Mr. Miyagi, [3])

In this last section a novel technique exploiting the sub-Laplacian $\Delta_H = X_1^2 + \beta^2 X_2^2$ is presented. Image restoration is performed in two steps: an hypoelliptic diffusion followed by a sharpening process.

The inspiration comes, yet again, from the heat equation. In a PDE lecture, after learning about the heat equation, one could wonder if it were possible to reverse the diffusion produced by the heat equation in order to predict the initial temperature profile at $t = 0$ given any instantaneous temperature profile at $t = T_1$. Unfortunately the short answer is no ([18], [40]): the heat flow is an irreversible process and solving the problem of going backwards in time yields chaos. Although a solution for some $t = T_0 \in [0, T_1]$ clearly exists.

The problem of going backwards in time reversing the effect of the heat flow is extremely susceptible to noise and will eventually diverge "blowing up". What we can do however is setting up a numerical approach to reverse the heat diffusion only up to a certain time T_0, and stop it before it blows up.

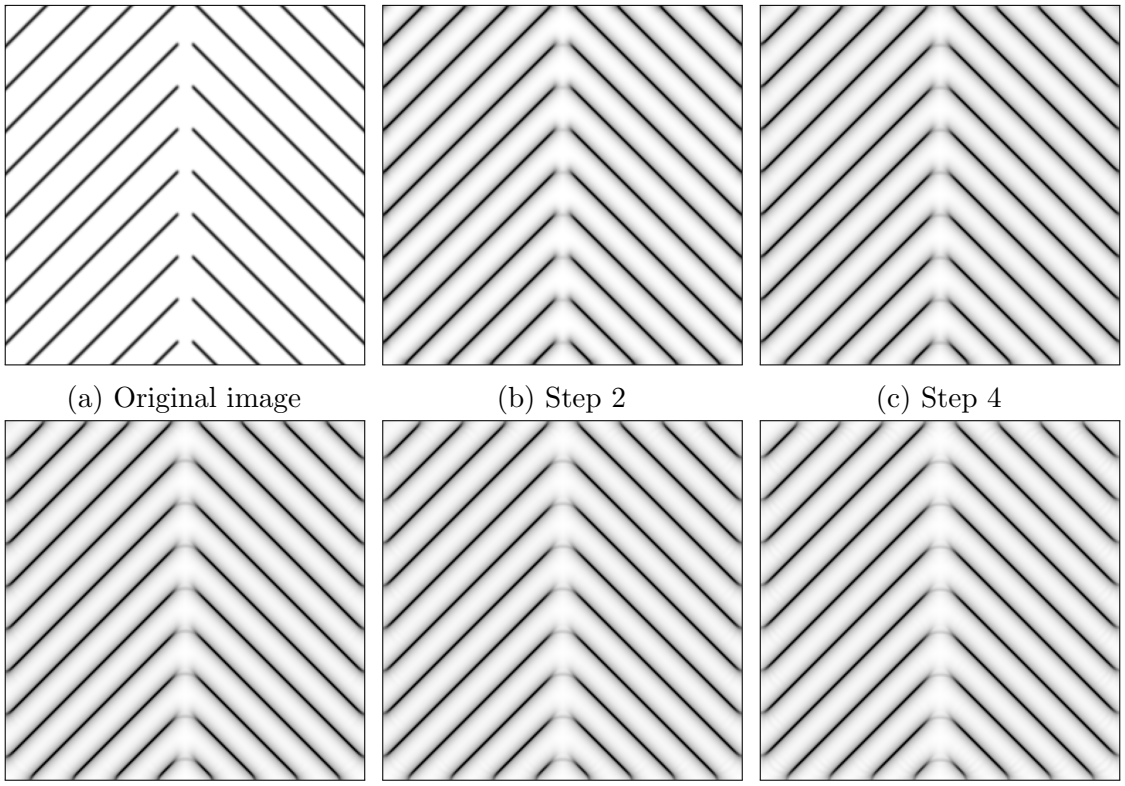
We can apply the same principle to the diffusion over $PT\mathbb{R}^2$ defined by Δ_H, and propose a new algorithm for image restoration: assuming that for small T we can recover the initial profile of neural activity by reversing the diffusion PDE, we can sharpen the restored image by applying such reverse diffusion considering a large value of β.

Thus we first "put the wax on", diffusing the image alongside the level curves with a small β, and then "get the wax off", sharpening using a larger value for β. In-between steps the image is projected to $\mathbb{R}^2$ and lifted back to $PT\mathbb{R}^2$ to avoid noise, which is mainly generated in the WaxOff phase of the algorithm.

The main differences with the methods proposed by Citti and Sarti [13] and by Boscain et al. [8] are a different lifting procedure that diffuses the neural activity on a broader family of association fields, the introduction of two parameters β for both diffusion and regression, and the replacement of non-maxima suppression in favor of a sharpening technique based on reversing the diffusion process.

(a) 1 step WaxOn (b) 1 step WaxOn-WaxOff (c) 2 steps WaxOn-WaxOff

Figure 5.7: Example of diffusion *WaxOn* on Gaussian lift (a) followed by regression *WaxOff* for small T (b). Multiple iterations of WaxOn-WaxOff are sequentially applied to produce (c)

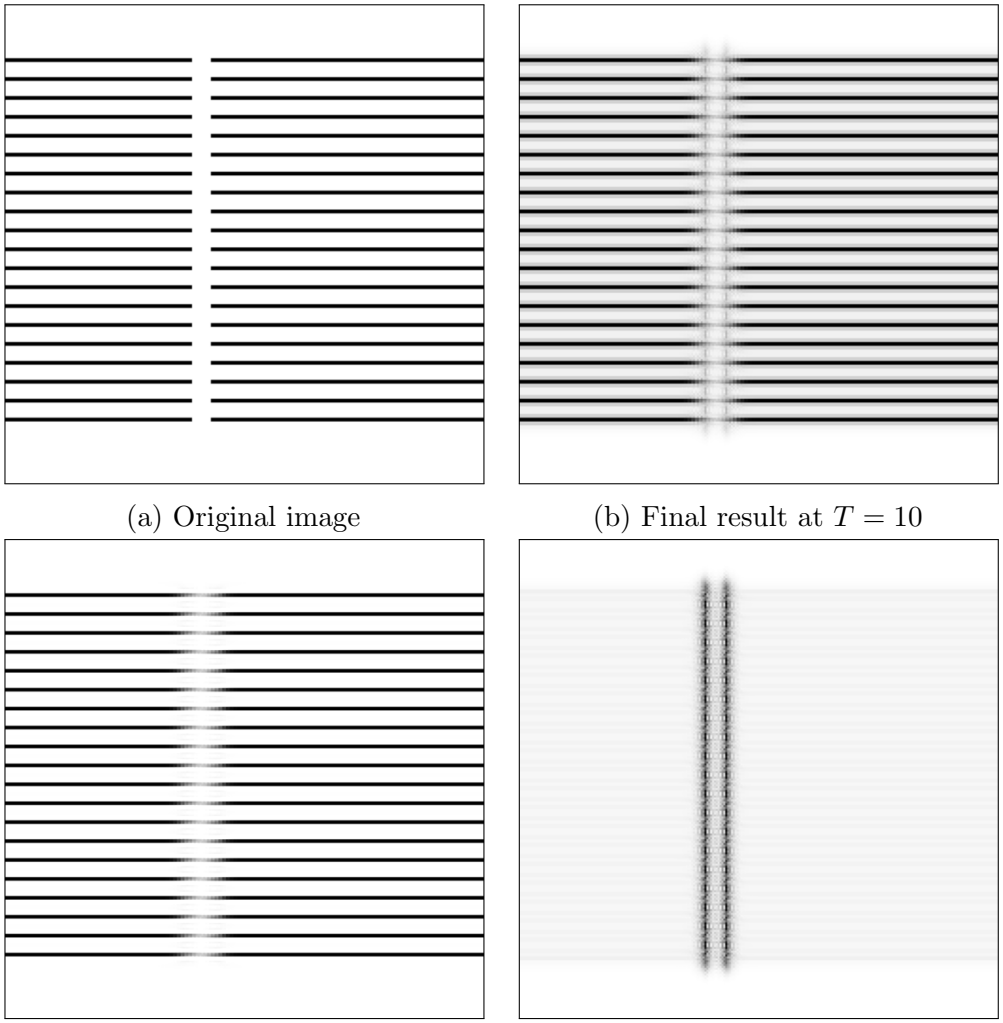


(a) Original image (b) Step 2 (c) Step 4

(d) Step 6 (e) Step 8 (f) Step 10

Figure 5.8: WaxOn-WaxOff diffusion obtained by iterating WaxOn for $T = 5$ with $\beta = 0.1$ and WaxOff for $T = 0.2$ with $\beta = 5$. After every iteration a sharpening kernel and a Gaussian smoothing kernel are used to additionally reduce noise. An arc between every pair of lines appears and becomes more visible after every iteration. This is the projection of a geodesic in $PT\mathbb{R}^2$ connecting every pair of lifted lines, as opposed to a geodesic in $\mathbb{R}^2$ that would be a straight line.

An interesting feature of this technique is that in the sharpening part of the algorithm both modal and amodal completions are clearly highlighted. See Figure 5.9 for an example.



(a) Original image (b) Final result at $T = 10$

(c) Modal completion (d) Amodal completion

Figure 5.9: Modal and amodal completion by Gaussian lift with $\sigma = 0.2$ and WaxOn-WaxOff with $\beta_1 = 0$ and $\beta_2 = 0.5$ (b). The two completions are obtained by erasing the neural activity distant from $\theta = 0$ (c) or close to $\theta = 0$ (d). The images are normalized.

Another application of this technique is perception-based image sharpening. By applying only the WaxOff part of the algorithm to a blurry image a sharpening technique is obtained.

(a) Original image

(b) Image blurred with 15px Gaussian kernel with $\sigma = 1.5$

(c) Regression with $T = 0.2$, $dt = 0.01$ after lift with $\sigma = 0.2$

Figure 5.10: A digital image (a) is first blurred with a Gaussian kernel and then sharpened using the WaxOff component of the WaxOn-WaxOff technique.

Remark: The problem with this approach is that the regression phase tends to blow up for big enough time intervals T. Additionally convergence and stability are not guaranteed although empirical results appear promising.

Chapter 6

A perception-based metric for Image Restoration

Assessing quantitatively the success of a restoration process is a difficult task. In the previous chapters many examples of convolution filters and image inpainting processes were shown without proposing a metric to assess the quality of the results. The reader might have been convinced just by the figures that the restoration process works. After all the human brain does this assessment intrinsically through neural perception. The aim of this chapter is however to present a metric that reflects such intrinsic neural mechanism to produce a quantitative assessment.

In the first two sections a few metrics that are well established in the field of Image Processing are presented (MSE, SSIM, GSSIM). Afterwords two new metrics are introduced (WGMSE, MSE^{σ}_{Gs}), which are part of the author's contribution and are presented for the first time in this work. At the end of the chapter a comparison is drawn on a set of examples that reflect how human perception works.

6.1 MSE

Mean Squared Error is the most common estimator of image quality, to the point that it is built-in specifically for images in MATLAB as the function `immse`. MSE is generally defined on a subset of $\mathbb{R}^n$ as

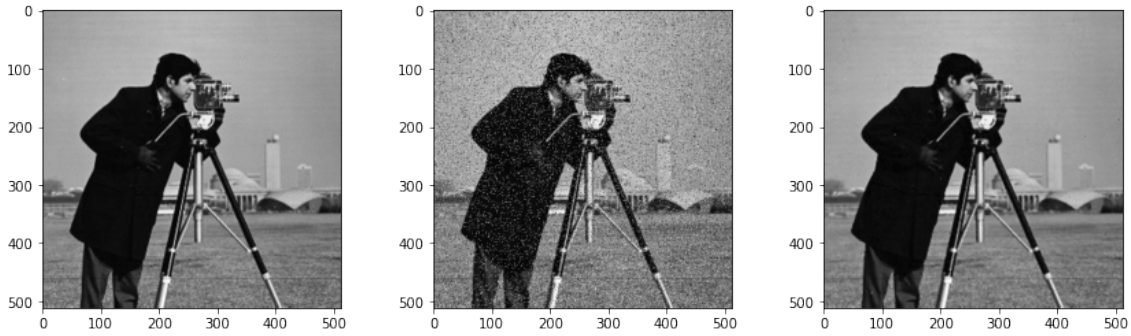
$$MSE(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=0}^{n} (x_i - y_i)^2$$

In the specific case of $N \times M$ images I, J it is defined as

$$MSE(I,J) = \frac{1}{MN} \sum_{y=0}^{M-1} \sum_{x=0}^{N-1} (I(y,x) - J(y,x))^2$$

using the convention of having the first index as the row, and second index as the column.

Although MSE is a good metric to measure the scale of corruption, it is generally not a good metric to measure the efficacy of restoration because it does not correlate well with the subjective quality the brain perceives. To see this, simply consider the following example of an image corrupted with salt and pepper noise, which is subsequently restored using a median filter (as it was introduced in Chapter 3).



(a) Original image, MSE=0

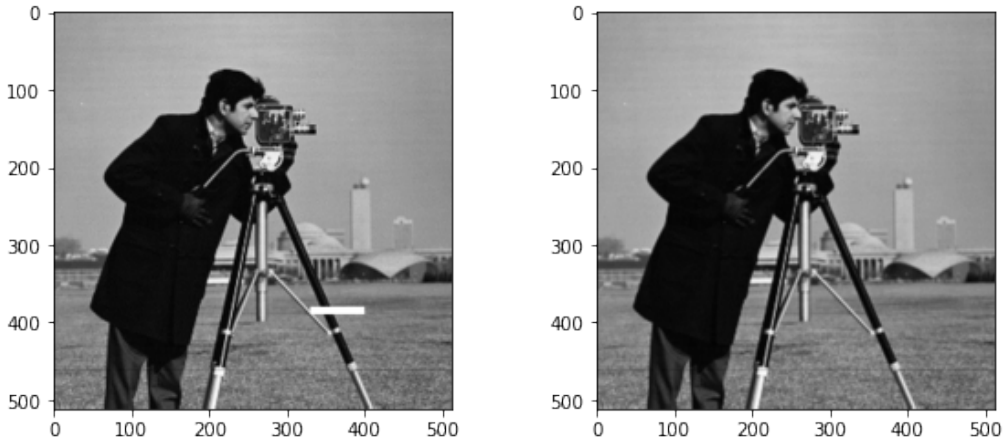
(b) Corrupted image, MSE=0.04

(c) Restored image, MSE=0.13

Figure 6.1: Example of MSE for image restoration assessment with Salt&Pepper noise: the original image (a) is corrupted with Salt&Pepper noise (b) which is restored using a median filter (c). Although (c) looks visually more appealing than (b) the MSE metric suggests otherwise.

Another example of MSE failing in the detection of a successful restoration can be presented in the case of a corrupted region with missing information. As seen in the following example (Figure 6.2) the error according to MSE is higher when considering a slightly blurred picture (almost imperceptibly) rather than in the case when the information from a whole region is missing.

Together these two examples aim to show that MSE is a very poor metric when assessing even very simple cases of image restoration.



(a) Corrupted image, MSE<0.01

(b) Gaussian blur $\sigma = 0.6$, MSE=0.06

Figure 6.2: The same image as in Figure 6.1 is corrupted with Missing information (a) and with an almost imperceptible Gaussian noise with $\sigma = 0.6$ (b).

6.1.1 SSIM and GSSIM

Other metrics such as SSIM (*Structural Similarity Index Measure*) and GSSIM (*Gradient-based Structural Similarity Index Measure*), which are briefly introduced in this section, appear to perform better from the point of view of human perception, as studied by Sara et al. [38].

SSIM was first introduced by Wang et al. [42] in 2004. It works by estimating the similarity index at each point (x, y) by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where on a rectangular neighborhood of the point (x, y) the quantities μ_x, μ_y are the sample means, σ_x^2, σ_y^2 are the sample variances and σ_{xy} is the sample correlation coefficient defined as

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$$

C_1 and C_2 are small coefficients used to avoid instability around points where the denominator approaches zero.

The index of the image is the mean of SSIM at all points in the image.

GSSIM was introduced by Chen et al. [12] to address SSIM failing on badly blurred images. It was developed based on the fact that edge information shall be regarded as

the most meaningful information when dealing with the Human Visual System.

Its definition is

$$GSSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{x'y'} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_{x'}^2 + \sigma_{y'}^2 + C_2)}$$

where x' and y' are the maps corresponding to the magnitude of the gradient.

6.1.2 Weighted gradient-based MSE (WGMSE)

In this section a new metric, part of the author's contribution, is presented.

A different more intuitive approach is to modify the MSE metric to also take into account the gradient of the image by weighting the difference in intensity and the difference in gradient. The assumption on which this is done is that our visual cortex is not only sensitive to the local intensity of the pixels, but also to the contrast that our brain perceives.

This extension of the MSE can be formulated by computing both the MSE of the intensity and the MSE of the gradient and weighting the sum, as follows

$$WGMSE(I,J,\alpha) = \alpha MSE(I,J) + \frac{1-\alpha}{2}(MSE(I_x,J_x) + MSE(I_y,J_y))$$

where the gradient components Ix, I_y, J_x and J_y are normalized with respect to the magnitude of the gradient.

(a) Original image, $MSE = 0$, $WGMSE = 0$

(b) Higher exposure, $MSE = 0.04$, $WGMSE < 0.01$

(c) Malfunctioning sensor, $MSE = 0.02$, $WGMSE = 0.47$

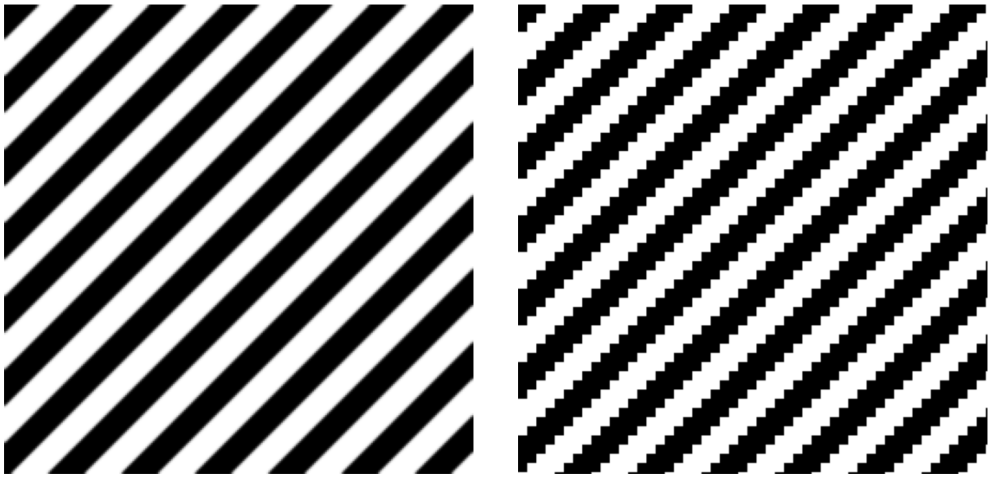
Figure 6.3: A digital image (a) is corrupted by an acquisition at an higher exposure (b) or by a malfunctioning sensing device (c). WGMSE appears to be a more effective metric at assessing corruption in relation to human perception.

6.1.3 MSE^{σ}_{Gs}: MSE on the visual cortex V1

In this section another new metric, part of the author's contribution, is presented.

Since the methods studied in this work are based on a completion model that takes place in the visual cortex V1 modeled by either $SE(2)$ or $PT\mathbb{R}^2$, it does seem natural to introduce an analogous for the MSE in this setting.

The reader might be tempted to take the classic definition of MSE over $\mathbb{R}^3$ and apply it over two lifted images on $SE(2)$ (or $PT\mathbb{R}^2$), but this approach results in a deeply flawed application: consider as an example two patterns of similar lines, one with smooth edges and one approximated only using straight vertical or horizontal edges, as it can often happen due to discretization errors (Figure 6.4). Comparing these two patterns with MSE over the visual cortex V1 seen as $\mathbb{R}^3$ yields maximum error when lifted using the direction of maximum stimulus as in the CPS model. The pixels at the borders of the lines are lifted at completely different angles and the error is at those points maximum.



(a) Smooth lines (b) Rough lines

Figure 6.4: Counterexample to the use of MSE as a metric for over $SE(2)$. The angles of the level curves in (a) are $\frac{\pi}{4}$ and $\frac{5\pi}{4}$ whereas in (b) the edges have angles 0, $\frac{\pi}{2}$, π and $\frac{3\pi}{2}$.

One approach is to change the preferred lifting procedure and, within the scope of this application, use a lift that intrinsically takes into account the excitatory connections between hypercolumns. A particularly well suited lift operator is the *Gaussian lift* which was introduced in Chapter 5.

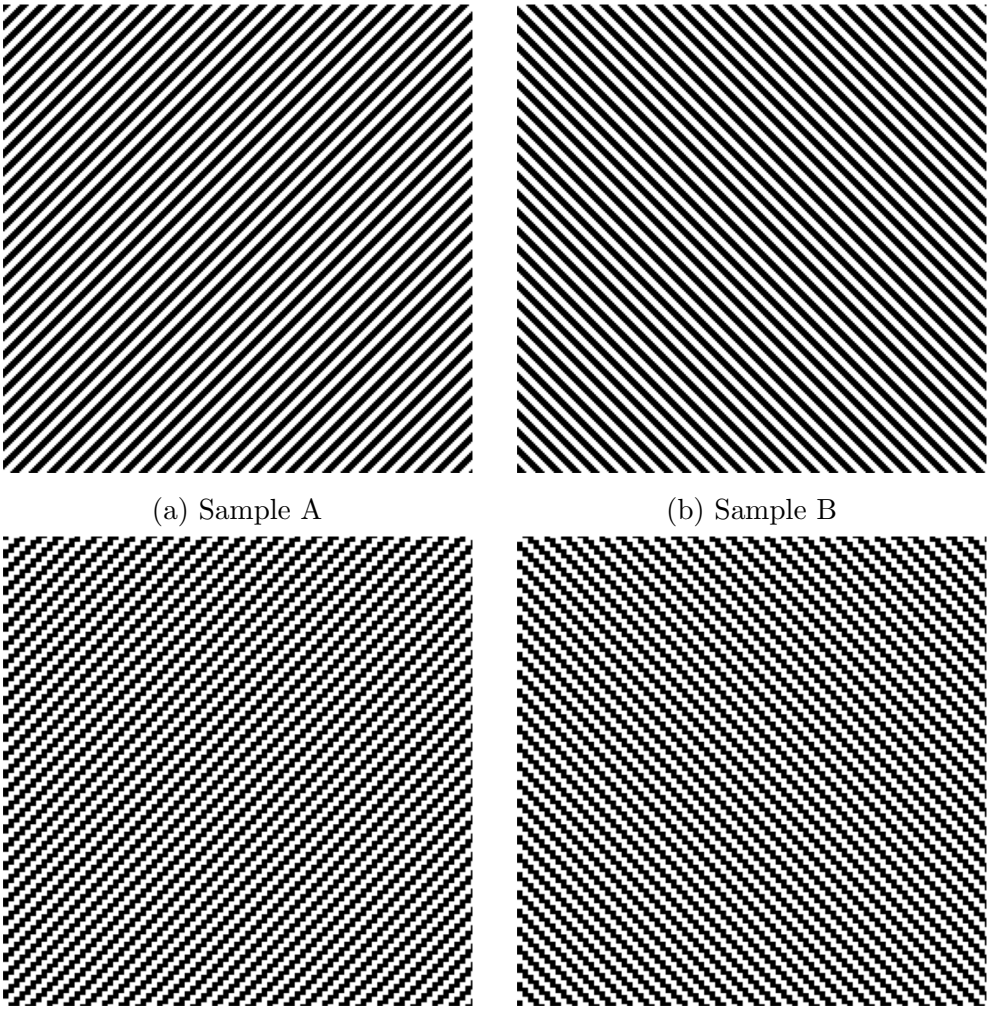
Denote by $\mathcal{L}^{\sigma}_{Gs}$ the Gaussian lift operator with standard deviation σ, then the MSE adapted to $SE(2)$ takes the form

$$MSE^{\sigma}_{Gs}(I, J) = MSE(\mathcal{L}^{\sigma}_{Gs}(I), \mathcal{L}^{\sigma}_{Gs}(J))$$

The analogous result can be obtained for $PT\mathbb{R}^2$ by taking into consideration $\mathcal{L}^{\sigma}_{Gs}$ as the lift to $PT\mathbb{R}^2$.

Although we obtain the same exact result when considering the lift over $SE(2)$ or $PT\mathbb{R}^2$ for small values of σ, from a computational point of view it is more convenient to adopt the latter as the number of operations is halved: the interval is reduced from $[0, 2\pi)$ to $[0, \pi)$.

6.2 Comparison of the metrics



(a) Sample A (b) Sample B

(c) Sample C (d) Sample D

Figure 6.5: Four samples intrinsically different. However, at the level of the visual cortex V1, sample A is similar to sample C and sample B is similar to sample D, whereas all the other combinations produce a very poor comparison.

We can compare the metrics on a specifically crafted set of images (figure 6.5) that makes use of the sensitivity to orientations that the visual cortex V1 provides. To the human brain (the reader can test this on themselves) sample A will be similar to sample C but different from samples B and D, and sample B will be similar to sample D but different from samples A and C. A good metric taking account of human perception should reflect this.

The results obtained are the following:

MSE	A	B	C	D
A	0	·	·	·
B	0.48	0	·	·
C	0.20	0.49	0	·
D	0.50	0.20	0.48	0

SSIM	A	B	C	D
A	0	·	·	·
B	0.04	0	·	·
C	0.60	0.02	0	·
D	0.02	0.60	0.04	0

WGMSE	A	B	C	D
A	0	·	·	·
B	0.78	0	·	·
C	0.30	0.84	0	·
D	0.84	0.30	0.90	0

MSE^{σ}_{Gs}	A	B	C	D
A	0	·	·	·
B	0.48	0	·	·
C	0.20	0.49	0	·
D	0.50	0.20	0.48	0

By computing the results we can see that all the metrics seem to be somewhat fit for the purpose, with $SSIM$ being particularly effective. Surprisingly there seems to be almost no difference (no difference at all when approximating to the second decimal) between the regular MSE and the modified version MSE^{σ}_{Gs} with $\sigma = 0.2$ that was introduced.

List of Symbols

$\mathbb{R}_+$	$[0,+\infty)$
$[a_{ij}]$	Matrix with entries a_{ij}
M_n	Space of n-dimensional matrices
$\mathbf{I}_n$	Identity of M_n
$\mathbf{0}_n$	Zero element of M_n
$GL(n)$	General linear group of dimension n with real entries
$T_p(M)$	Tangent space of M at a point $p \in M$
TM	Tangent space of M
$\mathfrak{X}(M)$	Space of C^∞ vector fields on M
$\gamma(t)$	Curve on a manifold parametrized by t
$F_{*,p}$	Differential at a point p
$\overrightarrow{X}_1$	A section on a manifold
X_1	Vector field associated to the section $\overrightarrow{X}_1$
$L_a(b)$	Left translation
H	Heisenberg group
$[\cdot,\cdot]$	Lie bracket
$\mathfrak{g}$	A Lie algebra (usually generated by the Lie group G)
$\tilde{A}$	Left-invariant vector field generated by A
$\mathcal{L}(G)$	Set of left-invariant vector fields of the Lie Group G
exp	Exponential map
log	Logarithm map
$SE(2)$	Space of rototranslations
$PT\mathbb{R}^2$	$\mathbb{R}^2 \times P^1$, the projective tangent bundle
$\mathcal{T}^k(M)$	Space of covariant tensor fields
$\mathcal{L}(I)$	Lift of the image I to either $SE(2)$ or $PT\mathbb{R}^2$
Δ_R	Anisotropic sub-Laplacian [13]
Δ_H	Anisotropic Laplacian [8]
I, J	Digital images
I_c	Corrupted image

I_s	Smoothed image
$\tilde{I}$	Lifted image (in $SE(2)$ or $PT\mathbb{R}^2$)
Σ_0	Surface over $SE(2)$ or $PT\mathbb{R}^2$
$G(x, y)$	Gaussian noise function
$E_i[\gamma], J[\gamma]$	Cost of the curve γ on $\mathbb{R}^2$
$\mathcal{C}[q]$	Cost of the curve q on $SE(2)$ or $PT\mathbb{R}^2$
$\mathcal{D}_i$	A set of curves
M	A manifold
$\mathcal{F}[g]$	Fourier transform of the function g
$\mathcal{L}^{\sigma}_{Gs}$	Gaussian lift with standard deviation σ
MSE	Mean Squared Error
MSE^{σ}_{Gs}	Mean Squared Error over $PT\mathbb{R}^2$ with lift $\mathcal{L}^{\sigma}_{Gs}$
$SSIM$	Structural Similarity Index Measure
$GSSIM$	Gradient-based Structural Similarity Index Measure
$\partial_x, \partial_t, \ldots$	Partial derivative
D_i	First finite differences derivative
D_{ij}	Second finite differences derivative
$\Delta x, \Delta y \ldots$	Discretization interval
div_R	Divergence in $SE(2)$
∇_R	Gradient in $SE(2)$
$H_R(\Sigma)$	Curvature of the surface Σ in $SE(2)$
ν_R	Normal vector in $SE(2)$
$Hess_R$	Hessian matrix in $SE(2)$

Bibliography

[1] Andrei Agrachev et al. "The intrinsic hypoelliptic Laplacian and its heat kernel on unimodular Lie groups". In: (June 2008). arXiv: 0806.0734 [math.AP].

[2] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing*. en. 2nd ed. Applied Mathematical Sciences. New York, NY: Springer, Dec. 2006.

[3] John G. Avildsen. *The Karate Kid*. 1984.

[4] M.R. Banham and A.K. Katsaggelos. "Digital image restoration". In: *IEEE Signal Processing Magazine* 14.2 (1997), pp. 24–41. DOI: 10.1109/79.581363.

[5] Sarah BM Bell, Fred C Holroyd, and David C Mason. "A digital geometry for hexagonal pixels". In: *Image and Vision Computing* 7.3 (1989), pp. 194–204. ISSN: 0262-8856. DOI: https://doi.org/10.1016/0262-8856(89)90044-9. URL: https://www.sciencedirect.com/science/article/pii/0262885689900449.

[6] U Boscain et al. "Hypoelliptic diffusion and human vision: A semidiscrete new twist". en. In: *SIAM J. Imaging Sci.* 7.2 (Jan. 2014), pp. 669–695.

[7] Ugo Boscain, Grégoire Charlot, and Francesco Rossi. "Existence of planar curves minimizing length and curvature". In: *Proceedings of the Steklov Institute of Mathematics* 270 (June 2009). DOI: 10.1134/S0081543810030041.

[8] Ugo Boscain et al. "Anthropomorphic Image Reconstruction via Hypoelliptic Diffusion". In: *SIAM j. control optim.* 50.3 (Jan. 2012), pp. 1309–1336.

[9] Ugo Boscain et al. "Cortical-inspired image reconstruction via sub-Riemannian geometry and hypoelliptic diffusion". In: *ESAIM Proc. Surv.* 64 (2018), pp. 37–53.

[10] Ugo V Boscain et al. "Highly corrupted image inpainting through hypoelliptic diffusion". en. In: *J. Math. Imaging Vis.* 60.8 (Oct. 2018), pp. 1231–1245.

[11] Alan Bundy and Lincoln Wallen. "Breadth-First Search". In: *Catalogue of Artificial Intelligence Tools*. Ed. by Alan Bundy and Lincoln Wallen. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, pp. 13–13. ISBN: 978-3-642-96868-6. DOI: 10.1007/978-3-642-96868-6_25.

[12] Guan-Hao Chen, Chun-Ling Yang, and Sheng-Li Xie. "Gradient-based structural similarity for image quality assessment". In: *2006 International Conference on Image Processing*. Atlanta, GA: IEEE, Oct. 2006.

[13] G Citti and A Sarti. "A cortical based model of perceptual completion in the Roto-translation space". en. In: *J. Math. Imaging Vis.* 24.3 (May 2006), pp. 307–326.

[14] G Citti et al. "Sub-Riemannian mean curvature flow for image processing". en. In: *SIAM J. Imaging Sci.* 9.1 (Jan. 2016), pp. 212–237.

[15] Gabriel Cruz, Murilo Baldi, and Rafael Amaro Rolfsen. *silentMoire*. URL: `https://github.com/gmelodie/silentMoire` (visited on 03/30/2022).

[16] Elizabeth J. Darrow. "Pietro Edwards and the restoration of the public pictures of Venice , 1778–1819: Necessity introduced these arts". English. PhD thesis. 2000.

[17] L E Dubins. "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents". In: *Amer. J. Math.* 79.3 (July 1957), p. 497.

[18] Lawrence C Evans. *Partial Differential Equations*. 2nd ed. Graduate studies in mathematics. Providence, RI: American Mathematical Society, Mar. 2010.

[19] David Field, Anthony Hayes, and Robert Hesst. "Contour Integration by the Human Visual System: Evidence for a Local "Association Field"". In: 1993.

[20] Bruno Franchi, Raul Serapioni, and Francesco Serra Cassano. "On the structure of finite perimeter sets in step 2 Carnot groups". In: *Journal of Geometric Analysis* 13.3 (Sept. 2003), pp. 421–466. DOI: `10.1007/bf02922053`.

[21] Rafael C Gonzalez and Richard E Woods. *Digital image processing*. 4th ed. Upper Saddle River, NJ: Pearson, 2018.

[22] Thomas Hawkins. "The geometrical origins of lie's theory". In: *Emergence of the Theory of Lie Groups*. New York, NY: Springer New York, 2000, pp. 1–42.

[23] Sigurdur Helgason. *Differential geometry, lie groups and symmetric spaces*. en. Graduate studies in mathematics. Providence, RI: American Mathematical Society, June 2001.

[24] David Hilbert. "Mathematical problems". en. In: *Bull. New Ser. Am. Math. Soc.* 8.10 (1902), pp. 437–479.

[25] Robert K. Hladky and Scott D. Pauls. "Minimal Surfaces in the Roto-Translation Group with Applications to a Neuro-Biological Image Completion Model". In: *Journal of Mathematical Imaging and Vision* 36.1 (Aug. 2009), pp. 1–27. DOI: `10.1007/s10851-009-0167-9`.

[26] Lars Hörmander. "Hypoelliptic second order differential equations". en. In: *Acta Math.* 119.0 (1967), pp. 147–171.

[27] J P Jones and L A Palmer. "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex". en. In: *J. Neurophysiol.* 58.6 (Dec. 1987), pp. 1233–1258.

[28] Shoshichi Kobayashi and Katsumi Nomizu. *Foundations of differential geometry, volume 1*. en. Wiley Classics Library. Nashville, TN: John Wiley & Sons, Feb. 1996.

[29] Rohit Kumar and Vishal Moyal. "Visual image quality assessment technique using FSIM". In: *Int. j. comput. appl. technol. res.* 2.3 (May 2013), pp. 250–254.

[30] John M Lee. *Riemannian Manifolds*. en. 1997th ed. Graduate Texts in Mathematics. New York, NY: Springer, Sept. 1997.

[31] Chaofeng Li and Alan C Bovik. "Three-component weighted structural similarity index". In: *Image Quality and System Performance VI*. Ed. by Susan P Farnand and Frans Gaykema. San Jose, CA: SPIE, Jan. 2009.

[32] S Marcelja. "Mathematical description of the responses of simple cortical cells". en. In: *J. Opt. Soc. Am.* 70.11 (Nov. 1980), pp. 1297–1300.

[33] Richard Montgomery. *A tour of subriemannian geometries, their geodesics and applications*. Mathematical Surveys and Monographs. Providence, RI: American Mathematical Society, Aug. 2006.

[34] Andrew Owens and Alexei A. Efros. "Audio-Visual Scene Analysis with Self-Supervised Multisensory Features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[35] Jean Petitot. *Neurogéométrie de la vision*. Jan. 2008.

[36] Jean Petitot and Yannick Tondut. "Vers une neurogéométrie. Fibrations corticales, structures de contact et contours subjectifs modaux". In: *Math. Sci. Hum. Math. Soc. Sci.* 145 (Mar. 1999).

[37] University of Queensland. *How do neurons work?* URL: `https://qbi.uq.edu.au/brain-basics/brain/brain-physiology/how-do-neurons-work` (visited on 03/30/2022).

[38] Umme Sara, Morium Akter, and Mohammad Shorif Uddin. "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study". In: *J. Comput. Commun.* 07.03 (2019), pp. 8–18.

[39] Richard Stallman. *Various Licenses and Comments about Them, new Python license*. URL: `https://web.archive.org/web/20211227002639/https://www.gnu.org/licenses/license-list.html%20#GPLv2` (visited on 05/02/2022).

[40] Walter A Strauss. *Partial Differential Equations*. en. 2nd ed. Chichester, England: John Wiley & Sons, Dec. 2007.

[41] Loring W Tu. *An introduction to manifolds*. en. 2nd ed. Universitext. New York, NY: Springer, Oct. 2010.

[42] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". en. In: *IEEE Trans. Image Process.* 13.4 (Apr. 2004), pp. 600–612.

[43] Frank W Warner. *Foundations of Differentiable Manifolds and Lie Groups*. en. Graduate texts in mathematics. New York, NY: Springer, Dec. 2010.