MASTER'S THESIS

UNTITLED THESIS

Frank William Hammond Espinosa

Master's Degree in Intelligent Systems (MUSI)
Specialization in Artificial Intelligence.

Centre for Postgraduate Studies

Academic year 2024/25

UNTITLED THESIS

Frank William Hammond Espinosa

MASTER'S THESIS

Centre for Postgraduate Studies

University of the Balearic Islands

Untitled thesis

Frank William Hammond Espinosa

Tutor(s): Julia Navarro Oliver and Ana Belén Petro Balaguer

Thesis of the Master's degree in Intelligent Systems (MUSI)

University of the Balearic Islands, 07122 Palma, Illes Balears, Spain

frank.william.hammond@gmail.com

Abstract—**This is a very abstract abstractionary abstract.**

I. INTRODUCTION

Blabla cites SOTA etc avorrit.

II. MATHEMATICAL MODEL

In this section, all of the mathematical machinery needed to understand and formalize the problem is presented, as well as the iterative mathematical methods used to solve it.

Although the mathematical exposition is, for the most part, either self-contained or specifically referenced, a certain degree of familiarity with modern mathematical analysis is assumed. A minimum familiarity with digital imaging is also assumed.

A. Theoretical grounding and notation

1) Image representations: Two representations of digital images will be used in this work, the equivalences between which will be described below.

The most common representation among computing frameworks is that of multidimensional arrays, sometimes also called *tensors*. We shall refer to this one as the **discrete** representation. Since *pytorch* will be used to implement the algorithms described in this text, the standard 4D representation will be used; that is, images will always have the *Shape* (B, C, H, W), where B is the batch size (1 for a single image), C is the number of channels (3 for an RGB image, 1 for a grayscale one), and H and W are the height and width of the image, respectively.

The other representation used will be called **continuous**[1]. It will be the preferred one in the mathematical formulas of this text. The idea is to think of images as either scalar or three-dimensional vector fields for grayscale or RGB images, respectively, defined on a rectangular domain $\Omega \subseteq \mathbb{R}^2$. Said fields are always piecewise constant on right-semiclosed unit squares with sides parallel to the coordinate axes.

In order to be able to comfortably extend continuous operations for images, it is useful to center the domain Ω with respect to the origin. Concretely, for a batched image I with shape (B, C, H, W), B fields are defined on the same domain Ω. Namely, $I^1, \ldots, I^B \colon \Omega \to \mathbb{R}^C$, where the domain Ω is given by

$$\Omega = (-\lceil \tilde{w} \rceil - 1, \lfloor \tilde{w} \rfloor] \times (-\lceil \tilde{h} \rceil - 1, \lfloor \tilde{h} \rfloor],$$

---

[1]The term *continuous* here refers to the domain of the representation. In fact, the functions involved will almost never be continuous.

with $\overline{w} = \frac{W-1}{2}$ and $\overline{h} = \frac{H-1}{2}$. The relationship between the two representations is given by

$$I[b][c][h][w] = I_{c+1}^{b+1}(w - \lceil \tilde{w} \rceil, h - \lceil \tilde{h} \rceil) \tag{1}$$

where the left term follows a standard 0-indexed array notation, and the right is standard mathematical function notation. Then, the functions I_c^b are extended to the rest of Ω by imposing that they be constant on all right-semiclosed squares $(x - 1, x] \times (y - 1, y]$.

It is also possible to extend the domain of the continuous representations to all of $\mathbb{R}^2$ by simply setting them to 0 outside of Ω. Reciprocally, for a given set of functions $I^1, \ldots, I^B \colon \mathbb{R}^2 \to \mathbb{R}^C$, it is possible to obtain a discrete representation of a digital image by fixing the desired dimensions H and W and sampling as in eq. (1).

It is immediate to see that this identification between the two representations respects pointwise operations (in particular, function sums and products). It is also possible to see that, for an appropriate translation, the convolution defined in *pytorch* with a kernel g coincides with the continuous convolution with (the continuous version of) the inverted kernel $\overline{g}(x) = g(-x)$.

2) L^p spaces: It will be useful to utilize mathematical terminology and machinery that is best exposed in function spaces. The most important and suitable function spaces that will be employed are the L^p spaces.

Definition 1. *Let Ω be a Borel measurable subset of $\mathbb{R}^n$ for some $n \in \mathbb{Z}^+$, and μ the Lebesgue measure on the Borel sets of $\mathbb{R}^n$.[2] For a given measurable function $f \colon \Omega \to \overline{\mathbb{R}}$, where $\overline{\mathbb{R}}$ is the extended real line equipped with its usual topology, define the quantity*

$$\|f\|_p = \left(\int_\Omega |f|^p \, d\mu \right)^{\frac{1}{p}}$$

for each $p \in [1, +\infty)$. Define the set of p-integrable functions as

$$\mathcal{L}^p(\Omega) = \left\{ f \colon \Omega \to \overline{\mathbb{R}} \mid \|f\|_p < +\infty \right\}.$$

It can be shown that $\mathcal{L}^p(\Omega)$ is a normed vector space when equipped with $\|\cdot\|_p$ as a norm.

---

[2]Properly defining the Borel sets and the Lebesgue measure is out of the scope of this work. All rectangles, either open, closed or semiclosed are Borel measurable subsets of $\mathbb{R}^2$. All integrals with respect to the Lebesgue measure coincide with their classical Riemann integral counterpart, whenever the latter is defined.

Finally, define $L^p(\Omega)$ by identifying functions in $\mathcal{L}^p(\Omega)$ that coincide almost everywhere with respect to μ and equipping the resulting equivalence classes with the norm of any of their elements. It is possible to show that $L^p(\Omega)$ is a separable Banach space for any p, and that it is a Hilbert space for $p = 2$.

It is trivial to see that the continuous representation of a digital image I with domain Ω is always an element of $L^p(\Omega)$ for any $p \in [1, +\infty)$.

3) Functional and convex analysis: It is possible to develop in a surprising generality a number of optimization algorithms which can later be applied for image processing problems. The continuous representation of digital images described above will allow the work with operators defined on function spaces in a more abstract fashion than with multidimensionals arrays. By doing so, the involved operators and formulae obtained often have much cleaner closed forms.

In this section, a few relevant definitions are presented. Most importantly, the direct method is introduced and proved. This theorem will be employed later to show the well-posedness of the presented minimization problems. Some results are given without proof; the interested reader can consult chapters 1 to 3 of [2].

Definition 2. *A normed vector space X is said to be a **Banach space** if it is complete under the metric induced by its norm. A **functional** on X is a function $F: X \to \mathbb{R} \cup \{+\infty\}$.*

Definition 3. *The **dual space** of X is defined as the set of all continuous linear functionals on X with codomain on $\mathbb{R}$, where continuity is to be interpreted with respect to the topology induced on X by its norm and the usual topology in $\mathbb{R}$. We denote it as X^*. It can be shown that X^* is a Banach space when equipped with the operator norm*

$$\|x^*\|_{X^*} = \sup_{x \in X, \|x\|=1} \|x^*(x)\|.$$

*We say that a given sequence $\{x_n\}_{n \in \mathbb{N}} \subseteq X$ **converges weakly** to a given $x \in X$ if the sequence $\{f(x_n)\}_{n \in \mathbb{R}}$ converges (in $\mathbb{R}$) to $f(x)$ for every $f \in X^*$.*

It is immediate to check that the usual convergence of a sequence in X implies its weak convergence to the same limit.

Definition 4. *A given functional F on a Banach space X is said to be **weakly lower semicontinuous** if, for each weakly converging sequence $\{x_n\} \subseteq X$ with weak limit $\lim_n x_n \in X$,*

$$F(\lim_n x_n) \leq \liminf_n F(x_n).$$

Importantly, the norm of X is a weakly lower semicontinuous functional.

Definition 5. *A given functional F on X is said to be **convex** whenever the condition*

$$F(x + (1 - \lambda)y) \leq \lambda F(x) + (1 - \lambda)F(y)$$

*holds for any $x, y \in X$ and any $\lambda \in (0, 1)$. If, additionally, the inequality is strict whenever $x \neq y$, F is said to be **strictly convex**.*

Definition 6. *A functional F is said to be **coercive** if, for every sequence $\{x_n\} \subseteq X$ such that $\lim_n \|x_n\| = +\infty$, then $\lim_n F(x_n) = +\infty$.*
*It is said to be **proper** if there exists some $x \in X$ such that $F(x) < +\infty$.*

Definition 7. *Let X be a Banach space. The **bidual** space of X, denoted by X^{**}, is the dual space of its dual space, that is, $X^{**} = (X^*)^*$. There is a canonical linear mapping from X to X^{**} given by*

$$x \mapsto (x^* \mapsto x^*(x))$$

*which can be shown to always be isometric and therefore injective. X is said to be **reflexive** whenever the canonical mapping above is also surjective.*

It is a direct consequence of the Banach-Alaoglu theorem (which is omitted here for brevity) that every bounded sequence in a reflexive Banach space has a weakly converging subsequence.

Theorem 1 (The direct method). *Let X be a reflexive Banach space and let F be a proper, coercive and weakly lower semi-continuous functional on X. Then, the minimization problem*

$$\min_{x \in X} F(x)$$

admits a solution. Moreover, if F is strictly convex, the solution is unique.

Proof. For the existence, note that since F is proper, $\inf_{x \in X} F(x) < +\infty$. It is therefore possible to construct a sequence $\{x_n\}_n \subseteq X$ such that $F(x_n) < +\infty$ for all n and $F(x_n) \to \inf_{x \in X} F(x)$. Since $\{F(x_n)\}_n$ is a converging sequence of real numbers, it is bounded. This, together with the coerciveness of F, implies that $\{x_n\}_n$ is bounded in norm. Since X is reflexive, it has a weakly converging subsequence. Denote the subsequence by $\{\overline{x}_n\}_n$ and its limit by $\overline{x} \in X$. Then, utilizing the weak lower semicontinuity of F:

$$\inf_{x \in X} F(x) \leq F(\overline{x}) = F(\lim_n \overline{x}_n) \leq \liminf_n F(\overline{x}_n) = \inf_{x \in X} F(x).$$

This shows that $\overline{x}$ is a minimizer of F.
The uniqueness for the strict convexity is immediate by contradiction. $\square$

Finally, a series of "bespoke" lemmas with simple proofs will be useful further ahead.

Lemma 1. *The sum of two weakly lower semicontinuous functionals is weakly lower semicontinuous.*

Proof. Immediate using the definition and the fact that $\liminf_n a_n + \liminf_n b_n \leq \liminf_n a_n + b_n$. $\square$

Lemma 2. *Let X and Y be Banach spaces. Let $F: Y \to \mathbb{R} \cup \{+\infty\}$ be a weakly lower semicontinuous functional, and $K: X \to Y$ a continuous function. Then, $F \circ K$ is weakly lower semicontinous.*

Proof. Since K is continuous, $K(\lim_n x_n) = \lim_n K(x_n)$ for any converging sequence $\{x_n\}_n \subseteq X$. The result is immediate by applying the definition of weak lower semicontinuity. $\square$

B. Convex optimization

Although necessary to establish the existence of solution, the direct method offers no way to actually compute the minimum of a posed problem. Other techniques, which often involve iterative algorithms, are useful for this. Notably, the Chambolle-Pock primal-dual will be used in this work. The rest of this section is devoted to briefly present the concepts necessary to enunciate it. For a more detailed explanation, the reader is referred to chapters 3 to 7 and section 8.4 of [2].

Definition 8. *Let X be a Banach space and F be a proper functional on X. Define $F^*\colon X^* \to \overline{\mathbb{R}}$ by*

$$F^*(x^*) = \sup_{x \in X} x^*(x) - F(x).$$

*This function is called the **Fenchel conjugate** or **convex conjugate** of F.*

If Y is another Banach space, and $A\colon X \to Y$ is a continuous linear operator, then a linear operator $A^*\colon Y^* \to X^*$ is defined by

$$(A^*y^*)(x) = y^*(Ax).$$

*This is called the **conjugate** operator of A[3].*

Definition 9. *Let X be a Banach space and F be a convex, weakly lower semicontinuous functional on X that is either bounded below or proper. For a given $\tau > 0$, define the **proximity operator** of F with step size τ as*

$$\mathrm{prox}_{\tau F}(x) = \operatorname*{argmin}_{y \in X} F(y) + \frac{\|x - y\|^2}{2\tau}.$$

This operator is well defined by theorem 1[4].

Theorem 2 (Chambolle-Pock algorithm). *Let $F\colon X \to \overline{\mathbb{R}}$, $G\colon Y \to \overline{\mathbb{R}}$ be proper, convex and lower semicontinuous functionals, and let $A\colon X \to Y$ be linear and continuous. If the problem*

$$\min_{x \in X} F(x) + G(Ax)$$

has a solution, then the sequence $\{x_n\}_n \subseteq X$ defined by the following iterates converges to it:

$$\begin{cases} x_{n+1} = prox_{\tau F}(x_n - \tau A^* z_n) \\ \tilde{x}_{n+1} = 2x_{n+1} - x_n \\ z_{n+1} = prox_{\sigma G^*}(z_n + \sigma A\tilde{x}_{n+1}) \end{cases} \qquad (2)$$

C. Physical model and formalization of the problem

1) Underwater Image Formation Model: When light travels in a water medium, two relevant physical phenomena different from those found in an aerial medium appear: first, the intensity in this light is absorbed by the water molecules in a significant manner. Such absorption is also dependent on the wavelength of the light, and is perceived by the human vision

---

[3]Although the notation and name used are the same, these are different concepts. In this text there will be no ambiguity with these notations because the codomain of a linear operator whose conjugate is needed will never be $\mathbb{R}$.

[4]It is possible to define this operator in greater generality, but doing so involves theory about subdifferentials and monotone operators that is out of the scope of this work.

---

system (HVS) as the red color dimming much faster than others. The second phenomenon is *scattering*, whereby light does not travel in a straight line from the object to the camera forming an infinitely thin beam, but it is reflected on water molecules instead, forming a cone. This is perceived by the HVS as the image presenting a glow-like or hazy appearance. Classical works in underwater imaging by McGlamery [4] and Jaffe [3] effectively quantify these phenomena and model the formation of an underwater image I as a linear superposition of three components, in what is often referred to as the Underwater Image Formation Model (UIFM) [?]:

$$I = I_d + I_{fs} + I_{bs}.$$

The I_d (*direct*) term corresponds to the light that travels in a straight line from the object to the camera; the I_{fs} (*forward scattering*) component corresponds to light that travels from the object to the camera but is scattered in a small angle; and the I_{bs} (*backward scattering*) term corresponds to light that travels directly from the light source to the camera by being scattered in a large angle.

Notably, the forward-scattering component can be obtained from the direct component via a convolution with a *point-spread function g.*

These classical models are based purely on physical properties and depend heavily on the conditions of the water and camera, presenting several constants that need to be estimated experimentally. This poses a disadvantage when working with underwater images taken in unknown conditions, as is often the case in general-purpose underwater image processing algorithms.

For this reason, some modifications and simplifications of the original models have been proposed over the years. Notably, similar notation to the following is used commonly in modern literature [?, ?]: let I be the captured image (i.e., the input), J the scene radiance (i.e., the desired image), t the transmission map, g the point-spread function and B the background light. The transmission map decays exponentially with the distance between the captured object and the camera, in a rate that is dependent on the light wavelength.

For an RGB image, the UIFM can then be expressed as follows, for each color channel $c \in \{r, g, b\}$ and pixel x,

$$I^c(x) = J^c t^c(x) + (g^c * (J^c t^c))(x) + B^c(1 - t^c(x)). \quad (3)$$

Each addend in the above formula corresponds to one component of the original UIFM, order of appearance both in the equation and the text.

In this work, a further modification of the model is made. By noting that summing the term $J^c t^c$ is equivalent to summing $\delta * J^c t^c$ (where δ denotes a Dirac delta), and using the linearity of the convolution, it is possible to combine the first two addends into one without losing generality. Additionally, a noise component is added, which is common in the literature [?]. The resulting equation is as follows:

$$I^c(x) = B^c(1 - t^c(x)) + (g^c * (J^c t^c))(x) + \xi^c(x), \quad (4)$$

where $\xi(x)$ is random noise on the pixel x, which will be modelled as white noise (i.i.d. centered gaussians). The

function g is no longer a point-spread function but a more abstract convolution kernel instead.

From an image processing standpoint, the desired image J is ideally obtained by inverting one of the formulas above from the captured image I. However, due to the ill-posedness and analytical untractability of the deconvolution problem involved, it is convenient to simplify the formula by assuming g^c to be a Dirac delta for each channel c. The resulting equation is

$$I^c(x) = B^c (1 - t(x)) + (J^c t)(x) + \xi^c(x), \quad (5)$$

This is is equivalent to ignoring the forward-scattering component and a common practice in the literature [?]. It is often called "simplified UIFM", and can still produce good results, but is nevertheless a biased model which completely neglects potentially impactful physical phenomena involved in the formation of the image.

D. Formalization of the problem

Our approach consists of firstly estimating B, t and a preliminary version of J with traditional methods by assuming the simplified equation eq. (5) holds. Then, we utilize this preliminary version as initialization of an unfolded variational approach to invert the full eq. (4).

More concretely, since t is fixed, we perform a substitution $Jt \rightarrow J$ (which is to be inverted later) and pose the following minimization problem, separately for each channel $c \in \{r, g, b\}$:

$$\underset{(J,g)\in D}{\operatorname{argmin}} E(J, g) = \frac{1}{2}\|J * g - I_2^c\|_2^2 + \mathcal{R}_1(J) + \mathcal{R}_2(g), \quad (6)$$

where $D = L^2(\Omega) \times L^2(\Omega)$, and $\mathcal{R}_1$ and $\mathcal{R}_2$ are some continuous regularization functions from $L^2(\Omega)$ to $\mathbb{R}$.

This problem is still mathematically intractable, since the resulting functional is not convex due to the convolution being bilinear w.r.t. the variable functions. A further simplification is made, whereby the functional is alternatively minimized by each of the two variables separately instead of jointly. To compensate for this simplification, the process is repeated a few times:

$$\begin{cases} g_{m+1}^c = \underset{g \in L^2(\Omega)}{\operatorname{argmin}} E(J_m^c, g) \\ J_{m+1}^c = \underset{J \in L^2(\Omega)}{\operatorname{argmin}} E(J, g_{m+1}^c) \end{cases} \quad (7)$$

Henceforth, the iterations resulting from this simplification will be indexed with the letter m and referred to as **greedy iterations**.

Due to the symmetry of the target functional E, both subproblems can be studied under the same schema. To see this, consider the following minimization problem: let $s, y \in L^2(\Omega)$ and $R: L^2(\Omega) \rightarrow \mathbb{R}$. Fixed those symbols, the idea is to solve for

$$\underset{x \in L^2(\Omega)}{\operatorname{argmin}} F(x) = \frac{1}{2}\|x * y - s\|_2^2 + R(x).$$

Both problems in eq. (7) can be trivially reduced to one of this form by appropriately setting s to I_2, and y and R to either J_m and $\mathcal{R}_2$ or g_{m+1} and $\mathcal{R}_1$. Thus, studying this problem will yield solutions for both subproblems above.

First, it is necessary to show that it is well-posed, meaning that the operations taking place are well-defined and a solution exists. Define the auxiliary linear operator $K: L^2(\Omega) \rightarrow L^2(\Omega)$ given by

$$Kx = x * y..$$

To see that the codomain of this function is, in fact, L^2, we employ the Young inequality together with the well-known fact that $L^2(\Omega) \subseteq L^1(\Omega)$ for a finitely-measured Ω:

$$\|x * y\|_2 \le \|x\|_2 \|y\|_1 < +\infty.$$

This also implies the boundedness of K.

Also define $N: L^2(\Omega) \rightarrow \mathbb{R}$ by $N(a) = \|a - s\|_2^2$, which is clearly strictly convex. In this way, we can express

$$F(x) = N(Kx) + R(x),$$

which is a problem of the same shape of that of theorem 2. Note that N is coercive, convex, lower semicontiuous and, in this case, always finite-valued. Convexity and lower semicontinuity imply weak lower semicontinuity.

Therefore, for a proper and weakly lower semicontinuous regularization R that is either coercive or bounded below, $F = N \circ K + R$ is coercive, proper and weakly lower semicontinuous by lemmas 1 and 2, and the problem admits a solution by theorem 1. For a non-null value of y, the function K is injective (cf. proposition 2 in the appendix), and therefore the strict convexity of N implies that of $N \circ K$, and if R is convex, then F is strictly convex. Therefore, the solution is unique. Moreover, the hypotheses of the Chambolle-Pock algorithm are satisfied. It yields the following iterates:

$$\begin{cases} x_{n+1} = \operatorname{prox}_{\tau R}(x_n - \tau K^* z_n) \\ \tilde{x}_{n+1} = 2x_{n+1} - x_n \\ z_{n+1} = \operatorname{prox}_{\sigma N^*}(z_n + \sigma K \tilde{x}_{n+1}) \end{cases} \quad (8)$$

Henceforth, the iterations in a primal-dual schema will be indexed with the letter n and referred to as **stages**.

Closed-form expressions for K^* and $\operatorname{prox}_{\sigma N^*}$ can be derived analytically (cf. propositions 3 and 4 in the appendix). They are given by

$$\operatorname{prox}_{\sigma N^*}(x) = \frac{x - \sigma s}{\sigma + 1} \quad \text{and} \quad K^* z = z * \overline{y}, \quad (9)$$

where $\overline{y}(x) = y(-x)$.

The only "missing" symbol is $\operatorname{prox}_{\tau R}$, but this depends entirely on the regularization function used and has no general closed form.

We now discuss precisely which regularization is to be used for each subproblem. Going forward, the variable g will be called *kernel* and the variable J will be called *image*[5].

---

[5] Here, J does not correspond to the radiance (i.e. "recovered *image*"), but J/t does. This is just convenient notation.

1) Kernel regularization: For the kernel g, it is desirable to impose three constraints:

1) That it is a convolution kernel, i.e., nonnegative and $\int_\Omega g \ d\mu = 1$. The assumption that the intensity of g be 1 is motivated by the empirical fact that the simplified UIFM, which is equivalent to setting $g = \delta$, often yields acceptable results. Convolving by a kernel with a different total intensity (in a practical sense, $\int_\Omega \delta \ d\mu = 1$) would alter the intensity of the output image J. This was the case in early experiments where this condition was not imposed.

2) That it vanishes outside a square much smaller than Ω. This condition is ubiquitous for convolution kernels in computer vision, as it corresponds to their discrete representations having much smaller sizes than the images'.

3) That it is a decreasing function of its radius. That is, that there exist a decreasing function $\varphi \colon [0, +\infty) \to \mathbb{R}$ such that $g(x) = \varphi(\|x\|)$ for all $x \in \Omega$. This assumption is motivated by two intuitions: first, that there are no privileged directions in an underwater setting, i.e. g is rotationally invariant, or, equivalently, a function of its radius; second, that the larger the angle of the scattering, the larger the distance that light must travel from the object to the camera, and therefore the more intensity is absorbed from it.

These constraints can all be expressed via a regularization function as follows: let

$$\mathcal{K} = \{f \in L^2(\Omega) | f \geq 0, \|f\|_1 = 1\}.$$

Then, condition 1 is equivalent to $g \in \mathcal{K}$. Additionally, for a fixed $r \in \mathbb{Z}^+$ such that $[-r, r]^2 \subseteq \Omega$, define

$$\mathcal{S}_r = \{f \in L^2(\Omega) | f \equiv 0 \text{ in } \Omega \setminus [-r, r]^2\}.$$

Condition 2 is equivalent to $g \in \mathcal{S}_r$ for some sufficiently small r. Finally, define

$$\mathcal{D} = \{\varphi \circ \| \cdot \|_2 \mid \varphi \colon [0, +\infty) \to \mathbb{R} \text{ is decreasing}\},$$

so that condition 3 is equivalent to $g \in \mathcal{D}$.

It is trivial to check that the sets $\mathcal{K}$, $\mathcal{S}_r$ and $\mathcal{D}$ are convex. It is also possible to see that they are closed in $L^2(\Omega)$ (cf. proposition 1), and hence their intersection is closed and convex. Thus, the indicator function given by

$$\delta_{\mathcal{K} \cap \mathcal{S}_r \cap \mathcal{D}}(g) = \begin{cases} +\infty, & \text{if } g \in \mathcal{K} \cap \mathcal{S}_r \cap \mathcal{D}, \\ 0, & \text{otherwise}, \end{cases}$$

is convex and weakly lower semicontinuous by Lemma 2.5.ii) of [2]. It is bounded below, and thus, taking $R = \delta_{\mathcal{K} \cap \mathcal{S}_r \cap \mathcal{D}}$ would guarantee existence of solution to the problem.

However, computing the proximity operator resulting would be challenging, so a final simplification is made. Simply, g is taken as a centered gaussian density truncated to $[-r, r]^2$ (that is, set to 0 outside of $[-r, r]^2$) and normalized so that its integral is 1. This is a sufficient condition for those outlined above, and allows for an important simplification of the optimization scheme: instead of performing primal-dual iterations, a simple gradient descent will be used to optimize the standard deviation of the gaussian density.

Such simplification may not reach the optimum guaranteed to exist with the conditions above, but will be empirically sufficient, likely faster to converge and much simpler from an implementation standpoint.

2) Image regularization: Instead of fixing a regularization for the image, the iterative schema eq. (8) will be unfolded. Concretely, the proximity operator $\text{prox}_{\tau R}$ will be replaced by neural networks that will be then trained end-to-end. Mathematically, the resulting schema is

$$\begin{cases} x_{n+1}^m = \mathcal{N}_n^m(x_n^m - \tau K^* z_n^m) \\ \tilde{x}_{n+1}^m = 2x_{n+1}^m - x_n^m \\ z_{n+1}^m = \text{prox}_{\sigma N^*}(z_n^m + \sigma K \tilde{x}_{n+1}^m) \end{cases}, \qquad (10)$$

where $\mathcal{N}_n^m$ is a neural network specific for stage n in the greedy iteration m. Although this deviates more from the original primal-dual schema than using a fixed neural network would, it has been shown to yield significantly better results [1].

Instead of iterating until convergence, the number of stages and greedy iterations will be fixed beforehand.

III. ALGORITHMIC APPROACH

A. First half of the problem (I_1).

Falta: explicar RCP; potser DCP

The way of inverting the formula in **??** is the same as in (CITAR ARTICLE REFERÈNCIA), although the implementation is fully self-made (the original implementation is written in MATLAB and, to the best of the author's knowledge, only publicly available in binary form).

First, the background light is chosen by iteratively splitting the image into four regions. Out of those regions, the one with the highest score, given by the average minus the standard deviation of the pixel values in the region, is selected and split again. The process stops once the selected region is smaller than a given threshold (16 colored pixels in our implementation). Then, the background light of the image is chosen as the pixel closest to white (in euclidean norm) in that small patch. This process is summarized in Algorithm 1.

Falta explicar com es calcula es transmission map i es guided filter. He posat s'algoritme pes transmission map a 2.

Once the background light B has been computed, it is possible to estimate the trasmission map t. We proceed in a similar fashion to (CITAR ART. REF): first, a coarse version of the transmission map t_0 is obtained via

$$t_0(x) = 1 - \min_{y \in \Omega_x} \left(\min \left(\frac{1 - I^r(y)}{1 - B^r}, \frac{I^g(y)}{B^g}, \frac{I^b(y)}{B^b}, \lambda S(y) \right) \right),$$

where the *saturation map* S is defined as 0 for black $(0, 0, 0)$ pixels and $S = 1 - \frac{\min(I^r, I^g, I^b)}{\max(I^r, I^g, I^b)}$ otherwise.

Then, a fine version of the transmission map, t, is obtained by passing t_0 through a guided filter with I^r as the guide. Cal explicar què fa es guided filter? O basta citar?

Finally, we estimate a first version of J as follows, for each color channel $c \in \{r, g, b\}$:

$$J^c(x) = \frac{I^c(x) - B^c}{\max(t(x), 0.1)} + (1 - B^c) \cdot B^c \qquad (11)$$

Algorithm 1: Estimate background light.

Data: I
Result: J
img $\leftarrow I$;
while *Size(img) > min_size* **do**
    quadrants $\leftarrow$ Split(img);
    max_score $\leftarrow 0$;
    for *q in quadrants* **do**
        $\mu \leftarrow$ Mean(q);
        $\sigma \leftarrow$ Std(q);
        score $\leftarrow \mu - \sigma$;
        if *score > max_score* **then**
            max_score $\leftarrow$ score;
            img $\leftarrow q$;
        end
    end
end

Algorithm 2: Estimate scene radiance and transmission map.

Data: Input image I, background light B, TM patch radius r_t, guided filter patch radius r_g
Result: Radiance estimation J_0, TM estimation t
; /* Compute coarse TM   */
$S \leftarrow$ Saturation(I);
Initialize t with the width and height of I;
for *pixel x* **do**
    ; /* Ω_x: patch around x, radius r_t */
    r_min $\leftarrow \min_{y \in \Omega_x}(1 - I^r[\mathrm{y}])/(1 - B^r)$;
    g_min $\leftarrow \min_{y \in \Omega_x}(I^g[\mathrm{y}])/B^g$;
    b_min $\leftarrow \min_{y \in \Omega_x}(I^b[\mathrm{y}])/B^b$;
    s_min $\leftarrow \min_{y \in \Omega_x}(S[\mathrm{y}]) \cdot \lambda$;
    $t[x] \leftarrow 1 - \min$(r_min, g_min, b_min, s_min)
end
; /* Refine transmission map   */
$t \leftarrow GuidedFilter(I^r, t; r_g)$;
; /* Estimate J_0 */
for *channel c* **do**
    $J_0^c \leftarrow (I^c - B^c)/(max(t, 0.1)) + (1 - B^c) \cdot B^c$
end
return J_0, t

B. Second half of the problem (I_2).

In order to solve the minimization subproblems in eq. (7), the iterative schema in eq. (8) with the formulae in eq. (9) is used, with the aforementioned symbolic substitutions. The missing operator $\text{prox}_{\tau R}$ is unfolded, meaning that it is substituted by a neural network instead of explicitly derived. The architecture and training of said neural network will be discussed later.

The final algorithm used for obtaining an approximate solution to eq. (6) is summarized in Algorithm 3.

Algorithm 3: Solve second half of the problem.

Data: I_2, t, J_0
Data: σ, τ
Result: J
$J \leftarrow J_0$;
Initialize g;
Initialize dual variables $\tilde{g}, \tilde{J}$;
for $n = 1$ **to** G **do**
    ; /* Fix J, estimate g.   */
    $\overline{J} \leftarrow DoubleFlip(J)$;
    for $s = 1$ **to** S **do**
        $tmp \leftarrow NeuralNet_{ns}^g \left(g - \tau \cdot \tilde{g} * \overline{J}\right)$;
        $g \leftarrow 2 \cdot tmp - g$;
        $\tilde{g} \leftarrow (\tilde{g} + \sigma \cdot (g * J) - \sigma \cdot I_2)/(\sigma + 1)$;
    end
    ; /* Fix g, estimate J.   */
    $\overline{g} \leftarrow DoubleFlip(g)$;
    for $s = 1$ **to** S **do**
        $tmp \leftarrow NeuralNet_{ns}^J \left(J - \tau \cdot \left(\tilde{J} * \overline{g}\right)\right)$;
        $J \leftarrow 2 \cdot tmp - J$;
        $\tilde{J} \leftarrow (\tilde{J} + \sigma \cdot (J * g) - \sigma \cdot I_2)/(\sigma + 1)$;
    end
end
return J / t

C. Implementation

Xerrar de U2Fold, la implementació en pytorch dels algoritmes d'abans i de la UNET (he de mirar bé si lo que faig és realment una UNET pq això de sumar en comptes de concatenar és raro), etc.

IV. EXPERIMENTATION

A. Algorithmic selection of hyperparameters

Given the high computational costs of training neural networks, it is unfeasible to perform a grid search on a substantial set of hyperparameter choices and train a network from scratch for each choice.

As an attempt to overcome this limitation, the following approach is proposed. For a given neural network architecture:

1) Determine, through a manual process, a sensible hyperparameter combination as a starting point. This includes the choice of **training-related** hyperparameters, such as the optimizer, learning rate scheduler, etc., and **architectural** hyperparameters, such as the number of layers, their sizes, activation functions...

2) Fix the training-related hyperparameters of the starting point, and for a given set of choices for the architectural hyperparemters (which should include the starting point), train a model on each choice for a small number of epochs. Select the "best" one by minimizing a target metric.

3) Fix the architectural parameters of the best neural network so far, and, similarly, select the best choice of training-related hyperparameters by training a model on each choice. This time, however, the number of epochs

should be larger, since some learning rate schedulers show performances greatly dependent on the number of epochs.

4) Finally, train the best model so far during many epochs as an attempt to get the best performance.

Of course, the target metric plays a central role in determining the output choice. Several metrics will be used to evaluate the quality of the model (cf. [TODO: AÑADIR SECCIÓN DE MÉTRICAS]), but it would be desirable to obtain a single number that will allow the automatic comparison of models. An obvious choice is a simple addition of the metrics, but this presents the problem [TODO: ACABAR]

Table I
CALIBRATION RESULTS FOR DIFFERENT METRICS AND LOSS TERMS WITH THE UIEB DATASET.

Metric or loss	Average	Standard deviation
UCIQEm	10.59041	2.16975
TV	0.05879	0.03238
PSNRm	0.05977	0.02899
DSSIM	0.11068	0.06204
MSE	0.02561	0.02265
CCSm	0.03598	0.03860
Fidelity loss	0.03710	0.02991

V. CONCLUSIONS

VI. APPENDIX

Proposition 1. *The sets $\mathcal{K}$, $\mathcal{S}_r$ and $\mathcal{D}$ are all closed.*

Proof. Let S be one of the sets above, and let $\{f_n\}_n \subseteq S$ denote a sequence converging in $L^2(\Omega)$ to a given limit $f \in L^2(\Omega)$. Showing that the set S is closed is equivalent to showing that $f \in S$.

A well-known result about L^p spaces states that L^p convergence implies pointwise convergence up to a subsequence. This result will be used in all three cases. For notational simplicity, simply take f_n as the converging subsequence. citar

First, consider $S = \mathcal{K}$. The result above immediately implies that f is nonnegative by taking limits on the inequality $f_n(x) \geq 0$. Another well-known result about $L^p(\Omega)$ spaces states that, whenever $\mu(\Omega) < +\infty$, the following inequality holds for $1 \leq p < q < +\infty$ and any measurable function g:

$$\|g\|_p \leq \mu(\Omega)^{1/p-1/q}\|g\|_q.$$

citar Thus, for $p = 1$, $q = 2$,

$$\|f_n - f\|_1 \leq \mu(\Omega)^{1/2}\|f_n - f\|_2.$$

Taking limits, this implies that f_n converges to f in L^1. As a consequence, $\|f\|_1 = \lim_n \|f_n\|_1 = \lim_n 1 = 1$.

Second, for $S = \mathcal{S}_r$, the result is immediate by taking limits on the converging subsequence outside of $[-r, r]^2$.

Finally, for $S = \mathcal{D}$, consider, for each f_n, the function ϕ_n such that $f_n = \phi_n \circ \|\cdot\|_2$. Take some $x \in L^2(\Omega)$ such that $\|x\|_2 = 1$. Then, for each $t \in [0, +\infty)$,

$$\phi_n(t) = f_n(tx).$$

This defines a function ϕ by $\phi(t) = \lim_n \phi_n(t) = \lim_n f_n(tx) = f(tx)$. Since x was arbitrary, it also shows that $f = \phi \circ \|\cdot\|_2$ as desired. □

Proposition 2 (Injectivity of convolution)**.** *Suppose that g is the continuous representation of a non-null digital image. Then, the function $K: L^2(\Omega) \to L^2(\Omega)$ given by $Kx = x * g$ is injective.*

Proof. Since the mapping has been shown to be well-defined and linear, it suffices to show that $ker(K) = \{0\}$; that is, that $Kx = 0$ a.e. implies $x = 0$ a.e.

This proof will make use of the Titchmarsh-Lions convolution theorem, which states that for any two compactly supported distributions T_1, T_2, the convex hull of $supp(T_1 * T_2)$ is the sum of the convex hulls of $supp(T_1)$ and $supp(T_2)$. citar libro Distributions and Fourier Transforms. Properly defining distributions, convolutions between them or their supports is out of the scope of this work, but it suffices to know that any $L^p(\Omega)$ function for $p \geq 1$ and a bounded measurable set Ω defines a distribution, and that the definitions given for distributions correspond exacty to those given for $L^p(\Omega)$ functions, the latter being much simpler.

The support of a function $f \in L^p(\Omega)$ in this context is defined as the smallest closed subset of $\mathbb{R}^2$ such that f is 0 almost everywhere outside of it. Formulaically:

$$supp(f) = \bigcap_{A \text{ closed}, f=0 \text{ a.e. in } A^c} A,$$

where f is extended to be 0 outside of Ω.

It is immediate to see that, for a bounded and measurable set $\Omega \subseteq \mathbb{R}^2$, an $L^p(\Omega)$ function has a compact support, and that said is empty if, and only if, the function is null almost everywhere. It is also immediate to see that a subset of $\mathbb{R}^n$ is empty if, and only if, its convex hull is empty.

Thus, if x is such that $x * g = 0$ a.e., its support is empty, and so is its convex hull. This implies that either the convex hull of $supp(g)$ or the convex hull of $supp(x)$ are empty. By hypothesis, g is not null, and hence it must be that x is zero almost everywhere. □

In the following two propositions, the usual identification between a Hilbert space and its dual via $x \mapsto (y \mapsto \langle x, y \rangle)$ is implicitly assumed.

Proposition 3. *Let $N: L^2(\Omega) \to \mathbb{R}$ be defined by*

$$N(x) = \|x - a\|_2^2$$

for some $a \in L^2(\Omega)$. Then,

$$prox_{\sigma N^*}(x) = \frac{x - \sigma s}{\sigma + 1}.$$

Proof. By the Moreau decomposition theorem,

$$\text{prox}_{\sigma N^*}(x) = x - \sigma \text{prox}_{N/\sigma}\left(\frac{x}{\sigma}\right). \quad (12)$$

Thus, it is sufficient to compute $\text{prox}_{\lambda N}$ for an arbitrary $\lambda > 0$ and then replace $\lambda = \frac{1}{\sigma}$.

For a fixed $z \in L^2(\Omega)$, consider the functional $J_{z,\lambda}(y) = \frac{\|y-z\|_2^2}{2\lambda} + \frac{1}{2}\|y - s\|_2^2$, so that

$$\text{prox}_{\lambda N}(z) = \text{argmin}_y J_{z,\lambda}(y).$$

Clearly, $J_{z,\lambda}$ is Gâteaux differentiable and

$$J_{z,\lambda}(y; h) = \langle h, \frac{1}{\lambda}(y - z)\rangle + \langle h, y - s\rangle.$$

Whence $\forall h\colon J_{z,\lambda}(y;h) = 0$ if, and only if, $y = \frac{z+\lambda s}{\lambda+1}$. Thus, by the Fermat principle, and taking $\lambda = \frac{1}{\sigma}$,

$$\mathrm{prox}_{N/\sigma}(z) = \mathrm{argmin}_y J_{z,1/\sigma}(y) = \frac{\sigma z + s}{\sigma + 1}.$$

Combining this last equation with eq. (12), we have that

$$\mathrm{prox}_{\sigma N^*}(x) = x - \frac{\sigma}{\sigma+1}(x+s) = \frac{x - \sigma s}{\sigma + 1}. \quad \square$$

Proposition 4. *Let* $K\colon L^2(\Omega) \to L^2(\Omega)$ *be defined by*

$$Kx = x * y$$

for some $y \in L^2(\Omega)$. *Then, the dual operator* K^* *is given by*

$$K^*x = x * \overline{y}.$$

Proof. By definition, K^* is the only operator satisfying the equality

$$\langle Ka, b \rangle = \langle a, K^*b \rangle$$

for all $a, b \in L^2(\Omega)$, where $\langle \cdot, \cdot \rangle$ is the usual scalar product in $L^2(\Omega)$: $\langle f, g \rangle = \int_\Omega fg\, dm$, where m is the Lebesgue measure restricted to Ω.

Note that, in order to properly define convolution between functions with domain $\Omega \subseteq \mathbb{R}^n$, it is first necessary to extend to $\mathbb{R}^n$ by setting them to 0 in Ω^c.

It is well-known that for any given $\varphi, \phi, \psi\colon \mathbb{R}^n \to \mathbb{R}$ such that their pairwise convolutions are defined and integrable, the following equality holds:

$$\langle \varphi * \phi, \psi \rangle = \langle \phi, \psi * \overline{\varphi} \rangle \tag{13}$$

where $\overline{\varphi}(x) = \varphi(-x)$. Therefore,

$$\langle a * y, b \rangle = \langle a, b * \overline{y} \rangle.$$

Thus, K^* is simply given by $K^*b = b * \overline{y}$. $\qquad\square$

REFERENCES

[1] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, 2018.

[2] C. Clason and T. Valkonen. Introduction to nonsmooth analysis and optimization, 2024.

[3] J. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111, 1990.

[4] B. L. McGlamery. A Computer Model For Underwater Camera Systems. In S. Q. Duntley, editor, *Ocean Optics VI*, volume 0208, pages 221 – 231. International Society for Optics and Photonics, SPIE, 1980.