
AFRL-AFOSR-CL-TR-2016-0008

# MATHEMATICAL MODELS FOR CAMOUFLAGE PATTERN ASSESSMENT

**Jaime Ortega**
**UNIVERSIDAD DE CHILE**

**02/17/2016**
**Final Report**



Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ IOS
Arlington, Virginia 22203
Air Force Materiel Command

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 01-04-2013 | Final Report | AUG,01,2011-OCT,31,2012 |

**4. TITLE AND SUBTITLE**

Mathematical Models for Camouflage Pattern Assessment

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
NO.FA9550-11-1-0170

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Jaime ORTEGA, Takeshi ASAHI, Fernando PADILLA, Matías GODOY

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

UNIVERSIDAD DE CHILE. Avenida Blanco Encalada 2120. Piso 7.
Santiago de Chile. Chile, Código postal: 8370459

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

ADMINISTRATIVE OFFICE:
OFC OF NAVAL RSCH (ONRRO)
BOSTON
BOSTON REGIONAL OFFICE
495 SUMMER STREET, ROOM 627
BOSTON MA 02210-2109, KELLYM@ONR.NAVY.MIL

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Distribution A

**13. SUPPLEMENTARY NOTES**

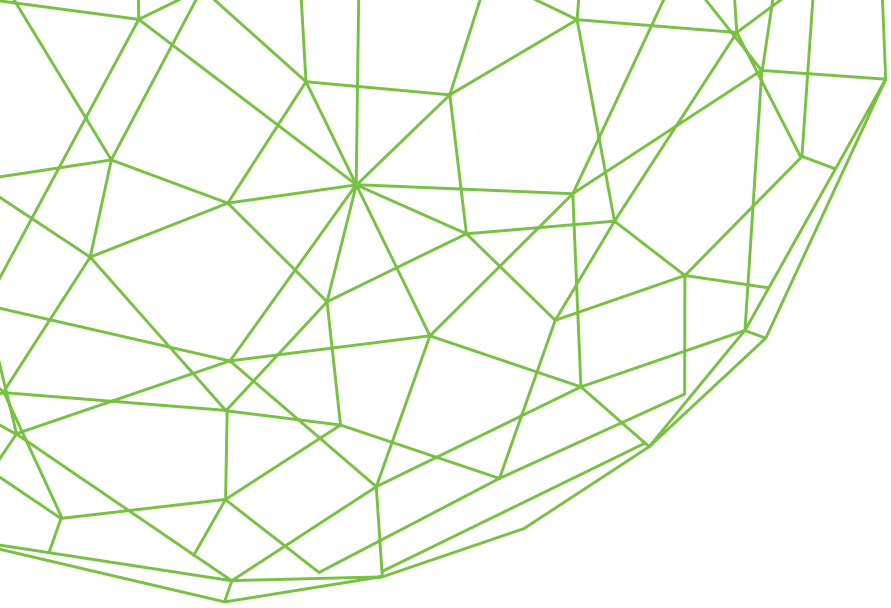
**14. ABSTRACT**

During this work we have developed a first approach to evaluate military camouflage pattern using advanced mathematical and image processing tools, in order to create indices for evaluating artificial patterns over a given background. This development is an innovative method which lay the foundations for evaluate and compare military camouflage patterns, this new approach can help to select the most effective camouflage for a given environment and situation, improving the security of the soldiers during a military operation. From the technical point of view, the main achievement is the development of a computational system, which allow us to differentiate between artificially generated textures, as well as natural ones, with the detection of objects within the image. The main tool is the use of mathematical methods to represent an image in terms of texture and the cartoon area in 2D examples. We improve the classical methods by detecting objects that have the same average of intensity, or may have the same texture but slightly shifted, since the classical discrimination lies on the comparison of the statistical distribution of the local image intensities, while for the proposed method the comparison is between the patches of the surrounding area. As a consequence, we automatically detect the periodicity of the texture and obtain a representation of the texture elements, as a dictionary of Gabor functions. So far, the latter has been implemented in 1D and we are extending it to 2D. This issue is important since a library of textures (backgrounds or camouflage patterns) may be obtained from a database of images. Therefore, it is possible to generate a classification and a common basic texture for each particular scenario, which allow us to measure the performance of a particular camouflage within the different scenarios. All the codes, for the calculation of indices, were implemented in Python and C (Open Source Code platforms) and they are available for distribution to the users.

**15. SUBJECT TERMS**
Image Processing, camoflage, segmentation

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | 77 | Jaime Ortega |
| | | | | | **19b. TELEPHONE NUMBER** *(include area code)* +56 2 2978 4631 |

**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**



CMM
Center for Mathematical Modeling

# Final Report

# on

# Mathematical Models for

# Camouflage Pattern Assessment

## AWARD NO.FA9550-11-1-0170

## April 2013

To: Col. Julian Williams, Commander, RDECOM Forward Element Command-Americas

From: Jaime Ortega, Center for Mathematical Modeling, University of Chile

Universidad de Chile
Centro de Modelamiento Matemático
Facultad de Ciencias Físicas y Matemáticas
http://www.cmm.uchile.cl



FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

University of Chile
Centro de Modelamiento Matemático
Facultad de Ciencias Físicas y Matemáticas

# Final Report: Camouflage Assessment

***January 2013***

**Abstract**

The main achievement is the development of a computational system which allow us to differentiate between artificially generated textures, as well as natural ones with the detection of objects within the image. The main tool is the use of mathematical methods to represent an image in terms of texture and the cartoon area in 2D examples. We improve the classical methods by detecting objects that have the same average of intensity, or may have the same texture but slightly shifted, since the classical discrimination lies on the comparison of the statistical distribution of the local image intensities while for the proposed method the comparison is between the patches of the surrounding area. As a consequence, we automatically detect the periodicity of the texture and obtain a representation of the texture elements, as a dictionary of gabor functions. So far, the latter has been implemented in 1D and we are extending it to 2D. This issue is important since a library of textures may be obtained from a database of images. Therefore, it is possible to generate a classification and a common basic texture for each particular scenario.

# Contents

# 1. Introduction

## 1.1 Background of the project

Camouflage plays an important role in order to increase the effectiveness of a mission and the survivability of soldiers. Camouflage can be defined as a method for concealing personnel or equipment from an enemy by making them appear to be part of the natural surroundings. Therefore, the purpose of a given camouflage pattern is to break the contour of a target in order to blend this target with the background where is placed. To this extent, the measurement of the performance of a camouflage pattern gives a quantitative base at the moment comparison between different patterns in specific scenarios.

The high cost that entails choosing and implementing a new camouflage pattern for armed forces, implies that the decision should be based on carefully designed patterns to ensure effectiveness in the largest set of possible scenarios, considering different lightness conditions, land surfaces, weather conditions, among other variables. Hence, the development of an automatic assessment system for camouflage pattern is a very useful and cost-saving tool, which complements human-visual techniques based on empirical testing.

Traditionally, the evaluation of camouflages is based on direct observation of an imagery collection by a group of observers, considering different conditions of lighting, background variation, noise and distance, among others. And recently, many methods have been proposed to assess camouflage pattern based on cluster analysis or the use of descriptor to quantify the similarity between the background and the camouflage texture (structural similarity).

In the present work, we propose a scheme using of mathematical modeling, image processing and human vision models to estimate the effectiveness of the camouflage patterns or the probability to detect the soldiers, under different parameters of design, background or illumination conditions. In particular we expect to develop a computational tool for assessing camouflage patterns to be used by the US Armed Forces.

Since this is a specific model for the assessment of camouflage, it is possible to process large quantities of images at high resolution and in the most diverse situations obtaining results similar to those that might result from an equivalent process with human evaluators. Contrasted with traditional human visual camouflage evaluations, where the personnel spend several days in the field, control over environmental variables turns in a severe drawback, including basic and unpredictable lightning conditions, weather, background variations, and many others. Setting new standards for computational image analysis, combining mathematical tools and several methodologies, could significantly reduce random or uncontrolled variables to a minimum by artificially exposing relevant information with the consequent improvement in results.

The aim of this project is to develop and evaluate indexes for the efectiveness of the camouflage of soldiers. This work may be considered as the starting point for a new type of camouflage assessment tool, reducing time, cost and almost every aspect on current standard methodologies based on human visual inspection at a final long-term integration. With the use of this evaluation tool, in the next steps, we plan to develop friendly software for the evaluation process of camouflages. Moreover, we will propose to create a computational camouflage design modulus. This computational tool will propose one (maybe several) patterns for different combat conditions.

## 1.2 Objectives and Goals

As stated in the proposal, the main objective of our research is *to develop a computational tool, based on mathematical models, computational algorithms and human vision models, to allow us the assessment of camouflage patterns, considering different environmental variables.*

From the main objective we can consider the following specific goals:

- Develop new image processing techniques for the representation lower stages of human vision.
- Develop and implement mathematical algorithms, based on variational models of image processing, for identifying images edges and textures.
- Build indexes for assessing the detectability of camouflage patterns.
- To implement the developed algorithms on a computational platform for assessing the effectiveness of camouflage patterns.
- In the next steps, to develop a desing modulus for new patterns and textures.

## 1.3 Summary

In the first report, we delivered a bibliographic survey on texture segmentation for using in camouflage assessment, considering a variational approach to this problem. In particular, we made a review of papers related to the following subjects:

- Stochastic-Variational Model for Soft Mumford-Shah Segmentation
- Nonlocal Image and Movie Denoising
- Nonlocal Linear Image Regularization and Supervised Segmentation
- Generalized Nonlocal Image Smoothing
- Nonlocal Mumford-Shah Regularizers for Color Image Restoration
- Index Camouflage Measurements.

We note that in the analysis of a camouflage, one of the main elements is the so-called texture. The texture can be seen as a repetition of basic texture elements called texels or textons, where the texels correspond to the fundamental unit of texture space whose placement obey some rule. In general, camouflage can be defined as a method for concealing animals, military vehicles, soldier or other objects to remain unnoticed by blending with their environment. A camouflage pattern will be successful if the visual system of the observer (or enemy) is unable to discriminate (or segment) the two textures of the background (environment) and the target. Then, an objective method to assess the effectiveness of camouflage pattern is the texture pattern analysis, in particular, the texture segmentation.

The objective of the algorithm is to recover both image structures by using our nonlocal approach.

As showed in the previous figure, the combination of the non-local means and the non-local Mumford-Shah regularization may improve the performance of finding the contours between regions with different textures.

For more details, we refers to the following sections of the report.

The two-dimensional case...

Index for the evaluation of the camouflage patterns...

# 2. The texture segmentation problem

As the final objective of the project is to calculate an index of the performance of camouflage, a powerful segmentation of texture algorithm is necessary. For a good segmentation algorithm, texture analysis is important in order to better achieve a relieable processing. Techniques having texture analysis may be divided into five groups [11]: structural, statistical, signal processing, model-based stochastic, morphology based, artificial intelligence and variational PDEs:

- Structural: where a given texture is generated by a set of texture primitives related by a set of rules
- Statistical: uses numerical indexes such as contrast, energy, entropy and relationship between pairs of pixels in a certain spatial orientation with given gray-levels.
- Signal Processing: usually is related with the use of the Fourier Transform where characteristics such as periodicity and directionality are considered. Many times, the use of filters is also included.
- Stochastic: where 2D ARMA models or Markov Random fields are usually applied.
- Morphology: through the morphological operators with structuring elements having different shapes are applied in order to discriminate between various textures.
- Artificial Intelligence: the use of an Artificial Neural Network trained with back-propagation has been used as classifiers.
- Variational PDEs (Partial Differential Equations): the minimization of a functional is used to obtain a non-linear PDE. The solution involves finding the contours of the different objects within the image.

For the present work, we will be focused on a combination of Variational PDE methods together with some statistical properties of the textures.

Other approches such as neural-networks will not be considered due to the fact that the training for detection have to be on a specific set of textures. On the other hand, it is important to recall that the final goal is to obtain a performance index, therefore *black-box* approaches are not that well suited for these kind of tasks.

## 2.1 Variational PDE models

As previously mentioned, the variational PDE models consists of defining a cost function (the functional), where the unknown variables are the pixels of an image or set of pixels of more than one function (e.g. image and contour). Because this problem is solved by obtaining the minimum and the number of variables is of the order of the size of the image, the use of efficicient optimization algorithms is necessary. Discretization issues should also be considered in order to obtain results of better quality. A general introduction is found in [6].

An example of these type of algorithms is the calculation of the Total Variation of a given function

$$\min_u TV[u, K|u_0] = \min_{K,u} \left\{ \int_\Omega (u - u_0)^2 d\Omega + \alpha \int_\Omega |\nabla u| d\Omega \right\} \tag{2.1}$$

as is exposed in [5]. The first term corresponds to the similarity term and is the one in charge of keeping the output image close to the input one. The second term is the smoothing (or regularization) term, in charge of the smoothness of the output image. The variable to minimize is the whole output image.

More mathematical details are to be foun in Appendix B and the summaries of the some state-of-the-art work involving non-local segmentation considering the variational approach (as in Report Dec 2011) are also included in the appendix.

## 2.2 The non-local means

The key idea here is that this formulation estimates the value of $x$ as an average of the values of all pixels whose Gaussian neighborhood looks like the neighborhood of $x$, this is important because with this formulation we can take advantage of the high degree of redundancy of natural images, or, in other words, we can detect textures and preserve it, instead of consider them as noise, like the usual algorithms for denoising does. For more detailed information please refer to section B.2 at the appendix.

## 2.3 Non-local variational image models

We will focus on the variational formulation of the segmentation problems where the (smoothness) regularization measurements consider the neighbourhood of the pixel [10]. The energy of the functionals define the forces of the active contours, but the non-local versions of some classical active contour approaches use the notion of patch distance instead of usual point-wise difference of luminosity.

Because the main modification is in the gradient, we focus on the classical definition of the derivative of the function $f : [a, b] \to (-\infty, \infty)$, in a point $x_0 \in [a, b]$, giving local information about the behavior of the function. Actually, $|f'(x_0)|$ is a measure of the function's variation in the point $x_0$. Then is natural to think that

$$|f'(x)|^2 = \int_a^b \left( \frac{f(x) - f(y)}{|x - y|} \right)^2 \delta_x(y) dy = \int_a^b (f(x) - f(y))^2 \left( \frac{\delta_x(y)}{|x - y|^2} \right) dy, \tag{2.2}$$

where $\delta_x$ is the Dirac function concentrated at $x$, the idea of the non-local is consider a more general expression for (2.2). Therefore, we replace $\delta_x(y) \longrightarrow w(x, y)$. If we choose a specific location x, the repetition of the texture will be at the $y$ positions where $w(x, y)$ has higher values.

# 3. The Index Measurement

One of the main issues of this research project, is the elaboration of an index for measuring, in some sense, wheter or not an object in a natural image, can be easily recogniced. Our aim is to develop an index associated to a given image under study, such that high values of the index means that the object(s) in the image can be easily recogniced by a human observer. We propose a definition and computation of an index based on the non-local Mumford-Shah framework. In this context, the two main factors that determines the capability of human vision to recognize objects in natural images, are the change in intensity level and change in texture of the object respect to the background. We define the Contour-Index as the change of texture and intensity level along the object's contour and we compute it based on the results from the non-local Mumford-Shah algorithm. We describe below the main components of the process and in the appendix section, we will provide the mathematical definitions and discuss implementation details.

## 3.1 Previous Steps

### 3.1.1 Implementation of the non-local Mumford-Shah regularization

Given an initial image $u_0$, we consider the variational regularization problem consisting in finding a regularized image $u^*$ and its associated soft-characteristic border function $v^* : \Omega \to [0,1]$ such that $(u^*, v^*)$ minimizes the Mumford-Shah (MS) cost functional

$$F_{MSNL}(u,v) = F_{sim}(u) + \alpha F_{RegL2NL}(u,v) + \gamma F_{AT}(v) \quad \text{(MS)}, \tag{3.1}$$

where $F_{sim}(u) = \int_\Omega (u-u_0)^2 dx$ is the similarity, $F_{RegL2NL}(u,v) = \int_\Omega v^2(x)|\nabla_w u|^2(x)dx$ is the MS regularization term, $F_{AT}(v) = \int_\Omega (\epsilon|\nabla v|^2 + \frac{(v-1)^2}{4\epsilon})dx$ is the Ambrosio-Tortorelli term and $\alpha, \gamma$ are positive parameters for controlling the relative importance among the different terms.

We also consider an alternative of the previous problem, based on the Total Variation (TV) cost functional

$$F_{TVNL}(u,v) = F_{sim}(u) + \alpha F_{RegL1NL}(u,v) + \gamma F_{AT}(v) \quad \text{(TV)}, \tag{3.2}$$

where $F_{RegL1NL} = \int_\Omega v^2(x)|\nabla_w u|(x)dx$ is the TV regularization term.

### 3.1.2 Minimization of MS cost functional $F_{MSNL}$

In this section we will provide numerical implementation details for the minimization of $F_{MSNL}$ defined in (3.1). Observe that $F_{MSNL}$ is differentiable and $F_{MSNL}(\cdot, v), F_{MSNL}(u, \cdot)$ are convex, therefore a gradient based alternating minimization scheme is well suited for minimizing $F_{MSNL}$. Our algorithm takes as input an initial image $u_0$ and an initial characteristic function $v_0$ (for instance, $v_0 \equiv 1$). In a first step, the weight function $w$ is computed, based on the initial image $u_0$. Then, a gradient descent alternating minimization scheme is carried out, considering $K$ outer iterations and $J$ inner iterations. For each (outer) iteration $k = 1, ..., K$, we compute $u_k$ by gradient descent minimization of $F(\cdot, v_{k-1})$ with step $\delta$ and $J$ (inner) iterations. Then we compute $v_k$ by gradient descent minimization of $F(u_k, \cdot)$ with the same step $\delta$ and the same number of inner iterations $J$. The algorithm is summarized in Algorithm 3.1.

### 3.1.3 Gradient of $F_{MSNL}$

When computing the Gateaux derivatives of the different terms that conforms $F_{MSNL}$ with respect to smooth functions $g, h : \Omega \to \mathbb{R}$ with compact support, the Gateaux derivative of $F_{sim}$ with direction $g$ is given by

$$DF_{sim}(u)(g) = 2\int_\Omega (u(x) - u_0(x))g(x)dx.$$

Also, $DF_{AT}(v)(h) = 2\epsilon \int_\Omega \nabla v(x)\nabla h(x)dx + \frac{1}{2\epsilon}\int_\Omega (v(x)-1)dx$. Now, recall the gradient-divergence adjoint relation $\int_\Omega f(x)\cdot\nabla h(x)dx = -\int_\Omega \text{div } f(x)h(x)$ where $f : \Omega \to \mathbb{R}^q$. Hence, taking $f = \nabla v$, we obtain

$$DF_{AT}(v)(h) = -2\epsilon \int_\Omega \Delta v(x)h(x)dx + \frac{1}{2\epsilon}\int_\Omega (v(x)-1)dx.$$

**Algorithm 3.1** Alternating-Gradient NLMS

**Require:** $u_0$ initial image and $v_0$ initial characteristic function.
**Ensure:** $u$ regularized image and $v$ final characteristic function.
Compute sparse weights $w$ using initial image $u_0$
**for** $k = 1, \ldots, K$ **do**
$u_k := u_{k-1}$ , $v_k := v_{k-1}$
**for** $j = 1, \ldots, J$ **do**
$g_u := D_u F_{MSNL}(u_k, v_k)$, according to (3.9)
$u_k := u_k - \delta g_u$
**end for**
**for** $j = 1, \ldots, J$ **do**
$g_v := D_v F_{MSNL}(u_k, v_k)$, according to (3.10)
$v_k := v_k - \delta g_v$
**end for**
**end for**
$u := u_K$ , $v := v_K$

On the other hand, the partial derivative of $F_{RegL2NL}$ respect to $u$ with direction $g$ is given by $D_u F_{RegL2NL}(u,v)(g) = 2\int_\Omega\int_\Omega v^2(x)\nabla_w u(x,y)\nabla_w g(x,y)dydx$. By the non-local version of the gradient-divergence adjoint relation $\int_\Omega\int_\Omega p(x,y)\nabla_w u(x,y)dxdy = -\int_\Omega \mathrm{div}_w p(x)u(x)dx$, we obtain

$$D_u F_{RegL2NL}(u,v)(g) = -2\int_\Omega \mathrm{div}_w(v^2\nabla_w u)(x)g(x)dx.$$

Now, the partial derivative of $F_{RegL2NL}$ respect to $v$ with direction $h$ in given by

$$D_v F_{RegL2NL}(u,v)(h) = 2\int_\Omega v(x)|\nabla_w u|^2(x)h(x)dx.$$

From the previous equations we obtain the partial derivatives of $E$:

$$D_u F_{MSNL}(u,v) = 2(u-u_0) - 2\alpha\mathrm{div}_w(v^2\nabla_w u) \tag{3.3}$$

$$D_v F_{MSNL}(u,v) = 2\alpha v|\nabla_w u|^2 - 2\gamma\epsilon\Delta v + \frac{\gamma}{2\epsilon}(v-1). \tag{3.4}$$

We recall that, the definition of the non-local gradient is

$$|\nabla_w u|^2 = \int_\Omega (u(y)-u(x))^2 w(x,y)dy \tag{3.5}$$

and taking $p(x,y) = v^2(x)\nabla_w u(x,y)$ in the definition of the non-local divergence $\mathrm{div}_w p = \int_\Omega (p(x,y) - p(y,x))\sqrt{w(x,y)}dy$, with $\mathrm{div}_w p : \Omega \to \mathbb{R}$ of a function $p : \Omega\times\Omega \to \mathbb{R}$, we have

$$\mathrm{div}_w(v^2\nabla_w u)(x) = \int_\Omega (u(y)-u(x))(v^2(x)+v^2(y))\sqrt{w(x,y)}dy \tag{3.6}$$

Local and non-local gradient based algorithms are capable of detecting edges in natural images. Local gradient is based only on local changes (with the neighbour positions) in intensity level. Therefore, one important limitation of the local gradient, is that it detects edges within a texture that might not correspond to the contour of an object. In contrast, non-local gradient is based on non-local information extracted from the image, therefore, it can detect a contour when there is a difference between the textures inside and outside the object, even if there is no apparent jump in gray level intensity. The non-local framework considers that textures are characterized by the repetition of a pattern within a determined neighborhood. Such patterns are associated to small connected regions or patches in the image, therefore it is necessary to define a proper notion of distance between patches for computing the non-local gradient of an image. Such distance, leads to the concept of weight function $w$, wich can be considered as an intrinsic descriptor of the texture changes in the image. In the appendix section we will discuss these concepts in more detail.

The norm of the non-local gradient $\|\nabla_w u(\vec{x})\|$ at a given point of $\vec{x}$ of the image $u$, is a measure of texture discrepancy within a neigborhood of the point. An small value of this measure, implies that the texture is homogeneous in a neighborhood of the point and a large value means that a texture change may ocurr in a neighborhood ot the point, indicating the presence of an adge contour.

### 3.1.4 Object Contour Detection:

We detect the object's contour based on the non-local Mumford-Shah algorithm. This algorithm is based on the concept of non-local gradient norm, described above, therefore, it is able to deal with images with texture changes between the object and background. The algorithm tries to identify regions in the image where $\|\nabla_w u(\vec{x})\|$ is small, leading to the identification of different regions according to the different textures in the image.

## 3.2 Computation of the Index

### 3.2.1 Contour-index Computation:

Once the object's contour has been detected, the following step is to calculate the contour-index $I$ that we define as the average of the norm of non-local gradient $\|\nabla_w u(\vec{x})\|$ along the contour. Observe that if a low value of $I$ means that the texture inside and outside the object's contour $\mathcal{C}$ is similar. Remmark that the contour-index $I$ depends on the weight function $w$ and the contour $\mathcal{C}$ and the correct estimation of them is a difficult task as it involves several parameters that model different aspect of the image textures.

### 3.2.2 Implementation of the calculation of the index

**The continuous setup** Given an image $u$ and the contour of the object $C \subset \Omega$, we define $I(u, \mathcal{C})$ as the mean-integral of the norm of the non-local gradient over the contour, that is,

$$I(u, \mathcal{C}) = \frac{1}{l(\mathcal{C})} \int_{\mathcal{C}} |\nabla_w u|(x) d\mu(x), \tag{3.7}$$

where $l(\mathcal{C})$ is the length of $\mathcal{C}$ and $\mu$ is the contour measure. We observe that, if the texture in the interior region of $\mathcal{C}$ is similar to the texture in the exterior region of $\mathcal{C}$ , then $I(u, \mathcal{C})$ would be small. Similarly, in the case of large texture's dissimilarities between the interior and exterior region of $\mathcal{C}$ then $I(u, \mathcal{C})$ will be large. We also note that the normalization factor $\frac{1}{l(\mathcal{C})}$ makes possible the comparison of the index of the object using different contour's length. We approximate $I(u, \mathcal{C})$ in 3.7 by

$$I(u^*, v^*) = \frac{\int_\Omega |\nabla_w u^*|(x) dx - F_{RegL2NL}(u^*, v^*)}{F_{AT}(v^*)},$$

where $(u^*, v^*)$ are the minimizers of the Mumford-Shah cost functional 3.1, $F_{RegL2NL}$ is the MS regularization term and $F_{AT}$is the Ambrosio-Tortorelli term. Hence, we have

$$I(u^*, v^*) = \frac{\int_\Omega (1 - v^{*2}(x)) |\nabla_w u^*|(x) dx}{\int_\Omega (\epsilon |\nabla v^*(x)|^2 + \frac{(v^*(x)-1)^2}{4\epsilon}) dx} \tag{3.8}$$

**The discrete setup** Now we will focus in the case of 2-dimensional images, i.e. $q = 2$. We follows the implementation of the non-local weights $w$ in [14]. For each pixel $i$ we consider a neighborhood $N_i$ including the $s$ most similar pixels (according to the distance $d$) within a search window of size $t$. Imposing symmetry of $w$ leads to at most $2s$ elements in $N_i$. By making $w[i, j] = 0$ if $j \notin N_i$, an sparse implementation of $w$ and the consequent savings in memory consumption are obtained. Let $u[i]$ be the image value at pixel location $i$ with $i = 1, \ldots, n$. The discretized version of the partial derivatives (3.9) and (3.10) are given by:

$$D_u F_{MSNL}(u, v)[i] = 2(u[i] - u_0[i]) - 2\alpha \mathrm{div}_w(v^2 \nabla_w u)[i] \tag{3.9}$$

$$D_v F_{MSNL}(u, v)[i] = 2\alpha v[i] |\nabla_w u|^2[i] - 2\gamma\epsilon \Delta v[i] + \frac{\gamma}{2\epsilon}(v[i] - 1). \tag{3.10}$$

From (3.5) and (3.6),

$$|\nabla_w u|^2[i] = \sum_{j \in N_i} (u[j] - u[i])^2 w[i, j]. \tag{3.11}$$

$$\mathrm{div}_w(v^2 \nabla_w u)[i] = \sum_{j \in N_i} (u[j] - u[i])(v^2[i] + v^2[j]) \sqrt{w[i, j]} \tag{3.12}$$

Similarly, the discretized version of 3.8 is

$$I(u^*, v^*) = \frac{\sum_i \sum_{j \in N_i} (1 - v^{*2}[i])(u^*[j] - u^*[i])^2 w[i,j]}{\sum_i (\epsilon |\nabla v^*[i]|^2 + \frac{(v^*[i]-1)^2}{4\epsilon})}. \tag{3.13}$$

# 4. Some numerical examples

## 4.1 A first approach in 1D

In order to better understand segmentation in textured images, we have to first study a simplified situation. For that reason we created some synthetic examples in 1D in order to have an initial idea of the influence of the different parameters on the result of the detection.

We started by having function with a 1D region (element) having an average value higher than the rest of the 1D domain (background). In the first stage we assume that the global reflected (signal) light intensity of the element is different from the background. If this object has no texture, assuming homogenous materials, we have that several different already proposed algorithms are capable of recovering the original edges of the objects. That is the case of the Total Variation (TV) filter, and the basic Mumford-Shah algorithm with the Ambrosio-Tortorelli approximation (MSAT).

The Total Variation filter consisits of an input image $u_0(\cdot)$ that will be regularized by a $L_1$ norm term for the gradient of the denoised image as in equation 2.1, where the problem consists of finding the image $u(\cdot)$ that minimizes the functional $F(u)$. On the other hand, the MSAT functional is the local specific version of the non-local Mumford-Shah functional, where $|\nabla_w u|^2$ is replaced by the usual gradient $|\nabla u|^2$.

Preliminary numerical implementation for non-local segmentation. For the first synthetic example, which is a one dimensional case, we can have a more graphical idea of the performance of the different algorithms. We start with an object with constant signal level (1.0) inserted in a black background (level 0.0). We add a simulated texture which is a sinusoidal signal with amplitude 0.1.

For the first synthetic example, in order to have a more graphical idea of the performance of the different algorithms, we start with an object with constant signal level (1.0) inserted in a black background (level 0.0). The difference with the usual examples is that we add a simulated texture, where we tested two cases:

- a sinusoidal signal with amplitude 0.1
- a periodic squared signal with amplitued 0.2

functions that were added to the previous generated signal. The above signal reproduce the two effects in the image. The big jump represents the discontinuity of the image that can be obtained by classical segmentation methods. The small oscilations of the signal represents the texture of the image which is a periodic structure.

We consider additive gaussian (white) noise in order to simulate a signal that would be acquired in a more real context. The corresponding generated signals are shown in Figures 4.1(a) and 4.2(a), respectively.

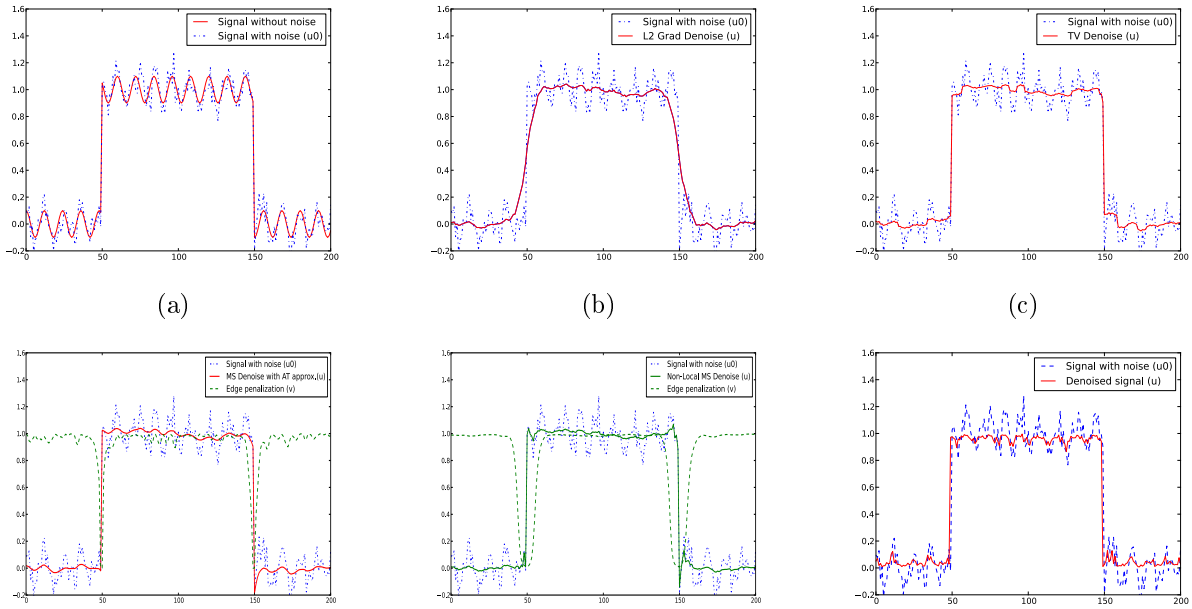
The denosing problem consists of recovering the original signal, filtering the noise, but keeping the sharp transitions, and the idea is to also recover the added periodic texture.

For the simulations and denoising, we implemented the following cases:

- regularization with the L2 norm of the gradient, corresponding to a convolution with a laplacian kernel,
- regularization with the L1 norm of the gradient, problem known as the Total Variation,
- regularization with the Mumford-Shah functional, using the Ambrosio-Tortorelli approximation,
- regularization with the non-local shift invariant wighted gradient, for the Mumford-Shah functional,
- the non-local means filtering.

From the results shown in Figures 4.1 and 4.2 we can obtain the following preliminary observations:

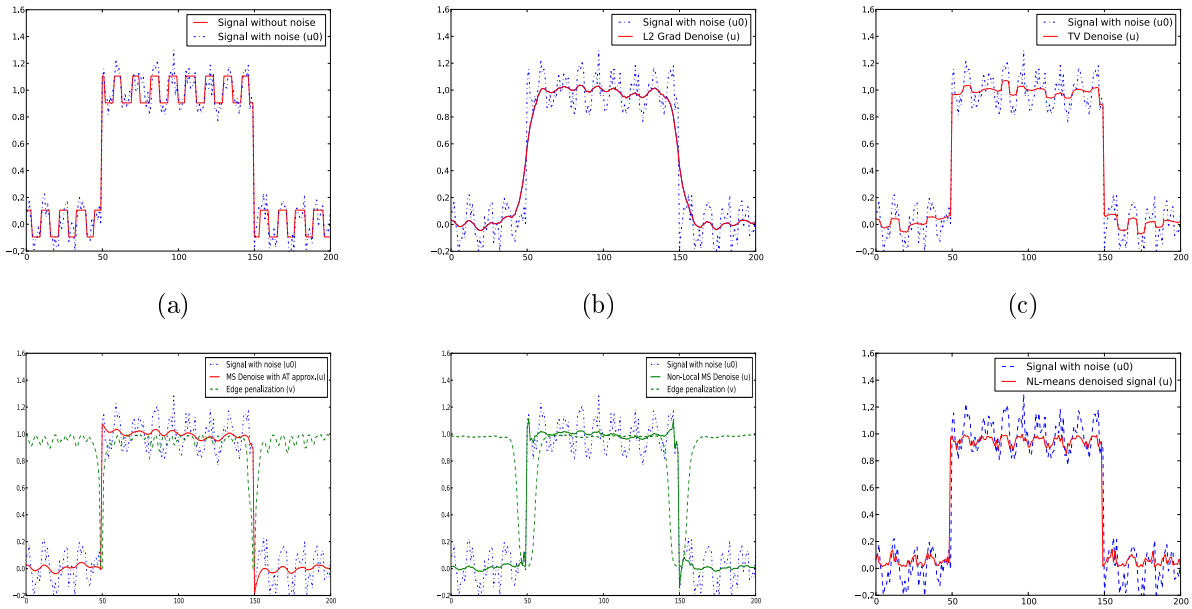
- the linear $L_2$ norm of the gradient regularization is not capable of recovering the sharp transitions (edges of the objects). Whereas, the non-linear filters such as the Mumford-Shah, Total Variation and Non-Local means are capable of keeping the sharp transitions.
- the Mumford-Shah based algorithms (local with Ambrosio-Tortorelli approximation and non-local shift-invariant gradient) are not capable of recovering the texture.
- The non-local means can recover the sharp transitions and is able to detect texture, but greately diminishes the contrast of this texture.




(a) (b) (c)

(d) (e) (f)

Figure 4.1: (a) original synthetic signal with and without noise, (b) regularization with the $L_2$ norm of the gradient ($\alpha = 20$), (c) regularization with the $L_1$ norm of the gradient ($\alpha = 0.5$) (d), regularization with Mumford-Shah ($\alpha = 50$,$\gamma = 1E-4$,$\varepsilon = 0.5$) (e) regularization with non-local Mumford-Shah ($\alpha = 10$,$\gamma = 1E-4$,$\varepsilon = 0.5$, quadratic spline $m = 3$), and (f) non-local means ($h = 0.5$, $a = 0.5$, $m = 10$)




(a) (b) (c)

(d) (e) (f)

Figure 4.2: (a) original synthetic signal with and without noise, (b) regularization with the $L_2$ norm of the gradient ($\alpha = 20$), (c) regularization with the $L_1$ norm of the gradient ($\alpha = 0.5$) (d), regularization with Mumford-Shah ($\alpha = 50$,$\gamma = 1E-4$,$\varepsilon = 0.5$) (e) regularization with non-local Mumford-Shah ($\alpha = 10$,$\gamma = 1E-4$,$\varepsilon = 0.5$, quadratic spline $m = 3$), and (f) non-local means ($h = 0.5$, $a = 0.5$, $m = 10$)

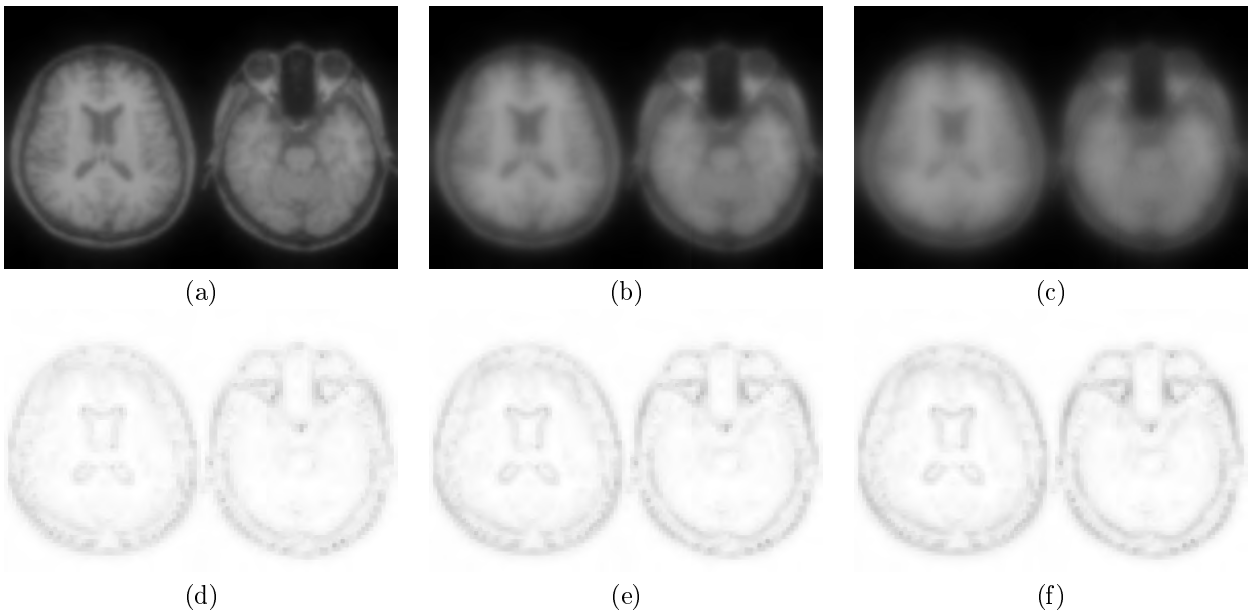


Figure 4.3: (a) image regularized with usual Ambrosio-Tortorelli approximation of MS functional, (b) with non-local gradient with B-spline of order n=4 and expansion m=3, non-local gradient with expansion $m$=5, (d), (e) and (f) the corresponding contours $v$ of (a), (b) and (c), respectively.

- The $L_1$ norm of the gradient regularization can, to a certain degree, recover the texture when its regions have constant gray levels (flat regions)
- The outputs of the local Mumford-Shah and non-local Mumford-Shah regularizations give more information than the other methods since we have the additional information of the contour of the objects within the image.

From the previous observations, we can conclude that a combination between the idea of the non-local means and the non-local Mumford-Shah regularization may improve the performance of finding the contours between regions with different textures.

## 4.2 The numerical solution for 2D

### 4.2.1 First approach with non-textured images

The images of the numerical implementation where we have the MS Local Regularizer, and the non-local regularized images. As we can see in Figure 4.3, the image is softened as we increase the width of function $f(\cdot)$, while the transition of the contour function $v$ is preserved, but having a wider width. This last consequence implies an easier contour finding procedure.

Given the preliminary results, in the future work, we will tackle the problem of regularizing images with texture. Therefore, we will continue with the study of the weight function $w(\cdot)$ for different cases. We believe that some clue of the texture information will be embedded in this weight function.

### 4.2.2 Experiments with Textured Images

The non-local MS regularization was applied to a set of artificial and natural images, the latter taken from the the Berkeley Segmentation Dataset [16]. Also, we illustrate the benefits of non-local MS over local MS. In our experiments, we realize that the quality of the results, highly depends on the parameters involved in the computation of the weight function: patch size ($m$), window search size ($s$), numbers of neighbors ($t$), scale parameter for weight function ($h$) and apodization function ($A$). These parameters model different aspects of the image texture, thus are intrinsic to each image and we have determined experimentally.

After we have computed the wight function $w$ we carry out the minimization of 3.1 using the gradient descent alternating minimization algorithm described in Section 3.1.2. This algorithm also depends on several

parameters that we determine experimentally for each image: weight of regularization term ($\alpha$), weight of Ambrosio-Tortorelli term ($\gamma$), gamma convergence parameter ($\epsilon$), gradient descent step ($\delta$ ), number of outer iterations ($K$) and number of inner iterations ($J$). We also compute the local MS segmentation in order to compare with the non-local MS results. Figure 4.4 shows 4 artificial images (first row) and the segmentation results after the non-local (second row) and local (third row) MS algorithm. Each image consist of an object's texture and a background's texture. We have created such texture by the regular replication of a given pattern. We observe that the non-local MS algorithm successfully detects the object's contour. In contrast, the local MS algorithm detects edges within texture so is not able to detect object's contour. The first row in Figure 4.5 shows the same images as in the first row of Figure 4.4 after the addition of random Gaussian noise. Second and thirds rows of Figure 4.5 show, non-local and local MS segmentation results respectively. We observe that the presence of noise affects the capability of the algorithm to detect the object's contour. Figure 4.6 shows 3 natural images (first row) and the segmentation results after the non-local (second row) and local (third row) MS algorithm. In contrast to the artificial images, natural images exhibits textures with a complicated pattern structure. Such patterns present differences in gray level, shape and sizes. To partially overcome this issue, a Gaussian smoothing filter preprocessing step is applied, to makes such features more homogeneous. We observe that non-local MS algorithm performs better than the local MS algorithm, in the identification of contours in areas with a change of texture. On the other hand, the local-MS seems to perform better in the identification of contours in areas with a change of intensity level.

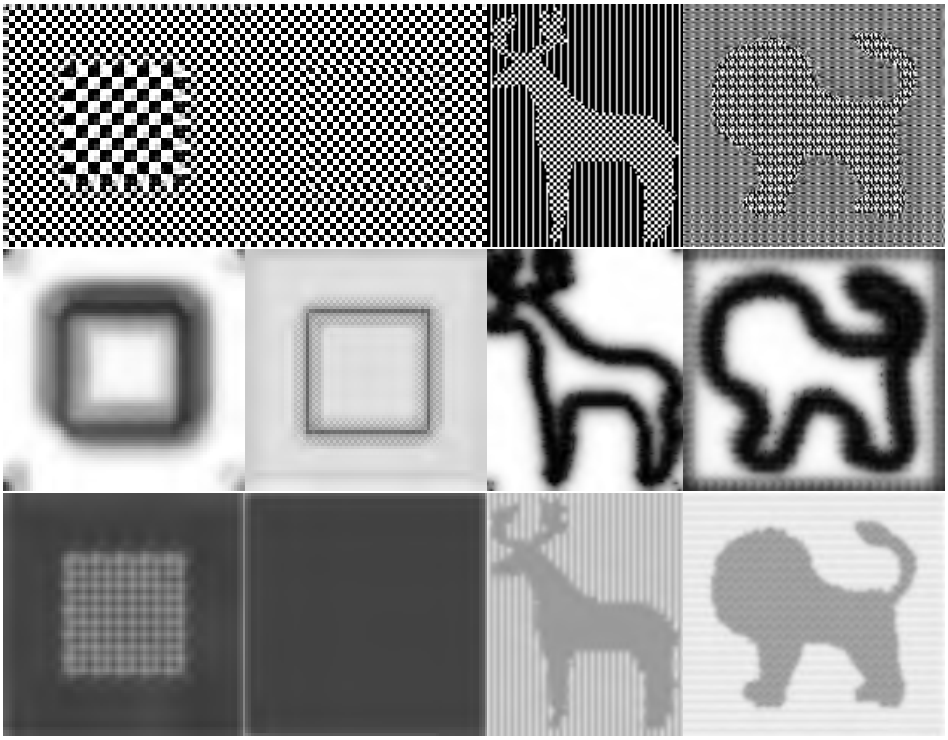


Figure 4.4: artificial image no noise.

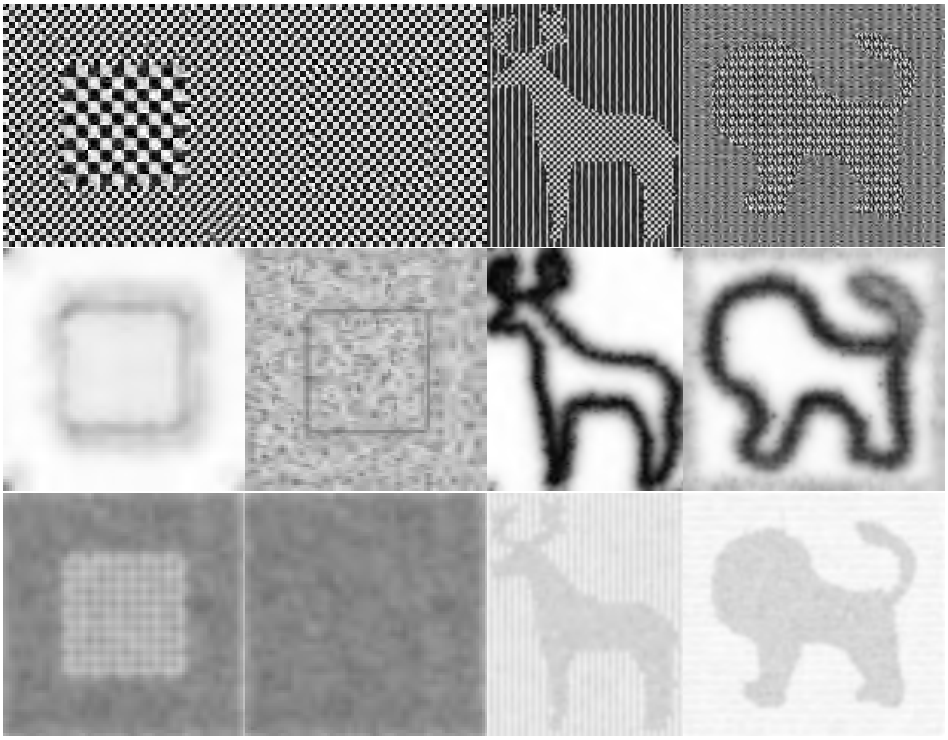


Figure 4.5: artificial image with noise.

Now, we asses behavior of the contour-index described in Section 3.2.2 by considering an artificial image with different levels of noise. Figure 4.7 exhibits from left to right, an artificial image with no noise and additive random Gaussian noise with variance 20, 40, 60 and 80 respectively. The respective contour-index values are 1.83, 0.58,0.19, 0.09 and 0.04. Observe that the contour-index value decrease with the level of noise and the object is more difficult to be distinguish visually.

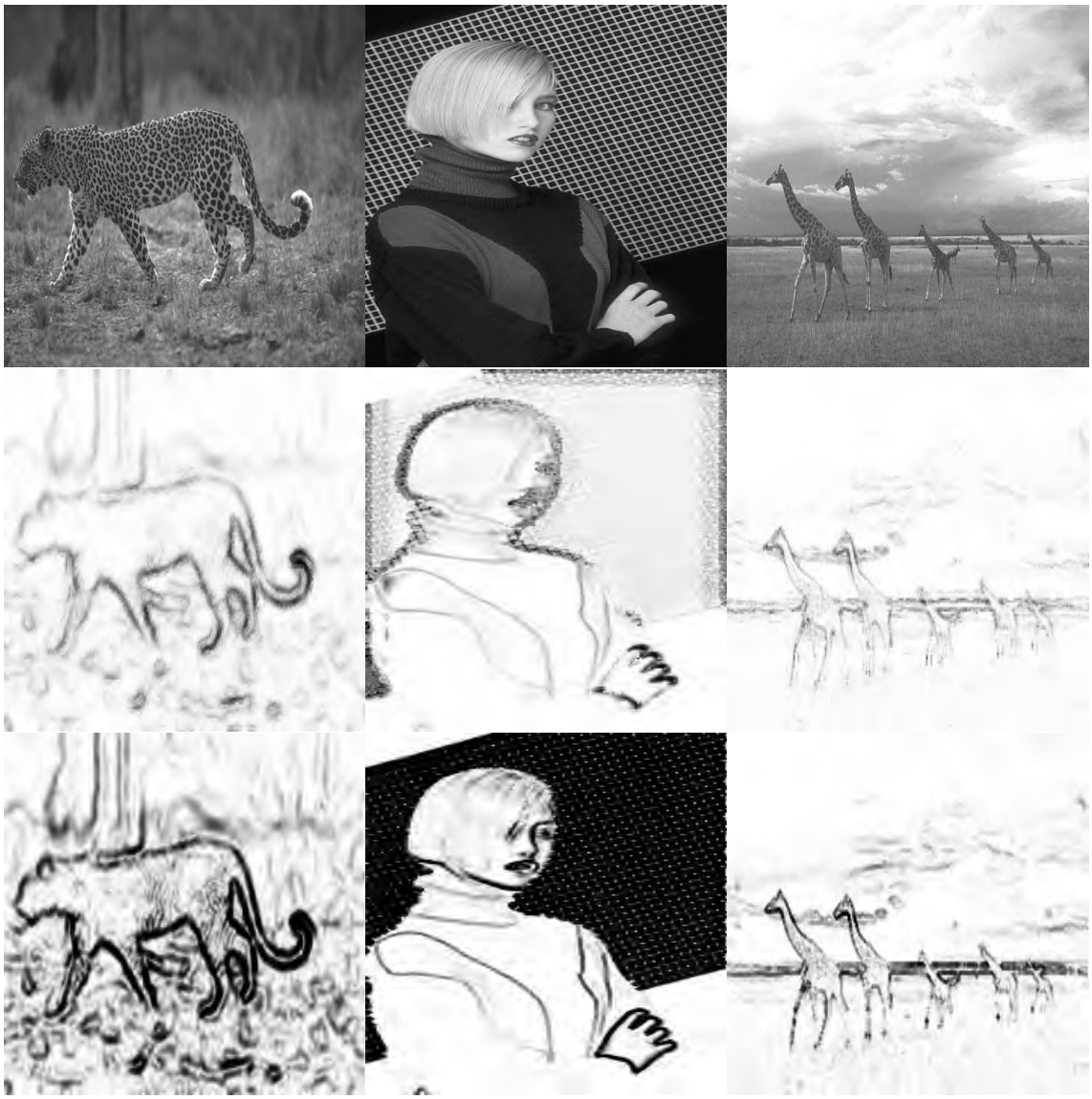


Figure 4.6: natural image no noise

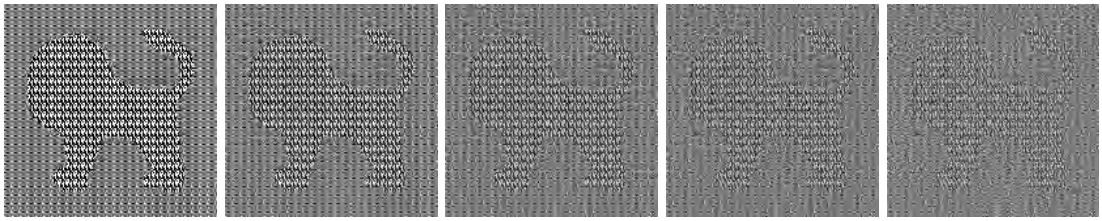


Figure 4.7: index noise

# 5. Concluding Remarks and Future Work

So far, none of the tested methods give a satisfactory result in terms of recovering the original texture, from an image with noise. It is expected that a combination of some of those methods would greately improve the performance. However, an efficient numerical implementation of such a combination of algorithms is not straightforward.

For the next stage we consider the issue of finding the local periodicity between the texture elements (textels). The local periodicity may be obtained from the information provided by the analysis of the weight function of the non-local Mumford-Shah functional. While the textels should be expressed in terms of a multiscale base such as the one provided by wavelets.

On the other hand, the images to be tested will have to include some camouflage-like textures.

## 5.1 Implementation of algorithms

As pointed out in Section 4, the next step involves the implementation of the minimization of non-local Mumford-Shah, including the optimization of the weight function $w(x, y)$. This would enable a more accurate definition of the countours of the objects with textures. Our assumption is that the optimization involving the weight function would improve the performance of the already proposed methods.

## 5.2 Segmentation Problems and Textured Images

In a future report some modifications will be added in order to consider the segmentation of different textured regions, but with a similar brightnes levels. In the current stage, we tested some of the algorithms in 1D with artificial texture, being sinusoidal or staircase, but different average gray levels. Since this analysis was based on the 1D simulations, still the extention to 2D has to be considered, since this extention is not always straightforward. The examples given in this work present less difficulties, since many algorithms are able to segment by taking advantage of the different mean levels of each texture. Therefore, more difficult images will be tested, where the biggest difference between the textures will be in the periodicity or the phase of the texels. Special attention will be payed to this last issue of change of phase in the texels, since this may be considered as a change in the texture. Therefore, this discontinuity would mean the border of an object within the measured image.

## 5.3 Calculation of the index of the camouflage

Once the countour is established by one of the algorithms described in the report, the numerical (dis)CONTINUITY index has to be calculated, considering the characteristics of the texels or difference between the textels along the border. The relation between this index and the reaction of a human observer has to be checked, since this is part of the original idea of the first stage of the design a camouflage pattern.

# A. Some concepts about texture

Try to characterize the texture concept takes relevance when we want to camouflage something, because our first impression is that the human eye recognizes the change between different textures, see Figure A.

Figure A.1: The change between different textures

Define texture is a difficult task, because we don't have a precise and unique mathematical definition or a clear concept. The basic idea is that the texture can be seen as a repetition of basic texture elements called texels or textons made of pixels whose placement obey some rule.

Let us give some recompilations ideas about textures:

1. The textured region can contain texture elements of various sizes, each of which can itself be textured.
2. The order consists in the nonrandom arrangement of elementary parts.
3. The parts are roughly uniform entities having approximately the same dimension everywhere within the textured region.
4. The parts are roughly uniform entities having approximately the same dimension everywhere within the textured region.
5. A region in an image has a constant texture if a set of local statistics or other properties of the picture function are constant, slowly varying, or approximately periodic.

# B. Mathematical aspects of the Variational PDE Models

In the framework of the local Variational PDE Models, the approaches are proposed with rather local based measurements (such as gradients) for the smoothness term. This is quite suitable for segmenting objects with constant luminosity characteristics. However for the cases where texture is present, these algorithms fail to output a good quality segmentation. For that purpose non-local regularity measurents have been proposed.

## B.1 The Mumford-Shah regularization

We recall that the segmentation problem, considering a variational approximation, was introduced by David Mumford and Jayant Shah in 1989 [19]. The problem addressed by the authors can be describe as follows: Given and image $I \subset \mathbb{R}^2$, we want to decompose the image in a finite number of subsets $R_i \subset \mathbb{R}^2$ and edges $K_i$, such that at the interior of each component $R_i$ the intensity of the image is almost constant and the edges are smooth.

This problem corresponds to an optimization problem of given and image $u_0$, where $u_0$ represents the intensity of the image, to find a function $u$ and a curve $K$, which minimizes the functional

$$\min_{K,u} MS[u, K|u_0] = \min_{K,u} \mathcal{H}^1(K) + \alpha \int_\Omega |\nabla u|^2 + \lambda \int_\Omega (u - u_0)^2,$$

where $\mathcal{H}^1(K)$ corresponds to the arc length of $K$.

To solve the above optimization problem, Luigi Ambrosio and Vincenzo Maria Tortorelli in 1990 (see Approximation of functionals depending on jumps by elliptic functionals via Γ-convergence, Comm. Pure Appl. Math., 43(8), 999–1036) introduced an aproximated proble for MS functional. This approximated problem have several advantages from theoretical and numerical point of view, for details of the presentation of these functional we refer to the report of bibliographic discussion.

In the case of texture segmentation, we want to use a similar approach to the one for the classic problem of segmentation, however we must consider the aspect of the periodicity of the patterns, that is, we must consider a non local approach. Nevertheless, it is necessary to include a term which allow us to compare texels in the image, and for that reason we consider a functional including the nonlocal efects. In this way, we introduce the non-local segmentation with the variational approach $G_\varepsilon : L^1(\Omega) \times L^1(\Omega) \to [0, +\infty]$, defined by

$$G_\varepsilon(u, v) = \begin{cases} \displaystyle\int_\Omega \left( \psi(v(\vec{x})) \, \|\nabla_w u(\vec{x})\|_\delta^2 + \frac{1}{\varepsilon} V(v(\vec{x})) + \varepsilon \|\nabla v(\vec{x})\|^2 \right) dx^N & \text{if } u, v \in H^1(\Omega), \\ & \text{and } 0 \le v \le 1 \ \ a.e. \\ \\ +\infty & \text{otherwise,} \end{cases}$$

where $V : [0, 1] \to [0, +\infty)$ be a continuous function vanishing only at the point 1, and let $\psi : [0, 1] \to [0, +\infty)$ be an increasing lower semicontinuous function with $\psi(0) = 0$, $\psi(1) = 1$, $\psi(t) > 0$ if $t \neq 0$.

## B.2 The non-local means

Recall that the non local means technique, based on the work of Buades et al., have been developed in the context of denoising filtering.

### B.2.1 Basics

The idea of Buades is defined as:

Consider a continuous noisy image $u_0$ defined on $\Omega$, a bounded domain of $\mathbb{R}^2$ then, the filtered image in the pixel $i$ is defined by:

$$NL(u_0)(x) = u(x) = \frac{1}{C(x)} \int_\Omega e^{-\frac{(G_a * |u_0(x+\cdot) - u_0(y+\cdot)|^2)(0)}{h^2}} u_0(y) dy = \frac{1}{C(x)} \int_\Omega w(x, y) u_0(y) dy$$

where $C(x)$ is a normalizing factor and $G_a(t)$ is a Gaussian Kernel with standard deviation $a$.
notice also that:

$$(G_a * |u_0(x+\cdot) - u_0(y+\cdot)|^2)(0) = \int_{\mathbb{R}^2} G_a(t)|u_0(x+t) - u_0(y+t)|^2 dt$$

In the discrete case (which is the one that we work in fact) we have the following formulation:
Consider a discrete noisy image $u_0 = \{u_0(i)/i \in \Omega\}$, in this context, the estimated value $NL(u_0)(i)$ is computed as a weighted average of all the pixels in the image, i.e. considering:

$$NL(u_0)(i) = u(i) = \frac{1}{C(i)} \sum_{j \in \Omega} w(i,j) u_0(j)$$

where $w(i,j)$ must be a sort of discretization of the continuous formula described before, it intends to measure the similarity between pixels $i$ and $j$; in a general context we have to consider $w(i,j)$ such that:

$$\sum_{j \in \Omega} w(i,j) = 1 \quad 0 \leq w(i,j) \leq 1$$

The (sort of) discretization of the weight function is:

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||u_0(\mathcal{N}_i) - u_0(\mathcal{N}_j)||^2_{2,a}}{h^2}}$$

where $Z(i)$ is a normalization factor, $||\cdot||_{2,a}$denotes the Euclidean weighted distance (by a Gaussian kernel of standard deviation $a$) and $\mathcal{N}_i$ denotes a "neighborhood" of the pixel $i$, centered at it, usually this neighborhood is a square of length $2m$, then $u_0(\mathcal{N}_i)$ is a vector, so, the idea of this discretization is to compare the patch centered on $i$ between the one centered on $j$.

The important element here is the weight function $w(x,y)$ used in the formula which perform the "averaging" term. This function is considered in a different context to perform segmentation (using it in the so called non-local gradient or weighted gradient as we seen before). It is important to follow the behaviour of this function in the task of denoising, specially in in terms of the pursued main characteristic of preservation of textures in denoising procedure. Then, is important to have an implementation of this algorithm, in order to have a checkpoint for next steps of this work.

### B.2.2 Implementation of Non local means

In order to implement the denoising algorithm for NL-means we have to try to reduce the number of calculations, as we seen before, the original filter involves a high number of calculations (at least $2N^2$ for each pixel if $N$ is the number of pixels of the image). In order to do that, we will follow the "faster" version of NL-means proposed by Buades, which try to reduce calculations:

Let $I$ a grid of pixels, choose a subset $\{i_1, \ldots, i_k\} \subset I$ . Consider $B = \{i/||i|| \leq m\}$, and then define: $W_k = i_k + B$. The idea is to divide $I$ in non-disconnected regions such that: $I = \cup_i W_i$ and $W_i \cap W_{i+1} \neq \emptyset$. The idea is to define NL-means for the $W_k$ objects (the so called vectorial NL-means) and then defining the NL-means for a fixed pixel as the average of the vectorial NL-means where this pixel belongs.

Let us define the vectorial NL-means, for each $W_k$ as:

$$NL(W_k) = \frac{1}{C_k} \sum_{j \in I} u_0(W_j) e^{-\frac{||u_0(W_k) - u_0(W_j)||^2_2}{h^2}}$$

where $C_k$ is a normalization parameter.

Notice that in this case, the norm involved is the usual, since we restore at the same time the whole neighborhood and do not want to give any privilege to any point in particular.

Finally, in order to restore the value at a pixel $i$, we must consider all $W_k$ containing $i$, so, if we define: $A_i = \{k \ / \ i \in W_k\}$, we have to define:

$$NL(u_0)(i) = u(i) = \frac{1}{|A_i|} \sum_{k \in A_i} NL(W_k)(i)$$

$$NL(u_0)(i) = u(i) = \frac{1}{|A_i|} \sum_{k \in A_i} NL(W_k)(i)$$

We attach an example of this implementation for the one dimensional case in numerical examples chapter, the example consists on a signal with texture (like a castle) which have been exposed to pseudo-random noise, the idea is to check if this implementation preserves the most important setting of the NL-means algorithm: remove noise without removing texture.

## B.3 Non-local variational image models

In other words, if we have a patch $p_x(t) = f(x+t), \forall t \in [-\tau/2, \tau/2]^2$, with $\tau$ the width and height of the patch. The $L^2$ patch distance will be:

$$d_a(p_x, p_y) = \int_\Omega g_a(t) |p_x(t) - p_y(t)|^2 dt$$

where $g_a(t)$ is a gaussian with variance $a$.

On the other hand, non-local formulation involves the use of a weighted gradient, that is the continuos equivalent of the graph-cut approach:

$$|\nabla_w u|^2(x) := \int_\Omega (u(y) - u(x))^2 w(x,y) dy$$

This formulation extends in a quite straight-forward way the original Mumford-Shah model to a non-local setting. The disadvantage lies on the extra complexity of the additional integral on the region. Computationally, it is difficult to determine the weight function $w(x,y)$ for each specific case. Some examples for this weight function are:

- Shift-invariant function such as a gaussian $w(x,y) = g_a(x-y)$ one.
- The patch distance $d_a(p_x, p_y)$ as previously mentioned.
- The NL-means weight, where the $w(x,y) = g_h(d_a(p_x, p_y))$.

## B.4 The basic model for obtaining the contour

By defining the square of the module of the non-local gradient as:

$$\left\| \nabla_w u(\vec{x}) \right\|^2 = \int_\Omega (u(\vec{x}) - u(\vec{y}))^2 w(\vec{x}, \vec{y}) dy^N$$

we denote the specific shift invariant cases as $w(\vec{x}, \vec{y}) = g(\vec{x} - \vec{y})$.

The non-local segmentation with the variational approach we have the formula

$$F_{NLG} = \int_\Omega \|\nabla_g u(\vec{x})\|^2 \, dx^N$$

while the gradient (*Gâteaux* differential) of this function $F_{NLG}$ with respect to function $u$.

$$\frac{\partial F_{NLG}}{\partial u} = 4 \, |g|_{L_1} \, u(\vec{l}) - 4 g * u(\vec{l}),$$

if parity of the function $g$ is imposed: $g(-\vec{x}) = g(\vec{x})$, and $*$ is the convolution $f * g(\vec{x}) = \int_\Omega f(\vec{y}) g(\vec{x} - \vec{y}) dy^N$.

On the other hand, the regularization term of the output function which is described as

$$F_{NLRegAT}(u, v) = \int_\Omega \|\nabla_g u(\vec{x})\|^2 \, v^2(\vec{x}) dx^N$$

has a *Gâteaux* differential of

$$\frac{\partial F_{NLRegAT}(u,v)}{\partial u} \{\delta u_{\vec{k}}\} = 2u(\vec{k}) g * v^2(\vec{k}) - 2g * (uv^2)(\vec{k}) - 2v^2(\vec{k}) g * u^2(\vec{k}) + 2 \, |g|_{L_1} \, u(\vec{k}) v^2(\vec{k})$$

As one of the objectives is to use B-spline functions in order to implement these procedures, the $g(\cdot)$ function is an expanded B-spline of order $n$: $g(x) = \beta^n(x/m)$, where $m$ is the scaling factor of expansion, where integer values were used instead. By using these kind of functions, it is possible to have efficient convolutions, since $b_m^n(x) = \beta^n(x/m) = b_1^n * u_m^n(x)$, where $u_m^n(x) = \frac{1}{m^n}\underbrace{b_m^1 * \cdots * b_m^1}_{\text{n times}}(x)$. The previous function may be regarded as applying $n$ times a moving average process of window width $m$. One characteristic of these families of fuctions is that they assimptotically converge to a gaussian function as $n$ grows. Gaussian functions are usually used since many effects on image acquisition such as bluring or other distortions may be modelled with this function.

## B.5 The convergence of the functional

Let us consider a regularity of the non-local gradient, given by

$$\|\nabla_w u(\vec{x})\|_\delta^2 = \int_\Omega (u(\vec{x}) - u(\vec{y}))^2 w(\vec{x},\vec{y}) dy^N + \delta \|\nabla u(\vec{x})\|^2,$$

where $\delta > 0$, the function $w \in \mathcal{C}^1(\Omega \times \Omega \setminus \{(x,y) : \ x = y\})$, with $w(\vec{x},\vec{y}) = w(\vec{y},\vec{x})$ and satisfying

$$\lim_{\vec{y}\to\vec{x}} \|\vec{x} - \vec{y}\|^2 w(\vec{x},\vec{y}) = d_0 > 0.$$

We introduce the non-local segmentation with the variational approach $G_\varepsilon : L^1(\Omega) \times L^1(\Omega) \to [0, +\infty]$

$$G_\varepsilon(u,v) = \begin{cases} \displaystyle\int_\Omega \left( \psi(v(\vec{x})) \, \|\nabla_w u(\vec{x})\|_\delta^2 + \frac{1}{\varepsilon} V(v(\vec{x})) + \varepsilon \|\nabla v(\vec{x})\|^2 \right) dx^N & \text{if } u, v \in H^1(\Omega), \\ & \text{and } 0 \le v \le 1 \ \ a.e. \\ \\ +\infty & \text{otherwise,} \end{cases}$$

where $V : [0,1] \to [0, +\infty)$ be a continuous function vanishing only at the point 1, and let $\psi : [0,1] \to [0, +\infty)$ be an increasing lower semicontinuous function with $\psi(0) = 0$, $\psi(1) = 1$, $\psi(t) > 0$ if $t \neq 0$.

We want to compute the $\Gamma$-limit of the $G_\varepsilon : L^1(\Omega) \times L^1(\Omega) \to [0, +\infty]$. Following the ideas of Ambrosio-Tortorelli, we proposed as $\Gamma$-limit as the functional $G : L^1(\Omega) \times L^1(\Omega) \to [0, +\infty]$,

$$G(u,v) = \begin{cases} \displaystyle\int_{\Omega \setminus S_u} \|\nabla_w u(\vec{x})\|_\delta^2 \, dx^N + 4 c_V \mathcal{H}^{n-1}(S_u) & \text{if } u \in SVB(\Omega), \\ & \text{and } v = 1 \ \ a.e. \\ \\ +\infty & \text{otherwise,} \end{cases}$$

with $c_V = \int_0^1 \sqrt{V(s)} ds$. The difference between the Ambrosio-Tortorelli's result is the non-local gradient in the above expressions.

This class of result says us that when we consider a numerical approximation of $\inf G_\varepsilon$ for $\varepsilon$ small, actually we are consider an approximation of $\inf G$.

# C. A Stochastic-Variational Model for Soft Mumford-Shah Segmentation[20]

## C.1 Introduction to the model.

Given an image $I$ in $L^2(\Omega)$, defined in $\Omega \subset \mathbb{R}^2$, which we assume to be open, bounded and smooth, we know that the goal of the classical segmentation problem is to find a closed 'edge set' $\Gamma$ and all the connected components $\Omega_i,\ i \in \{1,\dots,K\}$ associated to $\Omega \setminus \Gamma$, based in some desired visual measure (e.g. textures). Notice that with this process the image $I$ is discontinuous along $\Gamma$ and relatively soft or homogeneous inside each connected component $\Omega_i$.
In this context we will call to each patch $I_i = I|_{\Omega_i}$ a 'pattern' with an associated support $\Omega_i$.
We will call this standard procedure a 'hard' segmentation, because we make a partition of $\Omega$ based on the fixed borders contained in $\Gamma$, giving us a collection of disjoint supports $\Omega_i$, so, with this segmentation we have:

$$\mathbf{1}_\Omega(x) = \sum_{i=1}^{K} \mathbf{1}_{\Omega_i}(x)$$

The main idea of the soft segmentation is make a 'softer' or 'weak' partition of $\Omega$, in the following (formally) sense:

$$\mathbf{1}_\Omega(x) = \sum_{i=1}^{K} p_i(x)$$

Where $p_i$ are a continuous (or even smooth) functions, formally we can choose: $p_i = \mathbf{1}_{\Omega_i} * \rho_\varepsilon$ with $\rho_\varepsilon$ a standard regularization function.
We can connect this idea with the widely know 'mixture image models' in the following way:
Suppose that the image $I$ could be decomposed in $K$ patterns indexed by $\omega \in \{1,\dots,K\}$. Then, each pixel $x \in \Omega$ could be assigned to a pattern by the 'index random variable' $\omega(x) \in \{1,\dots,K\}$. In this context we can take $p_i(x) = \mathbb{P}(\omega(x) = i)$ with $i \in \{1,\dots,K\}$, this is the natural stochastic interpretation, also, we will call each $p_i$ the 'ownership' of the pattern $i$.
In contrast of the 'hard' segmentation, this 'soft' one allows to each pixel belong to all the patterns with some probability, then, this is a generalization of the 'hard' segmentation, and allows us to have more versatility with some natural phenomena when we usually don't have a 'clear' boundaries, instead of that we can assign probabilities in the transtition regions.

The model that we will study is based in the widely known Mumford-Shah, which is based in the following variational problem:

$$\min_{\Gamma,u} J[u,\Gamma|I] = \min_{\Gamma,u} \mathcal{H}^1(\Gamma) + \alpha \int_\Omega |\nabla u|^2 + \lambda \int_\Omega (u-I)^2$$

The main advantage of using a stochastic-variational modelation is that we have more universality in modeling due to stochastic approach, and also we have a rigorous mathematical analysis and better computational methods due to variational-PDE approach.

## C.2 The Soft Mumford-Shah (SMS) Model

Our aim is minimize the energy associated to $K$ patterns, i.e. we want to minimize:

$$\begin{aligned} J^{SMS}[P,U|I] &= \underbrace{\lambda \sum_{i=1}^{K} \int_\Omega (u_i - I)^2 p_i}_{F^{SMS}_{sim}} + \underbrace{\alpha \sum_{i=1}^{K} \int_\Omega |\nabla u|^2}_{F^{SMS}_{reg}} + \underbrace{\sum_{i=1}^{K} \int_\Omega \left( 9\varepsilon |\nabla p_i|^2 + \frac{(p_i(1-p_i))^2}{\varepsilon} \right)}_{F^{SMS}_{MM}} \\ &= \end{aligned}$$

with the constraint: $P : \Omega \to \Delta_{K-1}$ this is: $p_i \geq 0$, $i \in \{1,\dots,K\}$ and $\sum_{i=1}^{K} p_i = 1$.
Due to the structure of this energy we will need: $p_i \in H^1(\Omega)\ \forall i$. Then, under the supposition: $I \in L^2(\Omega)$ the energy will be well-defined in the following admissible set:

$$adm_K := \{(P,U) \mid p_i, u_i \in H^1(\Omega),\ i \in \{1,\dots,K\},\ P : \Omega \to \Delta_{K-1}\}$$

The construction of this energy and main mathematical results about it can be seen on the appendix.

## C.3 Euler Lagrange equations for the SMS Energy

### C.3.1 Euler Lagrange equations in the $(K-1)$ simplex

To minimize $J^{SMS}[P,U|I]$ we can use a gradient-descent method or the Euler Lagrange equations. We will compute this equations and bring a computational scheme to solve them.
The first-order partial variation on $U$ given $P$ leads to the following system of PDEs, for $i \in \{1,\dots,K\}$:

$$\begin{aligned} \alpha\Delta u_i + \lambda(I-u_i)p_i &= 0 \text{ in } \Omega & \text{(C.1)} \\ \frac{\partial u_i}{\partial n} &= 0 \text{ in } \partial\Omega & \text{(C.2)} \end{aligned}$$

the proof of this is standard and thus omitted.
Notice that the first equation could be rewritten in the form:

$$-\alpha\Delta u_i + (\lambda p_i)u_i = (\lambda p_i)I$$

which is a Poisson equation with variable coefficients (this is good because there exists numerical methods to solve this equation efficiently).
In the case of the first-orden partial variation on $P$ given $U$ we have the following system of PDEs: For $i \in \{1,\dots,K\}$

$$\begin{aligned} V_i(x) - \langle V\rangle(x) &= 0 \text{ in } \Omega & \text{(C.3)} \\ v_i(x) - \langle v\rangle(z) &= 0 \text{ in } \partial\Omega & \text{(C.4)} \end{aligned}$$

with $\langle w\rangle = \frac{1}{K}\sum_{i=1}^{K} w_i$, $V_i = \lambda(u_i - I)^2 - 18\varepsilon\Delta p_i + \frac{2}{\varepsilon}(1-p_i)(1-2p_i)$, $v_i = \dfrac{18}{\varepsilon}\dfrac{\partial p_i}{\partial n}$.
Details are given in the appendix.

Joining this results we have the following theorem

**Theorem C.3.1.** *[Euler Lagrange Equations of SMS]*
*We have the following system of equations: for $i \in \{1,\dots,K\}$*

$$\begin{aligned} -\alpha\Delta u_i + (\lambda p_i)u_i = (\lambda p_i)I \quad & \text{in } \Omega & \text{(C.5)} \\ -18\varepsilon\Delta p_i + 2\varepsilon^{-1}p_i(1-p_i)(1-2p_i) = \langle V\rangle - \lambda(u_i - I)^2 \quad & \text{in } \Omega & \text{(C.6)} \end{aligned}$$

*with $V = V(P,U) = (V_1,\dots,V_K)$ and Neumann condition in $u_i$ and $v_i$.*

### C.3.2 Computation of Euler Lagrange equations

To solve the equations of the last theorem we will use a algorithm called 'alternating minimization' (AM) which is related with the widely known algorithm called 'expectation maximization' (EM) used in statistics with 'hidden variables' [9][13]. In this context we will treat $p_i$ as a 'hidden variables'.
AM is a progressive algorithm: Given the actual best estimate ($t = n$) of $U^n = (u_i^n \mid i = 1,\dots,K)$, we solve:

$$P^n = argmin_P J^{SMS}[P|U^n, I] \quad \text{(C.7)}$$

or equivalently:

$$-18\varepsilon\Delta p_i + \frac{2}{\varepsilon}p_i(1-p_i)(1-2p_i) = \langle V^n\rangle - \lambda(u_i^n - I)^2 \; i \in \{1,\dots,K\}$$

with Neumann homogeneous conditions.
Then, after solving this, we get $P^n$, we get $U^{n+1}$ via:

$$U^{n+1} = argmin_U E[U|P^n, I] \quad \text{(C.8)}$$

or equivalently:

$$-\alpha\Delta u_i + (\lambda p_i^n)u_i = (\lambda p_i^n)I \ i \in \{1,\dots,K\}$$

with Neumann homogeneous conditions.
Notice that this last equations are a linear decoupled system, then, is easy to solve. The hard part is solve the first system of equations, because is coupled and non-linear, we need to linearize this system in the following way:
Let $e_i(x) := (u_i(x) - I(x))^2$ and $\vec{e} = (e_i \mid i \in \{1,\dots,K\})$, $V = V(P,U) = V(P,\vec{e})$.
We want to solve:

$$-18\varepsilon\Delta p_i + \frac{2}{\varepsilon}p_i(1-p_i)(1-2p_i) = \langle V\rangle - \lambda e_i$$

notice that:

$$\langle V(p,\vec{e})\rangle = \frac{1}{K}\sum_{i=1}^{K}(-18\varepsilon\Delta p_i + \lambda e_i + 2\varepsilon^{-1}p_i(1-p_i)(1-2p_i)) = \frac{\lambda}{K}\sum_{i=1}^{K}e_i + \frac{2}{\varepsilon K}\sum_{i=1}^{K}(2p_i^3 - 3p_i^2) + \frac{2}{\varepsilon K}$$

this is because $\sum p_i = 1$ and $\Delta(\sum p_i) = 0$
Notice also that: $p_i(1-p_i)(1-2p_i) = p_i(1-p_i)^2 - p_i^2(1-p_i)$.
Then, the non linear PDE can be solved in iterative scheme:

$$\dots \to P^{(j)} \to P^{(j+1)} \to \dots$$

via the following linealization:

$$-18\varepsilon\Delta p_i^{(j+1)} + \frac{2}{\varepsilon}p_i^{(j+1)}(1-p_j^{(i)})^2 = f_j^{(j)} \tag{C.9}$$

with $f_i^{(j)} = -\lambda e_i + \langle V(P^{(j)},\vec{e}\rangle + \frac{2}{\varepsilon}(p_i^{(j)})^2(1-p_i^{(j)})$ and Neumann homogeneous condition.
This linear system of Poisson equations can be solve using known solvers. The theoric convergence is still an open problem.

## C.4 Numerical Examples

## C.5 Appendix: Mathematical Facts

### C.5.1 Construction of the Energy for The Soft Mumford-Shah (SMS) Model

In this section we will discuss the main points of the construction of the SMS Energy. First of all, let $K$ the number of patterns involved (which can be a variable of the model to be found optimally, but in [21][22] they proved that is not necessary). Given $I = I(x)$ an image in $\Omega$ open, bounded, regular domain, our first objective is to compute the ownerships $p_1(x),\dots,p_K(x)$
Let $P(x) = (p_1(x),\dots,p_K(x))$ and $\Delta_{K-1} := conv(\vec{e}_1,\dots,\vec{e}_K)$ where $conv$ is the convex hull of the vectors $\vec{e}_i$ (The canonical basis of $\mathbb{R}^K$) this set is usually called the canonical $(K-1)$ simplex or probability simplex.
Then we have $P : \Omega \to \Delta_{K-1}$.
Associated with the label of each pattern $(\omega = i)$, we have $u_i(x) \in H^1(\Omega)$.
Define also $U(x) = (u_1(x),\dots,u_K(x))$.
Our aim is estimate the optimum pair of ownerships and patterns given $I$, mathematically we want the pair $(P^*,U^*)$ such that:

$$(P^*,U^*) = argmax_{(P,U)}\mathbb{P}(P,U|I)$$

But, due to the Bayes formula and taking que logaritmic likelihood (i.e. using $J[\cdot] = -\log\mathbb{P}(\cdot)$ known also as the Gibbs Energy) we deduce that our problem will have the following form:

$$argmin_{(P,U)}J[P,U|I] = argmin_{(P,U)}J[I|P,U] + J[P] + J[U]$$

Now, assuming that the pattern channels have independent components we can assume: $J[U] = J[u_1,\dots,u_K] = \sum_{i=1}^{K}J[u_i|i]$ this is motivated by the fact that a 2D image is a projection of the 3D environment where different objects in 3D are located independently in range and depth.
Then, for patterns in $H^1$, we can impose homogeneous energies (the idea is penalize changes in luminosity):

$$J[u_i|i] = J[u_i] = \alpha\int_\Omega |\nabla u_i|^2 \ i \in \{1,\dots,K\} = F_{reg}[u_i]$$

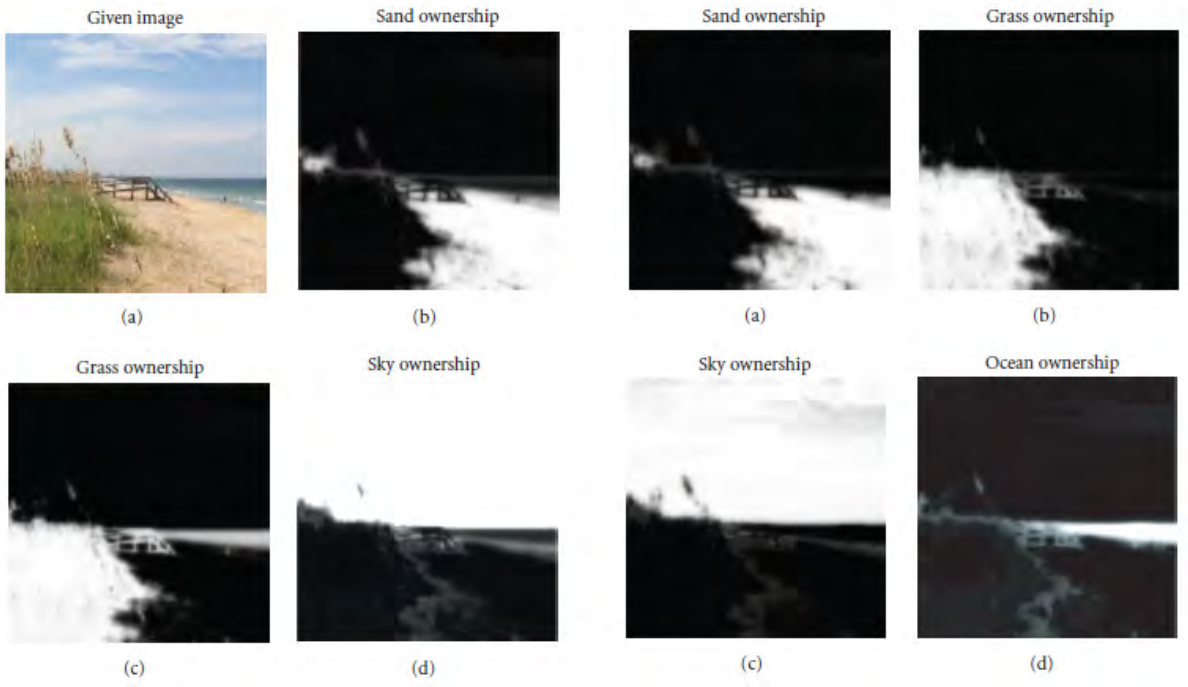



Figure C.1: SMS segmentation with three (left) and four phases (right). In the left we have the original image and the three ownership distributions: $p_1(x), p_2(x), p_3(x)$. In the right we have the four ownership distributions $p_1(x), p_2(x), p_3(x), p_4(x)$. Notice that the ocean pattern is "absorbed" into the grass pattern in the three phases segmentation due to the greenish color they happen to share. Also notice that if we add more patches we will have more softly segmented patterns.

Where $\alpha$ is a weight to model the 'visual sensitivity' to intensity roughness. Notice that, in contrast to original Mumford-Shah model, the energy of each channel is defined in all $\Omega$ instead of each $\Omega_i$. This is a good thing for the mathematical analysis because we are working in a fixed domain.

Now, for the term $J[I|P,U]$ we will study a particular case (Gaussian mixture with smooth mean fields). Assuming that all patterns are gaussian with means $u_1, \ldots, u_K$. WLOG we can assume that they all have the same variance $\sigma^2$, then, for each pixel $x \in \Omega$

$$(I|\omega(x) = i) \ \sim \ N(u_i(x), \sigma^2) \ i = 1, \ldots, K$$

Then, due to 'Total probabilities' property and applying the Gibbs Energy we have:

$$J[I|P,U] = J_\mu[I|P,U] = -\mu \int_\Omega \log(\sum_{i=1}^{K} g(I|u_i(x), \sigma) p_i(x)) \quad \mu > 0$$

Where $g$ denotes the Normal density function given by: $g(I|m,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(I-m)^2}{2\sigma^2}\right)$.
This formula is complicated in general, so the authors made the following reduction:
Considering each ownership $p_i(x) \sim \mathbf{1}_{\Omega_i}(x)$ so we have:

$$\begin{aligned}
-\log\left(\sum_{i=1}^{K} g(I|u_i(x),\sigma)p_i(x)\right) &\sim -\log\left(\sum_{i=1}^{K} g(I|u_i(x),\sigma)\mathbf{1}_{\Omega_i}(x)\right) \\
= -\sum_{i=1}^{K} \log g(I|u_i(x),\sigma)\mathbf{1}_{\Omega_i}(x) &\sim -\sum_{i=1}^{K} \log g(I|u_i(x),\sigma)p_i(x) \\
&= \frac{1}{2\sigma^2}\sum_{i=1}^{K}(I-u_i)^2 p_i(x) + Constant(K,\sigma)
\end{aligned}$$

This suggest the following model:

$$J[I|P,U] = \lambda \int_\Omega \sum_{i=1}^{K} (I - u_i(x))^2 p_i(x) = F_{sim}^{SMS}$$

Finally, the modelling of the term $J[P]$ (i.e. the energy model for the ownerships) will be done considering the following conditions:

- Each pattern ownership $p_i(x)$ have at most two phases: 'inside' ($p_i = 1$) and 'outside' ($p_i = 0$), the transition between them is narrow
- The boundaries (transition bands) are regular, not in zig-zag

With this considerations we will use the following energy, due to Modica and Mortola [18] with double potential: Given $p_i \in H^1(\Omega)$:

$$J_\varepsilon[p_i] = \int_\Omega 9\varepsilon |\nabla p_i|^2 + \frac{(p_i(1-p_i))^2}{\varepsilon} = F_{MM}^{SMS} \quad i \in \{1, \dots, K\} \quad \varepsilon << 1$$

Notice that $\varepsilon$ controls the transition band, and by this, we need also (to minimize the energy) that $p_i$ takes only 0 or 1 values (which is our first condition). Also, due to smallness of $\varepsilon$ we need regular $p_i$ (if not, the gradient part explodes), which is the second condition.
This energy is widely studied, in particular in the context of $\Gamma$-convergence [17], the main results for that studies made a strong link between SMS and classical Mumford-Shah functional, as we will see in the following theorems: Before showing the theorems we will need a few definitions:
Let $q \in L^1$, we will call the Total Variation (TV) of $q$ (as a Radon measure) to the following formula:

$$TV[q] := \int_\Omega |\nabla q| = \sup_{\varphi \in C_0^1(\Omega, B^2)} \langle q, \nabla \cdot \varphi \rangle$$

We also define, for $q \in L^1$:

$$J_0[q] := \begin{cases} TV[q] & \text{if } q = 0 \vee q = 1, \text{ a.e in } \Omega \\ +\infty & \text{if not} \end{cases}$$

Notice that by this definition we have: if $J_0[q] = +\infty$ then $q$ has only two phases, then $J_0[q] = TV[q] = Per(q^{-1}(1))$ where $Per(q^{-1}(1))$ denotes the perimeter of the support of $q$
Finally, consider the following subspace of $L^1(\Omega)$:

$$L^1_{[0,1]}(\Omega) := \{q \in L^1(\Omega) \mid q(x) \in [0,1] \ \forall x \in \Omega\}$$

After this definitions we can enounce the big theorems about the Modica and Mortola model:

**Theorem C.5.1.**
$\forall q \in L^1_{[0,1]} \setminus H^1(\Omega)$ *let* $J_\varepsilon[q] = +\infty$. *Then:*
$J_\varepsilon \to J_0$ *in the sense of the* $\Gamma$*-convergence in* $L^1_{[0,1]}(\Omega)$

**Theorem C.5.2.**
*Suppose we choose a subsequence* $p_\varepsilon$ *in an optimal way (i.e. they converge optimally to a two-phase pattern* $\mathbf{1}_V(x)$ *with boundary* $\Gamma = \partial V$*). Then*

$$J_\varepsilon[p_\varepsilon] \to length(\Gamma) = \int_\Omega |D\mathbf{1}_V(x)|$$

This last theorem reveals the strong connection with the SMS model with the original Mumford-Shah model.

### C.5.2 Problems and Solutions: Symmetry and 'Weak Supervision'

Let $S_K$ the permutation group of $\{1,\dots,K\}$. Given a $K$-tuple $F = (f_1,\dots,f_K)$ we define: $F_\sigma := (f_{\sigma(1)},\dots,f_{\sigma(K)})$ where $\sigma \in S_K$.
We have the following 'problematic' result, which leads to lose uniqueness (and even worse: using iterative algorithms we can have new intermediate solutions)

**Theorem C.5.3.** *[Hidden Symmetry of SMS]*
*Given $\sigma \in S_K$ we have: $J^{SMS}[P_\sigma, U_\sigma|I] = J^{SMS}[P,U|I]$*
*In particular, if we have: $(P^*,U^*) = argmin_{(P,U)\in adm_K} J^{SMS}[P,U|I] \Rightarrow \forall \sigma \in S_K\ (P^*_\sigma, U^*_\sigma)$ is also an optimal point*

The proof of this result is straightforward.
The obvious question is how to eliminate this symmetry, which lead us to lose uniqueness, the following technique, called 'weak supervision', lead us to 'break the symmetry':
We define $K$ patches: $Q_1,\dots,Q_K$ and then we impose the following restriction:

$$p_i|_{Q_j} = \delta_{ij}\ i,j \in \{1,\dots,K\}$$

where $\delta_{ij}$ is the Kronecker delta.

### C.5.3 Existence Results

**Theorem C.5.4.** *[Unsupervised SMS]*
*Suppose that $I \in L^2(\Omega)$. Then given any positive modeling parameters $(\lambda,\alpha,\varepsilon)$, a minimizer to the unsupervised soft Mumford-Shah (SMS) model must exist.*

**Theorem C.5.5.** *[Supervised SMS]*
*Suppose that $I \in L^2(\Omega)$. Then given any positive modeling parameters $(\lambda,\alpha,\varepsilon)$, a minimizer to the supervised soft Mumford-Shah (SMS) model must exist, assuming that each patch $Q_i$ has a positive Lebesgue measure.*

The proof of this theorems can be found on the original paper of Shen, anyway the proof is relatively standard except for one technical lemma.

### C.5.4 Details on the deduction of Euler Lagrange equations for the SMS Energy

The first calculate the first-order partial variation on $U$ given $P$ is standard.
In the case of the first-order partial variation on $P$ given $U$ we have to be more careful, because in this case we have the constraint $P : \Omega \to \Delta_{K-1}$, so, the derivative must take values on the tangent subspace of $\Delta_{K-1}$. To do this we will follow the technique developed by Chan and Shen [3]:
We first compute the first-orden variation in $P$, this give us to:

$$\delta J^{SMS} = \int_\Omega V \cdot \delta P dx + \int_{\partial\Omega} v \cdot \delta P dS$$

where $V = (V_1,\dots,V_K)$ and $v = (v_1,\dots,v_K)$ and $V_i = \lambda(u_i - I)^2 - 18\varepsilon\Delta p_i + \frac{2}{\varepsilon}p_i(1-p_i)(1-2p_i)$ defined in $\Omega$ and $v_i = 18\varepsilon\frac{\partial p_i}{\partial n}$ defined in $\partial\Omega$. So, the first partial variation without the constraint in $P$ is:

$$\frac{\partial J}{\partial_f P} = V|_\Omega + v|_{\partial\Omega}$$

Now, because $P \in \Delta_{K-1}$ we have to project appropriately, for this: Let $T_P\Delta_{K-1}$ the tangent space of $\Delta_{K-1}$ in $P \in \Delta_{K-1}$, and let $\pi : T_P\mathbb{R}^K \to T_P\Delta_{K-1}$ the ortogonal projection in the tangent space. Notice that, in this case, the normal direction of the tangent plane is given by:

$$\frac{\mathbf{1}_K}{\sqrt{K}} = \frac{(1,\dots,1)}{\sqrt{K}}$$

Then, the projection operator is given by:

$$\pi(w) = w - \frac{\mathbf{1}_K\langle w, \mathbf{1}_K\rangle}{K} = w - \langle w\rangle \mathbf{1}_K$$

where $\langle w \rangle = \frac{1}{K} \sum_{i=1}^{K} w_i$.
Finally, the restricted gradient of $E$ in $\Delta_{K-1}$ is:

$$\frac{\partial J^{SMS}}{\partial P} = \pi \left( \frac{\partial J^{SMS}}{\partial_f P} \right) = (V - \langle v \rangle \mathbf{1}_K)|_\Omega - (v - \langle v \rangle \mathbf{1}_K)|_{\partial\Omega}$$

In particular, this leads to the Euler Lagrange equations on $P$: For $i \in \{1, \ldots, K\}$

$$\begin{aligned} V_i(x) - \langle V \rangle(x) &= 0 \text{ in } \Omega \\ v_i(x) - \langle v \rangle(z) &= 0 \text{ in } \partial\Omega \end{aligned}$$

The following lemma allows us to take an homogeneous Neumann conditions:

**Lemma C.5.6.**
*Let $P : \Omega \to \Delta_{K-1}$. Then $\forall x \in \partial\Omega$ we have: $\langle v \rangle(x) = 0$*

**Proof:**

$$\langle v \rangle = \frac{1}{K} \sum_{i=1}^{K} v_i = \frac{18\varepsilon}{K} \sum_{i=1}^{K} \frac{\partial p_i}{\partial n} = \frac{18\varepsilon}{K} \frac{\partial}{\partial n} \left( \sum_{i=1}^{K} p_i \right) = \frac{18\varepsilon}{K} \frac{\partial}{\partial n} (1) = 0$$

The second equality is because $x \in \partial\Omega$ □

# D. Nonlocal Image and Movie Denoising - A. Buades, B. Coll and J.M. Morel [1]

## D.1 Introduction: Neighborhood Filters and NL-means

We will say that a filter (for images or video) is a neighborhood/NBH filter if this reduces the noise by averaging similar pixels. (anyway, we can use another statistical estimates like the median).
General CCD noise models (as we will see later) are signal dependent. Fortunately two pixels which received the same energy from the outdoor scene undergo the same kind of perturbations and therefore have the same noise model.
We will accept the following general assumption (which is the basic idea where this models relies):
**Assumption**: At each energy level the noise model is additive and white, then denoising can be achieved by first finding out the pixels which received the same original energy and then averaging their observed grey levels.

Since the original values of the image are lost the filters proceed by picking for each pixel $i$ the set of pixels $\mathcal{N}_i$ spacially close to $i$ and with similar grey value. The NBH filters proceed by replacing the grey level of $i$ (which will be denoted $u(i)$) by the average $NF(u(i)) = \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i} u(j)$ (again, we can use median also) Under the assumption that the pixels of $\mathcal{N}_i$ have the same energy as $i$, we have that $NF(u(i))$ is a denoised version of $u(i)$.
Most popular NBH filters are: $\sigma-$filter (Lee 1983), SUSAN (Smith, Brady 1997) and the bilateral filter (Tomasi and Manduchi 1998) where the neighborhoods are Gaussian in space and grey level.

In contrast, the Non-local means (proposed by Buades et al. in 2005) filter is based in the following idea:
Extend the concept of neighborhood to a wide class which they called non-local mans (NL-means). This algorithms class defines the neighborhood $\mathcal{N}_i$ of $i$ under the following condition:

$j \in \mathcal{N}_i \Leftrightarrow$ the grey level of a whole window around $j$is close to the grey level of the window around $i$

In simple words, we are relaxing the spatial constraint of the classical neighborhood filters.

### D.1.1 Comparison Principles

A systematic comparison of the huge variety of denoising methods is requested. The authors consider that visual comparison of artificially noisy images with their denoised version is subjective (which is a usual technique in papers), moreover this comparison methods depend strongly on the choice of the image and do not permit to address the main issues: the loss of image structure in noise and the creation of artifacts.
The authors propose three comparison principles aiming at more objective benchmarks.

The first principle asks that noise and only noise be removed from an image. It has to be perceptually tested directly on an image with no artificial noise added, then the idea is compare the difference between the image and its denoised version for each method. We will call this difference 'method noise'. With this comparison method is much easier to evaluate whether a method noise contains some structure removed from the image or not.

The second principle, which we will call 'noise to noise' relies on the idea that a denoising algorithm transforms a white noise into a white noise. This may be seen as a paradoxical requirement, but it is a good way to characterize artifact-free algorithms. Also, we have a powerful mathematical tools for testing: Mathematical analysis and Fourier spectrum testing.

The third principle, which we will call 'statistical optimality' is restricted to neighborhood filters, and is based to question if a given NBH filter is able or not to retrieve faithfully the neighborhood $\mathcal{N}_i$ of any pixel $i$.

As we will see later, the NL-means is the one with the best results on this principles.

## D.2 Noise Model

In this section we study barely a classic model for noise, this is the model for the CCD devices and the main result of this model is an hypothesis that will made the NBH filters useful for denoising.

In CCD devices, we have three kinds of noise:

1. Shot Noise: This noise is proportional to the square root of the number of incoming photons in the captors during the exposure time, namely:

$$n_0 = \sqrt{\Phi \frac{t}{h\nu} \cdot A \cdot \eta} = C\sqrt{\Phi}$$

   where $\Phi$ is the light power, $h\nu$ the photon energy, $t$ the exposure time, $A$ the pixel area, and $\eta$ the quantum efficiency. Joining all constants in $C$ we have the last formula. Where $\Phi$ can be understood as the true image.

2. Dark or Obscurity Noise: We will denote this noise as $n_1$, is due to spurious photons produced by the captor itself. We can assume the dark noise to be white, additive and with zero mean.

3. Read out Noise: We will denote this noise as $n_2$, is another electronic additive and signal independent noise. Can be assumed to have zero mean.

Also, we have to consider another correction, a "gamma" correction: a nonlinear increasing contrast change, we will denote it as a function $f$ applied to the noisy image. It is applied as an internal adjustment in rendering of images through photography, television and computer imaging. Usually we take: $f(x) = x^\alpha$ with $\alpha \in (0, 1)$
Summarizing we have:

$$u(i) = f(\Phi(i) + c\sqrt{\Phi(i)} + n_1(i) + n_2(i))$$

When $\Phi(i)$ is large the shot noise $\sqrt{\Phi(i)}$ dominates $n_1$ and the signal $\Phi(i)$ dominates $n_2$, this let us to expand $u(i)$:

$$u(i) \sim f(\Phi(i)) + f'(\Phi(i))(C\sqrt{\Phi(i)} + n_1(i) + n_2(i)) =: f(\Phi(i)) + n(i)$$

If instead $\Phi(i)$ is small with respect to $n_1(i) + n_2(i)$:

$$n(i) \sim u(i) \sim f(n_1(i) + n_2(i))$$

in the particular (and interesting case) of $\alpha = 1/2$ we have:

$$n(i) \sim \begin{cases} n_0(i) & \text{bright parts of the image} \\ \sqrt{n_1(i) + n_2(i)} & \text{dark parts of the image} \end{cases}$$

In all cases the noise is signal dependent but independent at different pixels.

In the following we aim at recovering $f(\Phi(i))$, ie. the true image up to the unknown gamma correction. The approximations we made for $u(i)$ and the white noise and independence assumptions on $n_0, n_1, n_2$ legitimate the following important hypothesis:

**Hyphotesis**: In a digital image, the noise model at each pixel $i$ only depends on the original pixel value $\Phi(i)$ and is additive. Let $\mathcal{N}_i$ the set of pixels with the same original value as $i$. Then $n(j), j \in \mathcal{N}_i$ are independent and identically distributed. (i.i.d.)

This hypothesis leads us to give a "proof" of the correctness of NBH (and NL-means) algorithms:
Given a pixel $i$, let $j \in \mathcal{N}_i$ all the pixels that follow the same model of $i$, i.e. $\forall j \in \mathcal{N}_i : \ u(j) = v(i) + n(j)$, where $v(i)$ is a deterministic function, $n(j)$ are i.i.d. noise. Then, considering the denoising filter:

$$NF(u(i)) := \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} u(j)$$

Thanks to the hypothesis and variance formula for independent variables we have:

$$NF(u(i)) = v(i) + \tilde{n}(i)$$

where:

$$Var(\tilde{n}(i)) = \frac{1}{|\mathcal{N}_i|} Var(n(i)) \leq Var(n(i))$$

i.e. these filters reduce the variance of the residual noise.

## D.3 General Neighborhood Filters

### D.3.1 Local NBH Filters

We will present these filters in order or complexity:
The first one, and then the most "primitive", is based in replacing the color of a pixel with an average of the nearby pixels colors, i.e. $\mathcal{N}_i$ is just a spatial neighborhood. The filtered value for the pixel $x$ is:

$$\mathcal{M}_\rho(u(x)) = \frac{1}{\pi\rho^2} \int_{\mathbb{R}^2} e^{-\frac{|x-y|^2}{\rho^2}} u(y) dy$$

Where $\rho$ is a parameter that controls (roughly) the size of the neighborhood. The problem of this filter relies on the case when a spatially closer pixels of the pixel $i$ don't have similar colors as $i$. When the color is replaced by an average of very distinct colors it produces blurring in the border of the transition of colors.
This suggest the neediness of a model which includes a weight to discard closer but 'too much' different pixels for the averaging, this is the idea for the Sigma filter (Lee 1983, Yaroslavsky 1985): Average neighboring pixels which also have a similar color value, the filtered value is given by:

$$NF_{h,\rho}(u(x)) = \frac{1}{C(x)} \int_{B_\rho(x)} e^{-\frac{|u(x)-u(y)|^2}{h^2}} u(y) dy$$

Only pixels inside $B_\rho(x)$ are averaged, $h$ controls the color similarity (is, roughly speaking, the tolerance for the color similarity) and $C(x)$ a is normalization factor.
Later, to avoid dependence of a "Ball Neighborhoods" we have the filters SUSAN (Smith and Brady 1997) and bilateral (Tomasi and Manduchi 1998) where the ball neighborhoods are replaced by exponential penalization on space, ie. we have "bilateral" Gaussian depending on both space and grey level:

$$SNF_{h,\rho}(u(x)) = \frac{1}{C(x)} \int_{\mathbb{R}^2} e^{-\frac{|x-y|^2}{\rho^2}} \cdot e^{-\frac{|u(x)-u(y)|^2}{h^2}} u(y) dy$$

Another way to avoid the blurring of the spatial filtering $\mathcal{M}_\rho$ is by a statistical correction due to Lee in 1980:

$$L\mathcal{M}_\rho(u(x)) = \mathcal{M}_\rho(u(x)) + \frac{\sigma_x^2}{\sigma_x^2 + \sigma^2}(u(x) - \mathcal{M}_\rho(u(x)))$$

where

$$\sigma_x^2 = \max\left(0, \frac{1}{\pi\rho^2} \int_{\mathbb{R}^2} e^{-\frac{|x-y|^2}{\rho^2}} (u(y) - \mathcal{M}_\rho(u(x)))^2 dy - \sigma^2\right)$$

The idea of this correction is based on the following observation: When the Gaussian mean is performed on an edge, the variance of the performed mean can become larger than the variance of the noise, this phenomena is not desired, and the correction tries to avoid this.
Bilateral filters anyway have a better performance than Lee's correction.
Notice that we can replace the mean operation by nonlinear operator like the median filter. Tukey in 1997 gives a median filter, this chooses the median value, that is, the value which has exactly the same number of grey level values above and below in a fixed neighborhood. It is equivalent to an average of the pixels in a direction orthogonal to the gradient, that is to an anisotropic diffusion or mean curvature motion.
A small comparison of this neighborhood filters that can be seen on the original paper gives us non fully acceptable results: Gaussian filtering don't maintain sharp edges, anisotropic filter removes small details and fine structures, Lee's statistical filter leave some areas untouched (then noisy), sigma and bilateral creates irregularities on the edges. This comparison make the neediness of consider a new model.

### D.3.2 Non Local Averaging

As we said before, the main idea of this model is based on the simply observation that the most similar pixels to a given pixel have no reason to be close of it (for example in periodic patterns), then the idea is to construct a filter which consider pixels with neighborhoods with similar average values as the neighborhood of the original pixel (then, we don't have spatial constraint). Then, the proposed formula is:

$$NL(u(x)) = \frac{1}{C(x)} \int_{\Omega} \exp\left(-\frac{g_\rho * |u(x+\cdot) - u(y+\cdot)|^2(0)}{h^2}\right) u(y)dy$$

where $g_\rho$ is a Gaussian kernel with standard deviation $\rho$, $C(x)$ is the normalizing factor, $h$ acts as a filtering parameter and:

$$g_\rho * |u(x+\cdot) - u(y+\cdot)|^2(0) = \int_{\mathbb{R}^2} g_\rho(t)|u(x+t) - u(y+t)|^2 dt$$

This last formula reveals the most important characteristic of this filter, NL replace the value of $u(x)$ by a weighted mean of $u(y)$. The weights is relevant only if a Gaussian window around $y$ is similar to the same window around $x$. This is the concept of self-similarity.
NL-means works great with text images, but is limited when an image have structured noise (like JPEG compression), in that case NL lose details. More specific information can be seen on the original paper.

## D.4 Principles for Denoising Algorithms Evaluation

We will enounce the formal assertions for this principles that we mentioned before in the introduction.

### D.4.1 Method Noise

As we said before, the idea of this principle is to evaluate if an algorithm just removes noise, or it also removes some structure of the image.
**Definition**: Let $u$ be a (not necessarily noisy) image and $D_h$ a denoising operator depending on $h$. The method noise of $u$ is the image difference:

$$n(D_h, u) = u - D_h(u)$$

and the formal principle will be:
**Principle 1**: For every denoising algorithm, the method noise must be zero if the image contains no noise, and should be in general an image of independent zero-mean random variables.
Examples of the evaluation of algorithms under this principle can be seen on the original paper, anyway, roughly speaking, the NL-means have the best results for this principle.

### D.4.2 Noise to Noise Principle

As we said before, the idea of this principle is; accepting that no algorithm can remove all the noise from an image, at least we want to transform noise in noise with less variance. This is a way to check if an algorithm reduces the noise, and also don't create artifacts on images.
**Principle 2**: A denoising algorithm must transform a white noisy image into a white noisy image (with lower variance).
As we said before, this principle have a good way to be checked: Studying the Fourier transform of denoised image, because we know that the Fourier Transform of a white Gaussian noise is a white Gaussian noise, so, visualizing the FT of the denoised image we will see if it remain as a white Gaussian noise, or it have changed in wrong way (creating artifacts).
Several algorithms have been checked with this principle (that can be seen on the original paper), and bilateral filters and NL-means report the best results.

### D.4.3 Statistical Optimality

We will understand statistical optimality as the ability of a generalized neighborhood filter to find the right set of pixels $\mathcal{N}_i$ for performing the average yielding the new estimate for $u(i)$, obviously this principle

will be useful just for neighborhood (or averaging en general) methods.
**Principle 3**: A generalized neighborhood filter is optimal if it finds for each pixel $i$ all and only the pixels $j$ having the same model as $i$. Obviously is impossible to check if the pixels in $\mathcal{N}_i$ satisfy $\Phi(j) = \Phi(i)$, then in general this condition is relaxed to check if the pixels $j$ are likely to have the same value as $i$. Examples are given in the original paper (anyway, this principle is more useful for movie denoising).

## D.5 Numerical Examples

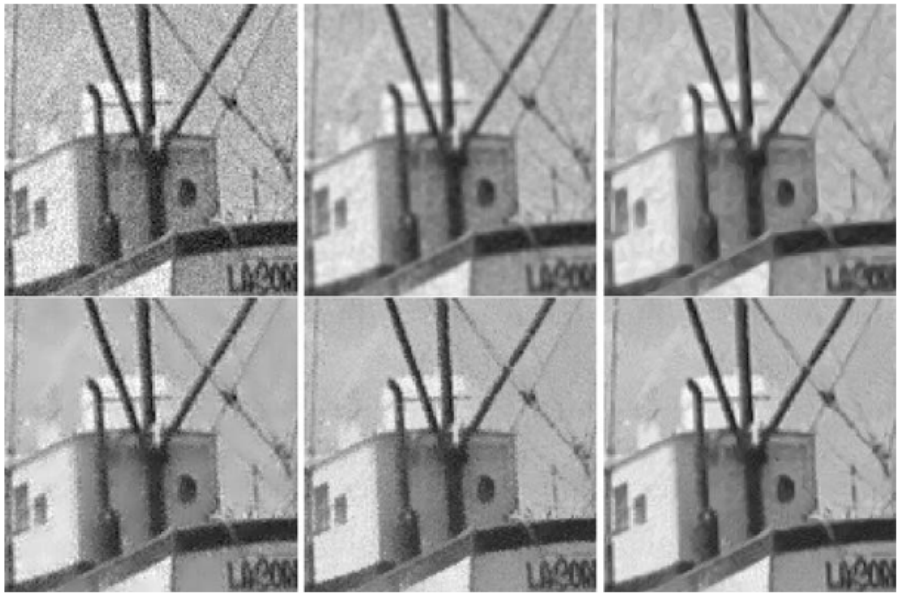


Figure D.1: Comparison of neighborhood filters. From top to bottom and left to right: noisy image (with gaussian noise with $\sigma = 15$), Gaussian filtering, anisotropic filtering, Lee's statistical filter, sigma or bilateral filter and the NL-means algorithm. All methods except the Gaussian filtering maintain sharp edges. However, the anisotropic filtering removes small details and fine structures. These features are nearly untouched by Lee's statistical filter and therefore completely noisy. The comparison of noisy grey level values by the sigma or bilateral filter is not so robust and irregularities are created on the edges. The NL-means better cleans the edges without losing too many fine structures and details.

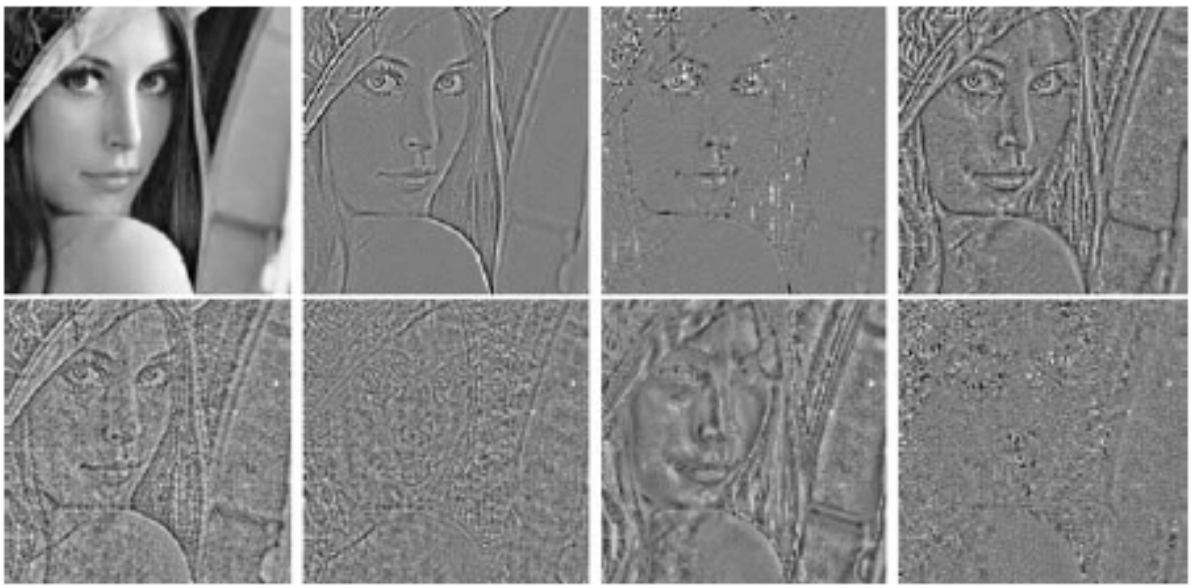


Figure D.2: Method noise experiment on Lena (gray levels only). From top to bottom and left to right: original image, Gaussian mean, mean curvature motion, total variation minimization, translation invariant soft and hard thresholding, bilateral filter and NL-means

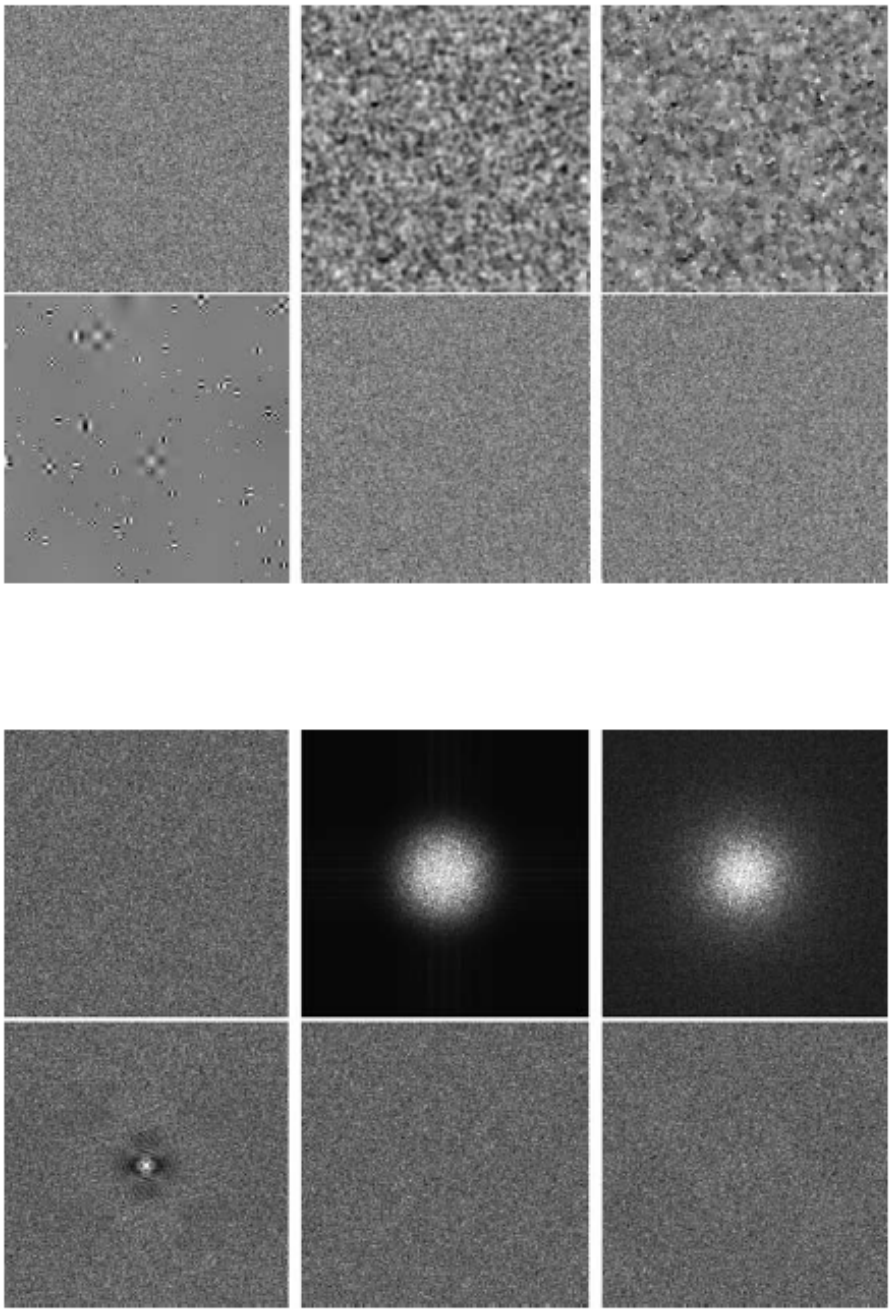


Figure D.3: Noise to noise principle: Upper images: Application of the denoising algorithms to a noise sample. From left to right and top to bottom: noise sample, filtered noise by the Gaussian filtering, total variation minimization, hard wavelet thresholding, bilateral filter and the NL-means algorithm. The parameters of each algorithm have been tuned in order to have a filtered noise of standard deviation 2.5. For the neighborhood or bilateral filter the research zone has been fixed to $21 \times 21$ and for NL-means we have used the whole image. Therefore, only the $h$ parameter has been tuned in order to obtain the desired standard deviation Lower Images: Noise to Noise principle: Fourier transforms of the filtered noises displayed in the upper images. The Fourier transform of a Gaussian white noise is a Gaussian white noise

# E. Nonlocal Linear Image Regularization and Supervised Segmentation - G. Gilboa, S. Osher

# Nonlocal Linear Image Regularization and Supervised Segmentation - G. Gilboa, S. Osher [7]

March 11, 2013

## E.1 Introduction: Objective, Problems on NL-means and Variational Point of view

The main objective of this paper is to give an unified approach to both denoising and segmentation tasks using nonlocal functionals and their respective nonlocal evolutions.

Recall the NL-means filter proposed by Buades, Coll and Morel in [1]

$$NL(u(x)) = \frac{1}{C(x)} \int_\Omega e^{-d_a(u(x),u(y))/h^2} u(y)dy$$

where

$$d_a(u(x), u(y)) = \int_\Omega g_a(t)|u(x+t) - u(y+t)|^2 dt$$

where $g_a(t)$ is a gaussian with variance $a$.
The normalization used here (performed by $C(x)$) does not guarantee that the mean value of the filtered image $u$ is the same as the mean value of the input image, this is not desired (for example when we have Gaussian white noise). Also, normalizing in this manner introduces some bias from the original distance between points with many similar regions (high $C(x)$) to more rare and singular points (low $C(x)$). Dividing by $C(x)$ tends to diminish this distinction. The authors will show a different normaliation, which retains symmetric similarities between points, preserve the mean value and does not tend to blur singular regions. They believe this may explain, in part, why their proposed iterative process outperforms the original filter.

Kindermann-Osher-Jones interpreted the NL-means (and NBH filters in general) as regularizations based on nonlocal functionals in the general form:

$$J_{KOJ}(u) := \int_{\Omega\times\Omega} f\left(\frac{|u(x) - u(y)|^2}{h^2}\right) w(|x-y|)dxdy$$

Particular cases are the Yaroslavsky functional $J_{Yar}$ where $f\left(\frac{|u(x)-u(y)|^2}{h^2}\right) = 1 - \exp\left(-\frac{|u(x)-u(y)|^2}{h^2}\right)$ or the NL-means functional $J_{BCM}$ where $f\left(\frac{|u(x)-u(y)|^2}{h^2}\right) = 1 - \exp\left(-\frac{d_a(u(x),u(y))}{h^2}\right)$
The idea is to solve a minimization problem using the above functionals to have a solution (which will be the filtered image). Notice that we have $w(|x-y|)$, in this cases this represents a simple symmetric window and $f(\cdot)$ is the most important part of the regularization. The main problem is that in general this functionals are not convex.
The authors follow this approach, but simplified to a quadratic functional by changing the roles of $f$ and $w$.

## E.2 Regularization Functional

The authors consider the following regularization functional:

$$F_{reg}(u) := \frac{1}{4} \int_{\Omega\times\Omega} (u(x) - u(y))^2 w(x,y)dxdy$$

This could be considered for $\Omega \subset \mathbb{R}^n$, and for images for $n = 2$. The weight function $w(x, y)$ is positive $(w(x, y) \leq 0)$ and symmetric: $w(x, y) = w(y, x)$.
For our purposes $w$ is based on image features and can be understood as the proximity between two points $x$ and $y$, based on features in their neighborhood. Notice the difference between the functional given before, the role of $w(x, y)$ is much more important now, it basically determinines the regularization.
Notice the corresopnding Euler-Lagrange descent flow is:

$$u_t(x) = -F'_{reg}(u)(x) = -\int_\Omega (u(x) - u(y))w(x,y)dy \tag{E.1}$$

Define now the following linear operator:

$$Lu(x) = \int_\Omega (u(y) - u(x))w(x,y)dy \tag{E.2}$$

Notice that $L$ can be seen as the continuous version of the graph Laplacian (where the undirected graph $G = (V, E)$ have weights in every arc). We will see that $L$ have similar properties with the operator $\div(c(x)\nabla)$ where $c(x)$ is a symmetric matrix, $c(x) > 0$.
Assuming our initial condition as the initial input image $f$, we can write the descent flow as the following linear (non local) diffusion problem:

$$u_t(x) = Lu(x), \quad u|_{t=0} = f(x) \tag{E.3}$$

## E.3 Variational Denoising

One can add a convex fidelity term to the convex functional $J$. For the $L^2$ fidelity we can consider:

$$J(u, f) = F_{reg}(u) + F_{sim}(u) = F_{reg}(u) + \frac{\lambda}{2}||u - f||_2^2 \tag{E.4}$$

then, $u$ satisfies the Euler-Lagrange equation:

$$-Lu + \lambda(u - f) = 0 \tag{E.5}$$

This can be seen as a constrained problem:

$$u := \arg\min F_{reg}(u), \quad s.t.\ ||u - f||_2^2 = |\Omega|\sigma_n^2$$

where $\sigma_n^2$ is the variance of an additive noise in a noisy image $f$. Then $\lambda$ can be viewed as a Lagrange multiplier and one can compute the constrained problem by initializing e.g. with $u|_{t=0} = f$, $\lambda = 1$ and iterating:

$$u_t = Lu + \lambda(f - u)$$
$$\lambda = \frac{1}{|\Omega|\sigma_n^2}\int_\Omega (u - f)Lu dx$$

## E.4 Multichannel signals

For the case when we have a multichannel signals: let $f(x) = (f^1, f^2, \ldots, f^M)(x)$ be a $M$ channel signal. A multi-valued affinity function (as we will see later) is used to compute $w(x, y)$ based on $f$ ($w(x, y)$ will be the same for all channels). Let $u(x) = (u^1, \ldots, u^M)(x)$ be the regularized signal. The regularizing functional in this case is:

$$F_{reg}^{mc}(u) := \frac{1}{4}\sum_{i=1}^{M}\int_{\Omega\times\Omega}(u^i(y) - u^i(x))^2 w(x, y)dy \tag{E.6}$$

and the multi-channel evolution for each channel $u^i$ is:

$$u_t^i(x) = \int_\Omega (u^i(y) - u^i(x))w(x, y)dy, \quad u^i|_{t=0} = f^i \tag{E.7}$$

## E.5 Properties of $L$

The following properties of the linear operator $L$ leads to establish some results for the flow (E.3) and for the variational problem (E.5)

**Proposition E.5.1.** *The linear operator $L$ defined by (E.2) satisfies the following properties:*

1. *If $u(x) = const$ then $Lu(x) = 0$. For $w(x, y) > 0$, $\forall x, y \in \Omega$, if $Lu(x) = 0$ then $u(x) = const$*
2. *Let $u(x_0) \geq u(x)$, $\forall x \in \Omega$, then $Lu(x_0) \leq 0$. For the minimum let $u(x_1) \leq u(x)$ then $Lu(x_1) \geq 0$.*
3. *$-L$ is a positive semidefinite operator i.e. $\langle -Lu(x), u(x)\rangle \geq 0$ where we consider the $L^2$ inner product.*
4. *$\int_\Omega Lu(x) = 0$*

We will need an additional technical condition on the weight function. The idea of this condition is that $w(x,y)$ can have zero values, but we need a certain level of connectivity, in the context of that there will not be any disjoint regions where no information is exchange between them throughout the evolution. Consider:

$$-L \text{has a zero eigenvalue of multiplicity 1.} \tag{E.8}$$

This condition is equivalent to state that $-L$ has only a constant function in its null-space. In graphs, this condition is equivalent to a connected graph (in the sense that we stated before) when the linear operator is the graph Laplacian. In our case the specific relation is given by the following lemma:

**Lemma E.5.2.** *Condition (E.8) holds if and only if for any two points $x, y$ there exists a sequence of points: $z_1, \ldots, z_k$ such that $w(x,z_1) \cdot \ldots \cdot w(z_k,y) > 0$*

With this technical condition and the properties of the linear operator $L$ we can prove some important properties of the flow (E.3):

**Proposition E.5.3.** *The flow defined by (E.3) satisfies:*

1. *Preservation of mean value:*
$$\frac{1}{|\Omega|}\int_\Omega u(x,t)dx = \frac{1}{|\Omega|}\int_\Omega f(x)dx$$
2. *The extremum principle holds*
$$\min_x f(x) \le u(x,t) \le \max_x f(x) \quad \forall x \in \Omega, \forall t \ge 0$$
3. *For $w(x,y)$ which admits condition (E.8), the solution converges to a constant*
$$u(x, t\to\infty) = const = \frac{1}{|\Omega|}\int_\Omega f(x)dx$$
4. *The following estimate holds:*
$$\frac{1}{2}\frac{d}{dt}||u(x,t)||_2^2 \le 0$$

Notice that from properties $(i)$ and $(iv)$ we have $\frac{d}{dt}var(u(t)) \le 0$, where $var(u)$ is the (empirical) variance of $u$, defined by:

$$var(u) := \frac{1}{|\Omega|}\int_\Omega \left(u(x) - \int_\Omega u(y)dy\right)^2 dx$$

Similar properties can be found for the variational formulation (E.5):

**Proposition E.5.4.** *The minimizer $u^\lambda$ of (E.5) admits the following properties:*

1. *Preservation of mean value:*
$$\frac{1}{|\Omega|}\int_\Omega u^\lambda(x)dx = \frac{1}{|\Omega|}\int_\Omega f(x)dx, \quad \forall \lambda \ge 0$$
2. *The extremum principle holds*
$$\min_x f(x) \le u^\lambda(x) \le \max_x f(x) \quad \forall x \in \Omega, \forall \lambda \ge 0$$
3. *For $w(x,y)$ which admits condition (E.8), the solution converges to a constant when $\lambda \to 0$*
$$\lim_{\lambda\to 0} u^\lambda(x) = const = \frac{1}{|\Omega|}\int_\Omega f(x)dx$$
4. *The following estimate holds:*
$$\frac{1}{2}\frac{d}{d\lambda}||f - u^\lambda||_2^2 \le 0$$

## E.6 Weights based on affinity functions

As we seen before, the weights $w(x,y)$ will define the regularization induced by the functional $J(u)$. The idea is to consider a basic affinity structure as the similarity between image features. For a point $x \in \mathbb{R}^2$ we will assign a feature vector denoted by $F_f(x)$ (a image feature could be grey level value, edge indicator, dominant direction, dominant frequency, etc.) We will consider also the region $\Omega_w(x) \subset \Omega$ as the points $y$ where $w(x,y) > 0$. Due to symmetry of $w(x,y)$ we will have the symmetry of this sets in the following sense: $y \in \Omega_w(x) \Leftrightarrow x \in \Omega_w(y)$.
Then, in general we will have the general weight function based on affinities:

$$w(x,y) = \begin{cases} h(F_f(x), F_f(y)) & y \in \Omega_w(x) \\ 0 & otherwise \end{cases} \tag{E.9}$$

where $h(s_1, s_2)$ is a similarity function with the following properties:

- Positive: $h(s_1, s_2) > 0$
- Symmetric: $h(s_1, s_2) = h(s_2, s_1)$
- Bounded: $h(s_1, s_2) \leq M < \infty$
- Maximal at equality: $h(s_1, s_1) \geq h(s_1, s_2)\ \forall s_1, s_2$

Then, in general, for features in a suitable Banach space, with a norm $||\cdot||$ a typical choice for similarity function is:

$$h(s_1, s_2) = e^{-(||s_1 - s_2||/m)^p} \tag{E.10}$$

where $m$ is a threshold parameter. The power $p \geq 1$, in general is set to $p = 2$ in case of Euclidean norm.

### E.6.1 Weights examples

Intensity, local

$$\begin{aligned} h(F_f(x), F_f(y)) &= e^{-(|F_f(x) - F_f(y)|/h)^2} \\ F_f(x) &= f(x) \\ \Omega_w(x) &= \{y \in \Omega \mid |y - x| \leq \Delta x\} \end{aligned}$$

where $\Delta x$ is the grid size. This results in a 4 nearest neighbors discretization.

Intensity, weighted, semi-local:

$$\begin{aligned} h(F_f(x), F_f(y)) &= e^{-(|F_f(x) - F_f(y)|/h)^2} e^{-|x-y|^2/(2\sigma_d^2)} \\ F_f(x) &= f(x) \\ \Omega_w(x) &= \{y \in \Omega \mid |y - x| \leq r\} \end{aligned}$$

where $\sigma_d$ controls the spatial decay and $r$ is the window radius (should be in order of $\sigma_d$)

For textures, let $K^1(x), \ldots, K^M(x)$ be $M$ linear filters of different directions and frequencies. Let $v^i := u * K^i$. The weights can be computed by:

$$\begin{aligned} h(F_f(x), F_f(y)) &= e^{-(|F_f(x) - F_f(y)|/h)^2} \\ F_f(x) &= (v^1, \ldots, v^M)(x) \\ \Omega_w(x) &= \{y \in \Omega \mid |y - x| \leq r\} \end{aligned}$$

The nonlocal version of Yaroslavsky affinity:

$$\begin{aligned} h(F_f(x), F_f(y)) &= e^{-(|F_f(x) - F_f(y)|/h)^2} \\ F_f(x) &= f(x) \\ \Omega_w(x) &= \Omega \end{aligned}$$

NL-means affinity:

$$\begin{aligned} h(F_f(x), F_f(y)) &= e^{-(|F_f(x)-F_f(y)|/h)^2} \\ F_f(x) &= f(x) \in B_x \text{ where } B_x\text{is a patch centered at } x \\ \Omega_w(x) &= \Omega \end{aligned}$$

## E.7 Discretization

We will require that the weights are sparse enough, this will made the complexity of the algorithm linear. This constraint is usually not very limitating since in the general case if we have a large window many connections have very low weight values which can be set to zero. If there are many connections with high weight values we suggest to take randomly only a part of them. The iterative process can usually compensate this random choice.
Let $u_k$ the value of the pixel $k$ in the image ($1 \leq k \leq N$), $w_{kl}$ is the sparsely discrete version of $w(x, y)$. Recall the neighborhood notation: $l \in \mathcal{N}_k = \{l : w : kl > 0\}$. The flow (E.3) is implemented by the explicit in time forward Euler approximation:

$$u_k^{n+1} = u_k^n + \Delta t \sum_{l \in \mathcal{N}_k} w_{kl}(u_l^n - u_k^n) \tag{E.11}$$

where $u_k^n = u_k(n\Delta t)$ All the coeficients on the right side are nonnegative if:

$$1 \geq \Delta t \sum_{l \in \mathcal{N}_k} w_{kl} \tag{E.12}$$

This is the well known CFL restriction on the time step $\Delta t$. This leads to maximum norm stability, in fact a maximum principle, for this approximation to (E.3) For the problem with fidelity term, we have the following discretization:

$$u_k^{n+1} = u_k^n + \Delta t \sum_{l \in \mathcal{N}_k} w_{kl}(u_l^n - u_k^n) + \lambda \Delta t(f_k - u_k^n) \tag{E.13}$$

and this implies the analog CFL condition:

$$1 \geq \Delta t \left( \sum_{l \in \mathcal{N}_k} w_{kl} + \lambda \right) \tag{E.14}$$

## E.8 Computing weights for non local means

The most important (and demanding in computational terms) part of the method is the approximation of $w(x, y)$ into a sparse discrete version $w_{kl}$ (where we need the conditions established before). The authors propose two algorithms: the first, a semilocal one, is proposed for denoising purposes; the second, a fully nonlocal (with random choices instead of checking all the possibilities) is intended for nonlocal segmentation.

#### E.8.0.1 Semi-local version

**Algorithm**. For each pixel $k$:

1. 1) Compute the similarity of all the patches in the window (Authors use $5 \times 5$ patch $B_x$ and $11 \times 11$ window $\Omega_w$). Construct $\mathcal{N}_k$ by taking the $m$ (Authors use $m = 5$) most similar and the four nearest neighbors (for conexity condition) of the pixel.
2. 2) Compute the weights $w_{kl}$, $l \in \mathcal{N}_k$ using the desired weight function and set to zero all the other connections. (i.e. $w_{kl} = 0$, $l \notin \mathcal{N}_k$
3. 3) Set $w_{lk} = w_{kl}$ (symmetry of weight)

For the CFL condition needed, if we have a weight function bounded by 1, we can choose $\Delta t = \frac{1}{2m+4}$ to satisfy this condition. If we add a fidelity term, this is also enough (if $\lambda < 1$).
The complexity of this algorithm is $O(N \times Window_{size} \times (Patch_{size} + \log m))$.

### E.8.1 Fast approximation for the fully nonlocal version

The following algorithm is based on ideas presented in [15]. It is simpler and faster but not accurate, anyway the results are better than the original fully nonlocal version.

**Algorithm**:

1. 1) Compute the mean and the standard deviation of all patches in the image. Create a two dimensional bin table such that all patches in a bin are within a specific range of mean and s.d. from each other. Both types of bins are spaced in $h/2$ increments.
2. 2) To construct the set $\mathcal{N}_k$: For each pixel $k$ we consider the 9 bins around it ($3\times 3$ window in the table, this ensures that patches which are very similar are taken into account). Pick randomly $3m$ patches from these bins, check their similarity to the patch of pixel $k$ and take the most similar $m$ of them. Add to $\mathcal{N}_k$ also the four nearest neighbors (for connectness)
3. 3) Compute $w_{kl}$ as in the local algorithm.

## E.9 Conclusions for denoising

The main conclusions for the denoising scheme proposed by the authors (based on the several examples that can be found on the original paper) can be summarized as follows:

1. 1) A semi-local search window $\Omega_w = \{y \in \Omega \ / \ |y - x| \leq r\}$ performs better than a fully nonlocal one (i.e. $\Omega_w(x) = \Omega$)
2. 2) The steepest descent flow performs better than the variational minimization for the same regularizer $J(u)$.
3. 3) The proposed flow performs better than the original NL-means filter as well as several well-known local PDE based regularizations.

## E.10 Algorithm for (Supervised) Segmentation

The following algorithm for nonlocal segmentation is based on method that are used in the field of classification and machine learning. The key point is realize that we use the same flow for this task, the only thing that change is the initial condition. For more information about the deduction of this model see the original paper.
Let $f$ the input image and $w(x, y)$ the corresponding weight function. Let $\Omega_0^O$ be an initial set which corresponds to the object to be segmentated, and let $\Omega_0^B$ be an initial set which corresponds to the background. In the following algorithm these regions are defined by the users, who marks them as an initial condition, specifying the object to be segmentated and the corresponding background.

**Algorithm**

1. 1) Initialize
$$u_0 := \begin{cases} 1 & x \in \Omega_0^O \\ -1 & x \in \Omega_0^B \\ 0 & \text{otherwise} \end{cases}$$
2. 2) Evolve for a duration $T$ the flow
$$u_t = Lu, \quad u|_{t=0} = u_0 \tag{E.15}$$
3. 3) Define $\Omega^O$, the set of nodes approximating the Object, by:
$$\Omega^O := \{x \in \Omega : u(x, T) > 0\}$$
and the Background by: $\Omega^B = \Omega \setminus \Omega^O$

In the case of multiple objects the algorithm can be generalized in the following way: Let $\Omega_0^1, \ldots, \Omega_0^M$ $M$ disjoint sets which are parts of $M$ regions to be segmented (including the background), this is the data defined by the user, then the algorithm is:

**Algorithm**

1. 1) Initialize a $M$ channel signal $u^i, i = 1, \ldots, M$ as follows:

$$u_0^i := \begin{cases} 1 & x \in \Omega_0^i \\ 0 & \text{otherwise} \end{cases}$$

2. 2) Evolve for a duration $T$ the flow

$$u_t^i = Lu^i, \quad u^i|_{t=0} = u_0^i \tag{E.16}$$

3. 3) Define $\Omega^i$, the set of nodes approximating the Object $i$, by:

$$\Omega^i := \{x \in \Omega : i = \arg\max_j u^j(x, T)\}$$

As a few remarks:

1. This algorithms can be related to image coloring, where a grey-scale image can be colored by user-provided coloring examples
2. A typical example where this algorithm fail is when the user marks are not even (geometrically speaking) and this marks dont sorround the object. An example of this can be found on the original paper.

## E.11 Numerical Examples

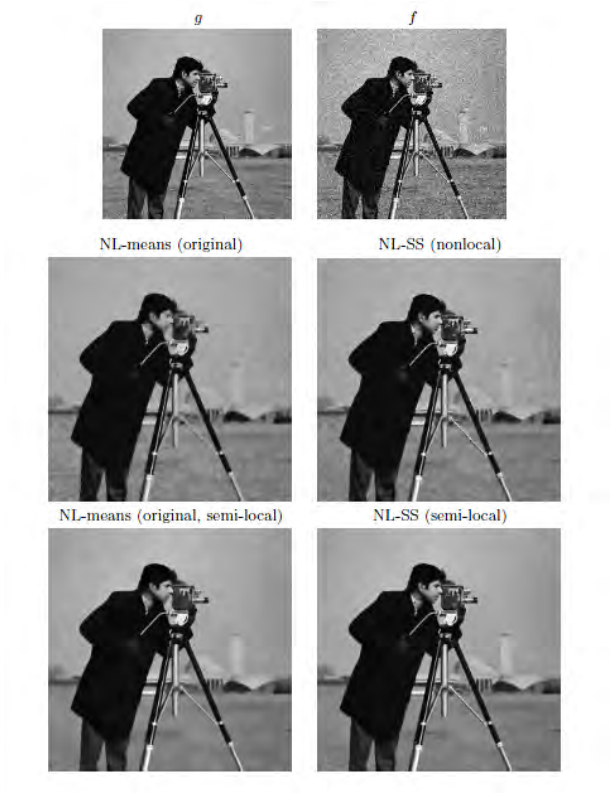



Figure E.1: Top row: Test images. Clean image $g$ (left), noisy image $f$. Cameraman image filtering result u. Second row: NL-means (nonlocal), and filtering by nonlocal scale-space (proposed flow). Third row: NL-means ($11 \times 11$ window), proposed nonlocal scale space ($11 \times 11$ window).

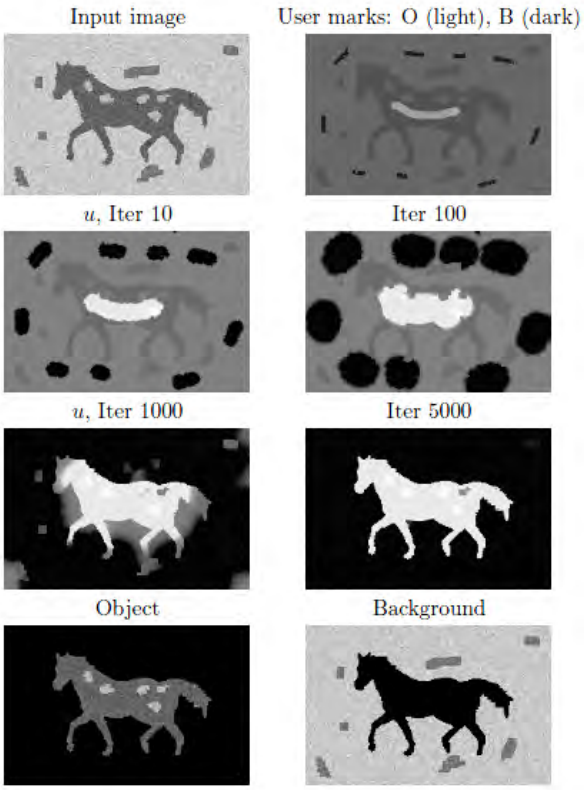



Figure E.2: Horse silhouette, with spots in both object and background, white Gaussian noise is added. 4-neighbor scheme. In the second and third rows, the advancement of the information with the iterations is illustrated

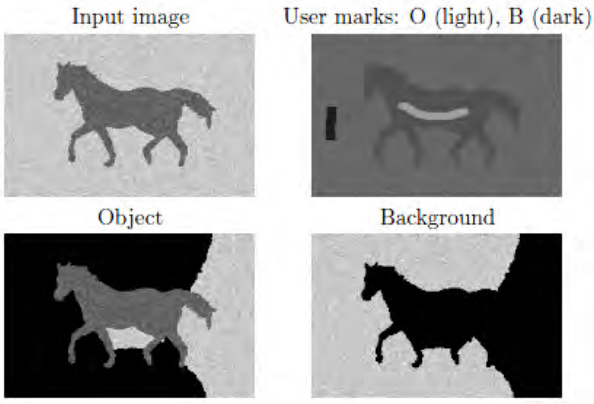



Figure E.3: Failure example: Marks of the background are too sparse and uneven.

# F. Generalised Nonlocal Image Smoothing - L. Pizarro, P. Mrázek, S. Didas, S. Grewenig, J. Weickert [12]

The main idea of this paper is to provide a discrete variational approach for image smoothing based on non local similarity and regularization constraints which penalize general dissimilarity measures defined in image patches.
The classical similarity measure is the weighted $L^2$ distance between patches, as we seen before in NL-means. For this purpose the authors first introduce the NDS model defined by Mrázek, which generalizes other models in the sense that with a choice of parameters this model reduces to known others.
After that, the authors propose a modification of the NDS model, the Generalised NDS model or GNDS, which includes the patch similarity ideas that can be seen on the NL-means filter which leads to more versatility and robustness for local structure of the image.

## F.1 NDS model: Nonlocal Data and Smoothness

Let $f, u : \Omega \to \mathbb{R}$ scalar images defined in the discrete domain $\Omega$, $f$ must be understood as the original (noisy) image while $u$ represents the "processed" version of $f$. Also, let $J = \{1, \dots, N\}$ the index set of all pixels in the images. We denote the pixel $i$ as $x_i$.
The discrete Energy $J^{NDS}$ of the NDS filter proposed by Mrázek in 2006 is a convex combination of a nonlocal data (or similarity) term $F_{sim}$ and a nonlocal smoothness (or regularization) term $F_{reg}$, which are given by:

$$F_{sim}(u) := \sum_{i,j \in J} \Psi_s(|u_i - f_j|^2) w_s(|x_i - x_j|^2) \tag{F.1}$$

$$F_{reg}(u) := \sum_{i,j \in J} \Psi_r(|u_i - u_j|^2) w_r(|x_i - x_j|^2) \tag{F.2}$$

Where $\Psi_\cdot : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ are increasing functions which penalizes large tonal distances (in greyscale).
The weights $w_\cdot : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ are non negative functions which penalizes large distances between pixels.
Then, the complete NDS model could be understood as a discrete nonlocal variational method which combine data and smoothness:

$$J^{NDS}(u) = (1 - \alpha) F_{sim}(u) + \alpha F_{reg}(u) \tag{F.3}$$

where $\alpha \in [0, 1]$ is a regularization parameter.

### F.1.1 Critical Points and Numerical Implementation

Is interesting to study the critical points of this functional, this leads to a fixed point scheme for minimizing the energy functional and, under suitable hypothesis, this leads to a stable procedure.
First of all, we need to calculate the derivatives of main terms:

$$\frac{\partial F_{sim}}{\partial u_k} = 2 \sum_{j \in J} \Psi_s'(|u_k - f_j|^2)(u_k - f_j) w_s(|x_k - x_j|^2) \tag{F.4}$$

$$\frac{\partial F_{reg}}{\partial u_k} = 4 \sum_{j \in J} \Psi_r'(|u_k - u_j|^2) w_r(|x_k - x_j|^2) \tag{F.5}$$

Notice that derivatives on $\Psi$ are respect of its arguments, then by linearlity:

$$\frac{\partial J^{NDS}}{\partial u_k} = (1 - \alpha) \frac{\partial F_{sim}}{\partial u_k} + \alpha \frac{\partial F_{reg}}{\partial u_k}$$

Then, for a critical point we want: $\nabla J^{NDS}(u) = 0 \Leftrightarrow \frac{\partial J^{NDS}}{\partial u_k} = 0 \quad \forall k \in J$.
If we call: $s_{i,j} := \Psi_s'(|u_i - f_j|^2) w_s(|x_i - x_j|^2)$
$r_{i,j} := 2\Psi_r'(|u_i - u_j|^2) w_r(|x_i - x_j|^2)$, then the critical point condition becomes to:

$$0 = (1 - \alpha) \sum_{j \in J} s_{i,j}(u_i - f_j) + \alpha \sum_{j \in J} r_{i,j}(u_i - u_j) \tag{F.6}$$

which can be rewritten in the fixed point form:

$$u_i = \frac{(1-\alpha)\sum_{j\in J} s_{i,j} f_j + \alpha \sum_{j\in J} r_{i,j} u_j}{(1-\alpha)\sum_{j\in J} s_{i,j} + \alpha \sum_{j\in J} r_{i,j}} \tag{F.7}$$

To have a well defined expression we will need that $\Psi.$ are increasing functions, also we need a positive weight functions (i.e. $w.(s^2) > 0$). Then, from the last expression, the authors propose the following fixed point scheme:

$$u_i^0 := f_i \quad u_i^{k+1} = \frac{(1-\alpha)\sum_{j\in J} s_{i,j}^k f_j + \alpha \sum_{j\in J} r_{i,j}^k u_j^k}{(1-\alpha)\sum_{j\in J} s_{i,j}^k + \alpha \sum_{j\in J} r_{i,j}^k} \tag{F.8}$$

or, in a compact form: $u^{k+1} = F(u^k)$ where $F : \mathbb{R}^N \to \mathbb{R}^N$ is a vector function defined on each coordinate by the right side expression of (F.7). This is usually called as "Nonlinear Jacobi Method"
This fixed point scheme has two simple but important properties:

**Proposition F.1.1.** *Under the hypothesis on $\Psi.$ and $w.$ as above, the scheme (F.7) satisfies a maximum-minimum principle:*

$$\min_{j\in J} f_j \le u_j^k \le \max_{j\in J} f_j \quad \forall i \in J, k \in \mathbb{N} \tag{F.9}$$

This proposition leads to an important theorical result, which guarantees the existence of minimizers (due to their behaviour of critical points as a fixed points of the operator $F$)

**Proposition F.1.2.** *The fixed point equation (F.7) has a solution.*

The proof relies on Brouwer fixed point theorem and continuity of the operator $F$ due to the first proposition.

Alternatively, the minimization of the NDS energy can be found using a gradient descent optimization, i.e. considering:

$$\frac{u_i^{k+1} - u_i^k}{\tau} = -\frac{\partial J^{NDS}}{\partial u_i^k} \quad \forall i \in J$$

where $\tau > 0$ is the step size. In the same way as the first iterative scheme, for this we have:

$$u_i^0 := f_i \quad u_i^{k+1} = (1-\tau)u_i^k + \tau \frac{(1-\alpha)\sum_{j\in J} s_{i,j}^k f_j + \alpha \sum_{j\in J} r_{i,j}^k u_j^k}{(1-\alpha)\sum_{j\in J} s_{i,j}^k + \alpha \sum_{j\in J} r_{i,j}^k} \tag{F.10}$$

Notice that if we set $\tau = 1$ we obtain the fixed point iteration (F.8)

### F.1.2 Important Cases

Let us rewrite the equation (F.7) in a more specific way (in the sense as we can see the dependances on parameters like the window size for $w$ (which we will denote by $\overline{w}$), for example)
Then, consider the following notation:

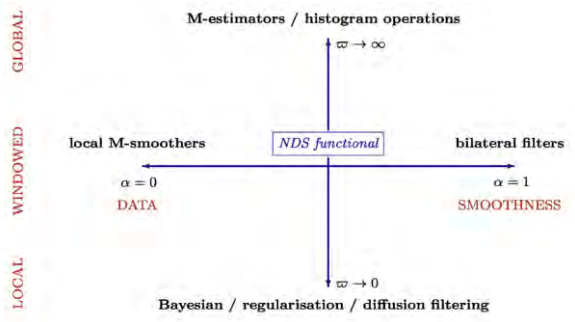
$$h_{i,j}^s := \Psi_s'(|u_i - f_j|^2) \quad h_{i,j}^r := 2\Psi_r'(|u_i - u_j|^2)$$

$$w_{i,j}^{s,\overline{w}} := w_s(|x_i - x_j|^2) \quad w_{i,j}^{r,\overline{w}} := w_r(|x_i - x_j|^2)$$

notice that a small $\overline{w}$ denotes a local operation (or a smaller window), and a larger one denotes a large-scale effects. The window size for the data and smoothness may differ (because the functions are not necesarily the same), with this notation (F.7) becomes to:

$$u_i = \frac{(1-\alpha)\sum_{j\in J} h_{i,j}^s w_{i,j}^{s,\overline{w}} f_j + \alpha \sum_{j\in J} h_{i,j}^r w_{i,j}^{r,\overline{w}} u_j}{(1-\alpha)\sum_{j\in J} h_{i,j}^s w_{i,j}^{s,\overline{w}} + \alpha \sum_{j\in J} h_{i,j}^r w_{i,j}^{r,\overline{w}}} \tag{F.11}$$

From this equation, and choosing apropiate configuration of $\overline{w}, \alpha$ and functions involved we can reduce this model to a variety of classical models. the following picture shows the landscape covered by the NDS model, we refer to the original paper for more detailed deduction for this reductions.




## F.2 Generalised Nonlocal Data and Smoothness (GNDS) Model

From the NDS model one can see that the tonal weights ($\Psi.$) just depend on differences between single pairs of pixels (i.e. the similarity measure just depends on single pixels), this single differences have limited ability to express local image structure and geometry, moreover, for practical purposes usually this differences applies only on a relatively small neighborhood.
The idea for this generalized model is to add the a more powerful measure to evaluate similarity, so the authors consider the concept of self-similarity of the whole image based on models like NL-means of Buades et al. to extend the previous model to a more versatile one. Then, the idea is to consider a measure which compares the similarity of a whole patch around a pixel between the whole patch around the other pixel, this idea is usually called patch-similarity.

### F.2.1 GNDS Functional and Its Minimization

Let us consider the following functions called "tonal distance": $d_s, d_r : \mathbb{R}^{2p} \to \mathbb{R}_0^+$ in the similarity and the regularization term. For example, in the similarity term, such a function calculates the distance between two image patches $u(\mathcal{P}_i)$ and $f(\mathcal{P}_j)$ of the initial image (in the case of the regularization term we change this by $u(\mathcal{P}_j)$). Where the index set $\mathcal{P}_i$ denotes the image patch as neighborhood of the pixel $i$. Both patches are assumed to have the same size $p \in \mathbb{N}$ and the same shape.
The standard distance function for this purpose is the weighted $L^2$ norm used in NL means algorithm, defined by:

$$|d(u(\mathcal{P}_i), f(\mathcal{P}_j))|^2 = \sum_p g_\sigma(p)(u_{i+p} - f_{j+p})^2$$

where $g_\sigma(p) := \exp(-p^2/(2\sigma^2))$.
With this definitions, the GNDS model is defined by:

$$\begin{aligned} J^{GNDS}(u) &= (1-\alpha)F^G_{sim}(u) + \alpha F^G_{reg}(u) \\ &= (1-\alpha)\sum_{i,j\in J} \Psi_s(|d_s(u(\mathcal{P}_i), f(\mathcal{P}_j))|^2) \cdot w_s(|x_i - x_j|^2) \\ &+ \alpha \sum_{i,j\in J} \Psi_r(|d_r(u(\mathcal{P}_i), u(\mathcal{P}_j))|^2) \cdot w_r(|x_i - x_j|^2) \end{aligned}$$

The minimization strategy is the same as for the NDS model, the idea is to find a fixed point form for this new functional. The minimizer of $J^{GNDS}$ necessarily satisfies:

$$\frac{\partial J^{GNDS}}{\partial u_i} = (1-\alpha)\frac{\partial F^G_{sim}}{\partial u_i} + \alpha \frac{\partial F^G_{reg}}{\partial u_i} = 0 \quad \forall i \in J$$

where:

$$\frac{\partial F^G_{sim}}{\partial u_i} = 2\sum_{j\in J} g_\sigma * \Psi_s'(d^2_{s;k-\cdot,j-\cdot})(0)(u_k - f_j) \cdot w_s(|x_i - x_j|^2)$$

$$\frac{\partial F^G_{reg}}{\partial u_i} = 4\sum_{j\in J} g_\sigma * \Psi_r'(d^2_{r;k-\cdot,j-\cdot})(0)(u_k - u_j) \cdot w_r(|x_i - x_j|^2)$$

Now if we consider the following notation:

$$h_{i,j}^{GS} := g_\sigma * \Psi_s'(|d_s(u(\mathcal{P}_i), f(\mathcal{P}_j))|^2)(0)$$

$$h_{i,j}^{GR} := 2g_\sigma * \Psi_r'(|d_r(u(\mathcal{P}_i), u(\mathcal{P}_j))|^2)(0)$$

we have the following fixed point equation for the GNDS model:

$$u_i = \frac{(1-\alpha)\sum_{j\in J} h_{i,j}^{GS} w_{i,j}^{s,\overline{w}} f_j + \alpha \sum_{j\in J} h_{i,j}^{GR} w_{i,j}^{r,\overline{w}} u_j}{(1-\alpha)\sum_{j\in J} h_{i,j}^{GS} w_{i,j}^{s,\overline{w}} + \alpha \sum_{j\in J} h_{i,j}^{GR} w_{i,j}^{r,\overline{w}}} \tag{F.12}$$

In the same way as in the NDS fixed point equation we can deduce a min-max principle and the existence of a fixed point, also, a minimizer can be found via gradient descent method.
The most important difference between this model and the NDS model is the use of patch distances instead of single difference between pairs of pixels, this induces a patch similarity measure, with is more powerful and versatile for our purposes.

### F.2.2 Double Weighting: Generalization of the model and Particular cases

Let us focus only on the similarity term (the same applies analogously for the regularization term) and expand the terms of the fixed point equation:

$$u_i = \frac{1}{M_{i,j}} \sum_{j,p\in J} g_\sigma(p) \cdot \Psi' \left( \sum_{q\in J} g_\sigma(q) |u_{i+p+q} - f_{j+p+q}|^2 \right) w_{i,j} f_j \tag{F.13}$$

where $M_{i,j}$ denotes the normalization term.
Notice that we have the Gaussian term $g_\sigma$ two times in the formula, once during the patch similarity calculation (sum over $q$), we will call this **inner weighting** of patch pixels, and then it appears in the sum over $p$, we will call this the **outer weighting** which is applied when summing the results of the function $\Psi'$.
Is interesting to see that what this formula means is: For pixels $u_i$ and $f_j$ we first calculate the patch distances of all patches at positions $i+p$ and $j+p$ taken with the offset $p$ around $u_i$ and $f_j$ respectively. Then, average this patches distances (transformed by the nonlinearlity $\Psi'$) using the outer weight $g_\sigma$. Thus, the pixel $f_j$ will contribute to the result $u_i$ with a high weight only if the patches around $u_i$ and $f_j$ are similar AND the patches around $u_{i+p}$ and $f_{j+p}$ are similar.

A natural question arises when we look at the expression (F.13), we have the two weights (inner and outer) with the same parameter ($\sigma$) but with different role in the expression, the parameter is the same because this becomes from the derivation of the energy functional, but, since the role between them is different, is natural to ask what happens if one change the parameter of the weights. If we consider that the weights have different parameters we have:

$$u_i = \frac{1}{M_{i,j}} \sum_{j,p\in J} g_\rho(p) \cdot \Psi' \left( \sum_{q\in J} g_\sigma(q) |u_{i+p+q} - f_{j+p+q}|^2 \right) w_{i,j} f_j \tag{F.14}$$

This simple observation makes a more generalized version of the model (obviously this is not derived from the original functional, but is the generalized version of this one since if we take $\rho = \sigma$ we are in the original model again), and establishes a family of filters from the GNDS model. This is the most general version of the model, and the following table shows the particular cases that reduces the model to a known other models. (by adjusting the parameters AND the involved functions).

Table F.1: Filtering Methods Belonging to GNDS Family

| Regularization parameter | Patch size | Integration Scale | Method |
|---|---|---|---|
| [0.5ex] $0 \leq \alpha \leq 1$ | $\sigma > 0$ | $\rho = \sigma$ | Original GNDS |
| $\alpha \in \{0, 1\}$ | $\sigma > 0$ | $\rho \to 0$ | NL-means (original) |
| $\alpha \in \{0, 1\}$ | $\sigma \to 0$ | $\rho > 0$ | NL-means (anisotropic) |
| $0 \leq \alpha \leq 1$ | $\rho \to 0$ | $\sigma \to 0$ | Original NDS |
| [1ex] | | | |

A remark on the complexity of this scheme: Given an image of $N$ pixels, a square search window $w$ with $s^2$ pixels and a circular patch of radius $r$, the computational complexity of the generalized filter family (i.e. the fixed point scheme in the most general version, with two different weights: $g_\rho, g_\sigma$ is: $O(N \times s^2 \times r_\rho^2 \times r_\sigma^2)$ And finally, a remark in the case of multichannel images: The extension of GNDS model to multichannel images is natural, we just need to consider a patch distance based in a vector way, computing the norm between vectors $\vec{u}$ and $\vec{f}$ (these are vectors with $d$ components, where each component correspond to a different channel), i.e.:

$$|d(\vec{u}(\mathcal{P}_i), \vec{f}(\mathcal{P}_j))|^2 = \sum_p g_\sigma(p) ||\vec{u}_{i+p} - \vec{f}_{j+p}||_2^2$$

using the euclidean norm. The optimality conditions are the same as the original model, just considering it on each channel.

## F.3 Numerical Examples


Figure F.1: Comparison to state-of-the-art methods. Top Row: Test images degraded by Gaussian noise with standard deviation 20. 2nd Row: Restored images by Dabov et al. (2007). 3rd Row: Restored images by Kervrann and Boulanger (2008). Bottom Row: Restored images by the proposed GNDS model.

# G. Nonlocal Mumford-Shah Regularizers for Color Image Restoration [14]

The main goal of this paper is to develop several functionals based on approximations of the original Mumford-Shah functional with nonlocal characteristics incorporated, this nonlocal characteristics are based on the work of Buades et al. and Gilboa et al.
Is important to incorporate the nonlocal characteristics in new models because this perform better than local methods in image denoising and restoration when the image have textures (for example, local methods usually consider textures as noise, and then, in denoising tasks this algorithms just remove the textures).
The authors presents non local extensions for the widely known approximations for the Mumford-Shah functional, this approximations are due to Ambrosio-Tortorelli and Shah with the primary objective of better restoration of fine structures and textures. Some applications (and algorithms) are presented for the following image tasks:

1. Color Image Deblurring and Denoising
2. Color Image Inpainting
3. Color Image Super-Resolution
4. Color Filter Array Demosaicing

## G.1 Introduction - Background

First of all, we need to recall some basic results and concepts about image regularization methods.

### G.1.1 Local Regularizers

The basic Mumford-Shah regularizating functional is used commonly in segmentation and restoration algorithms, it is given by the following formulation:
Given $u : \Omega \to \mathbb{R}$ and $K$ its edge set, the $MSH^1$ regularizer is:

$$J^{MSH^1}(u, K) = \beta \int_{\Omega \setminus K} |\nabla u|^2 dx + \alpha \int_K d\mathcal{H}^1$$

where $|\nabla u| = \sqrt{u_{x_1}^2 + u_{x_2}^2}$, $x = (x_1, x_2)$, $\mathcal{H}^1$ is the 1-D Hausdorff measure and $\Omega \subset \mathbb{R}^2$ is the open image domain. Notice that the first term enforces $u$ to be smooth everywhere except on the edge set $K$, and the second term enforce to minimize the total length of edges.
In general is very hard to minimize (in practice) this functional, due to its non convex behaviour. A way to solve this problem is to consider another functionals (with better structure) which approximate this one in some sense that asserts that the minimum points of this new functionals approximates the minimum points of the original one.
Ambrosio and Tortorelli approximated the Mumford-Shah functional by considering a sequence of more regular functionals denoted by $J_\varepsilon$ which converges to $J^{MSH^1}$ in the sense of $\Gamma$-convergence. The idea of this functional is to approximate the edge set $K$ by a smooth function $v$, the approximation is given by:

$$J_\varepsilon^{MSH^1}(u, v) = \beta \int_\Omega v^2 |\nabla u|^2 dx + \alpha \int_\Omega \left( \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx$$

where $0 \leq v(x) \leq 1$ represents the edges: $v(x) \sim 0$ if $x \in K = J_u$ (jump set of $u$), $v(x) \sim 1$ otherwise; $\varepsilon$ is a small positive constant, $\alpha$, $\beta$ positive weights. If we add a fidelity term to this functional we have that a minimizer $u = u_\varepsilon$ of $J_\varepsilon^{MSH^1}$ approaches a minimizer of $J^{MSH^1}$ as $\varepsilon \to 0$

Another approach is given by Shah using the total variation regularization proposed in image restoriation

mainly by Rudin, Osher and Fatemi, this is very useful due to its benefits of preserving edges and convexity. The total variation regularization is defined in the following way: given a locally integrable function $u$ define:

$$J^{TV}(u) = \sup\{\int_\Omega u\nabla\cdot\phi dx : \phi \in C_c^1(\Omega,\mathbb{R}), ||\phi||_{L^\infty(\Omega)\leq 1}\}$$

which coincides with $\int_\Omega |\nabla u| dx$ when $u \in W^{1,1}(\Omega)$.
Based on this, Shah proposed a modified version of Ambrosio-Tortorelli approximation by replacing the term $|\nabla u|^2$ by $|\nabla u|$ in the first term, then, the Shah approximation for the Mumford-Shah functional is given by:

$$J_\varepsilon^{MSTV}(u,v) = \beta\int_\Omega v^2|\nabla u|dx + \alpha\int_\Omega\left(\varepsilon|\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon}\right)dx$$

This functional $\Gamma$-converges to the $J^{MSTV}$ functional given by:

$$J^{MSTV} = \beta\int_{\Omega\setminus K}|\nabla u|dx + \alpha\int_K \frac{|u^+-u^-|}{1+|u^+-u^-|}d\mathcal{H}^1 + |D_c u|(\Omega)$$

Where $u^+, u^-$ denotes the values of $u$ 'at each side' of $K$ and $D_c u$ is the Cantor part of $Du$. This last functional is very similar with the total variation of $u \in BV(\Omega)$ that can be written for $K = J_u$ as:

$$J^{TV} = \beta\int_{\Omega\setminus K}|\nabla u|dx + \alpha\int_K |u^+-u^-|d\mathcal{H}^1 + |D_c u|(\Omega)$$

The only difference is that the MSTV regularizer does not penalize the jump part as much as the TV regularizer does.

This functionals are considered only for monochromatic images, but is naturally extended to color images by Blomgren and Chan, which propose a color TV regularization by coupling the channels, i.e. considering:

$$J^{TV} = \int_\Omega ||\nabla u||dx = \int_\Omega \sqrt{|\nabla u^R|^2 + |\nabla u^G|^2 + |\nabla u^B|^2}dx$$

Bar et al. extend this idea for the Mumford-Shah approximations for color images, by replacing $|\nabla u|$ by $||\nabla u||$ in $J_\varepsilon^{MSH}$ and $J_\varepsilon^{MSTV}$. Notice that the scalar-valued edge map $v$ is common for the three channels and provides the necessary coupling between colors.

### G.1.2 Nonlocal Methods

As we seen before in review of the paper of Buades et al. the importance of nonlocal methods is based on their well adaptation to texture denoising in contrast to standard local methods. Recall that the basic idea is to extend the concept of neighborhood filters which replace the value of a pixel with an average of its spatial neighbors, the nonlocal filters extend this concept to the one of patch-similarity, i.e. we will replace the value of a pixel for an averaging of pixels which have similar patch values. (and then, the spatial restriction is relaxed). The classical filter for this task is the NL-means filter due to Buades et al.:

$$NL(f(x)) = \frac{1}{C(x)}\int_\Omega \exp\left(-\frac{d_a(f(x),f(y))}{h^2}\right)f(y)dy$$

$$d_a(f(x),f(y)) = \int_{\mathbb{R}^2} g_a(t)||f(x+t)-f(y+t)||^2 dt$$

where $d_a$ is the patch distance, $f$ is the image to be filtered and $g_a$ is a Gaussian kernel with standard deviation $a$ which determines the patch size. For more information of this method you can see the chapter 3 of this review.

### G.1.3 Nonlocal Regularizers

The idea of this regularizers is to see the nonlocal filtering as a quadratic regularization based upon a nonlocal graph (a graph with weights). The most important contributions on this field are given by Gilboa

and Osher.
We will need some operators from this theory, the so called "non local differential operators over graphs" which are proposed by Gilboa and Osher.
Let $u : \Omega \to \mathbb{R}$ and $w : \Omega \times \Omega \to \mathbb{R}$ a non negative and symmetric weight function. We define the non local gradient vector $\nabla_w u : \Omega \times \Omega \to \mathbb{R}$ as:

$$(\nabla_w u)(x, y) = (u(y) - u(x))\sqrt{w(x, y)}$$

And the norm of the nonlocal gradient of $u$ is defined by:

$$|\nabla_w u|(x) := \sqrt{\int_\Omega (u(y) - u(x))^2 w(x, y) dy}$$

We also define the non local divergence of the vector $\vec{v} : \Omega \times \Omega \to \mathbb{R}$ by:

$$(div_w \vec{v})(x) := \int_\Omega (v(x, y) - v(y, x))\sqrt{w(x, y)} dy$$

Inspired in this operators, Gilboa and Osher propose the following general form for nonlocal regularizating functionals:

$$J(u) = \int_\Omega \phi(|\nabla_w u|^2) dx$$

where $s \mapsto \phi(s)$ is positive, increasing and convex in $\sqrt{s}$, and $\phi(0) = 0$.
If $\phi(s) = \sqrt{s}$, they propose the NL/TV (NonLocal Total Variation) regularizer:

$$J^{NL/TV}(u) = \int_\Omega |\nabla_w u| dx = \int_\Omega \sqrt{\int_\Omega (u(y) - u(x))^2 w(x, y) dy} dx$$

Which coincides, in the 2D local case to $J^{TV}(u) = \int_\Omega |\nabla u| dx$

## G.2 Proposed Nonlocal Mumford-Shah Regularizers

Based on the above, the authors propose nonlocal versions of the approximating functionals of Ambrosio-Tortorelli and Shah for the Mumford-Shah functional, and considering an appropiate fidelity term they develop functionals for the following image tasks:

1. Deblurring-Denoising
2. Inpainting
3. Super-Resolution
4. Demosaicing

Is important to recall that they also incorporate the vector case (i.e. color images) in their formulation, in the way as we seen above.

Then, the general model proposed by the authors is:

$$J^{NL/MS}(u, v) := \underbrace{\beta \int_\Omega v^2 \phi(||\nabla_w u||^2) dx}_{F^{NL/MS}_{reg}} + \underbrace{\alpha \int_\Omega \left( \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx}_{F_{AT}} = F^{NL/MS}_{reg}(u, v) + F_{AT}(v)$$

where $u : \Omega \to \mathbb{R}^3$, $v : \Omega \to [0, 1]$ and $\phi(s) = s$ or $\phi(s) = \sqrt{s}$ (the first choice correspond to $NL/MSH^1$ and the second to $MS/TV$) i.e.:

$$J^{NL/MSH^1}(u, v) := \underbrace{\beta \int_\Omega v^2 ||\nabla_w u||^2 dx}_{F^{NL/MS}_{regAT}} + \alpha \int_\Omega \left( \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx = F^{NL/MS}_{regAT}(u, v) + F_{AT}(v)$$

$$J^{NL/MSTV}(u,v) := \underbrace{\beta \int_\Omega v^2 ||\nabla_w u|| dx}_{F^{NL/MS}_{regS}} + \alpha \int_\Omega \left( \varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon} \right) dx = F^{NL/MS}_{regS}(u,v) + F_{AT}(v)$$

and recall that:

$$||\nabla_w u||(x) = \sqrt{\sum_{i=R,G,B} |\nabla_w u^i|^2(x)} = \sqrt{\sum_{i=R,G,B} \int_\Omega (u^i(x) - u^i(y))^2 w(x,y) dy}$$

as we said before, adding a fidelity term to this functionals we will be able to perform a specific restoration task, we will discuss this in the next section.
As an additional remark, in the practice the weight function that will be used is the classic NL-means weight (given an image $q$):

$$w(x,y) = \exp\left( -\frac{d_a(q(x), q(y))}{h^2} \right)$$

and we use search windows $S(x) = \{y \ / \ |x-y| \leq r\}$

## G.3 Image Restoration Tasks with NL/MS Regularizers

As we said before, choosing an appropiate fidelity term we will construct functionals for several tasks of image restoration, also sometimes we will need to do a pre-process to image in order to construct the weight function appropriately.

### G.3.1 Deblurring-Denoising

The standard degradation model for deblurring and denoising is:

$$f = k * u + n \quad (f^i = k * u^i + n^i, i = R, G, B)$$

where $k$ is the associated (known) blurring kernel and $n$ is an additive noise (which can be Gaussian, Laplacian or impulse noise (in the last case a preprocessing is needed, because the model is no longer valid in this case)). Then, in the case of Gaussian noise, the $L^2$ fidelity term is commonly used (this is led by maximum likelihood estimation)

$$\Phi(f - k * u) = \int_\Omega \sum_i |f^i - k * u^i|^2 dx$$

For the impulse noise (or Laplacian) the $L^1$ fidelity term is more appropriate (due to its better results on removing outlier effects), then, considering the case of independent channels noise we have:

$$\Phi(f - k * u) = \int_\Omega \sum_i |f^i - k * u^i| dx$$

Then, the proposed types of total energy for color image deblurring-denoising are:

$$J^{Gau}(u,v) = \underbrace{\frac{1}{2} \int_\Omega \sum_i |f^i - k * u^i|^2 dx}_{F^{Gau}_{sim}} + J^{NL/MS}(u,v) = F^{Gau}_{sim}(u) + J^{NL/MS}(u,v)$$

$$J^{Imp}(u,v) = \underbrace{\int_\Omega \sum_i |f^i - k * u^i| dx}_{F^{Imp}_{sim}} + J^{NL/MS}(u,v) = F^{Imp}_{sim}(u) + J^{NL/MS}(u,v)$$

The proprocess needed for the construction of a image $\overline{g}$ needed for the construction of weight function $w$ is given by:

1. Initialize: $r^i_0 = 0$, $g^i_0 = 0$, $i = R, G, B$

2. do (iterate $m = 0, 1, 2, \ldots$)

   - $g_{m+1} = median(f + r_m, [s\ s])$
   - $r_{m+1} = r_m + f - k * g_{m+1}$
     while $\sum_i ||f^i - k * g_m^i||_1 > \sum_i ||f^i - k * g_{m+1}^i||_1$

### G.3.2 Inpainting

In this problem we have an observed image $f$ given by:

$$f = u \text{ in } \Omega \setminus D$$

Where $D = D^0$ is the region where the input data $u$ is lost. Inspired in the work of Chan and Shen [4], the authors propose the following energy functional for inpainting:

$$J^{Inp}(u,v) = \underbrace{\frac{\lambda}{2}\int_\Omega \mathbf{1}_{\Omega\setminus D}(x)\sum_i |f^i - u^i|^2 dx}_{F_{sim}^{Inp}} + J^{NL/MS}(u,v) = F_{sim}^{Inp}(u) + J^{NL/MS}(u,v)$$

Also, the weights $w$ are updated only in the damaged region $D$ in the $m$-th iteration for $u$ using the modified patch distance:

$$d_a^R(u(x), u(y)) = \int_{\mathbb{R}^2} \mathbf{1}_{\Omega\setminus R}(x+t) g_a(t) ||u(x+t) - u(y+t)||^2 dt$$

where $R \subset D$ is a unrecovered region (still missing region).
Then, the missing region $D = D^0$ can be recovered by the following iterative algorithm, which produces regiones $D^i$, $i = 0, 1, 2, \ldots$ con $D^n \subset \ldots \subset D^0$
**Algorithm**

1. Compute weights $w$ for $x \in \Omega$ such that $P(x) \cap (\Omega \setminus D^0) \neq \emptyset$ using $d_a^{D^0}(u^0(x), u^0(y))$ with $u^0 = f$ in $\Omega \setminus D^0$ and $\infty$ in $D^0$, a patch $P(x)$ centered at $x$, and $y \in S(x) \cap (\Omega \setminus D^0)$.

2. Iterate $n = 1, 2, \ldots$ to obtain a minimizer $(u, v)$ starting with $u = u^0$:

   (a) For fixed $u$, update $v$ in $\Omega$ to obtain $v^n$

   (b) For fixed $v$, update $u$ in $\Omega$ to obtain $u^n$ with a recovered region $\Omega \setminus D^n$: at every $m$-th iteration, update weights $w$ only in $x \in D^0$ subject to $P(x) \cap (\Omega \setminus D^{m,n}) \neq \emptyset$ with

   $$d_a^{D^{m,n}}(u(x), u(y))$$

   where $y \in S(x) \cap (\Omega \setminus D^{n,m})$, $D^{n,m}$ is and un-recovered region in $D^0$ until $m$-th iteration with $D^n = D^{n,n} \subset \cdots \subset D^{n,2m} \subset D^{n,m}$

### G.3.3 Super-Resolution

The idea of this process is to recover a high resolution image from a filtered and down-sampled image (This process is usual in a sequence of images in video). Our observed data is

$$f^i = D_k(h * u^i) \quad i = R, G, B$$

where $h$ is a low-pass filter, $D_k : \mathbb{R}^{n\times n} \to \mathbb{R}^{p\times p}$, where $p = [n/k^2]$ is the down-sampling operator with a factor $k$ on each axis. We want to recover $u \in (\mathbb{R}^{n\times n})^3$, this could be done minimizing:

$$J^{Sup}(u,v) = \underbrace{\frac{1}{2}\int_\Omega \sum_i |f^i - D_k(h * u^i)|^2 dx}_{F_{sim}^{Sup}} + J^{NL/MS}(u,v) = F_{sim}^{Sup}(u) + J^{NL/MS}(u,v)$$

In addition, for the computation of the weights $w$ we need a super-resolved image $\overline{g} \in (\mathbb{R}^{n\times n})^3$ obtained by bicubic interpolation of $f$. Notice that $\overline{g}$ is only used for the computation.

### G.3.4 Demosaicing

The idea is to reconstruct a full color image from the incomplete color samples output from an image sensor overlaid a color filter array (CFA). A CFA is a mosaic of color filters in front of the image sensor (we use here the Bayer filter [2]). Since each pixel of the sensor is behind a color filter, the output is and array of pixel values, each indicating a raw intensity of one of the three filter colors. In this variational framework, we consider the observed data $f$ as:

$$f^i = H^i \cdot u^i \quad i = R, G, B$$

where $\cdot$ is the pointwise product, and $H^i$ is the down-sampling operator; $H^G$ has alternating 1 and 0 values for odd rows and alternating 0 and 1 values for even rows, $H^R$ has alternating 0 and 1 values for odd rows and 1 and only 0 values for even rows, $H^B$ has only 0 values for odd rows and alternating 1 and 0 values for even rows. The authors propose the following minimization problem to recover a full color image $u$:

$$J^{Demo}(u,v) = \underbrace{\frac{1}{2}\int_\Omega \sum_i |f^i - H^i \cdot u^i|^2 dx}_{F^{Demo}_{sim}} + J^{NL/MS}(u,v) = F^{Demo}_{sim}(u) + J^{NL/MS}(u,v)$$

Moreover, for the computation of the weight function $w$, the authors use the interpolated image obtained by applying Hamilton-Adams algorithm [8] for the green channel and bilineal interpolation for R-G and B-G.

### G.3.5 Numerical Discretizations

For minimize the proposed functionals in $u$ and $v$ we will consider the Euler Lagrange equations, notice that:

$$\frac{\partial J^{Gau,Im,Inp,Sup,Demo}}{\partial v} = 2\beta v\phi(||\nabla_w u||^2) - 2\varepsilon\alpha\Delta v + \alpha\left(\frac{v-1}{2\varepsilon}\right) = 0$$

$$\frac{\partial J^{Gau}}{\partial u} = \tilde{k} * (k * u - f) + L^{NL/MS}u = 0$$

$$\frac{\partial J^{Im}}{\partial u} = \tilde{k} * sign(k * u - f) + L^{NL/MS}u = 0$$

$$\frac{\partial J^{Inp}}{\partial u} = \mathbf{1}_{\Omega\setminus D}(u - f) + L^{NL/MS}u = 0$$

$$\frac{\partial J^{Sup}}{\partial u} = \tilde{h} * (D_k^T(D_k(h * u) - f)) + L^{NL/MS}u = 0$$

$$\frac{\partial J^{Demo}}{\partial u} = H \cdot (H \cdot u - f) + L^{NL/MS}u = 0$$

where $\tilde{k}(x) = k(-x)$, $\tilde{h}(x) = h(-x)$, $D_k^T$ is the transpose of $D_k$ (i.e. the up-sampling operator) and:

$$L^{NL/MS}u = -2\int_\Omega \{(u(y) - u(x))w(x,y) \cdot [(v^2(y)\phi'(||\nabla_w u||^2(y)) + v^2(x)\phi'(||\nabla_w u||^2(x))]\}dy$$

To solve two Euler-Lagrange equations simultaneously the alternate minimization (AM) approach (which we seen in the second chapter) is applied. Since the energy functionals are not convex in $(u, v)$ we may compute only a local minimizer. The proposed iterative algorithm is:

- Initialization: $u^0 = f, v^0 = 1$
- Iterate $n = 0, 1, 2, \ldots$ until $(||u^{n+1} - u^n||_2 < \nu||u^n||_2)$
  1. Solve the equation for $v^{n+1}$ using Gauss-Seidel Scheme:
  $$(2\beta\phi(||\nabla_w u^n||^2) + \alpha/(2\varepsilon) - 2\varepsilon\alpha\Delta)v^{n+1} = \alpha/(2\varepsilon)$$
  2. Set $u^{n+1,0} = u^n$ and solve for $u^{n+1}$ by iterating on $l$:
  $$u^{n+1,l+1} = u^{n+1,l} - dt \cdot \frac{\partial J}{\partial u}(u^{n+1,l}, v^{n+1})$$

where $\nu$ is a small positive constant.
The discretization of the functions involved is: Let $u_k^i$ the value of a pixel $k$ in the image ($1 \leq k \leq N$) with channel $i = R, G, B$ i.e. the discretization of $u^i(x)$, let $p_{k,l}^i$ be the discretized version of $p^i(x, y)$ with $x, y \in \Omega$). Let $w_{k,l}$ the sparsely discrete version of the weight function $w$, this is constructed by the same algorithm described by Gilboa and Osher in chapter 4. We follow the same notation for neighborhood sets as in Gilboa and Osher paper (and in Buades paper): $l \in \mathcal{N}_k\{l : w_{k,l} > 0\}$. Then, we define $\nabla_{wd}$ and $div_{wd}$, the discretizations of $\nabla_w$ and $div_w$ as:

$$\nabla_{wd}(u_k^i) := (u_l^i - u_k^i)\sqrt{w_{k,l}} \quad l \in \mathcal{N}_k$$

$$div_{wd}(p_{k,l}^i) := \sum_{l \in \mathcal{N}_k} (p_{k,l}^i - p_{l,k}^i)\sqrt{w_{k,l}}$$

Then we have:

$$|p^i|_k = \sqrt{\sum_l (p_{k,l}^i)^2} \text{ the magnitude of } p_{k,l}^i \text{at } k$$

and then, the discretization of $||\nabla_w u||^2$:

$$||\nabla_{wd} u||^2 = \sum_{i \in \{R,G,B\}} |\nabla_{wd} u_k^i|^2 = \sum_i \sum_l (u_l^i - u_k^i)^2 w_{k,l}$$

For details on the construction of $w$ we refer to the algorithm given in Gilboa and Osher paper, on chapter 4.

## G.4 Numerical Examples

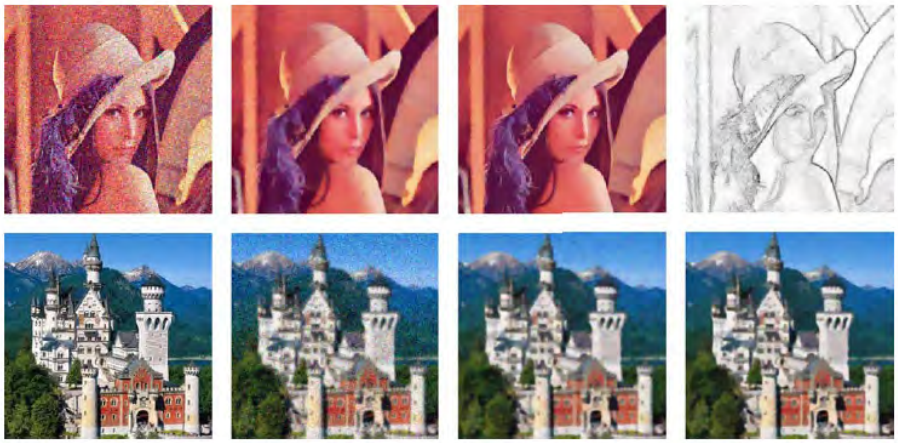


Figure G.1: DENOISING OR DEBLURRING IN THE PRESENCE OF GAUSSIAN NOISE. Top: (first) noisy image ($f = u + n$) with noise variance ($\sigma = 0.02$), recovered images using (second) MSTV and (third and fourth) NL/MSTV with edge set $v$. Bottom: (first) original, (second) blurry-noisy data ($f = k * u + n$) and noise variance ($\sigma = 0.02$) recovered images using (third) MSTV and (fourth) NL/MSTV.

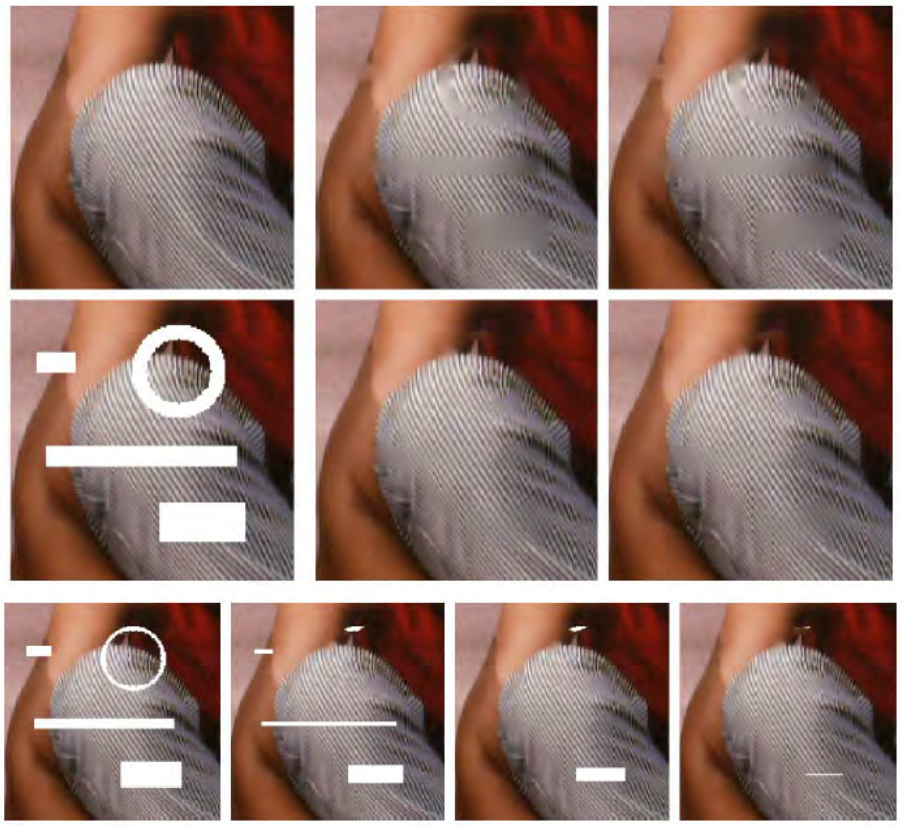


Figure G.2: INPAINTING of $150 \times 150$) size image. First column: (top row) original, (middle) data $f$ . Second and third columns: recovered images using (top) $MSH^1$, MSTV, (middle) $NL/MSH^1$, NL/MSTV, with $51 \times 51$ search window and $9 \times 9$ patch. Bottom row: process of inpainting with $NL/MSH^1$ in 100th, 200th, and 350th iterations.


Figure G.3: SUPER-RESOLUTION of a still image. Top: original image of size $272 \times 272$, blurred image $h * u$ with out of focus blur kernel $h$ with radius $r = 3$, down-sampled data $f = D_k(h * u)$ of size $68 \times 68$ with $k = 4$, preprocessed (up-sampled) image $\bar{g}$ using bicubic interpolation. Bottom: recovered images using (left) MSTV, (right) NL/MSTV.

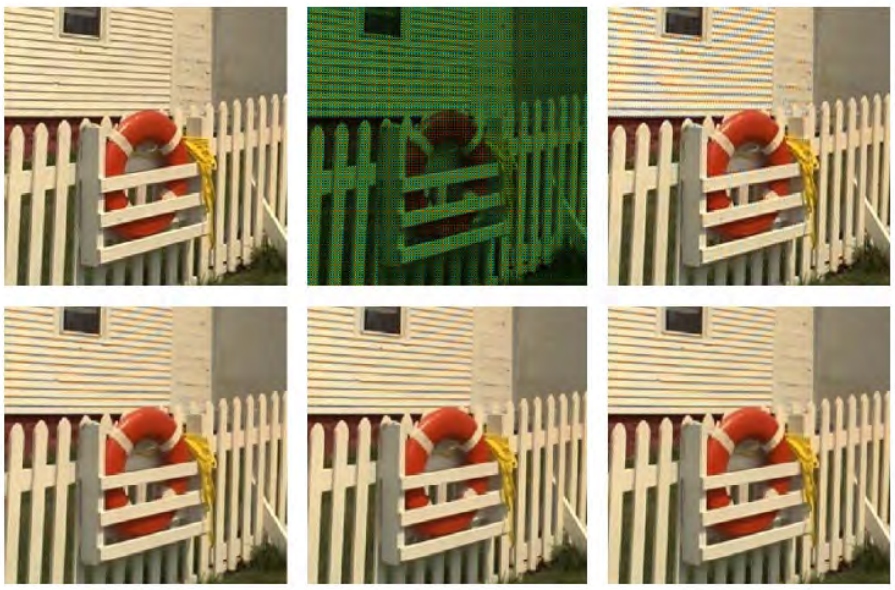


Figure G.4: DEMOSAICING using NL/MSTV with iterative algorithm. Top: original image, data $f$ , interpolated image using Hamilton-Adams for green and bilinear interpolation for $R-G$ and $B-G$. Bottom: demosaiced images with decreasing sequence of $h = \{16, 8, 4\}$

## G.5 Summary and Conclusions

After several testings to check the performance of the methods proposed by the authors and other common methods (results can be seen on the Numerical examples, and more information can be seen on the original paper) the authors conclude:

1. $NL/MSH^1$ and $NL/MSTV$ outperform the local one methods on all the applications.
2. For deblurring-denosing $NL/TV$ and $NL/MSTV$ provide the best recovered images.
3. $NL/MSTV$ brings the best results in super-resolution.
4. $NL/MSH^1$ and $NL/MSTV$ provide superior results to local models by better recovering textures and large missing regions.
5. $NL/MSTV$ reconstructs images best in demosaicing.

To sum up, $NL/MSTV$ produces the best results in general. The next step of the project is to consider the numerical implementation of these models.

# H. Index Measurements

## H.1 Camouflage measure models

Camouflage is used to prevent detection of the individual combat soldier and his equipment by both visual observation and other sensors, and thereby to increase the survivability of all soldiers and the mission effectiveness. Camouflage achieves its purposes by blending the soldier with the background, and disrupting the cues for perceiving the soldier as a single object (Killian & Hepfinger, 1992).

Since the mid 1990's, there has been a need to improve the understanding of how low contrast targets, such as camouflage, are perceived by observers and acquired in complex surrounds. Although there has been a long history of attempting to predict the probability of psychophysical task (Detection, Recognition, Identification) for target acquisition by means of electro-optical systems. Most of the models developed for predicting psychophysical task performance of a given target have been expressed in terms of the model's target signature metric and an ensemble average of observers. These models are not highly accurate in predicting the psychophysical performance of a unique target in a unique surround due to the absence of a sufficiently adequate description of the target signature and natural surround and the interactions of the target signature with the local surround (Desmond, 2004). Figure 3 shows target acquisition models that were developed for military purposes, with the primary emphasis on sensor performance (Desmond, 2004)

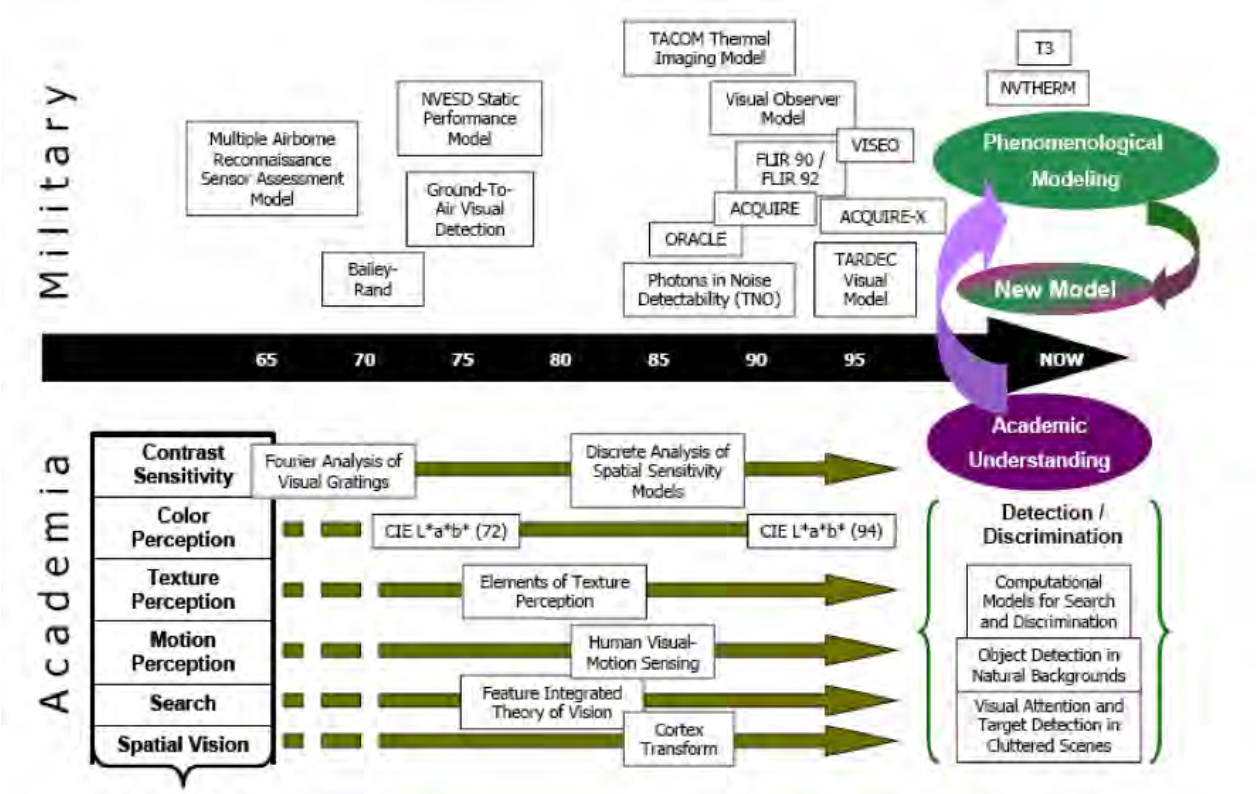



Figure H.1: Military and Academia Progress Toward Modeling Observer Target Acquisition Performance (Desmond, 2004)

### H.1.1 S-CIELAB Metric

The original digital image is decomposed into a color space based on opponent-colors, and each color separation is convolved with a Gaussian shaped, two-dimensional separable spatial filter to simulate the blurring of the human visual system. Zhang and Wandell pointed out those psychophysical experiments suggest the human visual system's representation of simple colored patterns is *pattern-color separable*. Thus,

the image can be pre-processed with separate color and spatial filters which correspond to the human visual system prior to transformation to CIELAB color space. The delta E color difference is calculated between the original and its copy to quantitatively describe any reproduction error. Spatially correlated color error maps or histograms of color error can be generated using the 1978 CIELAB color space.

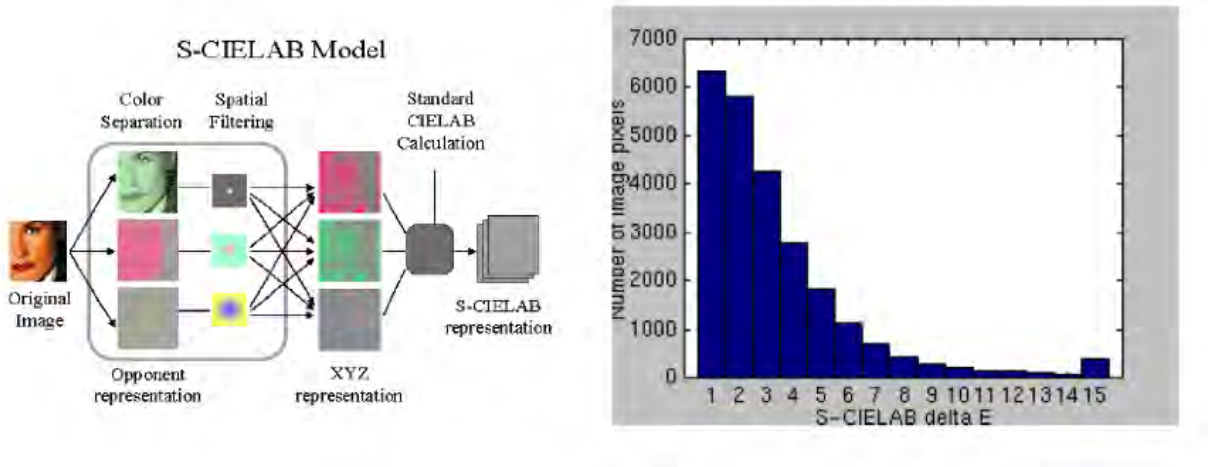



Figure H.2: The S-CIELAB Color Methodology

### H.1.2 The Sarnoff Visual Discrimination Model

The Sarnoff model, requires two images as input for analysis and produces an output in terms of Just Noticeable Difference (JND).

An image pair with one JND has a 75% probability of discrimination, two JNDs have has a 93% probability of discrimination, and three JNDs have a 98% probability of discrimination. this model quantifies an observer psychophysical discrimination task, i.e., measuring the difference of comparing two similar images, rather than measuring an observer military detection task

## H.2 Patterns evaluation system

### H.2.1 MACE (mobile army camouflage evaluation Killian & Hepfinger (1992))

The MACE system consists of hardware and software for acquiring, reducing and analyzing images of camouflage-in-background scenes. Analysis with MACE system includes transformation from a set of monochromatic images to standard color description coordinates, and direct comparisons of first and second order statistical measures between the camouflaged soldier and the local background.

### H.2.2 Data reduction

Is a transformation from a set of monochromatic images to standard color description coordinates. We note that MACE system uses the standard system CIE (Commision Internationale de L'Eclairage) for color matching built around Grassman's tristimulus laws for color mixing:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = k \int_0^\infty R(\lambda) S(\lambda) \begin{pmatrix} \varphi_x(\lambda) \\ \varphi_y(\lambda) \\ \varphi_z(\lambda) \end{pmatrix} d\lambda,$$

where:

- $R(\lambda)$ is the surface reflectance of an object under observation.
- $S(\lambda)$ is the spectral radiance of the illuminant.

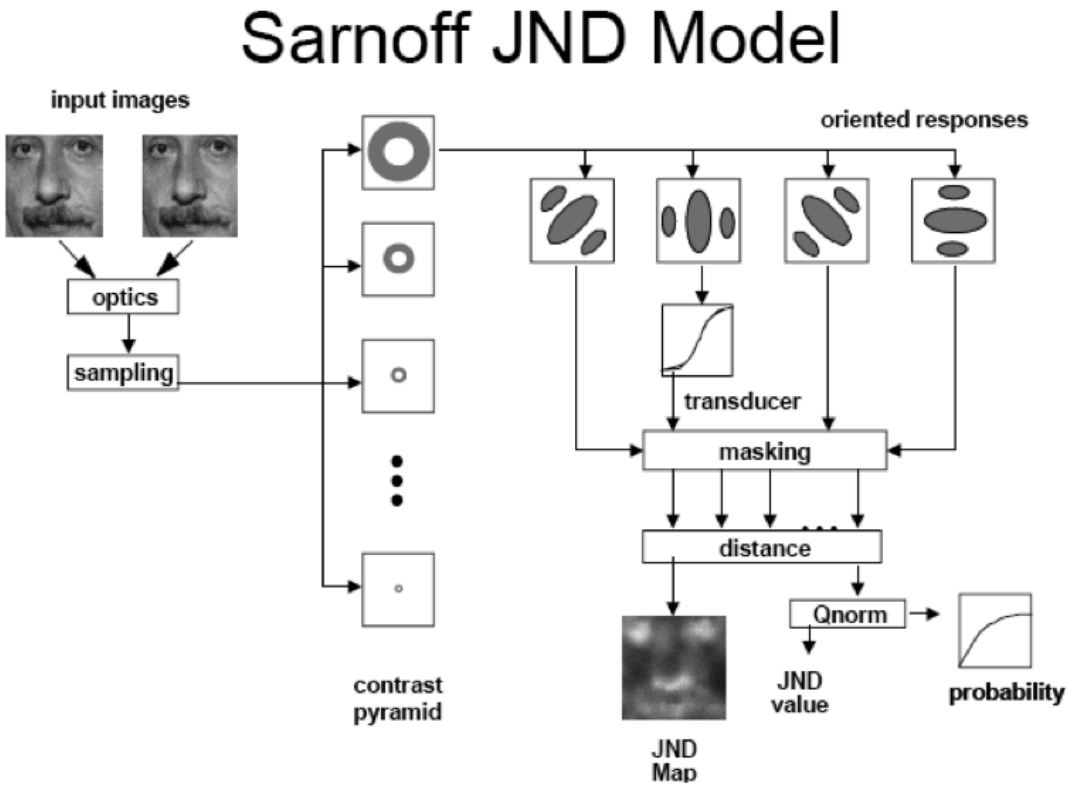



Figure H.3: The Sarnoff JND Model

- ?($\lambda$) are the tristimulus response functions for standar observer.
- $k$ is a normalization constant.

The data reduction process consists of three steps: frame registration, reflectance extraction, and finally the transformation, using the above equation.

## H.3 Measures of similarity

Killian & Hepfinger ( 1992) implemented a set of object/background similarity measures that are sensitive to transitions in intensity, color, and texture as a basis for camouflage evaluation. The measured the difference in mean, standard deviation, minimum and maximum value, coefficient of skewness, coefficient of kurtosis, and Bhattacharyya distance between the first order distribution of objet and background. They also used measures based on spatial, gray level co-ocurrence matrix (SGLCM) and the spatial gray level difference distribution of Weszka. All of these features are determined for each color-component image over a range in scale space. The texture features are also calculated for four orientations of the displacement operator. The final step of MACE system is to rank order a series of camouflage patterns, based on its features.

### H.3.1 Georgia Tech Vision GTV

The Georgia Tech Research Institute developed a simulation of human vision and visual cognition that is capable of automatically detecting and identifying targets and other types of visual features in visible, IR, and SAR imagery. The simulation, called GTV (Georgia Tech Vision), was originally developed for the Army AMCOM for the purpose of evaluating camouflage and IR suppression. It has since proven to be a powerful tool when used in a number of ATR scenarios, including identification of defects in agricultural products, finding areas of interest in multi-spectral overhead imagery, and identification and classification of lesions in biomedical imagery.

### H.3.2 Synthetic Image Analyst (SynIA)

SynIA is a GTV-based software that incorporates a model of vision and visual cognition. Like a human image analyst, SynIA must experience each target in a number of different views and contexts before it can reliably discriminate the targets from other objects and background clutter.

The normal sequence of operations in interpreting images is:

- **Analyze** the both the training and test images into their visual features.

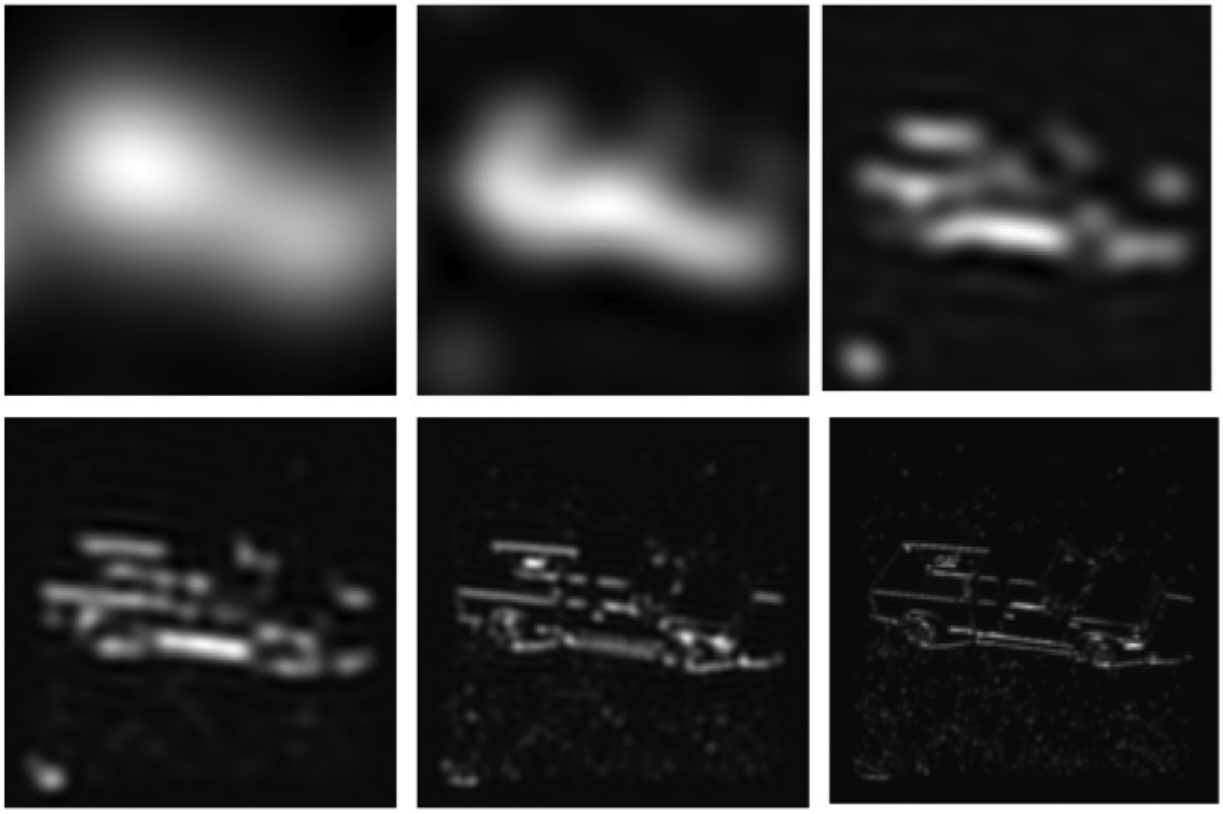


Figure H.4: Examples of spatial frequency channel outputs for the Òbare vehicleÓ input image shown in Figure 5

- **Train** SynIA by using the outputs of the SynIA Analysis for the training set. This generates *target knowledge* files. In training, the targets are specified by *mask images*, in which the targets pixels are set to non-zero values and other pixels are set to zero.
- **Interpret** the test images by using the outputs of SynIA analysis and specifying the target knowledge files generated by training SynIA with similar images.

In order to quantify the degree of similarity between signature of the various vehicle configurations, it used a figure of merit called the Spatial Pattern Angle. This measure is an analog of the spectral angle. The spectral angle, commonly used in hyper-spectral signal analysis, is the resultant angle between two points in multi-dimensional space made up of the various spectral dimensions. Analogously, the measure used in this study is the resultant angle between points. The spatial pattern angle is defined in a hyper-space defined by the 24 spatial-filter channel outputs, as:

$$\theta = \cos^{-1}\left(\frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|}\right),$$

where $\vec{a}$ and $\vec{b}$ are vectors representing the 24 filter channel outputs for two different target configurations.

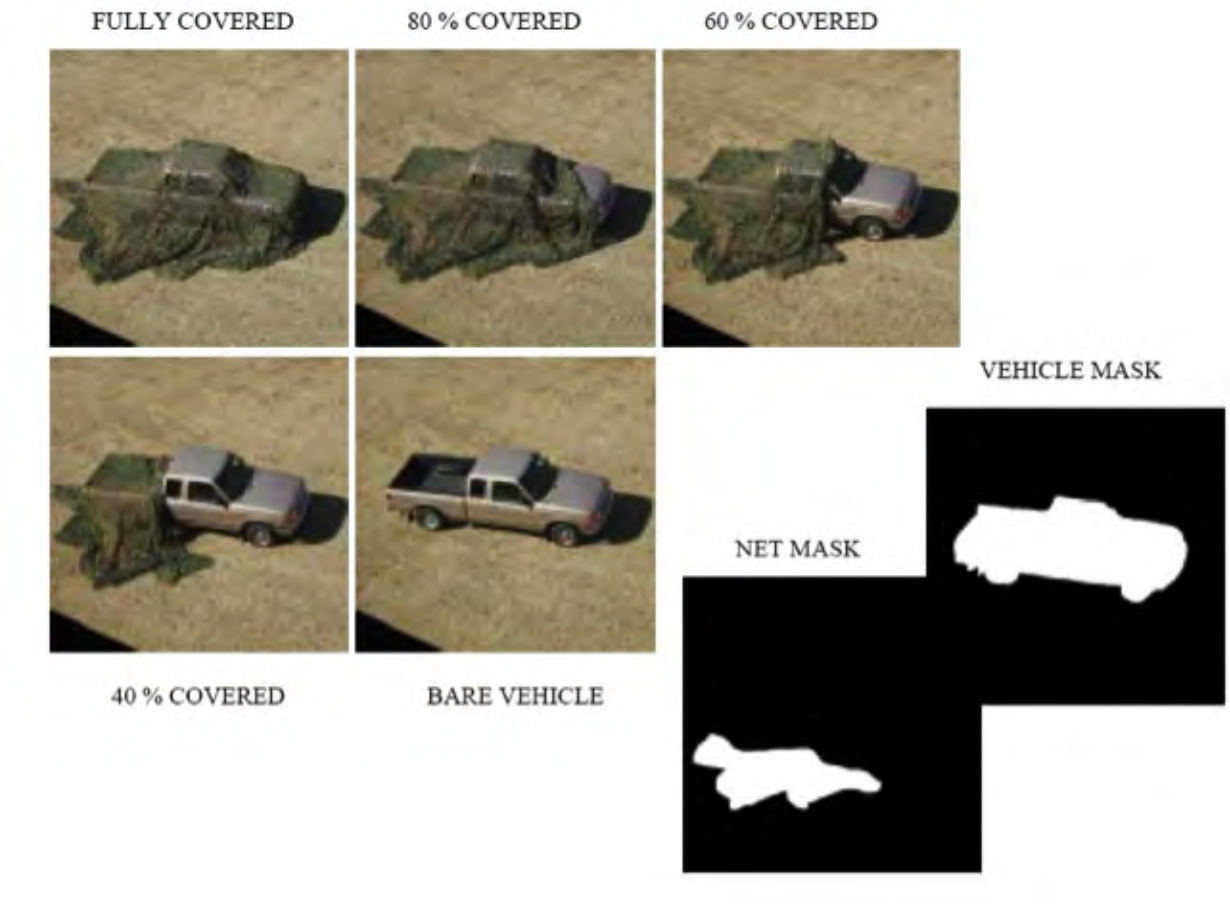



Figure H.5: Input images showing target configurations and masks.

# Bibliography


[1] B. Coll A. Buades and J.M. Morel. Nonlocal image and movie denoising. *Int. J. Computer Vision*, 76:123–139, 2008. (document), D, E.1

[2] B.E. Bayer. Color imaging array. 1976. G.3.4

[3] T. F. Chan and J. Shen. Variational restoration of non-flat image features: models and algorithms. *SIAM Journal on Applied Mathematics*, 61:1338–1361, 2000. C.5.4

[4] T.F. Chan and J. Shen. Mathematical models for nonlocal texture inpaintings. *SIAM Journal of Applied Mathematics*, 62:1019–1043, 2002. G.3.2

[5] T.F. Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *Image Processing, IEEE Transactions on*, 1998. 2.1

[6] Tony F. Chan, Jiahong (Jackie) Shen, and Luminita Vese. Variational pde models in image processing. *Notices Amer. Math. Soc*, 50:2003, 2003. 2.1

[7] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling and Simulation*, 6:595–630, 2007. E

[8] J.F. Hamilton and J.E. Adams. Adaptive color plan interpolation in single sensor color electronic camera. 1997. G.3.4

[9] A. D. Jepson and M. J. Black. Mixture models for image representation. *PRECARN ARK Project Technical Report ARK96-PUB-54*, 1996. C.3.2

[10] Miyoun Jung, Gabriel Peyré, and Laurent D. Cohen. Texture segmentation via non-local non-parametric active contours. In *Proceedings of the 8th international conference on Energy minimization methods in computer vision and pattern recognition*, EMMCVPR'11, pages 74–88, Berlin, Heidelberg, 2011. Springer-Verlag. 2.3

[11] YH Tay JY. Tou and PY Lau. Recent trends in texture classification: A review. 2

[12] S. Didas S. Grewenig L. Pizarro, P. Mrázek and J. Weickert. Generalised nonlocal image smoothing. *International Journal of Computer Vision*, 90:62–87, 2010. (document), F

[13] R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley and Sons, 2002. C.3.2

[14] T. Chan M. Jung, X. Bresson and L. Vese. Nonlocal mumford-shah regularizers for color image restoration. *IEEE Transactions on Image Processing*, 20:1583–1598, 2011. (document), 3.2.2, G

[15] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12:839–842, 2005. E.8.1

[16] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 4.2.2

[17] G. Dal Maso. *An Introduction to* $\Gamma$*-convergence*. Birkhauser, 1992. C.5.1

[18] M. Modica and S.Mortola. Un esempio di gamma-convergenza. *Bollettino della Unione Matematica Italiana B*, 5-14:285–299, 1977. C.5.1

[19] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989. B.1

[20] J. Shen. A stochastic-variational model for soft mumford-shah segmentation. *Int. J. Biomedical Imaging*, 2006, 2006. (document), C

[21] Z. Tu and S. C. Shu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:657–673, 2002. C.5.1

[22] S. C. Zhu and A. Yuille. Region competition: unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:884–900, 1996. C.5.1