Example-Based Regularization in Inverse Problems

Dmitry Datsenko
Example-Based Regularization in Inverse Problems

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science

Dmitry Datsenko

Submitted to the Senate of
the Technion — Israel Institute of Technology

Tishray 5766 Haifa October 2006
This research thesis was done under the supervision of Dr. Michael Elad.

I thank him for his dedicated guidance and assistance in this research.

A special "thank you" to my wife Katya for her love and encouragement.

The generous financial help of the Technion is gratefully acknowledged.
# Contents

Abstract .......................... 1

Notation and Abbreviations ........ 3

1 Introduction ........................ 5
   1.1 The Need for Regularization .... 5
   1.2 Regularization Techniques ....... 6
   1.3 The Use of Examples for Regularization ... 7
   1.4 This Work Contribution ........... 7
   1.5 This Thesis Structure ............ 8

2 Background on Regularization ...... 9
   2.1 The Maximum-Likelihood Estimator .... 9
   2.2 The Bayesian Point of View and Regularization ... 11
   2.3 Evolution of Image Priors .......... 13

3 Using Examples – Survey of Techniques 17
   3.1 Learning Prior Parameters via Examples ... 18
   3.2 Learning the Posterior Directly via Examples ... 20
   3.3 Building the Regularization with Examples ... 24
   3.4 Using Examples: A Summary ........... 26
List of Figures

3.1 Influence zone of pixels in a degraded patch. For a 5 × 5 patch in the low-resolution grid, and assuming a degradation that includes a 3 × 3 footprint blur and a 2 : 1 decimation in each axis, the corresponding maximal patch size in the high-resolution grid is of size 11 × 11. .......................... 22

4.1 Training Stage Algorithm .................................................. 29
4.2 Searching Stage Algorithm .................................................. 31
4.3 Reconstruction Stage Algorithm ............................................ 34
4.4 Comparison between different high and low quality sizes. These results were validated over the test case shown in Figure (5.2). The value n is the low quality patch size, MaxHQSize stands for the maximal high quality patch size with respect to n (based on Equation (4.1)). The graphs show the influence of the number of pixels used from the high quality patch on the overall MSE. .......................... 34

4.5 Outliers among valid examples. Top: The high quality image (left), and the corresponding measurements (right). Both 11 × 11 and 9 × 9 blocks are marked. Bottom: the 50 nearest neighbors found, their RMSE in the low-resolution and the high-resolution (9 × 9) domains. As can be seen, while all examples are close in the low-resolution, many of them are in fact outliers. .......................... 36

4.6 Pruning Stage Algorithm .................................................. 39
5.1 Example database for Simulation #1 - text image reconstruction. . . 44
5.2 Simulation #1 - text image reconstruction results. ................. 44
5.3 Simulation #1 – a text image: the pruning effect on the reconstruction MSE. .................................................... 44
5.4 Example database for Simulation #2 – text image reconstruction. . 45
5.5 Simulation #2 – a text image. ........................................... 45
5.6 Example database for Simulation #3 – formula image reconstruction. 45
5.7 Simulation #3 – an image with formula. .............................. 46
5.8 Simulation #3 – an image with formula: the pruning effect on the reconstruction MSE. ................................. 46
5.9 Example database for Simulation #4 – lines image reconstruction. . 46
5.10 Simulation #4 - lines image reconstruction results. ................. 47
5.11 Simulation #4 – a portion of a graph: the pruning effect on the reconstruction MSE. ................................. 47
5.12 Example database for Simulation #5 – a drawing image. ......... 47
5.13 Simulation #5 – a drawing image. (a) Original image of size 119 × 69; (b) Degraded image; (c) Bi-cubic interpolation result (MSE=1118.9); (d) MMSE reconstruction (MSE=463.2); (e) Reconstruction after 2880 pruning iterations (MSE=366.0). .................................... 47
5.14 A block diagram of the training algorithm when using the higher frequencies as features. .................. 48
5.15 A block diagram of the reconstruction algorithm using the high-frequencies features. ................................. 48
5.16 Comparison between an algorithm that operates on raw data and one that uses high frequencies only. Scale factor is 2. ................. 48
5.17 The ORL face database, containing 40 different people, each with 10 images of size 112 × 92 pixels. .................. 49
5.18 Face #1 results. (a) Original image; (b) Degraded image; (c) Bicubic interpolation result (MSE=149.64); (d) MMSE reconstruction (MSE=75.4) based on 9 images; (e) MMSE reconstruction (MSE=97.1) based on 390 images; (f) MMSE reconstruction (MSE=79.9) based on 399 images; (g) Pruned (1970 steps) reconstruction (MSE=62.38) based on 9 images; (h) Pruned (3060 steps) reconstruction (MSE=88.01) based on 390 images; (i) Pruned (2030 steps) reconstruction (MSE=72.71) based on 399 images.

5.19 Face #2 results. (a) Original image; (b) Degraded image; (c) Bicubic interpolation result (MSE=159.25); (d) MMSE reconstruction (MSE=65.24) based on 9 images; (e) MMSE reconstruction (MSE=102.97) based on 390 images; (f) MMSE reconstruction (MSE=68.90) based on 399 images; (g) Pruned (2280 steps) reconstruction (MSE=60.08) based on 9 images; (h) Pruned (5320 steps) reconstruction (MSE=92.66) based on 390 images; (i) Pruned (3780 steps) reconstruction (MSE=65.61) based on 399 images.
Abstract

Regularization plays a vital role in inverse problems in image processing, and especially in ill-posed ones. Along-side to classical regularization techniques based on smoothness, entropy, and sparsity, an emerging powerful regularization is one that leans on image examples. Today’s literature offers a variety of example-based techniques for regularization. Generally speaking, these techniques can be divided into two main categories: (i) methods using parametrized priors, with a learning method for setting these parameters; and (ii) methods that use examples directly. This work starts by reviewing existing contributions, and carefully pointing to their shortcomings. More specifically, the first family of methods are often too limited in grasping the richness conveyed by the examples database, which naturally lead to inferior performance. The use of examples directly, as the second family of methods advocates, is often coupled with lack of a clear global quality criteria, leading unavoidably to heuristic schemes.

In this work we propose an efficient scheme for using image examples as driving a powerful regularization, applied to several classic inverse problems in image processing - denoising, deblurring, and super-resolution. The proposed framework fuses existing elements from example-based reconstruction algorithms, as described above, and overcome many of their shortcomings. The proposed algorithm starts by assigning per each location in the degraded image several candidate high-quality patches. Those are found as the nearest-neighbors in an image-database that contains pairs
of corresponding low- and high-quality image patches. The found examples are used for the definition of an image prior expression, merged into a global MAP penalty function. We use this penalty function both for rejecting some of the irrelevant outlier examples, and then for reconstructing the desired image.

The proposed algorithm strives to enjoy from benefits of both example-based approaches mentioned above, in order to achieve a solution that would be both analytically and visually pleasing. Along with promising results on the denoising and the deblurring problems, this work also addresses the single-image super-resolution problem - a far more challenging task. We demonstrate our algorithm on scanned text, graphics, and drawings and face images, showing high quality outcome.
Notation and Abbreviations

SR — Super-Resolution.

PDF — The Probability Density Function.

(M)MSE — (Minimum) Mean Squared Error.

MLE — Maximum Likelihood Estimator.

MAP — The Maximum A-posteriori Probability.

FIR — Finite Impulse Response.

PDE — Partial Differential Equation.

RML — Regularized Maximum Likelihood.

DB — Database.
Chapter 1

Introduction

1.1 The Need for Regularization

The conventional super-resolution (SR) process uses a multitude of measured low-quality images in order to produce the super-resolved outcome. It is well known that such SR process may lead to higher optical (i.e., true) resolution. The higher-frequencies in the resulting image, which represent the newly-introduced details, are in fact available in the measurements in an aliased form. The SR process recovers these high frequencies by exploiting the various given images, each exhibiting a different aliasing effect. This explains why such resolution improvement is possible in the first place. However, for such a process to succeed, sufficient number of low-resolution images are needed, so as to enable the recovery of the aliased frequencies uniquely.

Based on the above reasoning, one might be led to the natural conclusion that super-resolution based on a single measured image is impossible. Is it indeed so? The answer depends on the available information the reconstruction process has access to. Clearly, one type of information that is made available to the reconstruction process is the measured image(s). Those alone could suffice if enough of them are available, as described above. If only one image is given, an alternative source of information is necessary, so as to compensate for the lack of data. An a-priori knowledge about the objects in the image could be proposed
as such source of information. This leads naturally to the concept of regularization.

Regularization plays a vital role in inverse problems, and especially in ill-posed ones, where insufficient data is available. One way to interpret the regularization is a way of gaining an algebraic stability in the reconstruction process. However, regularization is much more than a mere stabilization technique. A Bayesian point of view interprets such addition to inverse problems as a way of exploiting the probability density function (PDF) of images – the prior. This way, a properly chosen regularization can direct the solution towards a better quality outcome, by bringing into account the proper behavior of the desired image. Indeed, in the extreme case, super-resolution from a single measured image – the image scale-up problem – can be made possible and successful due to such well-chosen prior.

1.2 Regularization Techniques

Much of the progress made in the past two decades on inverse problems in image processing can be attributed to the advances in forming or choosing the way to practice the regularization. The simplest regularization practiced was based on Tikhonov’s idea [37], enforcing spatial smoothness uniformly on the output image. This option leads to the well-known Wiener filter [38] for image restoration, and is known to over-smooth image edges. Introduction of spatially adaptive smoothness priors was shown to lead to better results, leaning first on a weighted least-squares scheme [14], and later on robust statistics techniques [23]. In fact, much of the activity that brought partial differential equations (PDE) to the realm of image processing has to do with ways of defining edge-preserving regularization terms [32, 35]. In parallel to those techniques, sparsity of transform coefficients (e.g. wavelet) has been also used as a way of forming regularization in inverse problems [10, 7].

Common to all the above regularization methods is the use of closed-form simplistic mathematical expressions in defining the PDF of images. One must ask: can the wealth of image content be grasped by such simple expressions? Judging by the quality of results obtained in challenging inverse problems (e.g. deblurring and super-resolution) that employ these regularization methods, the answer is unfortunately negative. While such methods
perform much better than previously practiced reconstruction algorithms, the quality of the results is typically far from being satisfactory. Realizing this, in recent years there has been a trend of seeking better and more complex priors of various sorts.

### 1.3 The Use of Examples for Regularization

One fascinating and promising such direction is the use of examples, basically suggesting that instead of arbitrarily and intuitively defining the PDF, let image examples help in defining it. This work focuses on this *example-based* approach, describing how examples can and have been used effectively for regularization, reviewing the main contributions along these lines in the literature. As it turns out, there are two main effective ways to exploit examples in inverse problems – use them to fine-tune the parameters of previously defined regularization expressions or use them directly for the reconstruction procedure. These two options are described in details in this thesis. A fusion of these two approaches could be suggested in all sorts of ways, and indeed this is the path taken in this work.

The use of examples becomes much more effective when handling narrow family of images, such as scanned documents, drawings or face images. Indeed, in this work we concentrate on documents and face images as specific image families for which general purpose priors leave much room for improvement.

Beyond the offered review on the use of examples in inverse problems in general, this work presents a novel algorithm for image denoising, deblurring, and scale-up, fusing ideas from the above described three families of algorithms.

### 1.4 This Work Contribution

In this work we describe an effective technique for reconstruction of the linearly degraded grayscale images. Instead of leaning on the conservative way of defining pure analytic regularization functional, which describes the PDF of images, either general ones or belonging to a specific family, we refer to image examples, allowing them to form an expression that
practically bypasses the need for an explicit designing of a regularization term. After building this example-based prior, it can be easily plugged into a global quadratic optimization problem, which can be solved numerically.

Along with the framework providing image reconstruction as a solution of an optimization problem, we propose a way for discarding initially selected examples, which do not fit the global reconstructed image context. These outliers are sequentially detected and pruned, refining the examples-based prior and leading to better reconstruction results. The proposed algorithm is demonstrated on scanned documents with text, equations, and graphics, and also on face images. The results obtained are state-of-the-art, confirming the potential of the developed method.

### 1.5 This Thesis Structure

In Chapter 2 we give the necessary background on linear inverse problems and statistical estimators. We show how the Bayesian approach derives the need for regularization term followed by a tour of regularization techniques evolution.

In Chapter 3 we give a survey on existing work that deploy example databases, describing how examples can be used in the construction of the prior. As already mentioned above, there are two major techniques that use image examples for reconstruction, and these two families of techniques treat the examples in a different way.

A third approach could be suggested, where a fusion of the above two options is done, combining their strength. Our algorithm belongs to this group of methods. In Chapter 4 we describe our technique for example-based image reconstruction with its algorithmic contributions and outliers rejection scheme.

In Chapter 5 we provide results of applying our approach to reconstruction of images from different families – scanned documents, drawings, and face images.

In Chapter 6 we draw conclusions about the presented approach and our contribution, along with marking areas for future research and discussion.
Chapter 2

Background on Regularization

2.1 The Maximum-Likelihood Estimator

A fundamental signal processing problem is the recovery of a signal $x \in \mathbb{R}^n$ from a measurement vector, $y \in \mathbb{R}^m$, related to it through

$$ y = Hx + \nu. $$

(2.1)

In this equation, the matrix $H \in \mathbb{R}^{m \times n}$ represents some linear degradation operation (in the context of image processing, it may represent blur, decimation, geometrical warp, and more), and $\nu \in \mathbb{R}^m$ stands for an additive noise, assumed to be a zero-mean and white Gaussian random vector with probability density function given by

$$ \text{Prob}(\nu) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left\{ -\frac{\nu^T \nu}{2\sigma^2} \right\}. $$

(2.2)

We assume that the degradation operator $H$ and the noise variance $\sigma^2$ are known. This degradation model is very common in image processing and includes such particular cases, such as:

- Additive noise only, where $H$ an identity matrix. This leads to the image denoising problem.
• Convolution with some blurring kernel and additive noise, where $H$ represents a blurring FIR filter. This leads to the well studied image deblurring problem.

• The operator $H$ could also include a downscaling effect to it, turning this estimation problem to the image scale-up (or super-resolution) problem.

• $H$ could represent a sequence of projection operations, leading to the tomography problem.

An inverse problem consists of using the measurements $y$ to infer the values of the original image $x$, and this is the general problem we target in this work.

The maximum likelihood estimator (MLE) suggests to choose the signal $x$ that leads the conditional probability $\text{Prob} \left( y | x \right)$ – termed the likelihood function – to maximum. This means that we choose the signal that makes the measurements the most likely to take place, and thus the name of this method. Clearly, such method exploits the measurements alone in forming the estimated result.

Considering the model in Equation (2.1), based on the Gaussianity of the noise and the fact that $x$ is assumed to be known, the measurement vector is also a Gaussian random vector with a shifted mean. Thus, the likelihood function becomes

$$\text{Prob} \left( y | x \right) = \text{Prob} \left( Hx + v | x \right) \quad (2.3)$$

and so

$$\text{Prob} \left( y | x \right) = \frac{1}{(2\pi)^{n/2}\sigma^n} \cdot \exp \left\{ -\frac{1}{2\sigma^2} \cdot \|y - Hx\|_2^2 \right\}. \quad (2.4)$$

Therefore, the MLE result is given by

$$\hat{x}_{ML} = \arg \max \limits_{x} \text{Prob} \left( y | x \right) = \arg \min \limits_{x} \|y - Hx\|_2^2. \quad (2.5)$$
In cases where the Gram matrix $H^T H$ is positive definite, the problem is considered well-posed, and there is a unique solution to the above minimization, being

$$\hat{x}_{ML} = (H^T H)^{-1} H^T y.$$  \hspace{1cm} (2.6)

In cases where $H^T H$ is singular, there are infinitely many possible solutions, caused by the null-space of the matrix $H$. In such a case, the problem is considered ill-posed, and more information is necessary in order to tune the reconstruction towards a unique solution. This leads naturally to the notion of regularization. From a purely algebraic point of view, regularization of the MLE is done by turning the penalty function into a strictly convex one, thus guaranteeing a unique solution. A simple way of achieving this goal is via Tikhonov’s approach,

$$\hat{x}_{RML} = \text{Arg min}_{\hat{x}} \|y - H\hat{x}\|_2^2 + \lambda \|S\hat{x}\|_2^2 = (H^T H + \lambda S^T S)^{-1} H^T y,$$ \hspace{1cm} (2.7)

where $S^T S$ is assumed to be positive definite. The new solution corresponds to a regularized ML approach (RML). Notice that an arbitrary quadratic term $\|S\hat{x}\|_2^2$ has been added here, and while it removes the ill-posedness of the original problem, it is unclear at all whether it helps in getting a proper result.

The above discussion might lead to the wrong impression that regularization is necessary only if the problem is ill-posed. Considering the signal denoising problem, where $H = I$, the matrix $H^T H$ is positive definite and thus the problem is well-posed. Nevertheless, the MLE result due to Equation (2.6) is $\hat{x}_{ML} = y$, which unveils the weakness of the MLE.

### 2.2 The Bayesian Point of View and Regularization

The Bayesian approach starts with the replacement of the likelihood function with the posterior probability $\text{Prob}(x|y)$. With this seemingly minor change comes a revolutionary
perception of the problem at hand, because now $x$ is assumed to be random as well. The
Bayes formula ties the above two conditional probabilities by

$$\text{Prob}(x|y) = \frac{\text{Prob}(y|x) \text{Prob}(x)}{\text{Prob}(y)}.$$  \hfill (2.8)

Generally speaking, there are two ways to practice the Bayesian approach and lead to a
constructive estimate of $x$ - the maximum a-posteriori probability (MAP) and the minimum
mean squared error (MMSE) methods. The simpler method is the MAP one, choosing the
$x$ that maximizes $\text{Prob}(x|y)$. Using Equation (2.8) this reads

$$\hat{x}_{\text{MAP}} = \underset{x}{\text{Arg max}} \text{Prob}(x|y) = \underset{x}{\text{Arg max}} \text{Prob}(y|x) \text{Prob}(x).$$  \hfill (2.9)

Observe that the denominator $\text{Prob}(y)$ has been removed from consideration, since it is
considered as a constant with respect to the optimization task. For reasons to be clear
shortly, a convenient way to describe the probability density function of $x$ is the Gibbs
distribution, which represents $\text{Prob}(x)$ as an exponential term

$$\text{Prob}(x) = \text{Const} \cdot \exp\{ -\alpha A(x) \}.$$  \hfill (2.10)

Such a description loses no generality, as every non-negative function can be written in such
a format. The constant in-front of the exponential is a normalization factor, guaranteeing
that the integral over all $x$ leads to 1. The term $A(x)$ is a non-negative energy function,
supposed to be low for highly probable signals and high otherwise. Using this, and the
expression we already have for the likelihood function in Equation (2.4), we obtain

$$\hat{x}_{\text{MAP}} = \underset{x}{\text{Arg max}} \text{Prob}(y|x) \text{Prob}(x) = \underset{x}{\text{Arg min}} \| H \hat{x} \|_2^2 + 2\alpha \cdot A(x).$$  \hfill (2.11)

We see that the MAP method leads naturally to the concept of regularization as already
described in Equation (2.7), only this time giving a probabilistic meaning to the additional
expression $A(x)$, rather than settling with the gained algebraic stability.

A second, more involved, way to practice the Bayesian approach is the MMSE estimator.
This option chooses the expected value of $x$ based on its conditional density $\text{Prob}(x|y)$, i.e.

$$\hat{x}_{\text{MMSE}} = E\{x|y\} = \int_{x_1} \int_{x_2} \cdots \int_{x_n} x \cdot \text{Prob}(x|y) \, dx.$$  \hspace{1cm} (2.12)

It is easily seen that this solution leads to the minimizer of the expression

$$E\{\|\hat{x} - x\|_2^2 | y\} = \int_{x_1} \int_{x_2} \cdots \int_{x_n} \|\hat{x} - x\|_2^2 \cdot \text{Prob}(x|y) \, dx,$$ \hspace{1cm} (2.13)

which is the mean-squared-error measure. This explains the name chosen for this estimator. Since the integral is $n$-dimensional (as the dimension of $x$), such an approach is typically prohibitive for high-dimensional cases.

Whichever method chosen, MMSE or MAP, the estimation of $x$ using the Bayesian approach requires a clear definition of the energy function $A(x)$. When dealing with images, this energy function is essentially describing how natural images behave. In the next subsection we describe the main choices made for $A(x)$ in the past 2-3 decades, showing the evolution of reasoning practiced.

### 2.3 Evolution of Image Priors

Assuming the Gibbs distribution for images, as in Equation (2.10), what should $A(x)$ be so as to reflect the distribution of natural images? This question poses one of the most fundamental problems in image processing. This enigma has drawn a considerable research attention in the past 2-3 decades, and is still considered an open question. In this Section we briefly describe the main milestones in this arena, showing how priors are getting smarter and more complex, all in the attempt to better describe image content.

This simplest and the earliest, but not necessarily the most natural option, is to choose the minimal-energy solution, setting $A(x) = \|x\|_2^2$. This prior makes a doubtful assumption about the reconstructed signal being located near the origin. The Tikhonov regularization presented earlier is also among the firsts to be practiced, generalizing the above energy consideration to an affine warp of the space, choosing $A(x) = \|Lx\|_2^2$. In image processing...
$L$ is often chosen as the Laplacian operator, leading to the promotion of spatial smoothness across the image in a uniform way. The uniformity is a key feature of this choice, as it leads to numerical convenience in the classic deblurring problem, where $H$ is a linear space-invariant blurring operation. In such a case, the matrix inversion required in Equation (2.7) can be easily done in the frequency domain, since $H^T H + \lambda L^T L$ is block-circulant (or could be approximated as such, after proper boundary treatment). This becomes the well-known Wiener filter algorithm, which for many years was the leading approach in image deblurring.

The choice of the Laplacian operation for measuring smoothness, both here and in later priors proposed, is not the only possibility, and similar regularization expressions can be practiced with other derivatives. For example, the choice $A(x) = \|D_h x\|_2^2 + \|D_v x\|_2^2$, with $D_h$ and $D_v$ being horizontal and vertical derivatives, can also be used. The difference is in the kind of smoothness that is expected from the outcome – while first order derivatives promote constant values, second derivatives such as the Laplacian allow tilted planes (and saddle points) as well.

By the late 80’s and early 90’s it became clear that the Wiener filter is not producing good enough results, and better ones are within reach when avoiding the enforced spatial uniformity. This basic idea of forcing smoothness adaptively across the image found many manifestations in various proposed image priors. One of the simplest ways is the weighted least-squares expression – $A(x) = (L x)^T W (L x)$. The matrix $W$ is a diagonal one with positive entries along the main diagonal being 1 for smooth regions, and close to 0 for edge or texture zones. This matrix can be build based on the measurements $Y$, assumed to contain enough information to yield such a segmentation. One positive feature of this choice is the fact that the MAP estimator remains linear, although frequency domain solutions are no longer relevant. Thus, iterative restoration techniques came to be prevalent and unavoidable.

Since the reconstruction process is iterative, one could update the weight matrix $W$ based on the current solution (assumed to be better than the measurements), and this way direct the solution towards a better result. As it turns out, this option is effectively obtained when exploiting concepts from robust-statistics. The field of robust statistics
focuses on estimation in the presence of outliers. In the regularization expressions we have seen above, edges are causing outliers. While most of the regions in an image provide low energy after the Laplacian operation (due to their smoothness), edges cause very high and exceptionally different values. Penalizing those using the $\ell^2$-norm leads to a very strong penalty, which is avoided by smoothing out the edges, as indeed happens. The alternative is to use robust measures such as $\ell^1$-norm, the Hubber-Markov or the Cauchy functions, etc.

Then the choice of $A(x)$ becomes $A(x) = 1^T \rho(Lx)$, where $\rho(\cdot)$ is a scalar robust (having a sub-linear derivative) function. When operating on a vector it is applied entry-wise, and thus the above is simply the sum over all the entries. Clearly, with this choice of prior the overall reconstruction algorithm becomes non-linear. If $A(x)$ is convex then a unique solution is guaranteed, and can be found via an iterative procedure. This leads to systematic ways of designing non-linear filtering techniques, as indeed required in images due to their non-homogeneity.

A vast amount of activity in image processing, which seems to be independent of the above discussion, is the introduction of partial differential equations (PDE’s) into image processing filtering techniques. As it so happens, contributions such as the Perona-Malik scheme [30], the total variation (TV) by Rudin, Osher and Fatemi [32], the Beltrami flow due to Sochen, Kimmel, and Malladi [35], the directional filter due to Weickert [40], and many more, are directly related to the robust-statistics techniques, although formulated in the continuum. For example, the TV suggests the energy function $A(x) = \|\nabla x\|_1$, which clearly uses a derivative and a robust integration measure. To this date, the TV and its variants are considered among the best regularization techniques available, and are very often used in image processing.

In parallel to the impressive progress made on the use of PDE’s in image processing in defining regularization expressions, the field of approximation theory contributed its own techniques as well for this purpose, and in particular via the use of the wavelet transform. Empirical observations suggested that after a wavelet transform, the coefficients of natural signals tend to be sparse, i.e., many of them are zero or near zero [10]. This led to a proposed regularization expression of the form $A(x) = \|T x\|_p^p$, where $T$ represents the
wavelet transform on the signal, and the $\ell^p$-norm (with $p \leq 1$) comes to sum over these coefficients in a way that promotes sparsity [24, 26, 34].

While substantially different from the previous options discussed, this regularization also applies some sort of derivatives, followed by a robust measure. The wavelet transform performs an inner product of the signal with zero-mean vectors that can be interpreted as multi-scale derivatives. While the TV and the like use a fixed scale and shift-invariant derivative (i.e. a derivative that applies uniformly across the signal), the wavelet option suggests a multi-scale set of derivatives, but without the shift-invariance property. More recent works deploy redundant transforms, leading to longer sequence of coefficients, so as to obtain both scale and shift invariance. Such constructions are considered among the best available methods today [12, 16, 25, 31, 36].

In summary, the quest for better regularization expressions for images is very much active today, with many new contributions that extend the above list of options and improve on them. Using the above rationale in forming the regularization, one must question the fundamental ability of a simple analytical expression $A(x)$ to grasp the complexity and wealth of general image content. This brings us to the main part of this thesis, presenting a new way of forming the regularization, based on image examples.
Chapter 3

Using Examples – Survey of Techniques

An emerging powerful regularization that is drawing research attention in recent years is one that leans on examples. Rather than guessing the image PDF and forcing a simple expression to describe it, we let image examples guide us in the construction of the prior. Examples can be used in a variety of ways, and the various proposed methods can be roughly divided into two categories:

1. **Learning prior parameters:** If we are generally pleased with the above-described analytical priors, those can be further improved by learning their parameters.

2. **Learning the posterior directly:** Rather than learn the image prior and then plug it in a MAP/MMSE reconstruction penalty term, one can use the examples to directly learn the posterior probability density function, and then use it for the reconstruction.

A mid-way between the above two options is to build a regularization expression with examples. This is a fusion of the above two techniques, where examples are found as part of the reconstruction process, and then plugged directly into an explicit regularization expression.
In the following subsections we expand on each of those families of methods, and describe related work found in the literature.

3.1 Learning Prior Parameters via Examples

Considering the vast progress made on the formation of regularization expressions, as described above, the most natural way to introduce examples into inverse problems is to keep the use of those expressions, and use examples to tune some parameters that control these priors. Thus, the regularization expression is $A(x, \theta)$ where $\theta$ are the parameters to be found.

A pioneering work by Zhu and Mumford considered this approach, steered to train a Markov random field (MRF) prior [43]. The energy function considered is

$$A(x) = \sum_{k=1}^{n} \lambda_k \rho_k(L_k x). \tag{3.1}$$

This function leans on a weighted average of robust measures of smoothness, using different robust functions $\rho_k(\cdot)$, analyzing filters $L_k$, and weights. All these can be learned in principle using a large corpus of high-quality image examples, $\{x_t\}_{t=1}^T$.

There can be many ways to tune the prior parameters, each considering a different objective. The work reported in [43] suggests to learn the parameters such that the marginals of the prior fit empirical observations, while maximizing the entropy of the probability density function, so as to consider a worst-case scenario. A different method of similar flavor has been proposed recently by Roth and Black, addressing the same energy function [33]. Their approach, termed fields of experts, aims to minimize the Kulback-Leibler distance between the empirical distribution of the example set and the prior trained.

Still using a database of high-quality images, the work by Buccigrossi and Simoncelli propose a prior learning for natural images, based on the statistics of such images in the wavelet domain [6]. While the classic use of wavelet promotes sparsity and assumes an independence between the coefficients, their work considered learning of the joint probabil-
ity of neighboring wavelet coefficients (space, orientation, or scale-neighborhoods). Their algorithm is far simpler than the ones in [43, 33], owing to the simplicity gained by the wavelet transform that allows for simple marginals to describe the required addition.

An entirely different approach for learning prior parameters is the one reported by Haber and Tenorio in [22]. Whereas the previous methods learned the prior and based this on a set of high-quality images only, the work in [22] uses pairs of $T$ images, $\{x_t, y_t\}_{t=1}^T$, representing the high quality image and its degraded version, using the same degradation (and noise) to be overcome in the inverse problem at hand. Thus, the images $y_t$ are generated by simulating the degradation effects,

$$y_t = Hx_t + \nu_t.$$  

(3.2)

Since the role of regularization is primarily to get better reconstruction, Haber’s way of finding the parameters is to minimize the reconstruction error on the above training set. Considering a MAP reconstruction formula

$$\hat{x}_{MAP}(\theta) = \text{Arg min}_{\hat{x}} \|y_t - Hx\|_2^2 + \lambda \cdot A(x, \theta),$$

(3.3)

the results are functions of the parameters $\theta$. By minimizing

$$\sum_{t=1}^T \|\hat{x}_{MAP}(\theta) - x_t\|_2^2$$

(3.4)

with respect to $\theta$, we tune the parameters to lead to the minimal mean-squared-error (MSE) in an empirical sense. Such interesting mixture of MMSE and MAP methods is very effective, and breaks the pure Bayesian interpretation of the energy function $A(x)$, since now it is related to the degradation operation. Just as before, here as well one faces a complicated optimization task, which can be handled comfortably only in simple parametric forms.

One last family of techniques that falls under the same regularization learning methodology is the one that targets the quest for a dictionary that yields sparse representations [1, 2, 12, 15, 29]. These methods are based on the assumption that the signal in mind, $x$,
could be created as a sparse linear combination of columns from the dictionary $D$, namely, $x = D\alpha$. The matrix $D$ is full-rank, having more columns than rows. This implies that there are infinitely many ways to construct $x$ as linear combination of columns from $D$. Among all these possibilities, we consider the sparsest – the one that fuses the smallest number of columns in such construction. Thus, handling the general inverse problem posed in Section 2, the MAP approach in this case leads to [7],

$$\hat{x}_{MAP} = D \cdot \text{Arg min}_\alpha \|y - HD\alpha\|_2^2 + \lambda \cdot \|\alpha\|_1.$$  \hspace{1cm} (3.5)

In this penalty function we describe the desired signal as $x = D\alpha$ and force its representation vector $\alpha$ to be sparse. The work reported in [1, 2, 12, 15, 29] all consider the question of training the dictionary $D$ to perform best in such inverse problems, with differences in the numerical methods proposed, or the way to fuse local and global relationships in the spatial domain.

Common to all the above methods is the fact that a parametric energy function is used and its parameters are tuned by the examples. Also, all these methods call for an involved optimization procedure, but one that should be done off-line. Once the regularization expression is ready, it can be deployed for use in the inverse problem in mind. Haber’s work restricts the parameters to the same inverse problem trained on, while the other techniques are more general, allowing the use of the prior found in every inverse problem. Generally speaking, the more parameters involved in the definition of the prior, the more flexible it becomes. However, with such gain in flexibility comes the optimization complications.

### 3.2 Learning the Posterior Directly via Examples

The above-described approach uses examples indirectly, by training the regularization parameters. An entirely different way of exploiting examples is to use them directly within the reconstruction process. In such an approach, the examples are gathered to a database and used explicitly in the on-line reconstruction algorithm. These gathered examples may be regarded as samples from the posterior $\text{Prob}(x|y)$, and as such they offer a direct way
of performing reconstruction.

The example database organization is similar to the way described in Haber’s method – gathering a set of high quality patches from example images and a corresponding set of degraded versions thereof, obtained by applying the degradation sequence on each. This gives us the set of patch pairs \( \{x_t, y_t\}_{t=1}^{T_s} \). Thus, the obtained database is tightly coupled to the type of degradation process that characterizes the inverse problem to be solved.

The basic idea behind the direct use of examples is one of pattern matching, i.e., given a low quality image \( y \), seek in the database similar low-quality examples. Taking their corresponding high-quality pairs, those could be used for the reconstruction as they provide the high-quality content that fits the measurements. Clearly, such a process cannot be operated on large size images, since this implies an impossibly large database, so as to guarantee that every possible content encountered in the measurements can be found in the database. Therefore, the above method is applicable for small patches of images, with typical size of the low quality images ranging between \( 5 \times 5 \) to \( 15 \times 15 \) pixels. This also implies that the above process is operated locally, or even on a pixel-by-pixel basis. Thus, given the image pairs described above, we sweep through the high-quality image set, and extract all image patches of size \( m \times m \) (with full overlaps, since we are interested in all possible examples contained in the training set). This gives a very large set of example patches, denoted as \( \mathcal{X} = \{x_t\}_{t=1}^{T_s} \) (there is no typical database size \( T_s \) - it varies according to the image content, and may range from \( 10^4 \) to \( 10^5 \) examples for text images up to \( 10^6 \) to \( 10^7 \) examples for pencil drawings). For each ”clear” patch \( x_t \in \mathcal{X} \), there is a corresponding patch of size \( n \times n \) that is attained by putting \( x_t \) through the degradation sequence. We denote these corresponding degraded patches as \( \mathcal{Y} = \{y_t\}_{t=1}^{T_s} \).

The choice of \( n \) and \( m \) is not trivial and depends on the image context – choosing too small \( n \) means that the low-resolution patch is too small, and thus many irrelevant examples join the reconstruction process and divert it. Too large \( n \) may lead to no adequate examples in the database, and thus to failure again. Along with that, the use of large \( n \) increases the computational complexity, decreasing the algorithm’s speed. As to the choice of \( m \), it depends on \( n \), and on the degradation operation. A critical value of \( m \) is the one
that contains all the pixels in $x_t$ that are involved in constructing the measurement $y_t$. For example, for $n = 5$, and a degradation that includes a $3 \times 3$ blur followed by $2 : 1$ decimation in each axis, we get $m = 11$, as Figure 3.1 shows. Choosing a smaller value for $m$ wastes an information within the corresponding measurements, and choosing a larger value for $m$ implies that the high-resolution patch relies on the spatial context, rather than the measurements alone, and as such, it may be misleading. Interestingly, the various works reported in [8, 11, 18, 19, 27, 39] all assume much smaller $m$. We will return to this point later, showing that indeed using $m$ near the critical value described above leads to better results.

Figure 3.1: Influence zone of pixels in a degraded patch. For a $5 \times 5$ patch in the low-resolution grid, and assuming a degradation that includes a $3 \times 3$ footprint blur and a $2 : 1$ decimation in each axis, the corresponding maximal patch size in the high-resolution grid is of size $11 \times 11$.

Once the database $\{\mathbf{X}, \mathbf{Y}\} = \{(x_t, y_t)_{t=1}^{T_x}\}$ is ready, it can be used directly in the re-
construction algorithm. Note that this data-set may contain simple gray-scale images, as described in [9], or high-frequency (and possibly multi-scale) content as has been commonly used in [3, 18, 19, 27]. This choice of representation is tightly coupled with the type of images we target.

Turning to the reconstruction process, consider a given low quality image \( y \), known to be damaged by \( H \) and by an additive white Gaussian noise of strength \( \sigma^2 \). Per every location \( p \) in the image we extract a patch of size \( n \times n \), denoted as \( y_p \). At the heart of the reconstruction process lies the need to find the nearest neighbors of \( y_p \) from \( \mathcal{Y} \). We consider all the candidate examples \( y_t \) in \( \mathcal{Y} \) satisfying

\[
\|y_p - y_t\|_2^2 \leq Th
\]

as possible matches (or the single nearest among them, if only one is desired). The threshold \( Th \) depends on the patch size and the noise variance (e.g., \( Th = 4n^2\sigma^2 \)). Define \( \Omega[p] \) as the set of indices of the found nearest neighbors, it’s a subset of \( \{t\}_{t=1}^{T_s} \). Having found this subset of examples, \( \mathcal{Y}^p = \{y^p_k\}_{k \in \Omega[p]} \), their corresponding pairs \( \mathcal{X}^p = \{x^p_k\}_{k \in \Omega[p]} \) are the candidate patches to be used for the reconstruction.

Given the reference vector \( y^p \) of length \( n^2 \) and the database \( \mathcal{Y} \) that contains \( T_s \) examples, the above-described search should be done efficiently, as it is part of the online algorithm. Some methods that have been considered in the literature for speeding-up the search include clustering techniques, principal component analysis, and other fast nearest-neighbor methods [28].

The above process is performed for every location \( p \) in \( y \), or with possible jumps to reduce computational complexity. Assuming a full-overlap approach, for every location \( p \) there is a set of candidate high-resolution \( m \times m \) patches \( \mathcal{X}^p = \{x^p_k\}_{k \in \Omega[p]} \). There are several ways one can use these results. Defining an output canvas \( \hat{z} \) as expanding the low-resolution image, we need to fill-in the pixel values. Every example found, \( \hat{z}^p_k \), has a known footprint on this canvas, and thus there are several intuitive ways to proceed:

1. **Scalar MMSE estimate:** Considering the pixel at location \( p \) in the output canvas
\( \hat{x} \) has many contributions, coming from all patches in \( \mathcal{X}^p \) that overlap it. By simply averaging these values we essentially perform an approximate MMSE estimate. This is because these values can be considered as samplings from the posterior \( \text{Prob}(x|y) \). By creating a histogram of these values, we get a 1D approximate description of this posterior, and the expected value can be computed by a simple mean of the samples.

2. **Scalar MAP estimate:** The above procedure is susceptible to outliers. Using the very same histogram of those values, one can seek its peak, and this will be the MAP estimation for the desired output. From a practical point of view, it is likely that this histogram is too poor to work with because of insufficient data, and curve fitting or smoothing will be needed.

3. **A special case – Non-Overlap and 1-NN:** If this algorithm extracts only the nearest neighbor, and if the patches used are taken with no overlap, we get only one value per each location \( p \) in the reconstructed canvas, and then the above two methods coincide, suggesting that the output at this location is simply the candidate value.

All these are pixel-based reconstructions, and as such, they are easy to implement. However, their simplicity comes with a price – the examples found contain many outliers, and those may divert the desired result. As we shall see in the next chapter, in some cases, for some locations, the number of outliers may exceed the number of proper ones, and in those cases, even the MAP method may deteriorate.

The works reported in [18, 19, 27] employ the non-overlapping option with 1-NN. Freeman et. al. [18, 19] also considered some (not full) overlaps, in the spirit of the MMSE approach described above. Other algorithms that lean on similar rational for texture synthesis, denoising, or inpainting (filling in holes) are found in [8, 11, 39, 41, 42]. This set of works is also markedly different in the origin of the examples – rather than taking them from a separate set of images, the examples are drawn from the given image itself. Another very related recent example-based work of extreme importance is the one reported in [4, 5]. These papers present an example-based image denoising algorithm, using examples from
the corrupted image itself, averaged via weighting in order to obtain denoising.

3.3 Building the Regularization with Examples

The last family of techniques is one that fuses the above two approaches, and thus improves on both. On one hand, a regularization expression that considers the entire unknown image as a whole is better than a local treatment, and as such, should be preferred. Furthermore, when joined to the likelihood term, the influence of the measurements and the regularization can be merged in a clear way to define the objective of the reconstruction procedure. On the other hand, a local treatment enables to parallelize the algorithm, simplifies the overall algorithms, and provides a direct way to use the examples in the reconstruction, rather than leaning on a guessed expression.

One can enjoy both worlds when merging the two techniques. First, operate locally as described above, and find per every pixel \( y_p \) its relevant nearest neighbor patches \( X_p \). However, instead of a simple operation such as voting or averaging, as proposed above, plug these examples into an especially tailored regularization expression. Such expression would represent a tighter description of the forces the unknown image is supposed to satisfy, and in a holistic way that considers all the image. This idea has been practiced successfully in several recent works [3, 9, 17, 19, 18]. Beyond the expected improvement caused by handling the reconstruction process globally, such an approach is able to better handle outliers in the found examples.

Interestingly, the regularization obtained in the above fused technique deviates from the classic Bayesian point of view. The above-proposed regularization cannot be considered as a general image prior, because it is a much narrower point of view of the image in mind. Furthermore, this expression is heavily dependent on the measurements, from which we have obtained the high-resolution nearest-neighbors, and as such, this expression “sees” much more than just the ideal signal behavior. One could consider this prior term as an attempt to model the true image prior in the vicinity of its true values, and as such being local in the signal space.
The pioneering work by Baker and Kanade [3] was the one to fully practice the above set of ideas. In handling the super-resolution problem, Baker and Kanade formed an explicit regularization expression that requires proximity between the spatial derivatives of the unknown image to those of the found examples. The examples in their work are found by a pyramidal derivative set of features, which means that rather than using the raw data directly and an $\ell^2$ measure of distance, a weighted $\ell^2$ is effectively used. Every location obtains one example, being the nearest-neighbor, and all these forces are merged into one global expression.

Freeman et. al. [18, 19] also considered a similar approach, but with some important differences. Rather than forming an explicit regularization expression, their MAP method adopts a Bayesian network point of view. Their algorithm defines local probabilities that take into account the proximity between the low-resolution measurements and the database patches (these parallel the likelihood term), and the agreement between high resolution neighboring patches between themselves (which parallel the regularization). The proposed algorithm remains local, as it does not consider the unknown image as a whole. Indeed, rather than concentrating on the true unknown $x$, the focus is on the network interpretation of the data, discovering the nearest neighbors that survive a Bayesian belief propagation algorithm, using those in the formation of the solution.

### 3.4 Using Examples: A Summary

In this section we have seen many ways to practice the use of examples in forming a regularization for inverse problems. The major questions one faces when designing such algorithms are:

- Which examples to use? the examples can be taken from the corrupted image itself or from other images. Also, one could work with pairs of low and high quality images, or only high quality images.
- Which estimator to use? we have seen the MMSE and MAP being used, and a related question is whether to work globally or locally.
• How to use the examples? we have seen them used indirectly by training a regularization parameters, directly in constructing the reconstruction result, or plugged into a tailored regularization expression.

• How to represent examples? beyond the natural use of raw data, one can extract features, as high frequencies, multi-scale derivatives, and more.

• How to organize the examples? in algorithms that employ on-line searches for the nearest neighbor a pre-organization is mandatory for a fast search. We have mentioned clustering methods, and PCA brought to use.

• Which inverse problem to consider? the overall algorithm depends on the type of inverse problem addressed.
Chapter 4

The Proposed Approach

Our work (also reported in [9, 13]) is inspired by the above algorithms, and considers a simplified MAP method that targets scanned document and face images. Similar to Baker and Kanade’s approach [3], an explicit regularization expression is formed, although using the raw data directly, instead of complicated features. As we show here, for the specific images handled in this work, this approach leads to better results. Also, instead of using a single example per location, our algorithm uses a multitude of them. A key feature in the proposed scheme is a pruning mechanism that draws its strength from the very same MAP expression mentioned above. Such pruning is paramount in handling outlier examples, often encountered in example-based techniques. We now turn to describe this algorithm in details.

Our algorithm considers the image scale-up problem, assuming that the following sequence of degradations has been applied to some ideal image:

1. Blurring with a given linear space invariant FIR kernel $B$ of size $b \times b$.

2. Subsampling by factor $d$ in each axis with arbitrary starting offset. While this work concentrates on integer values of $d$ for simplicity of notations, all the discussion given here could apply to arbitrary ratios just as well.

3. Adding a zero-mean and white Gaussian noise with variance $\sigma^2$. 
The denoising and deblurring problems are special cases of the above degradation model.

Along with knowing the degradation characteristics, we ought to have a set of images, belonging to the same family as the corrupted one. The reconstruction algorithm is divided into four major stages:

- Gathering a database of example pairs - executed on the provided set of examples, once for each image family.
- Searching for legitimate examples - executed once for every corrupted image.
- Building a reconstructed image according to the found examples.
- Pruning irrelevant examples to improve the reconstruction quality.

While the first stage is done off-line, all the remaining ones are parts of the on-line reconstruction algorithm.

The third stage appears both independently and also as a part of the pruning stage, since as it will be shown later, reconstruction is an intermediate part of the pruning stage, which is executed repeatedly until a stopping condition is reached. As opposed to the work described in [19, 18, 3], gray-scale values are used directly, simplifying the overall algorithm.

4.1 Training Stage

The routine Training Stage in Figure 4.1 describes the off-line building of the examples dictionary.

Given the desired low-quality patch size $n \times n$, blurring kernel size $b \times b$ (we assume, that $n$ and $b$ are odd numbers - for symmetry reasons) and the downsampling factor $d$, we can calculate the minimal (critical) $m$, so that the high-quality patch of size $m \times m$ would become exactly of size $n \times n$ after the degradation:

$$m = d(n - 1) + b.$$  \hspace{1cm} (4.1)

High quality patches of size $m \times m$ are extracted over all pixel locations in all provided
training images (with overlaps). Matching low-quality patches are acquired by applying
the controlled degradation consisting only of blurring and downsampling (i.e, without ad-
ditive noise\(^1\)). The gathered set of high- and low-quality patches are denoted by \(\mathcal{X}\) and \(\mathcal{Y}\) respectively, as described in the previous chapter.

As the obtained database of low- and high-quality patches is to be used for extensive
searches in consequent steps of the algorithm, an efficient structure is required for its storage
for later use. In our work we have chosen to use the K-D-Tree algorithm that will be
explained in the next section.

The routine Training Stage is run only once, and its outputs can be used repeatedly
for different reconstruction tasks, as long as we are handling the same degradation operators
and the same image family. This stage is a preparation for the reconstruction task and is
executed offline, before actually analyzing the degraded image.

<table>
<thead>
<tr>
<th>TRAINING STAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Set of clean images (S); blurring kernel (B) of size (b \times b); scale factor (d); low-quality patch size (n).</td>
</tr>
<tr>
<td><strong>Output:</strong> K-D-Tree (LQT)ree, containing degraded patches (\mathcal{Y}); clean patches (\mathcal{X}).</td>
</tr>
<tr>
<td><strong>Assign:</strong> (\mathcal{X} = {}); (\mathcal{Y} = {});</td>
</tr>
<tr>
<td><strong>Set High-Resolution Patch size:</strong> (m = d(n - 1) + b).</td>
</tr>
<tr>
<td><strong>Start Loop:</strong> For each image (s) in (S) and each pixel location (p),</td>
</tr>
<tr>
<td>(\bullet ) Extract high-quality patch (x^p) of size (m \times m) centered in (p) from image (s).</td>
</tr>
<tr>
<td>(\bullet ) Blur and decimate (x^p) by the proper degradation steps and obtain (y^p).</td>
</tr>
<tr>
<td>(\bullet ) Update the database (\mathcal{X} = \mathcal{X} \cup x^p)</td>
</tr>
<tr>
<td>(\bullet ) Update the database (\mathcal{Y} = \mathcal{Y} \cup y^p)</td>
</tr>
<tr>
<td><strong>End Loop.</strong></td>
</tr>
<tr>
<td><strong>Build K-D-Tree:</strong> (LQT)ree = buildOptTree((\mathcal{Y}))</td>
</tr>
</tbody>
</table>

Figure 4.1: Training Stage Algorithm

\(^1\)In this stage noise characteristics are not considered, but they will be taken into account later
in reconstruction stage.
4.2 Database Structure: K-D tree

Since the algorithm performance is highly dependent on the efficiency of the search queries for examples stored in the database, its organization is a question of great importance. In this work we chose to use the K-D-Tree algorithm [20], which organizes the database off-line to enable a fast nearest neighbor search, by defining an optimal binary tree of thresholds on the input coordinates. The low-quality examples are represented as one-dimensional vectors (by raster scan, for example). The tree is built in the following way: considering the low-quality patches as vectors, the spreadest entry in this dataset is found. The examples are split into two equal groups, by a hyper-plane orthogonal to the chosen dimension and crossing it in its median value, calculated over the examples. This process is repeated recursively, each time splitting the spreadest dimension.

For a dataset containing \(T_s\) examples, each of dimension \(n^2\), this pre-organization requires an \(O(n^2 \cdot T_s \log T_s)\) in computations and \(O(T_s)\) in memory. The thresholds in this algorithm are chosen optimally so as to expedite the search, and indeed, the K-D tree algorithm leads to an \(O\{\log T_s\}\) expected number of distance evaluations in the quest for any predetermined number of the closest neighbors. By choosing a large number of neighbors, we guarantee to find all the relevant ones, satisfying inequality (3.6).

This functionality is implemented in two main procedures:

- \(KDTree = buildOptTree(Y)\)
  
  - **Input:** \(Y\) – two dimensional array, containing data vectors (in our case it will be of size \(n^2 \times T_s\)).
  
  - **Output:** \(KDTree\) – the tree data structure for further accessing.

- \(Y_i = kdOptNNQuery(KDTree, y^p, J)\)
  
  - **Input:** \(KDTree\) – the tree data structure; \(y^p\) – queried vector of size \(n^2 \times 1\); \(J\) - number of nearest neighbors to find\(^2\).

\(^2\)Instead of \(J\), threshold of error could be supplied, and then all relevant examples are brought, with \(J\) a varying depending on the example given and the database content. As this option was not implemented in this work, we restrict the discussion to the constant \(J\) per pixel.
– **Output:** $\mathcal{Y}_t$ – the found nearest neighbors.

## 4.3 Searching Stage

For a given low-quality image we run the routine **Searching Stage** in Figure 4.2, as it was previously described in Section 3.2. Since the high-resolution patches are gathered at every pixel location in the degraded image (full overlapping), their corresponding footprints on the high-quality canvas will be centered at one out of every $d$ pixels both for rows and columns ($d$ is the down-sampling ratio) - still with overlaps.

**SEARCHING STAGE**

**Input:** Degraded image $y$; K-D-Tree $LQT_{ree}$, containing degraded patches $\mathcal{Y}$; clean patches $\mathcal{X}$; number of nearest neighbors to fetch $J$.

**Output:** Cell array of found high-quality examples for each pixel $\theta$.

**Assign:** $\theta = \{\}$; $n = \text{dimension of vectors in } \mathcal{Y}$;

**Start Loop:** For each pixel location $p$ in image $y$,

- Extract low-quality patch $y^p$ of size $n \times n$ centered in $p$ from image $y$.
- Find $J$ nearest neighbors indices for $y^p$ in $LQT_{ree}$:
  
  $T_{y^p} = \text{kdOptNNQuery}(LQT_{ree}, y^p, J)$

- Store high-quality patches corresponding to the found indices:
  
  $\theta[p] = \mathcal{X}[T_{y^p}]$

**Stop Loop.**

Figure 4.2: Searching Stage Algorithm

## 4.4 Reconstruction Stage

As we stated earlier, the blurring kernel $B$ is of size $b \times b$ and the downscaling factor is $d$. Thus we can incorporate these two sequential degradations into a uniquely built matrix $H$ of size $m^2 \times n^2$. Recalling Haber and Tenorio formulation [22] and given the found examples, we plug them directly into an explicit, MAP-like looking penalty functional of the form
\[ g(x) = \|y - Hx\|_2^2 + \lambda \cdot A(x, \theta), \quad (4.2) \]

where

\[ \hat{x}_{MAP} = \text{Arg min}_x g(x). \quad (4.3) \]

Here the regularization expression \( A(x, \theta) \) is built according to the found examples (incorporated in the parameter \( \theta = \{\{x^p_k\}_{k \in \Omega[p]}\}_p \)). As such, this term is dependent on the training set \( S \) and the degraded image \( y \). Because \( A(x, \theta) \) not only estimates the log probability of the reconstructed image (the log of the prior), but also aware of the measure \( y \), this approach is not Baysian per-se. The regularization term we propose to use is given by

\[ A(x, \theta) = \sum_p \sum_{k \in \Omega[p]} \|Q (R^p \hat{x} - x^p_k)\|_2^2. \quad (4.4) \]

This expression is defined through the use of the examples found per each pixel location \( p \) of the degraded image. Since the reconstructed and degraded images differ in sizes due to the down-sampling ratio \( d \), the summation is done over the reconstructed image locations \( p \) with respect to the relative patch locations in the degraded source. The vectors \( x^p_k \) are the examples found in a previous stage. The operator \( R^p \) extracts a block of size \( m \times m \) pixels from the image \( \hat{x} \) that matches the footprint of the corresponding high-quality examples. So if the reconstructed image has size of \( r \times s \) pixels, the operator \( R^p \) is represented as a matrix of size \( m^2 \times r \cdot s \). The matrix \( Q \) is a diagonal weighting matrix, giving higher weight to pixels in the patch center. The inner summation is done over all found nearest neighbors, their indices are taken from the set \( \Omega[p] \). The outer summation runs through all pixels in the low-resolution image. According to its formulation, \( A(x, \theta) \) requires that the reconstructed image should locally agree with every found example in every pixel location.
A similar concept appears in [3], where a multi-scale derivatives are matched, rather than direct gray-values, as done here.

From the optimization point of view the minimization of $g(x)$ as defined above is simple, as $g(x)$ is a plain quadratic equation. The use of conjugate-gradient (CG) or even the steepest descent (SD) [21] iterative algorithms for its minimization leads to a fast solution, obtained after 10 iterations. More details on this solver are shown in Figure 4.3. More specifically, the Hessian and gradient of $g(x)$ is given respectively by

$$K = \nabla^2 g(x) = 2 \left( H^T H + \lambda \cdot \sum_p |\Omega[p]| (R^p)^T Q^T Q R^p \right)$$

(4.5)

and

$$\nabla g(x) = K x - 2 \left( y^T H + \lambda \cdot \sum_p \sum_{k \in \Omega[p]} (x^p_k)^T Q^T Q R^p \right)^T,$$

(4.6)

and thus the SD algorithm, for example, requires an update formula of the form

$$x_{i+1} = x_i - \mu_i \nabla g(x_i),$$

(4.7)

where

$$\mu_i = \frac{\nabla g(x_i)^T \nabla g(x_i)}{\nabla g(x_i)^T K \nabla g(x_i)}.$$  

(4.8)

The CG algorithm differs from the above by simply holding two last gradient residuals, and by faster convergence (see [21] for more details).

Note that in the actual implementation of the CG algorithm, we do not need to explicitly use the matrices $H$ and $R^p$, but rather apply them and their adjoints as simple operations on images.
RECONSTRUCTION STAGE

**Input:** Degradation operator $H$; degraded image $y$; regularization parameter $\lambda$; cell array of found high-quality examples $\theta$; weighting matrix $Q$.

**Output:** Reconstructed image $\tilde{x}$; value at the solution, $g(\tilde{x})$.

**Assign:** $x = \text{mean}(\theta)$ \{similar to scalar MMSE estimate in Section 3.2\}

Initial direction and error: $v = \epsilon_{curr} = \nabla g(\tilde{x})$ – according to Equation (4.6)

**Define:** $K = \nabla^2 g(\tilde{x})$ – according to Equation (4.5)

**Start Loop:** While $\|\nabla g(\tilde{x})\|_2 > 10^{-6}$,

- Calculate image update coefficient: $\alpha = \frac{v^T \epsilon_{curr}}{\epsilon_{prev}^T K v}$. \{Note: $K$ is never calculated explicitly, but rather $K v$ is acquired by simple operations on $v$\}

- Update image: $\tilde{x} = \tilde{x} + \alpha \cdot v$.

- Update gradient residual: $\epsilon_{prev} = \epsilon_{curr}$; $\epsilon_{curr} = \nabla g(\tilde{x})$.

- Calculate direction update coefficient: $\beta = \frac{v^T \epsilon_{prev}}{\epsilon_{prev}^T \epsilon_{prev}}$.

- Update direction: $v = \epsilon_{curr} + \beta \cdot v$.

**Stop Loop.**

**The Solution:** is given in the vector $\tilde{x}$.

---

**4.5 Defining A Weighting Matrix**

We now return to the issue of the patch sizes, and show how it affects the weighting matrix $Q$, which is part of our formulation. Figure 4.4 describes how the results (the testcase details are exactly as those described at the beginning of Section 5.1 in Simulation #1, where the image is degraded by a convolution with a 3-by-3 kernel followed by 2:1 decimation) are affected by the change of the patch sizes in the low-resolution ($n$) and the high-resolution ($m$). Per each choice of ($n; m$) the basic algorithm RECONSTRUCTION STAGE is run and the resulting MSE according to the original image is recorded. Instead of presenting the
Figure 4.4: Comparison between different high and low quality sizes. These results were validated over the test case shown in Figure (5.2). The value $n$ is the low quality patch size, $\text{MaxHQSize}$ stands for the maximal high quality patch size with respect to $n$ (based on Equation (4.1)). The graphs show the influence of the number of pixels used from the high quality patch on the overall MSE.

Performance as a function of $n$ and $m$, it is described as several curves parameterized by $n$, and as a function of the relative value of $m$. Each value of $n$ implies a maximal patch size in the high-resolution, which contains all the high-resolution pixels influencing the low-resolution patch. This size is defined as $\text{MaxHQSize}$. Since we initially fixed the value of $m$ in Equation (4.1), we can emulate its smaller values by building the weighting matrix $Q$ containing on its diagonal weight ‘1’ for the corresponding pixels that belong to the high-res. patch of size $\tilde{m} \times \tilde{m}$ ($\tilde{m} < m$) and ‘0’ otherwise.

As can be seen from Figure 4.4, a patch size close to the maximal one should be preferred. We also see that a slightly smaller size is performing better. This outcome is simple and can be explained intuitively, since the greyscale values of the pixels that reside on the high-res. patch borders have relatively small contribution to the greyscale values of the pixels in the resulting low-res. patch. For example, assuming a uniform blurring kernel of size $b \times b$ and no downsampling, the pixel in the center of the high-res. patch will contribute equally to $b^2$
different pixels in the low-res. patch, whereas the pixel in the corner of the high-res. patch will contribute only to a single pixel in the degraded patch.

The above leads to the proposal to use a weighting matrix $Q$ that assigns a weight per each pixel in the example patch based on its overall contribution in the degraded image. We propose to recover this weighting matrix in a reverse order: for every location in the low-quality patch of size $n \times n$ we determine which locations in the high-quality patch contributed to the examined pixels, and what were their weights. Those weights are accumulated for every location in high-res. while sweeping over all locations in low-res.

### 4.6 Pruning Outliers Stage

The above-described process is sensitive to outliers, and indeed, among the many examples found, some are irrelevant. Here is a simple experiment that illustrate this. Figure 4.5 (top row from the right) describes a low-resolution patch of size $5 \times 5$ taken from a degraded text image (the degradation includes a 3-by-3 blur, a 2:1 decimation, and finally additive white Gaussian noise with $\sigma = 8$). Figure 4.5 (top row from the left) also presents the original high-resolution corresponding patch of size $11 \times 11$ with a rectangle selecting the internal part of the patch of size $9 \times 9$. Searching for the nearest neighbors of the low-resolution patch in a database with 197,000 examples, taken from a similarly scanned printed page, Figure 4.5 shows the high-resolution patches of size $9 \times 9$ corresponding to the closest 50 low-quality patches found (the low-resolution ones are not shown, since they are not used in the reconstruction process, and so are out of our interest, except for their proximity to the supplied degraded sample). All are well within the required distance to assure a proper proximity (in the low-resolution domain). However, when computing the root-mean-squared-error (RMSE) between the chosen high resolution patches and the original content (that is available for us in this case) we see that many of the chosen examples are outliers with irrelevant content and there is no direct correlation between high-res. and low-res. RMSEs.

The remedy to the above-described outliers problem is to exploit the coherence we
Figure 4.5: Outliers among valid examples. Top: The high quality image (left), and the corresponding measurements (right). Both $11 \times 11$ and $9 \times 9$ blocks are marked. Bottom: the 50 nearest neighbors found, their RMSE in the low-resolution and the high-resolution ($9 \times 9$) domains. As can be seen, while all examples are close in the low-resolution, many of them are in fact outliers.

expect to have between overlapping high-quality patches. However, in order to exploit this potential, we have to abandon the pixel-based methods. As said above, using a global penalty function that ties the examples to each other may help in addressing this problem. The method shown above use the found examples to define a global image regularization. This by itself is not sufficient for robustness against outliers. Thus, we use the emerging MAP penalty function to choose the problematic patches and prune them out.

The above-proposed penalty functional in Equation (4.2) is using the local examples in order to define a global regularization for the unknown image. However, unfortunately this is not enough. In order to get an intuition for this expression, when $\lambda \rightarrow \infty$, its minimization leads to the simple pixel-based averaging algorithm described earlier. Furthermore, for a general value of $\lambda$ and when considering the denoising problem (where $H = I$), the minimizing result is also a simple averaging, including the measurement at this pixel. While it is an improvement over the MMSE algorithm we had before, we have clearly failed to force spatial coherence between the patches, as desired. In fact, this also implies that the algorithm described in [3] has no robustness to outliers as well.
Some degree of outlier-resistance can be achieved by replacing the $\ell^2$ norm in the prior terms with an $\ell^1$ one. However, considering the denoising problem again, such change replaces the mean by a median, and for too many outliers as often happens, this method still fails. Furthermore, rather than discarding complete patches, upon discovering that they are misleading, the outliers will be handled on a pixel-by-pixel basis, which loses much of the existing potential.

The solution we propose is to assign a weight to every example, so that those examples ‘living in harmony’ with their surroundings are weighted high, while others are down-weighted. Thus, the alternative MAP penalty becomes

$$g(\hat{x}) = \|H\hat{x} - y\|^2_2 + \lambda \sum_p \sum_{k \in \Omega(p)} w_p^k \|Q(R_p^p \hat{x} - x_p^k)\|^2_2.$$

(4.9)

There are many ways to estimate or choose these weights. Indeed, the work in [19, 18] offers a Bayesian belief propagation as an attempt to prune the various examples. Here we shall concentrate on a simplified and yet very effective case, where the weights are binary: ‘0’ for a bad example and ‘1’ for a good one. One has to make sure, however, that not all the examples in a specific location get a zero weight, because then we may get a hole in our reconstruction. As we are about to show, the MAP functional itself will serve in evaluating these weights. Since a zero weight is never re-weighted back to ‘1’, the whole process may be considered as sequential pruning of examples from the initial set $\theta$.

The proposed algorithm starts addressing all found examples. In a sequential process, these examples are pruned one patch at a time. For the current set of examples, the minimizer of (4.9) is computed, and the value of $g(\hat{x})$ at the minimum serves as a reference value. Per each patch $x_p^k$, that is still active (e.g. exists in $\theta$), we compute the optimal output image minimizing the modified MAP function

$$g_p^k(x) = g(x) - \lambda \|Q(R_p^p \hat{x} - x_p^k)\|^2_2.$$

(4.10)

This MAP functional is simply the same one defined earlier, with the omission of one example. Clearly, the value of this penalty term is necessarily smaller than the reference
one. Among all these examined patches, we prune the one that gives the largest difference between the reference penalty value, and the modified MAP penalty value. We denote those differences as $\Delta_k^p$:

$$\Delta_k^p = \min_x g(x) - \min_x g_k^p(x).$$

(4.11)

For example, if we try to reconstruct a noisy originally pure white image, with learned database containing only white patches, then $\Delta_k^p$ will be zero for all found examples. In the pruning scheme, the patch discarded is considered to be the least compatible with the remaining patches. The whole algorithm is shown in Figure 4.6, where subroutine buildReconstructedImage denotes direct execution of algorithm defined in Figure 4.3 as Reconstruction Stage.

A side benefit of this process is that we obtain a sequence of output images, one after each pruning step. Thus, beyond the first step that computes the optimal output image globally, all remaining steps are local and of low-complexity. As the pruning process proceeds, the value of the MAP penalty in (4.9) is consistently decreasing. An efficient stopping rule for this process is the dynamic range found in the set $\Delta_k^p$ – we consider the ratio between the maximal value of $\Delta_k^p$ to its median:

$$ratio = \frac{\max_{k,p} \Delta_k^p}{\text{median}_{k,p} \Delta_k^p}$$

(4.12)

When this ratio gets below $C$ (should be learned in some way on the similar images), all remaining patches are considered as positive contributors, and the algorithm is stopped. Alternatively, the removal of patches can be stopped when per each location $p$ we have one example remaining. Since the algorithm prunes sequentially patches from the found set, and since their number is finite, the proposed process necessarily stops at some point.
**Pruning Stage**

**Input:** Degradation operator \( H \); degraded image \( y \); regularization parameter \( \lambda \); cell array of found high-quality examples \( \theta \); weighting matrix \( Q \).

**Output:** Reconstructed image \( \hat{x} \).

**Start Loop:** while not reached stopping condition,

- \([\hat{x}, g] = \text{buildReconstructedImage}(H, y, \lambda, \theta, Q)\)
  
  **Start Loop:** For each pixel \( p \) in image \( y \) and each example \( x^p_k \) in cell \( \theta[p] \),

  - **Build new \( \theta \) without \( x^p_k \):**
    \[
    \theta_{\text{new}} = \theta, \quad \theta_{\text{new}}[p] = \theta_{\text{new}}[p] \cap x^p_k.
    \]

  - **Get new regularization value:**
    \[
    [\hat{x}_{\text{new}}, g_{\text{new}}] = \text{buildReconstructedImage}(H, y, \lambda, \theta_{\text{new}}, Q)
    \]

  - **Estimate regularization value change:**
    \[
    \Delta^p_k = g - g_{\text{new}}
    \]

  **End Loop.**

- **Find possible outlier location:)**
  \[
  [k_{\text{max}}, p_{\text{max}}] = \text{Arg max}_{k,p} \Delta^p_k
  \]

- **Remove the outlier from the examples set:**
  \[
  \theta[p] = \theta[p] \cap x^{p_{\text{max}}}_{k_{\text{max}}}
  \]

- **Reevaluate stopping condition.**

**End Loop.**

Figure 4.6: Pruning Stage Algorithm
4.7 Implementation Details

It is clear that the algorithm, described in Figure 4.6, implies a computationally heavy process. For a degraded image with $P$ pixels and $J$ candidate patches per location, the complexity of the reconstruction stage (Figure 4.3) is governed by the calculation of $\nabla g(\mathbf{x})$, which requires $O(JP)$ operations at the least. The later stage of pruning, as described in Figure 4.6 requires reevaluation of $\Delta^p_k$ over all pixel locations in the degraded image support. Thus each pruning iteration has complexity of $O(J^2P^2)$.

This high complexity is caused by an implicit assumption that each $\Delta^p_k$ is strongly influenced by all candidate patches found for all pixel locations, so they all have to be considered in $\Delta^p_k$ calculation. Also, due to this influence, removal of an arbitrary example patch requires an update of $\Delta^p_k$ for all pixel locations and all the found examples therein. However, as we have empirically verified, the effect of a removed patch on the reconstructed image is local, and exponentially decreasing outside its support. Thus, instead of calculating $\Delta^p_k$ using all remaining candidate patches, we restrict the algorithm to consider only the patches in some predefined vicinity of pixel $p$. Along with that, after some candidate patch at pixel $q$ is pruned, we do not need to update $\Delta^p_k$ in all pixel locations anymore, but rather recalculate $\Delta^p_k$ in the very same vicinity of pixel $q$. The size of this local area is dependent both on the downsampling ratio and the size of blur kernel.

Exploiting the above insight, each pruning step has complexity $O(J^2)$ and is independent of the degraded image size. The reconstructed image is updated piece-wise, according to the influence of the removed candidate patch.

Beyond the above obvious computational gain, we should note that since the removal of a single example patch at a time is not expected to influence the output image by much, an iterative solution based on the conjugate direction and an initialization with the previously obtained image are found to lead to a rapid convergence (2-5 iterations), further reducing the required computations.
Chapter 5

Simulation Results

We demonstrate the various reconstruction methods discussed above by showing the results on several simulations involving scanned documents with different content types – text, equations, graphics, and then show how it is extendable with some modifications to face images.

5.1 Examples on Scanned Documents

The above algorithm has been tested on scanned documents, where the use of raw gray values in representing the examples seems to perform the best [9]. Here we provide some examples to illustrate the behavior of the algorithm on text images, drawings, and equations.

The first two simulations involve a text image. In these text simulations we used $m = 11$ (high-quality patch size) and $n = 5$ (low quality patch size). The degradation operator used is a 2D low-pass separable 3-tap blur $[0.25, 0.5, 0.25]$, a scale factor of 2, and an additive white Gaussian noise with $\sigma = 8$.

The image shown in Figure 5.1 was used for extracting nearest-neighbor examples. Patches of size $11 \times 11$ taken from this image form the example database for Simulation #1 - text image reconstruction. For the reconstruction, a weighting matrix $Q$ in Equation (4.2) is built in order to give the equal treatment to the pixel block of $9 \times 9$ ($\tilde{m} = 9$) and to ignore all border pixels (practically reducing the referenced high-quality patches size to
9 × 9). Overall, there are $T_s = 197,000$ examples in the DB.

We see that the robust optimal solution we have built “costs money” – it promises a profit of just $8,295 (cf. with the profit of $8,826 promised by the nominal optimal solution). Note, however, that the robust optimal solution remains feasible whatever are the realizations of the uncertain data from the uncertainty set in question, while the nominal optimal solution requires adjusting to these data and, with this adjusting, results in the average profit of $7,843, which is by 5.4% less than the profit $8,295 guaranteed by the robust optimal solution.

Note also that the robust optimal solution is significantly different from the nominal one: both solutions prescribe to produce the same drug DrugI (in the amounts 17,467 and 17,552 packs, respectively), but from different raw materials, RawI in the case of the robust solution and RawII in the case of the nominal one. The reason is that although the price per unit of the active agent for RawII is slightly less than for RawI, the content of the agent in RawI is more stable, so that when possible fluctuations of the contents are taken into the account, RawI turns out to be more profitable than RawII.

In some cases, it is natural to assume that the perturbations affecting different uncertain data entries are random and independent of each other. In these cases, the Robust Counterpart based on the interval model of uncertainty seems to be “too conservative” – why should we expect that all the data will be simultaneously driven to their “most unfavorable” values and immune the solution against this highly unlikely situation? A less conservative approach is offered by the ellipsoidal model of uncertainty. To motivate this model, let us look what happens with a particular linear constraint.

Figure 5.1: Example database for Simulation #1 - text image reconstruction.

Figure 5.2 shows several images referring to a test done on a portion of a text image. This figure shows the original high-resolution image, the degraded one, and several reconstruction results. As can be seen, the pixel-based MMSE. When using the proposed global scheme, the initial result should be similar to the pixel-based MMSE one, but due to the introduction of $\lambda = 1e-3$ (chosen manually), it is somewhat better. As pruning takes place, the result improves substantially, due to the removal of 940 outlier examples out of the original 15,000 patches.

The stopping rule used here is the one described above (testing the dynamic range of $\Delta_k^p$ with threshold $C = 0.25$). Figure 5.3 presents the reconstruction RMSE as a function of the pruning stages. Overlaid on this graph is the stopping rule curve, showing the dynamic range of $\Delta_k^p$ as a function of the pruning stages, and the threshold it meets in the stopping rule. As can be seen, while not perfect, the proposed stopping rule does succeed...
in stopping the algorithm in the vicinity of the best MSE.

![Images of original, degraded, bi-cubic interpolation, MMSE reconstruction, and reconstruction after pruning images](image)

(a) Original image of size $35 \times 251$
(b) Degraded image
c (c) Bi-cubic interpolation result (MSE=1720.9)
(d) MMSE reconstruction (MSE=314.5)
(e) Reconstruction after 940 pruning iterations (MSE=258.3)

Figure 5.2: Simulation #1 - text image reconstruction results.

The images shown in Figure 5.4 were used for extracting nearest-neighbor examples for Simulation #2 - text image reconstruction. This time the weighting matrix $Q$ is setting pixel’s weights in high-quality patch proportionally to pixels contribution to the low-quality patch formation.

Figure 5.5 presents the original image, its degraded version, and its reconstruction results using bi-cubic interpolation, pixel based MMSE estimator, and the result after pruning. In this simulation we have used one example per location with full overlaps, implying that per every pixel we have 25 candidate values. We fixed $\lambda = 1.6e-2$, and performed 1090 pruning stages out of the overall 6,860 initial examples. In fact the values of stopping threshold $C$ and normalization coefficient can be learned on the provided example images by taking the

\footnote{Here and elsewhere, the number of pruning stages is governed by a stopping threshold.}
Figure 5.3: Simulation #1 – a text image: the pruning effect on the reconstruction MSE.

Figure 5.4: Example database for Simulation #2 – text image reconstruction.
best values of leave-one-out training runs.

Figure 5.5: Simulation #2 – a text image.

The next three simulations were performed on scanned documents containing graphics and drawings. Figure 5.6 gives the image from which examples have been drawn for Simulation #3 - formula image reconstruction. In this simulation there are \( T_s = 82,000 \) such examples. The degradation assumed in this simulation includes the same blur, followed by a scale-down factor of 3. Once again, low quality patch size is chosen to be \( n = 5 \), and according to Equation (4.2), high quality patch size is set to be \( m = 15 \).

The reconstruction results are shown in Figure 5.7. As before, the pruning algorithm leads to a better MSE result. Figure 5.8 presents the stopping rule, again successfully leading to near optimal MSE performance.

The fourth simulation shows reconstruction results related with an image containing a portion of a graph. Figure 5.9 shows the training image, Figure 5.10 shows the reconstruction results in this case, and Figure 5.11 demonstrate the stopping rule behavior. Overall,
there are $T_s = 110,000$ examples in the DB for Simulation #4. The degradation details are virtually the same as in Simulation #3. The results are generally the same as in previous simulations, showing the strength of the pruning technique proposed.

Figures 5.12 and 5.13 show similar training information and reconstruction results for the simulation that considers a drawn cartoon. In this simulation the noise power was chosen to be $\sigma = 2$, and $\lambda = 1e - 3$. Patches of size $11 \times 11$ taken from the four images in Figure 5.12 and rotated version of them in 5deg (full circle) form the example database for Simulation #5 of drawn cartoon image reconstruction. Overall, there are $T_s \approx 3e + 6$ examples in the DB for Simulation #5.

We conclude this results sub-section with a simulation that returns to the text content images, and the database as defined for Simulation #1. Our aim is to show that for the content dealt with here, the best approach is the use of the raw information and not high/medium frequencies, as often deployed on natural images. Figure 5.14 shows how the training is done, when aiming to use a pre-process of removing the low-frequencies. The low-pass filter we have used here is a simple $5 \times 5$ Gaussian filter of width $\sigma = 1.5$. Figure 5.15 presents the way this database is deployed to the actual reconstruction. Figure 5.16 shows the reconstruction results with a low-pass filter and without it (our method), for a
noisy case and noiseless one. As can be seen, while all results are good, better performance is obtained using the plain raw data, as we recommend.

Figure 5.7: Simulation #3 – an image with formula.

\[ b_i(x_i) = \prod_{k} m_{ki}(x_i) \phi_i(x_i,y_i) \]
Figure 5.8: Simulation #3 – an image with formula: the pruning effect on the reconstruction MSE.

Figure 5.9: Example database for Simulation #4 – lines image reconstruction.
(a) Original image of size $69 \times 291$

(b) Degraded image

(c) Bi-cubic interpolation result (MSE=1.4972)

(d) MMSE reconstruction (MSE=312.2)

(e) Reconstruction after 1100 pruning iterations (MSE=263.4)

Figure 5.10: Simulation #4 - lines image reconstruction results.
Figure 5.11: Simulation #4 – a portion of a graph: the pruning effect on the reconstruction MSE.

Figure 5.12: Example database for Simulation #5 – a drawing image.
Figure 5.13: Simulation #5 – a drawing image. (a) Original image of size 119 × 69; (b) Degraded image; (c) Bi-cubic interpolation result (MSE=1118.9); (d) MMSE reconstruction (MSE=463.2); (e) Reconstruction after 2880 pruning iterations (MSE=366.0).

Figure 5.14: A block diagram of the training algorithm when using the higher frequencies as features.
Figure 5.15: A block diagram of the reconstruction algorithm using the high-frequencies features.

(a) Noise ($\sigma = 8$), using high frequencies (MSE=406).
(b) Noise ($\sigma = 8$), using raw data (MSE=367).
(c) No noise, using high frequencies (MSE=354).
(d) No noise, using raw data (MSE=322).

Figure 5.16: Comparison between an algorithm that operates on raw data and one that uses high frequencies only. Scale factor is 2.
5.2 Treatment of Face Images

When turning to handle face images, a change is needed in the way the examples are fitted. Using the raw gray-values directly leads to no feasible neighbors in many cases, as the diversity of the image patches is large. Following the idea of using high-derivatives as promoted in [3, 19, 18], we consider the image patches with their mean removed. This means that in the database construction stage, the pairs \( \{ y_t, x_t \}_{t=1}^{T_s} \) are gathered without their mean, i.e. \( 1^T y_t = 0 \) and \( 1^T x_t = 0 \) for all \( t \). Given the measured low quality patch \( y^p \), its mean

\[
d^p = \frac{1}{n^2} 1^T y^p
\]

is also removed and kept aside for later use. The nearest-neighbors are found as before with the mean-less patches, and we obtain the sub-set of high-quality patches \( X^p \) as before. Those are used in our formulation in Equation (4.9), with a constant \( d^p \) added to them.

The above method resembles the example-search technique proposed in [19, 18], but there are few differences. Their method suggests an application of a low-pass filter on the measured image \( y \) to obtain the low-frequencies of the reconstruction \( y_{LPF} \). This image is up-sampled by a plain (e.g., bi-cubic) interpolation to the high-resolution canvas, and serves as the low-frequencies in the destination image. The reconstruction process is applied on the residual, by interpolating also \( y - y_{LPF} \) to the higher grid, and fitting examples to patches in this image. Thus, their method requires more computations, as it works on a larger image and larger patch sizes.

We tested the above-described algorithm on the ORL face image database, which contains 400 images of 40 people, as shown in Figure 5.17.

As before, the results shown in the tests include a bi-cubic interpolation, the MMSE (averaging the gray-values of the found examples per each location), and the pruned result. The tests reported here assume a Gaussian blur of width 5 × 5 pixels with \( \sigma = 1.5 \), a downscaling factor of 3:1, and an additive noise with \( \sigma = 2 \) gray values. The reconstruction is based on 10 nearest-neighbors per pixel, using \( \lambda = 2e - 4 \), and patch sizes \( n = 4 \) and...
Figure 5.17: The ORL face database, containing 40 different people, each with 10 images of size $112 \times 92$ pixels.
In the example-based reconstruction tests we tested 3 options: (i) using all 399 remaining faces; (ii) using the 9 images of the same person, and; (iii) using 390 images of all other people in the database, excluding the same person. The results for two different people (chosen in random) in the database are shown in Figures 5.18 and 5.19, and support the need for pruning as proposed here. We can also see that when reconstructing the image based on the same person’s images, the results are the best, but deteriorate only slightly when using the entire database of faces.
Figure 5.18: Face #1 results. (a) Original image; (b) Degraded image; (c) Bicubic interpolation result (MSE=149.64); (d) MMSE reconstruction (MSE=75.4) based on 9 images; (e) MMSE reconstruction (MSE=97.1) based on 390 images; (f) MMSE reconstruction (MSE=79.9) based on 399 images; (g) Pruned (1970 steps) reconstruction (MSE=62.38) based on 9 images; (h) Pruned (3060 steps) reconstruction (MSE=88.01) based on 390 images; (i) Pruned (2030 steps) reconstruction (MSE=72.71) based on 399 images.
Figure 5.19: Face #2 results. a) Original image; (b) Degraded image; (c) Bi-cubic interpolation result (MSE=159.25); (d) MMSE reconstruction (MSE=65.24) based on 9 images; (e) MMSE reconstruction (MSE=102.97) based on 390 images; (f) MMSE reconstruction (MSE=68.90) based on 399 images; (g) Pruned (2280 steps) reconstruction (MSE=60.08) based on 9 images; (h) Pruned (5320 steps) reconstruction (MSE=92.66) based on 390 images; (i) Pruned (3780 steps) reconstruction (MSE=65.61) based on 399 images.
Examples can be used for getting an effective regularization in inverse problems involving images. This is especially true for specific type of images, such as faces or scanned documents, as demonstrated in this work. The use of examples provides a step forward in reconstruction quality, compared to the classic priors or regularizations that have been proposed in the past decades. There are several ways to exploit examples in inverse problems: three techniques have been described in this work – learning parameters of the regularization expression, a direct use of the examples in forming the posterior, or in merging these techniques. This work has described a specific method that belongs to the later family of algorithms, focusing on the need to prune outlier examples to fine-tune the outcome.

In deploying the above set of ideas to inverse problems, there are many open questions that are yet to be addressed. Here we list few of those:

- **Real super-resolution experiment?** The results provided in this thesis are simulated with artificially degraded and restored images in order to measure restoration success by means of MSE. The next step should be an experiment with initially degraded data that was not synthesized from clear image.

- **Blind restoration?** The previous question raises another issue: when the data is provided without any characteristics of degradation process, what should be changed in the algorithm to cope with such cases? How sensitive is the algorithm towards
using database trained on different blurring kernel than the one that was applied in degradation process?

• Multiscale treatment? It seems natural to consider a multiscale method that considers image patches of varying sizes in seeking fitting examples. One way to implement this idea is by choosing the maximal size that gives sufficient number of examples. However, a fast nearest-neighbor algorithm that can cope with varying patch sizes should be devised.

• Theoretical foundations? While all the above discussion makes a lot of sense and seems intuitive, using examples should be strengthen by a supporting theoretical study. No such study has been proposed as of yet.

• How big should the DB be? This question is tightly coupled with the previous one. We are sampling a specific distribution, and we need sufficient examples so as to claim reasonable proximity to every instance that can be encountered. Thus, the richness (or entropy) of the image distribution should be taken into account in gathering the database.

• Regularization or prior? We have seen examples used both as a way to drive a prior (i.e. practice a pure Bayesian approach), or for forming a measurement-dependent regularization expression. It is unclear at all which of the two techniques is better, and how these two methods could be compared.

• What about general content images? When the inverse problem deals with a general content image, the amount of examples should grow dramatically, and perhaps leading to the point of requiring an impractical algorithm. Are example-based techniques at all fitted for handing general images?

These and many more related questions will be probably studied by researchers in the coming years. The potential of examples in handling inverse problems better is unquestionable, and as such, the interest in this field is expected to grow.
Bibliography


65


66


67


מעבר להיעשות האופייניים יותר-thankees זה. הערת נכסות זול, או סכום ש.LinkedList
للעובדה נספתי על המנהל לפרט דרכו לטפח החוד ושפרני צורתה הוא. ניואט- multi-
מרתפת המסדה והקבלת הלוביצן שיפור נכס. לעד שיפורים מעשיים אטרקטיביים. שייך scale
לدنيי התרבותי משאלות ייחודיים công עד הקמתו המהפכים, עותי.
הכימיות והשכיות לשתי המשפות של הוחזרו עלי הממציאים עלי חלקי כופר פציונרי.

האלגוריתמים המוצעים מחולק לארבעה שלבים עיקריים: איסוף הדוגמאות להתוכנות נקיה, מלקטון_transform את ל, ובנוסף

העפרונות החeterminate. אנשי הדומע והדומע這裡 לשני ממצאים של שני פסים תומכות, כל כנס

כל פיסת תמונתacroית פסיכולוגית. השלב הזה מעבירהفعה את הפות הת合わות השחורה.

בנוסף תמונות פוטו. התחל השחורה מצורה בסיסית לה לפתור התוכנות הפוטו 맞א מתכון הדומעספריאות אשת המשך לא יותר.

הלוח ההתרבות 시행ב (לOutOfBounds ישיא '-') פסיפס התוכנות הסיכובט חפם.

הכיס המתורן. התואמותprises במתן התוכנות הפוטו, ולא הדומעStoryboard ישאר נגזרה.

הטיה האלק התואמות לחמצן הגבורה. כל הדומעStoryboard מסיום והמשתקפת לחדר ביב

טי המתחק את פונקציהיןAstrovit הפורל של התוכנות השחרורית. הביסים מעقضيית חלק

הכלול מתבגרים ייח הליצור פונקציה כאשר MAP עלבולעה. פונקציה היא מהווה את השר ביבית.

משחררי את התחרי השחורה בשני איפנים. ראשית, עי מישורונות גים כל התואמות השחרורית שוור יתי ש/ioutilה בניסים מבוקרים, לא כל הדומעStoryboard באיתון הזרה. שטמא בוש.

בל החרית, חסרandraות למדים של מוקור התוכנות הפוטו, לפי יש צור יוכל תוכס - שלב

שני הדומעStoryboard כלא לותריות. הפונקציהımız מצויה מדויק אספány של כל התואמות השחרורית שפרמאשר כי מאפיית סין על מדת שכר את אייאת השחרור הממד המכחיב באפ סוחה

עי מוחות הסכמה של כל דומע العام השכינכם של התוכנות הפוטו.

משחררי של האלגוריתמים המוצע איה.Linear מתר CONTRACTION של כל אתת משתי המשפות תשלומית

של שיטוטי הגרוזיצים ממסכים הדומעStoryboard המפורץ ליגי, כל אחרעל מתכת חסין פתרון

שהיה אופטימלי בג מבחן אנליטי ושקבמחת ויוראליט. יובא את התואמות מצא Hancock

עביר המדריך يستתור תشاب, עבירה והתקפת באת אcação הסבר-יווהיזג הוב

ונוחת המוחה בודד ממקמה מטריקט שלוש תמונות מונדלות, מתר�폴ג גולש, או

מגדים את התואמות האלגוריתמים על סקטס סקר, ציור, ורטוטו למידות פמי כל-

אלו, כל שיש מסייבה התואמות ברמות איקサイト בᠣ.Collectors.

הרייך הבסיסי של שיטוטי הדומעStoryboardبعית בשחרר שלニックו ובכזר חלבעיתינא
בעדות של מיכל אהרן ומיכל אָלְעָי מוחלטת האחתה ב-27 וב היא שהיכת למספה
ו היא יишא העולים ה- Prior-ה荞מאט הלייטמ ב.SelectSingleNode
של התמונה הקטן ליצוגים דילים ויתרъה המימונים יוש באלל לבל כפרמטאוכי

המשפה השניה של蕖 מתארת השינה שתגעה דודגמואת מכילה שיתפוה שאמפשאתה
התרחבה במטרה עלה ריצי ספורט פילופ התומנה להשתתפתי בודגמואת השיר
בותהיל השזור התומנה. בקומה התומנה פעמים פעימת פילופ גז_sibling של תמונת טאיה
היתוך בתקף הקלקול. בנקים פיקניזת פילופ גזSibling של תמונת מושחותר לתליסות
נפרשת ביגוליריציה. חיישות האלה בודר כלש טגית לבגון פרתור ברגר טקמיית
וליפ פיסת קטות עלה התמונה הפונה. תקריל הליד גנישת אזות אופטימי של
הتكنונמקימיות בתמונת הדגייה למוהר עיירה הקدخول. לפתחיו עצה על שטהות שניים
את סידוריה העבורה התומנהottie ב-1,30,31,32.

ובזכות אט מאיתיל הבקסירה של תמונת קימוט בחזרה והתנישות חזרה ללקויים
המשתתפיים בטח. חוסר יזה פופר, המשפחתי התפשותיות של התศูนย์טוס סובל בברדכ
ծמוכובלאי ייכלד תלים אצ התשע התומן בנסף התומנה. חס שלבשבוע טביי מוביל
לידיה ביצועים. חיישות יישר עדגמואת. לעמות גא. מופיעה בר לאל מוך
לאיכות הפרזה, דר הפרצר בניק ס çözümוי וורlèveוי.

בנוסף היא אנשי הפסים טסים עילית לימויה בודגמואת הקלקוח לתמונת יצירתי
גזרותיניות עילית המזימה עוצר המספר בין מיבת הקלאסיות בין מיבת התמונה צומת: התרד
ירשות החזרת ישירה לעלות תמונת. הגישה המופיעה משכבה האלמנטים מתואריהם.
Tzóir

The ill-posed problem of matching an image to a given meshed model may or may not have a solution, depending on the algorithm and the model.

The matching process engages a mathematical model of the meshed object. The matching process is divided into two main stages: (i) matching the meshed object to the model, and (ii) matching the model to the meshed object. The first stage is performed by the meshing algorithm, and the second stage is performed by the matching algorithm. The matching algorithm uses a cost function to measure the similarity between the meshed object and the model. The cost function is a function of the meshed object and the model, and it is used to determine the best correspondence between the two.

The matching algorithm uses a cost function to measure the similarity between the meshed object and the model. The cost function is a function of the meshed object and the model, and it is used to determine the best correspondence between the two.
רשימת אורות

22 ............................................. 3.1
איור השפעות של פיקסלים משניים לפישת מקורות

31 .................................................. 4.1
אלגוריתם של שאב אימוץ

33 .................................................. 4.2
אלגוריתם של שאב הימורים

36 .................................................. 4.3
אלגוריתם של שאב שחרור

37 .................................................. 4.4
תשובה של גלגל שגיכו של פיקסלים לפיתוח وأكدת קהלר

39 .................................................. 4.5
דוגמאות ואתができる בקננים

42 .................................................. 4.6
אלגוריתם של שאב סינון

46 .................................................. 5.1
מאגר דוגמאות לניסוי 1 - שחרור של פיקסלים

47 .................................................. 5.2
ניסוי 1 - תユーザーות השחרור של פיקסלים

48 .................................................. 5.3
ניסוי 1 - פיקסלים: השפעה של פיקסלים על ייעוד השחרור

48 .................................................. 5.4
מאגר דוגמאות לניסוי 2 - שחרור של פיקסלים

49 .................................................. 5.5
ניסוי 2 - תユーザーות השחרור של פיקסלים

50 .................................................. 5.6
מאגר דוגמאות לניסוי 3 - שחרור של פיקסלים

51 .................................................. 5.7
ניסוי 3 - תUSER שחרור של פיקסלים

52 .................................................. 5.8
ניסוי 3 - نتيجة תโรค: השפעת סינון על שיגיון השחרור

52 .................................................. 5.9
מאגר דוגמאות לניסוי 4 - שחרור של פיקסלים

53 .................................................. 5.10
ניסוי 4 - תUSER שחרור של פיקסלים

54 .................................................. 5.11
尼斯י 4 - גרף תโรค: השפעת סינון על שיגיון השחרור

54 .................................................. 5.12
מאגר דוגמאות לניסוי 5 - שחרור של פיקסלים

<table>
<thead>
<tr>
<th>מספר</th>
<th>תקציר</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>ת agréמשה המרצות</td>
</tr>
<tr>
<td>4.1</td>
<td>שלב אימון</td>
</tr>
<tr>
<td>4.2</td>
<td>ממנה נתונים: K-D tree</td>
</tr>
<tr>
<td>4.3</td>
<td>שלב חיפוש</td>
</tr>
<tr>
<td>4.4</td>
<td>שלב שיוור</td>
</tr>
<tr>
<td>4.5</td>
<td>הדרת מציאות המופקות</td>
</tr>
<tr>
<td>4.6</td>
<td>שלב סיני</td>
</tr>
<tr>
<td>4.7</td>
<td>פרטיו מופשר</td>
</tr>
<tr>
<td>5</td>
<td>תוצאות של ניתוחים</td>
</tr>
<tr>
<td>5.1</td>
<td>תוצאות של מסמכים אחרים</td>
</tr>
<tr>
<td>5.2</td>
<td>סיכום התמונות של五一 אדם</td>
</tr>
<tr>
<td>6</td>
<td>מסמכים נוספים俞ית</td>
</tr>
<tr>
<td>65</td>
<td>תומר יחומים</td>
</tr>
</tbody>
</table>
تكلم לבנגלית

הדורות הספורים

مبادئ 1

1.1 הור desar ברגליציה
1.2 שיטה דרגליציה
1.3 שしようと דוגמאות לזר לדרגליציה
1.4 תורת✅ על העבורה האואית
1.5 המבנה של העבורה

רקט על הרגליות 2

2.1 Maximum-Likelihood זכרון
2.2 הוספת כלים ביסטרים הרגליות
2.3 ההתקוות של פוכנציאט פילוג האופטימאל של המגון

שימור בדגם - סקר על שיטות 3

3.1ernes ברגליציה של פוכנציאט פילוג האופטימאל של המגון כשל המגון
3.2 זוגראות.
3.2 זוגראות.
3.3 בנימ הרגליות על סְּמִך זוגראות.
המחק הבנוי בהגותי ד"ר מקראโต אלעיג בפקולטה למדעי המחשב.

ברקע של זהות של הכתיבת המשורר מעלה מגוון של שמלודות ומוחלט העצמות.

המודה המוחודה לאישה קטינה על אוהבהصادות.

אני מודה לבעניין על התפונה הכספית והריבעה בשכנות.
רגולריות מבוססת רגרמאות宓יעות הפך

תפקיד על מבקר

לשם מילוי חלקי של הדירישות לקבלת הňוגור

מוניטור שלמדים במדעי המחשב

דימהר דגנוב

חוגם לסיים פעמיים - מאן טכנולוגיות לישראל

תשיי תשס’ו

אוקטובר 2006
רגולריות מבוססת דוגמאות בקצוף

דמתיי דנוק