HES-SO // MASTER
Switzerland

Thesis for the degree of
MASTER OF SCIENCE IN ENGINEERING
with orientation in
INDUSTRIAL TECHNOLOGIES

A Fast SRR Algorithm Based on Recursive Least Square Estimation and Simultaneous Image Registration

by
Stéphane Kirchner

Under direction of
Prof. Dr. Stephan Robert (HEIG-VD, Switzerland)

Members of the jury:

Prof. Dr. Stephan Robert (HEIG-VD, Switzerland)
Prof. Dr. Hervé Dedieu (HEIG-VD, Switzerland)
Asst. Prof. Dr. Vorapoj Patanavijit (AU, Thailand)
Asst. Prof. Dr. Kittiphan Techakittiroj (AU, Thailand)

Project realised at the Department of Computer and Network Engineering,
Faculty of Engineering
at Assumption University (AU), Bangkok, Thailand
under supervision of Asst. Prof. Dr. Vorapoj Patanavijit

August 4, 2011

Contents

List of Figures

List of Tables

Abstract

A Fast SRR Algorithm Based on
Recursive Least Square Estimation and Simultaneous Image Registration
by
Stéphane Kirchner

This thesis proposed a Super Resolution Reconstruction (SRR) algorithm based on recursive least square estimation and simultaneous image registration. This recursive least square estimation algorithm is computationally fast and effective, and the simultaneous image registration algorithm is more efficient for real practical use. So this thesis will try to present the advantages by combining these two algorithms into a single framework.

The thesis is structured into five main parts:

- The first chapter introduces the super resolution reconstruction (SRR).
- Later, the chapter 2 examines the super resolution reconstruction algorithm using the stochastic regularization approach.
- The chapter 3 reviews and examines three SRR algorithms using the stochastic regularization approach (Classical SRR algorithm, Fast SRR algorithm based on Recursive Least Square and SRR algorithm based on Simultaneous Image Registration) and also the **proposed algorithm** based on recursive least square estimation and simultaneous image registration.
- The chapter 4 gives some experimental comparisons among different algorithms.
- And finally, the chapter 5 will give the conclusion and the future works directions.

Acknowledgements

To begin, I would like to thank Prof. Dr. Stephan Robert from HEIG-VD (Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud, Switzerland) for offering me this opportunity: to do my Master's thesis at Assumption University in Bangkok, Thailand. I also thank him for the interest he has kept all along my work, even away from thousands of miles.

My thanks also to Ms. Sylvie Villa, the person responsible of the field "Engineering and Architecture" at the HES-SO Master, Switzerland. She accepted my application to go made my thesis at Assumption University in Bangkok, even if I gave my application after the deadline.

I also want to thank Asst. Prof. Dr. Kittiphan Techakittiroj who was the link at Assumption University (Department of Mechatronics Engineering) and has also made possible the exchange. In addition, it also looked through my work's progress.

I thank the Assumption University and its faculty of engineering, for giving me this place of work in the "High Speed Communication Network Laboratory" for the thesis.

My thanks also go to Mr. Glen Chatelier, director of the Office of International Affairs at Assumption University, which also helped me find an apartment near the workplace.

I also thank Pham Hong Ha, Ph.D.c (Ph.D. candidate) working in the same laboratory, for helping me on some issues discovered during my work. He also gave me many advices for the life in Bangkok.

Finally, I want really to express my gratitude to Asst. Prof. Dr. Vorapoj Patanavijit, who was my supervisor during this work, for offering me this place (he work in the Department of Computer and Network Engineering at Assumption University). He is also the one who had the idea to combine these two algorithms into a single framework. He helped me greatly throughout the work at "High Speed Communication Network Laboratory" and he always had time to devote to my questions. Moreover, his usual good humor made the work more than pleasant.

Stéphane Kirchner

Bangkok

August 4, 2011

Chapter 1

Introduction

There is great demand for high resolution images, and it continues to grow. It is perfectly understandable why an image of high resolution (HR) is preferable to an image of low resolution (LR) for a doctor who looks at a radiography. And the fields are varied, from the visual pleasure of a HR image created by a private camera to the HR images given by a drone where the best resolution can provide details that may be crucial in choosing an intervention of the army.

The easiest way to increase the resolution is either to increase the pixels number of the sensor, or to increase the pixel density on the sensor. In the first case, increasing the sensor size causes an increase in capacitance (electronic capacitance of the sensor) and therefore decrease the speed of transfer. Moreover, this is contrary to the current policy of searching to miniaturize everything. For the second solution, we have already reaches its limits. Indeed, by decreasing the size of the sensors, the light received decreases as well. This brings up shoot noise that reduces image quality. There is a limit of magnitude without too great noise. This optimal size has already been reached [1].

Consequently, alternative approach for increasing the resolution is an image reconstruction that is more useful today. The idea of the image reconstruction is that the acquisition of images can be seen as some degraded processes performed on an original high resolution image (motion, added blur, down-sampling and adding noise). Image reconstruction is to find the baseline image from images of poor quality (filtering noise, up-sampling, removing blur and inverse motion).

There are two different uses of SRR:

- Reconstruction of a HR image from a LR image.
- Reconstruction of a HR image from multiple LR images of the same scene (frames).

1.1 Objective

This thesis proposed a Super Resolution Reconstruction (SRR) algorithm based on recursive least square estimation and simultaneous image registration. This recursive least square estimation algorithm is computationally fast and effective, and the simultaneous image registration algorithm is more efficient for real practical use. So this thesis will try to present the advantages by combining these two algorithms into a single framework.

1.2 Thesis structure & scope

In this thesis, several image reconstruction algorithms using stochastic regularization approach are reviewed. This thesis is not meant to use an exhaustive list of the completed literature on image restoration, but take several of the important methods [2] [3]. It will describe each algorithm (and the proposed one) and find out the advantages and weak points. Each algorithm will be used with: "Laplacian Regularization", "MRF (Markov Random Field) Regularization" [4] and "BTV (Bilateral Total Variation) Regularization" [4] (each with L1 and L2 norms estimation). It will compare them with different cases of inputs (one or multi frames), different kind of images (different texture) and with different kind of added noises. This work will only uses grayscale images.

This thesis is not going to compare the use of the regularization functions or that of the norm estimations, but only give few examples of their uses. **The main purpose of the work is to compare the SRR algorithms.**

Structure:

- This first chapter introduces the super resolution reconstruction (SRR).
- Later, the chapter 2 examines the super resolution reconstruction algorithm using the stochastic regularization approach. This thesis will also present three different regularization functions (Laplacian, Markov Random Field and Bilateral Total Variation) and two norm estimations (L1 and L2) for it.
- The chapter 3 reviews and examines three SRR algorithms using the stochastic regularization approach (Classical SRR algorithm, Fast SRR algorithm based on Recursive Least Square and SRR algorithm based on Simultaneous Image Registration) and also the **proposed algorithm** based on recursive least square estimation and simultaneous image registration.
- The chapter 4 also give some experimental comparisons among different algorithms, the using regularization function and norm estimation choice. These comparisons use several cases (different noises, one input image or frames and the kind of texture of the image(s)).
- The chapter 5 gives the conclusion and the future works directions.

In Appendix A, a short presentation of the toolbox created for Matlab used in the experimental comparisons will be given, and how to use it.

1.3 Introduction of SRR

The super resolution reconstruction techniques create a high resolution (HR) image from a sequence of low resolution (LR) and noisy images (frames). This is only possible because these LR images are aliased (they do not respect the Shannon theorem and are sampled under the Nyquist rate [5]). Details of the real scene are available in small parts in each LR frames. This thesis will work with the method of super resolution reconstruction using stochastic regularization approach. This method work in spatial domain and not as some methods in frequency domain. The SRR has three main phases, the motion estimation, the images (frames) fusion and the deblurring.

1.3.1 General Model for Image Acquisition

Image acquisition can be seen as blurring, down-sampling and noising that degrades the quality. A low resolution (LR) image $\boldsymbol{Y}$ can be see as:

$$Y = AX + \varepsilon \tag{1.1}$$

where $\boldsymbol{X}$ is the true high resolution (HR) image, $\boldsymbol{A}$ the wrap, blurring and down-sampling operator and $\boldsymbol{\varepsilon}$ the additive noise. The matrix $\boldsymbol{A}$ can also be separate as:

$$A = DB \tag{1.2}$$

or

$$A_k = DBM_k \tag{1.3}$$

where $\boldsymbol{D}$ represent the downsampling matrix and $\boldsymbol{B}$ the blurring matrix. $\boldsymbol{M_k}$ is a wrap matrix (translation, rotation, etc.) used in the general model for image HR reconstruction with multiple LR images of the same scene $(Y_1, Y_2, \ldots, Y_N)$. The matrices $\boldsymbol{D}$ and $\boldsymbol{B}$ can also be different for every LR image $\boldsymbol{Y_k}$ (it will be $\boldsymbol{D_k}$ and $\boldsymbol{B_k}$), but in this paper, the assumption that they are constant will be used (it is mostly the case if acquired with an unique camera).

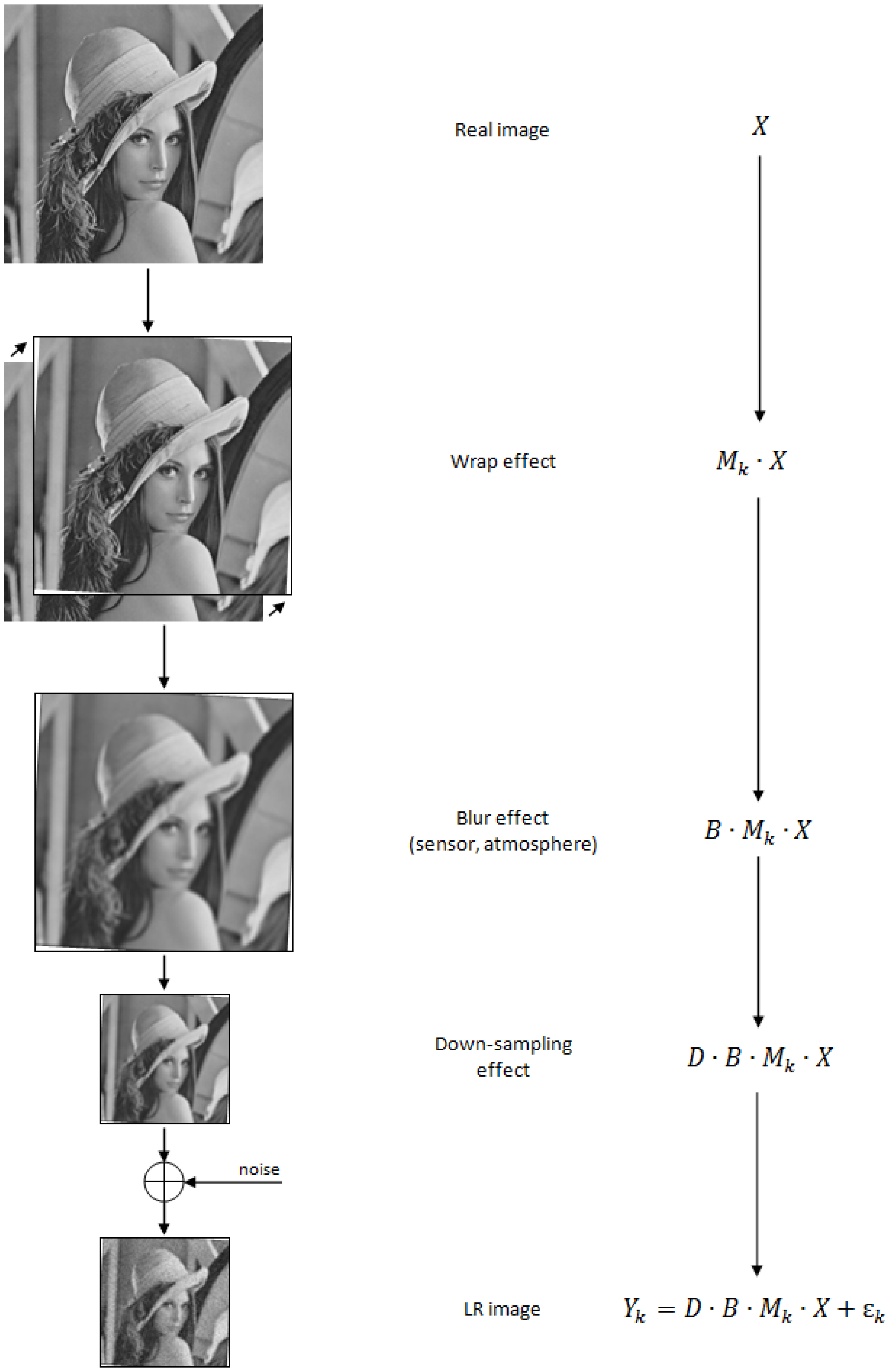


Figure 1.1: General model for image acquisition.

1.3.2 Motion estimation

In fact, the motion of frames is usually unknown. The motion estimation is about to find the motion informations between each frames and the reference one, and this with an accuracy of sub-pixel. There are many motion model of this estimation: only translation, translation with rotation and resizing or motion estimation by parts (separate images into several parts with a motion estimation for each parts). The first will be used for a simple scene shift, the second for a case with for example a rotation of the camera when shooting, and finally the last, for a case where several different parts (for example objects) in the scene have different motions. For every kind of motion estimations, it exist some algorithms to do it.
The experimental comparisons (chapter 4) use the displacement by translation only. For this, it use a function called "dftregistration()" created by Manuel Guizar [6]. This function is an efficient subpixel image registration by cross-correlation.

1.3.3 Frames fusion

Once the motion estimation is done, it is therefore necessary to merge the LR images into a HR image. The motion is with an accuracy of sub-pixel, so it is necessary to interpolate the images before the fusion. This will be done with well known methods such as bilinear, bicubic or spline interpolations. Then, the images are merged. The interpolation and fusion is commonly called the method of Shift-and-Add [7].

1.3.4 Deblurring

As seen in the general model for image acquisition (section 1.3.1), the LR images are usually blurred. To remedy this, there are special filters (the Wiener filter for example). But in the case of super resolution reconstruction by using stochastic regularization approach, it is integrated in the mathematical process.

Chapter 2

Stochastic regularization approach for SRR

A classical type of estimators for SRR algorithm is the ML-type estimators [8] [9]. The SRR algorithm can be mathematically expressed as the following minimization problem:

$$\hat{X} = \arg\min_{X}(\rho(DBX - Y)) \tag{2.1}$$

or

$$\hat{X} = \arg\min_{X}(\rho \sum_{k=1}^{N}(DBM_kX - Y_k)) \tag{2.2}$$

where $\boldsymbol{\rho}()$ is the norm estimation. To minimize this cost function, the reconstruction image $\hat{\boldsymbol{X}}$ have to be closely equal to the original image $\boldsymbol{X}$ (therefore each pixel intensity of the reconstructed image is closely equal to each pixel intensity of the original image). To compute it, we will use estimation of the matrix $\boldsymbol{B}$ (assumptions about the blur). Better these assumptions are, better the result will be. But this algorithm is an ill-posed problem, small noise in the observed image $\boldsymbol{Y}$ will give large change in the result. Therefore, it is necessary to add a regularization term in (2.1) and (2.2) as following:

$$\hat{X} = \arg\min_{X}\Big[(DBX - Y) + \lambda\left\{\Upsilon X\right\}\Big] \tag{2.3}$$

or

$$\hat{X} = \arg\min_{X}\Big[\sum_{k=1}^{N}(DBM_kX - Y_k) + \lambda\left\{\Upsilon X\right\}\Big] \tag{2.4}$$

where $\boldsymbol{\lambda}$ is the regularization parameter (weighting the first term, the similarity term, against the second, the regularization cost) and $\boldsymbol{\Upsilon}$ is the regularization cost function. Moreover this function will improve the rate of convergence by compensating the missing informations by some general prior information about the desirable HR solution of the image $\hat{\boldsymbol{X}}$. This mathematical framework is called SRR using stochastic regularization approach.

For the rest of the paper, equations will only be written for multi LR images as input (general model for SRR).

2.1 The steepest descent method

The steepest descent method, also called the gradient descent method, is an optimization algorithm that is usually used in the SRR frameworks. It finds the minimum (or local minimum) of a function by using the gradient. Fig. 2.1 shows steps of the gradient descent algorithm for a two variables problem. The algorithm follow the direction of the biggest gradient to go to the global minimum (or at least the local minimum).

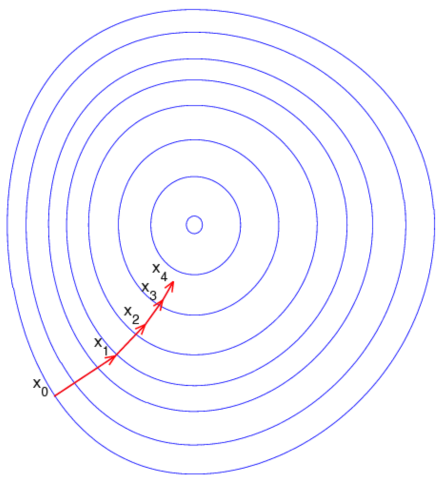


Figure 2.1: Illustration of the gradient descent (image source: `http://en.wikipedia.org/wiki/Gradient_descent`)

The SRR algorithm frequently used the steepest descent method in order to determine its solutions, such as the solution of Eq. (2.4) can be mathematically expressed as (written for the L2 norm estimation):

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \left[\sum_{k=1}^{N} (D^T B^T M_k^T (D B M_k \hat{X}_n - Y_k)) - \lambda \cdot \left\{ (\Upsilon^T \Upsilon) \hat{X}_n \right\} \right] \quad (2.5)$$

2.2 Norm estimation

The L1 and L2 norm (error norm) estimations are traditionally used as the fidelity term (or similarly term) in the steepest descent method (optimization). L2 influence function goes faster and is more efficient than L1 norm, but with high noised images, L2 norm has also more chance to have less good results (local minimum instead the global minimum).

If the fidelity term is:

$$\begin{bmatrix} 12 & -6 & 1 & 9 \\ 11 & -18 & -1 & -6 \\ 21 & 4 & 2 & 3 \\ -8 & -2 & 3 & 2 \end{bmatrix} \tag{2.6}$$

It become:

$$Fidelity\_term\_for\_L1 = \begin{bmatrix} 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \tag{2.7}$$

$$Fidelity\_term\_for\_L2 = \begin{bmatrix} 12 & -6 & 1 & 9 \\ 11 & -18 & -1 & -6 \\ 21 & 4 & 2 & 3 \\ -8 & -2 & 3 & 2 \end{bmatrix} \tag{2.8}$$

L1 takes only the signs of the values of the fidelity term while L2 take them as they are.

It exists also many other norm estimations in the literature (L1-L2 hybrid norm, Euclidean norm, Lq norm, L_∞ norm, etc.), but they will not be used in this paper.

2.3 Regularization term

The regularization functions work like a high-pass filter to improve the rate of convergence by compensating the missing informations by some general prior informations.

The following section will shortly describe the different regularization functions used in this paper. It will also give the expressions used in the SRR algorithm with a little difference: The given expressions will not be normalized, but they are in the experimental comparisons (see section 4 for more information about the normalization).

2.3.1 Laplacian

The Laplacian regularization function is a common high-pass filter. $\boldsymbol{\Upsilon}$ is then replaced by a discrete Laplace operator (often used in image processing). It can be given as convolution with the following kernel:

$$\Gamma = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.9}$$

By using the stochastic regularization approach for SRR, the steepest descent method and the L1 and L2 norm estimation, the problem can be written as follow:

Implementation with L1 norm:

$$\hat{X} = \arg\min_X \Big[\sum_{k=1}^{N} |DBM_kX - Y_k| + \lambda \left\{\Gamma X\right\}^2\Big] \tag{2.10}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[\sum_{k=1}^{N} (D^T B^T M_k^T sign(DBM_k\hat{X}_n - Y_k)) - \lambda \cdot \Big\{(\Gamma^T\Gamma)\hat{X}_n\Big\}\Big] \tag{2.11}$$

Implementation with L2 norm:

$$\hat{X} = \arg\min_X \Big[\sum_{k=1}^{N} (DBM_kX - Y_k)^2 + \lambda \left\{\Gamma X\right\}^2\Big] \tag{2.12}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \left[\sum_{k=1}^{N} (D^T B^T M_k^T (DBM_k \hat{X}_n - Y_k)) - \lambda \cdot \left\{ (\Gamma^T \Gamma) \hat{X}_n \right\} \right] \quad (2.13)$$

Eq. 2.11 and Eq. 2.13 are the equations used in the super resolution reconstruction algorithms.

Example:

The next figure is an example of a Laplacian regularization term for a given image:

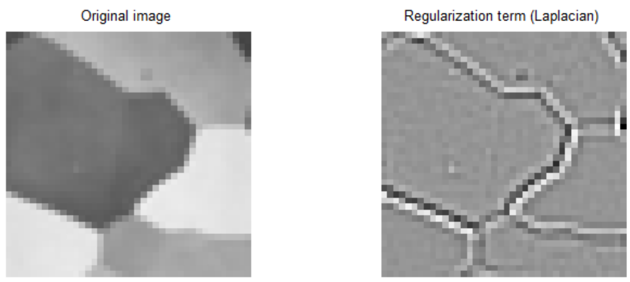


Figure 2.2: Example of a Laplacian regularization term for a given image

2.3.2 Markov Random Field (MRF)

Markov Random Fields are a kind of statistical model [10]. They are widely used for vision problems in digital image processing.

Implementation with L1 norm:

$$\hat{X} = \arg\min_{X} \Big[\sum_{k=1}^{N} |DBM_k X - Y_k| + \left\{ -\frac{1}{2\beta_{MRF}} \sum_{c \in C} \rho_\alpha(d_c^t X) \right\} \Big] \tag{2.14}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\begin{aligned} \hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[& \sum_{k=1}^{N} (D^T B^T M_k^T sign(DBM_k \hat{X}_n - Y_k)) \\ & - \lambda \cdot \left\{ \sum_{c \in C} \rho'_\alpha(d_c^t \hat{X}_n) \right\} \Big] \end{aligned} \tag{2.15}$$

where $\rho'_\alpha(\cdot)$ is defined as:

$$\rho'_\alpha(\cdot) = 2x \;; if \;\; \rho_\alpha(\cdot) \; is \; a \; quadratic \; function \tag{2.16}$$

$$\rho'_\alpha(\cdot) = \begin{cases} 2x & ; \mid x \mid \leq T_{Huber} \\ 2T_{Huber} \cdot sign(x) & ; \mid x \mid > T_{Huber} \end{cases} \;; if \;\; \rho_\alpha(\cdot) \; is \; a \; Huber \; function \tag{2.17}$$

Implementation with L2 norm:

$$\hat{X} = \arg\min_{X} \Big[\sum_{k=1}^{N} (DBM_k X - Y_k)^2 + \left\{ -\frac{1}{2\beta_{MRF}} \sum_{c \in C} \rho_\alpha(d_c^t X) \right\} \Big] \tag{2.18}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\begin{aligned} \hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[& \sum_{k=1}^{N} (D^T B^T M_k^T (DBM_k \hat{X}_n - Y_k)) \\ & - \lambda \cdot \left\{ \sum_{c \in C} \rho'_\alpha(d_c^t \hat{X}_n) \right\} \Big] \end{aligned} \tag{2.19}$$

Eq. 2.15 and Eq. 2.19 are the equations used in the super resolution reconstruction algorithms.

Example:

The next figure is an example of a MRF regularization term for a given image:

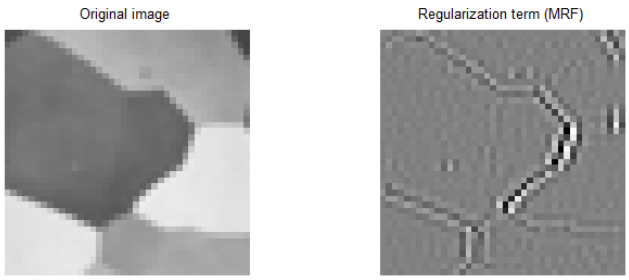


Figure 2.3: Example of a MRF regularization term for a given image (MRF temperature = 5)

2.3.3 Bilateral Total Variation (BTV)

BTV regularization is based on total variation (TV) criterion [11] and the bilateral filter (see Appendix A).

Implementation with L1 norm:

$$\hat{X} = \arg\min_{X} \Big[\sum_{k=1}^{N} |DBM_k X - Y_k| + \lambda \left\{ \sum_{l=-P}^{P} \sum_{m=0}^{P} \alpha^{|m|+|l|} \, \big|\big| X - S_x^l S_y^m X \big|\big| \right\} \Big] \tag{2.20}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[\sum_{k=1}^{N} (D^T B^T M_k^T sign(DBM_k \hat{X}_n - Y_k)) - \lambda \cdot \{ \sum_{l=-P}^{P} \sum_{m=0}^{P} \alpha^{|m|+|l|} (I - S_x^l S_y^m) \cdot sign(\hat{X}_n - S_x^l S_y^m \hat{X}_n) \} \Big] \tag{2.21}$$

where operator matrices S_x^l and S_y^m shift X by l and m pixels in horizontal and vertical directions, presenting several scales of derivatives. The scalar α $(0 < \alpha < 1)$ is applied to give a spatially decaying effect to the summation of the regularization term.

Implementation with L2 norm:

$$\hat{X} = \arg\min_{X} \Big[\sum_{k=1}^{N} (DBM_k X - Y_k)^2 + \lambda \left\{ \sum_{l=-P}^{P} \sum_{m=0}^{P} \alpha^{|m|+|l|} \, \big|\big| X - S_x^l S_y^m X \big|\big| \right\} \Big] \tag{2.22}$$

With the steepest descent method, the SRR solution can mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[\sum_{k=1}^{N} (D^T B^T M_k^T (DBM_k \hat{X}_n - Y_k)) - \lambda \cdot \{ \sum_{l=-P}^{P} \sum_{m=0}^{P} \alpha^{|m|+|l|} (I - S_x^l S_y^m) \cdot sign(\hat{X}_n - S_x^l S_y^m \hat{X}_n) \} \Big] \tag{2.23}$$

Eq. 2.21 and Eq. 2.23 are the equations used in the super resolution reconstruction algorithms.

Example:

The next figure is an example of a BTV regularization term for a given image:

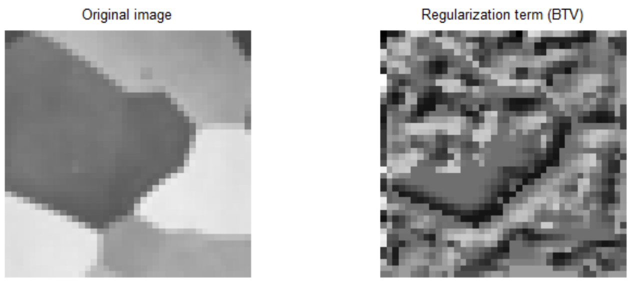


Figure 2.4: Example of a BTV regularization term for a given image (range of neighbour pixels = 1, scalar weight = 0.7)

Chapter 3

Review of the SRR or Super Resolution Reconstruction algorithms

3.1 Classical SRR algorithm

For the first step of the SRR algorithm, the algorithm has to made a motion estimation ($\boldsymbol{M_k}$) between each frame and the first one (in sub-pixel accuracy). The motion estimation will be used in the fidelity part of the SRR algorithm as shown in the following figure. The next figure 3.1 is expression (2.5) with few notes about the content:

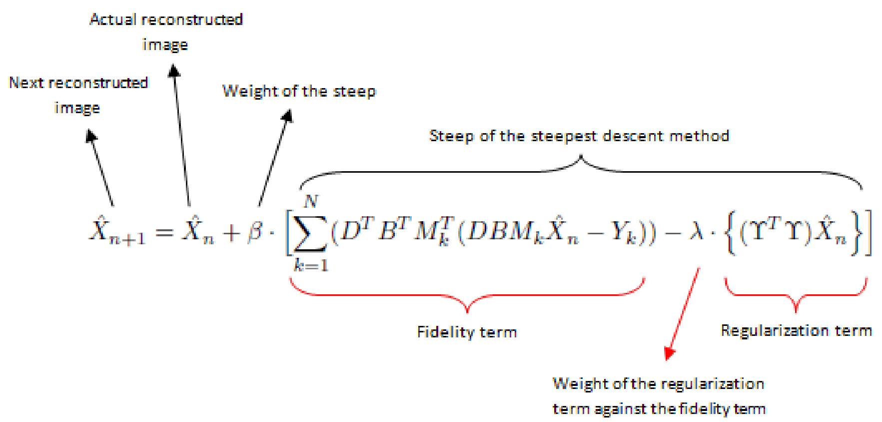


Figure 3.1: Expression (2.5) in details

The figure 3.2 shows the block diagram of the classical algorithm. The blocks F and R represent the fidelity and regularization terms respectively (see fig. 3.1).

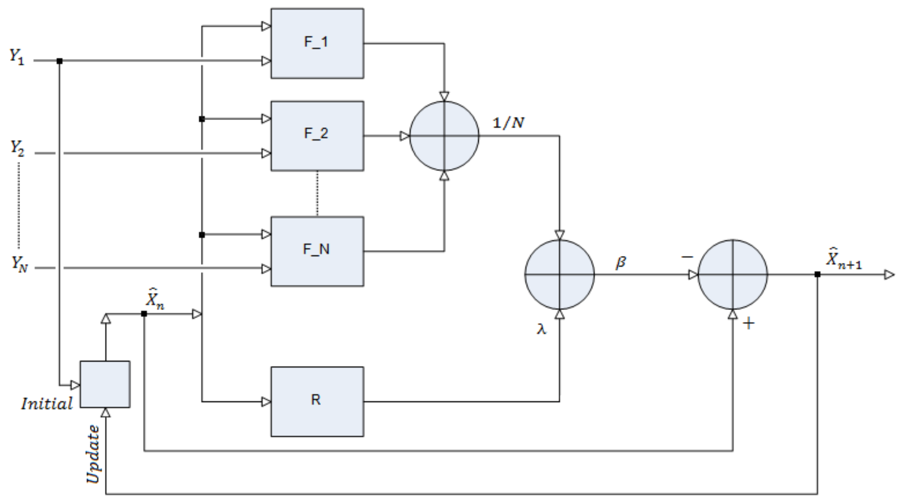


Figure 3.2: Representation of the "Classical SRR algorithm" in block diagram

For the initial state of the steepest descent, the algorithm (in this work) use the interpolation of the degraded image as the initial value of the SRR algorithm or $\boldsymbol{\hat{X}_1 = resize(Y_1)}$. In the classical algorithm, $\boldsymbol{Y_1}$ is the reference image of the LR frames. The motion estimation will first be estimated between every frame and the reference frame, but the first frame $\boldsymbol{Y_1}$ (or the reference frame) has no motion with the reconstructed image and, just by resize it (the algorithm use "imresize" of the Image Processing Toolbox by MathWorks), it give a good start point for the steepest descent method.

The registration process or the motion estimation is estimated from each degraded images that are low resolution and noisy hence. This registration information is not accurate and its may degrade the SRR performance.

3.2 Fast SRR algorithm based on Recursive Least Square

This SRR algorithm works like the "Classical SRR algorithm" (3.1) but this SRR algorithm is separated into 2 parts to be faster [7] [12] [13]:

- Data fusion (non iterative, Shift-and-Add method) and creation of the blurred image $\hat{\boldsymbol{Z}} = \boldsymbol{B}\hat{\boldsymbol{X}}$.
- Estimating the deblurred HR image $\hat{\boldsymbol{X}}$ from $\hat{\boldsymbol{Z}}$ (iterative method, similar to the classical SRR algorithm).

But for this data fusion, the blur and decimation matrix have to be the same for each degraded images (it is the case for frames created by one camera). This robust data fusion gives the blurred version of the ideal image $\boldsymbol{X}$. This removes some operations in the steepest descent.
First, the algorithm computes the motion estimation between each frame and the first one (in sub-pixel accuracy) as the classical SRR algorithm. With this information, it use the Shift-and-Add method to create $\hat{\boldsymbol{Z}}$ (resize to the final size, motion & interpolation and finally, fusion of the frames).
The second part of the "Fast SRR algorithm based on Recursive Least Square" (iterative deblurring) can be mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \left[(B^T(B\hat{X} - Z)) - \lambda \cdot \left\{ (\Upsilon^T \Upsilon)\hat{X}_n \right\} \right] \tag{3.1}$$

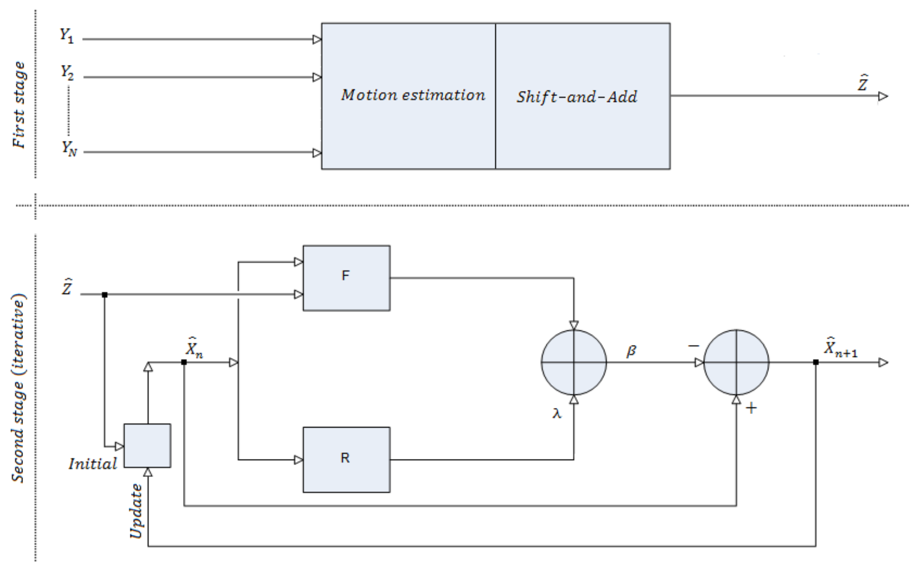


Figure 3.3: "Fast SRR algorithm based on Recursive Least Square", in block diagram representation

The initial state for the algorithm can be the blurred HR image ($\hat{\boldsymbol{X}}_{\boldsymbol{1}} = \hat{\boldsymbol{Z}}$).

3.3 SRR algorithm based on Simultaneous Image Registration

In the classical SRR framework, the motion estimation is most of the time not precise (in larger part because the motion estimation is applied on degraded LR images). Algorithm [14] tries to overcome this problem. So, when the reconstruction is finished, it makes a new motion estimation between the reconstructed image and other degraded frames to have a better estimation. If this estimation is different than the current one (or not close enough), then it will run again the SRR algorithm with the new motion estimations. And at the end, it computes again the motion. The motion is estimate iteratively as the SRR algorithm estimation until the motion estimation (j+1) is the same as the motion estimation before (j), or close enough (as shown in figure 3.4).

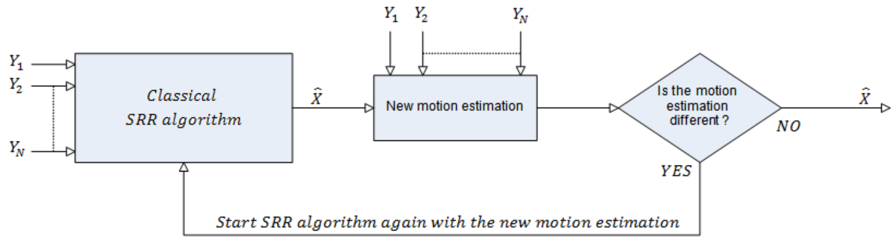


Figure 3.4: Representation of the “SRR algorithm based on Simultaneous Image Registration” in block diagram

3.4 Proposed SRR algorithm Based on Recursive Least Square Estimation and Simultaneous Image Registration

The proposed algorithm is based on the idea of the "Fast SRR algorithm based on Recursive Least Square" (3.2) and the idea of the "SRR algorithm based on Simultaneous Image Registration" (3.3). First it use the two parts same as the "Fast SRR algorithm based on Recursive Least Square":

- Data fusion (non iterative, Shift-and-Add method) and creation of the blurred image $\hat{\boldsymbol{Z}} = \boldsymbol{B}\hat{\boldsymbol{X}}$.

- Estimating the deblurred HR image $\hat{\boldsymbol{X}}$ from $\hat{\boldsymbol{Z}}$ (iterative method, similar to the classical algorithm).

But for this data fusion, the blur and decimation matrix have to be the same for each degraded images (it is the case for frames created by one camera). This robust data fusion gives the blurred version of the ideal image $\boldsymbol{X}$. This removes some operations in the steepest descent.
First, the algorithm computes the motion estimation between each frame and the first one (in sub-pixel accuracy) as the classical SRR algorithm. With this information, it use the Shift-and-Add method to create $\hat{\boldsymbol{Z}}$ (resize to the final size, motion & interpolation and finally, fusion of the frames).
The second part of the "Fast SRR algorithm based on Recursive Least Square" (iterative deblurring) can be mathematically expressed as:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \left[(B^T(B\hat{X} - Z)) - \lambda \cdot \left\{ (\Upsilon^T \Upsilon)\hat{X}_n \right\} \right] \tag{3.2}$$

After, it makes a new motion estimation between the reconstructed image and each input frames after the SRR is finished. The motion is estimated iteratively until the motion estimation (j+1) is the same as the motion estimation before (j), or close enough (see figure 3.5).

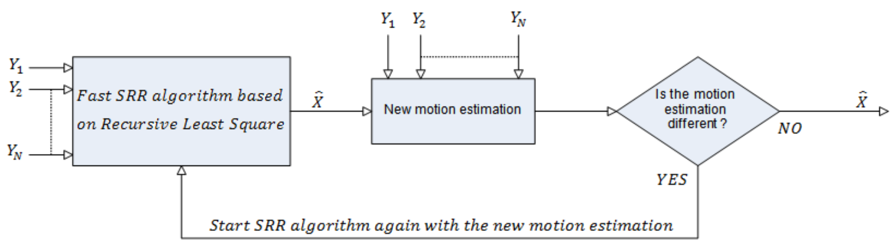


Figure 3.5: Representation of the Proposed SRR algorithm Based on Recursive Least Square Estimation and Simultaneous Image Registration in block diagram

Chapter 4

Experimental results

For every SRR algorithms and for every kind of regularization term, λ (weigh of the regularization term again the fidelity term) will be normalized. So, the weight of the regularization term for a SRR with one input image or multi frames will be the same. The next expression is an example of the mathematical modification for the normalization on the expression 2.13:

$$\hat{X}_{n+1} = \hat{X}_n + \beta \cdot \Big[\sum_{k=1}^{N} \frac{\mathbf{1}}{\boldsymbol{N}}(D^T B^T M_k^T (DBM_k \hat{X}_n - Y_k)) - \lambda \cdot \Big\{(\Gamma^T \Gamma)\hat{X}_n\Big\}\Big] \tag{4.1}$$

Several noises can be observed on real images (depend mostly on the capturing method). So it is more than useful to simulate this noise to test the reaction of an algorithm on a known added noise. The next noise cases will be used in all the experimental comparisons:

Noise case	Type of noise
1	No noise
2	Gaussian noise, SNR = 35
3	Gaussian noise, SNR = 25
4	Gaussian noise, SNR = 15
5	Poisson noise
6	Salt & Pepper noise, d = 0.015
7	Salt & Pepper noise, d = 0.030
8	Speckle noise, v = 0.01
9	Speckle noise, v = 0.03

Table 4.1: Noise cases used in the experimental comparisons (for Salt & Pepper, d is the noise density, and for the Speckle noise, v is the variance)

4.1 Description of the experimental comparisons.

Experimental comparisons for one input image		
No	Name	Objective
4.2.1	Parameters for the SRR	To demonstrate the impact of the SRR parameters on the steepest descent
4.2.2	Rough estimation of the best SRR parameters	To estimate the best parameters for the SRR (such β, λ, etc) and the regularization function for different images and noise models
4.2.3	Regularization term	To test the performance of different regularization functions with different image and noise cases
4.2.4	Norm estimation	To give the best choice between L1 and L2 norm estimations for different cases

Experimental comparisons for multi-frame images (synthetic frames)		
No	Name	Objective
4.3.1	Algorithms experimental results (in qualitative measurement)	Experimental comparisons between algorithms to analyse the performances in order to examine the accuracy of the simultaneous registration algorithm
4.3.2	Algorithms experimental results (in subjective measurement)	Somme examples of the results with the reviewed algorithms and other SRR methods for comparison
4.3.3	Computing times	Experimental comparisons between algorithms to analyse the computing time

Experimental comparisons for multi-frame images (real frames)		
No	Name	Objective
4.4.1	The algorithm of iterative termination technique	How to find the best reconstructed image without the real image X (so when to stop the SRR algorithm)
4.4.2	Experimental comparisons on real frames set No 1	Analyse of the algorithm performances on real frames
4.4.3	Experimental comparisons on real frames set No 2	Analyse of the algorithm performances on real frames by using the SRR algorithms on separate parts of the frames (with overlap)

Table 4.2: Table of the experimental comparisons

4.2 Experimental comparisons for one input image

This section will analyses the SRR results with one input image. The selected images (all in grayscale) are:

- “Lena”, famous standard picture in image processing selected for its great textures.
- “cameraman”, also a famous standard picture in image processing. Selected for sharp edge between foreground and background.
- “alumgrns”, a grey scale image with less texture, and sharp edge between the regions.

The reconstructions will be made with the “Classical algorithm” (section 3.1), other algorithms have no interest for the case of one input image.

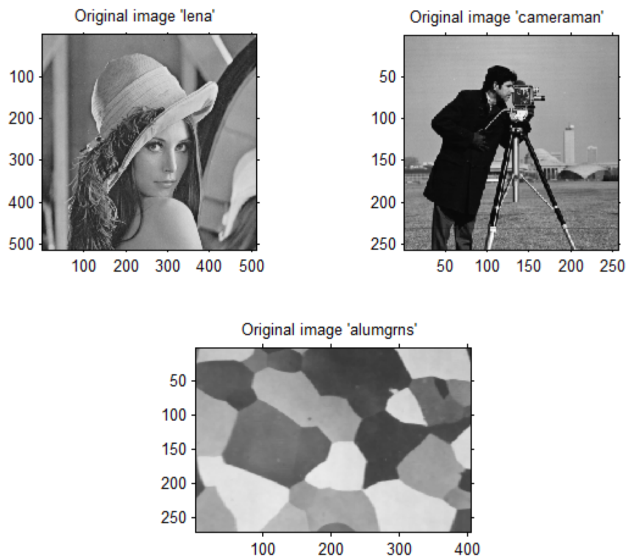


Figure 4.1: Selected images for experimental comparison for one input image

Figure 4.2 shows the chosen parts (will be the original HR image $\boldsymbol{X}$ for the SRR algorithm) of the selected images:

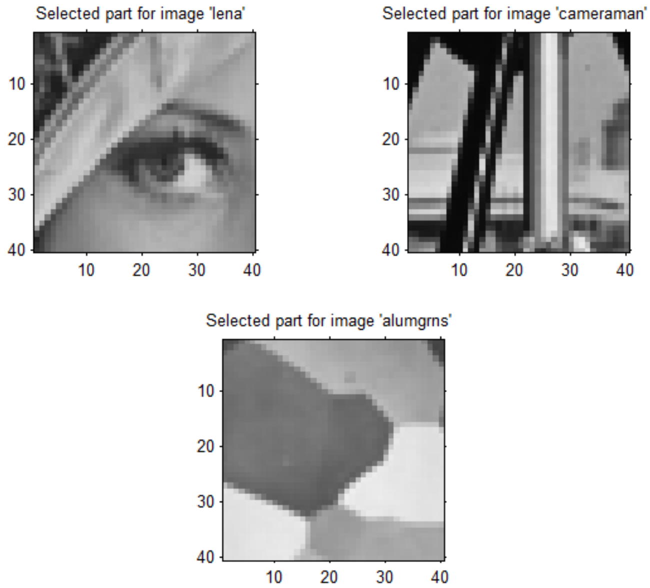


Figure 4.2: Selected parts for experimental comparison for one input image

The following figures illustrate the LR images $\boldsymbol{Y}$ created with the original HR images $\boldsymbol{X}$ for the different cases of noise:

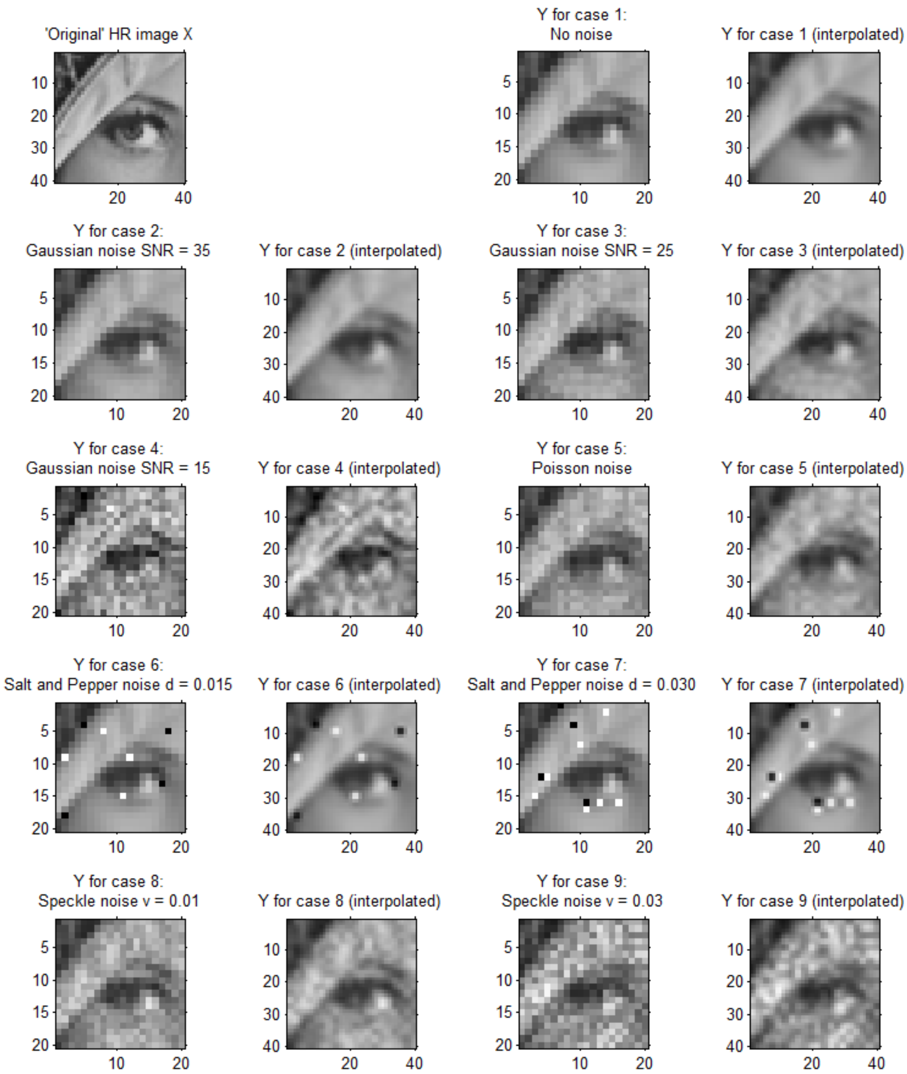


Figure 4.3: LR images (Y) created with the lena image for experimental comparisons for one input image and the corresponding interpolations (bicubic)

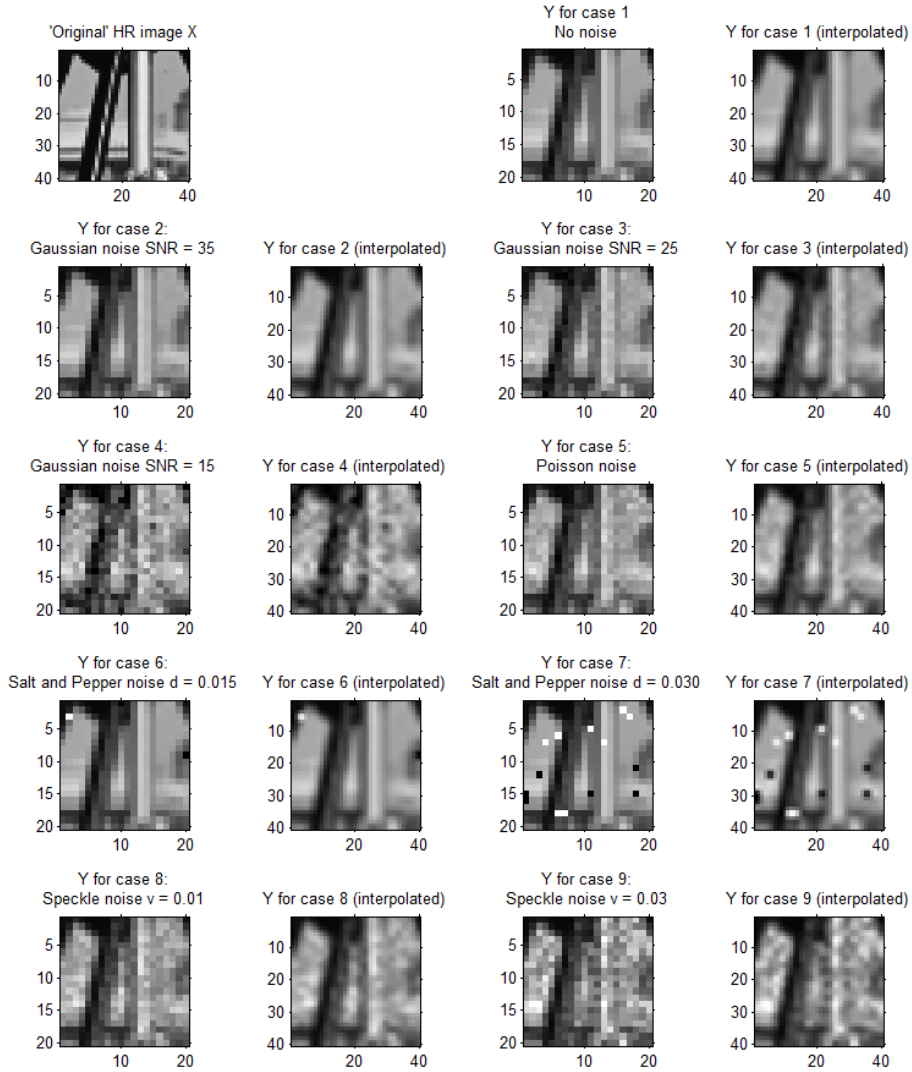


Figure 4.4: LR images (Y) created with the cameraman image for experimental comparisons for one input image and the corresponding interpolations (bicubic)

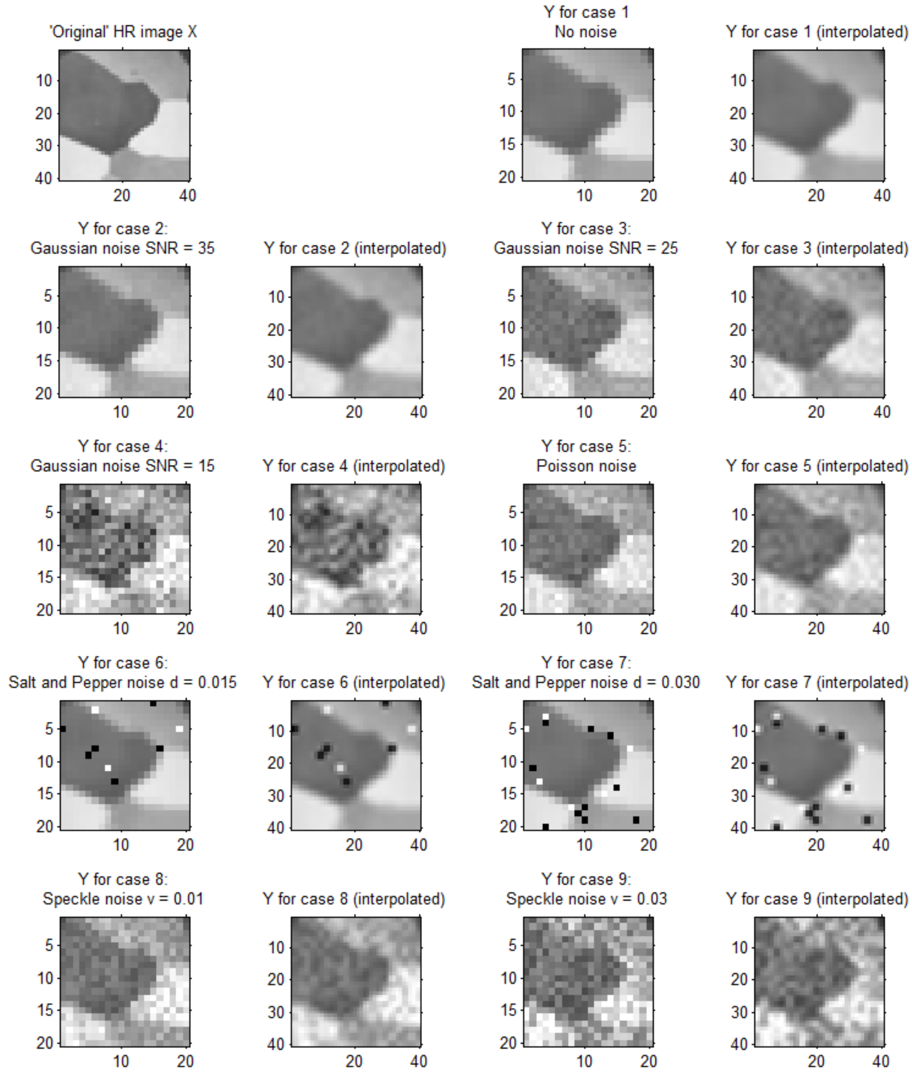


Figure 4.5: LR images (Y) created with the alumgrns image for experimental comparisons for one input image and the corresponding interpolations (bicubic)

4.2.1 Parameters for the SRR

Every parameters (both Super Resolution Reconstruction algorithm parameters and regularization function parameters) had to change with the kind of the image, noise models and the noise power, therefore there is no proposed algorithm to determine the best parameters exactly.

An example of the impact of the SRR parameters (Laplacian regularization) in the SRR performance using the steepest descent is shown in the next figure (cameraman image with added Poisson noise).

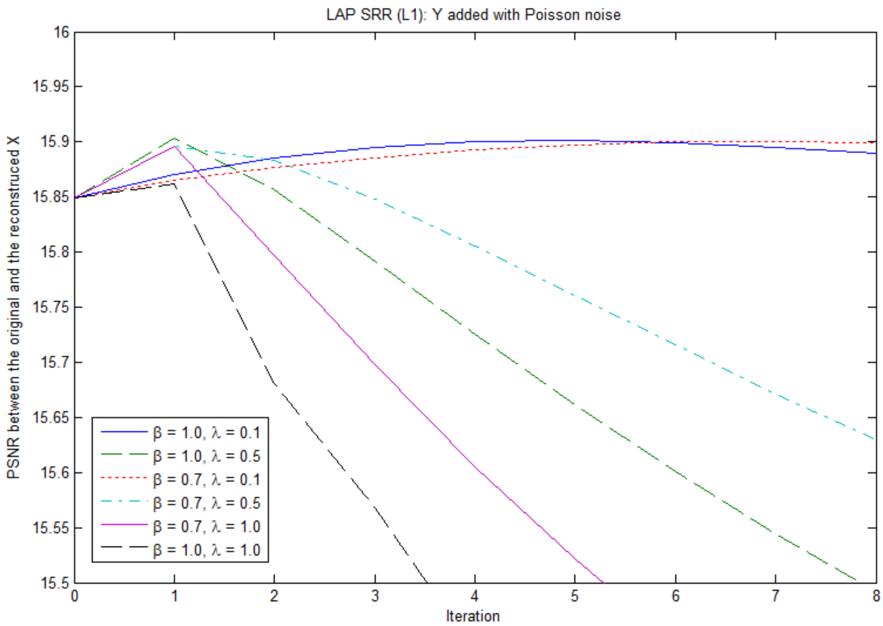


Figure 4.6: Example of the impact of the SRR parameters

Experimental analysis

The graph (fig. 4.6) shows how the choice of parameters is important (both β and λ). This concerns only the parameters related to the SRR algorithm and not the regularization function which has also to be chosen. A wrong choice may give a poor performance, which might have been avoided with an another choice of parameters. Therefore, the next section present hoe to choose the best parameters.

4.2.2 Rough estimation of the best SRR parameters

Some simulations were used to have a **rough estimation** of the best parameters for the SRR algorithm. These simulations used the input image (one frame) of section 4.2. It computes the classical SRR algorithm by varying different steps β (from 0 to 1 by 0.1 steps), weight of the regularization λ and regularizations parameters. The best given parameters would be a good start to find the optimal parameters in each of the experimental comparisons.

	Noise case	1	2	3	4	5	6	7	8	9
	Signal/Noise [dB]	∞	35	25	15	22	18.5	17.5	20	15.5
LAP	β	0.8	0.9	1	0.9	0.9	1	0.8	0.7	0.2
	λ	0	0	0.1	0.3	0.1	0.1	0.1	0.1	0.8
MRF	β	1	1	0.9	0.2	0.7	1	1	1	0.1
	λ	0.1	0.2	0.1	0.5	0.1	0.1	0.1	0.1	1
	temp	15	13	10	5	6	5	5	5	5
BTV	β	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	λ	0	0	0.1	0.2	0.1	0.2	0.1	0.1	0.1
	Ksize	1	1	1	5	1	1	1	1	4
	s. weight	0	0	0	0.7	0.3	0.2	0.5	0.9	0.8

Table 4.3: Rough estimation of the best parameters for the lena image

	Noise case	1	2	3	4	5	6	7	8	9
	Signal/Noise [dB]	∞	35	25	15	21.5	22	14	20.5	15.5
LAP	β	1	1	0.7	1	0.9	1	0.9	1	0.8
	λ	0	0	0	0.1	0.1	0.1	0.1	0.1	0.9
MRF	β	1	0.9	0.9	1	1	1	0.9	0.9	0.2
	λ	0.1	0.3	0.2	0.1	0.2	0.1	0.1	0.1	0.6
	temp	17	16	16	11	14	14	10	15	5
BTV	β	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
	λ	0	0	0	0.2	0	0	0.1	0	0.5
	Ksize	1	1	1	1	1	1	1	1	1
	s. weight	0	0	0	0	0	0	0	0	0

Table 4.4: Rough estimation of the best parameters for the cameraman image

		1	2	3	4	5	6	7	8	9
	Noise case	1	2	3	4	5	6	7	8	9
	Signal/Noise [dB]	∞	35	25	15.5	22.5	19	15	20	15.5
LAP	β	0.9	1	1	0.4	0.6	1	1	0.8	1
	λ	0	0	0.6	1	1	0.1	0.6	1	0.4
MRF	β	1	0.6	0.2	0.1	0.1	1	1	0.1	0.1
	λ	0.2	0.2	0.5	1	1	0.1	0.1	1	1
	temp	16	13	5	5	5	5	5	5	5
BTV	β	0.5	0.5	0.4	0.4	0.3	0.5	0.5	0.4	0.4
	λ	0	0.1	0.2	0.2	0.2	0.1	0.1	0.2	0.2
	Ksize	1	1	1	5	3	4	5	3	5
	s. weight	0	0	1	0.8	0.7	0.8	0.9	0.8	0.8

Table 4.5: Rough estimation of the best parameters for the alumgrns image

"β" is the step for the steepest descent and "λ" the weight of the regularization term against the fidelity term. For the MRF case, "temp" mean the temperature parameter. For the BTV case, "Ksize" mean the range of the neighbor pixels and "s. weight" the scalar weight.

The signal to noise ratio (SNR) is directly computed from the created images. It gives a good idea of the amount of noise. But it is not the same for the salt and pepper noise (noise cases No 6 and 7) who is totally different to the others. The salt and pepper noise simulates errors of the sensors (no data, saturation or failure from a sensor).

It is important to note that if the best parameters for a regularization case include $\lambda = 0$, it mean the algorithm do not use the regularization term and so the regularization parameters given have no meaning.

All the "best parameters" are computed with the L1 norm estimation.

Experimental analysis

LAP :
First, it easy to notice that the step is often 1.0 (or close). So the assumption can be made that for any not to noisy image, a 1.0 step will give good results, but a step between 0.5 and 1.0 will also give good results for a noisy image.
The weight of the regularization is zero for cases with low noise, because the purpose of regularization is to provide a priori information to make easier the steepest descent with a case of noisy image. For low-noisy images, a regularization weight of about 0.1 looks good. And for very noisy (as noise cases 4 and 9), a weight of 0.2 up to 1.0 is sometimes necessary.

MRF :
For the MRF regularization, a good steep is similar to the LAP one. 1.0 for image without or with low noise and between 0.1 and 0.2 for high noise.
For the regularization weight, it should be near 0.1 for image without or with low noise (like Laplacian regularization). More noised image might need a λ

from 0.3 up to 1.0 sometimes.
The temperature parameter should be smaller for high noised image than with low noised one. But this parameter also change a lot with the texture of the image.

BTV :
As for the Laplacian regularization function, in the case of an image without noise, it is not necessary to use regularization.
It should be noted that a steep close to 0.5 is almost always optimal.
The weight of the regularization function is, when it is used, close to 0.1-0.2.
The scalar weight should be near 0.7 and less for not to noised image.
The range of the neighbor pixels is most of the time 1. But for image with less texture, it can be bigger. For the alumgrns image, it goes up to 5 sometimes.

All these analysis just give an idea to find the appropriate parameters for every experimental image. Consequently, these informations are used to estimate roughly the expected magnitude of a parameter in order to find the optimum one.

4.2.3 Regularization term

This section present the comparative performance of the three regularization functions (LAP, MRF and BTV) in the SRR framework.
The following figures will show the steepest descent (PSNR between the reconstructed image X and the original one) for the "Classical SRR algorithm" (L1 norm estimation) with Laplacian, MRF and BTV regularization functions (using the rough parameters estimations of section 4.2.2).

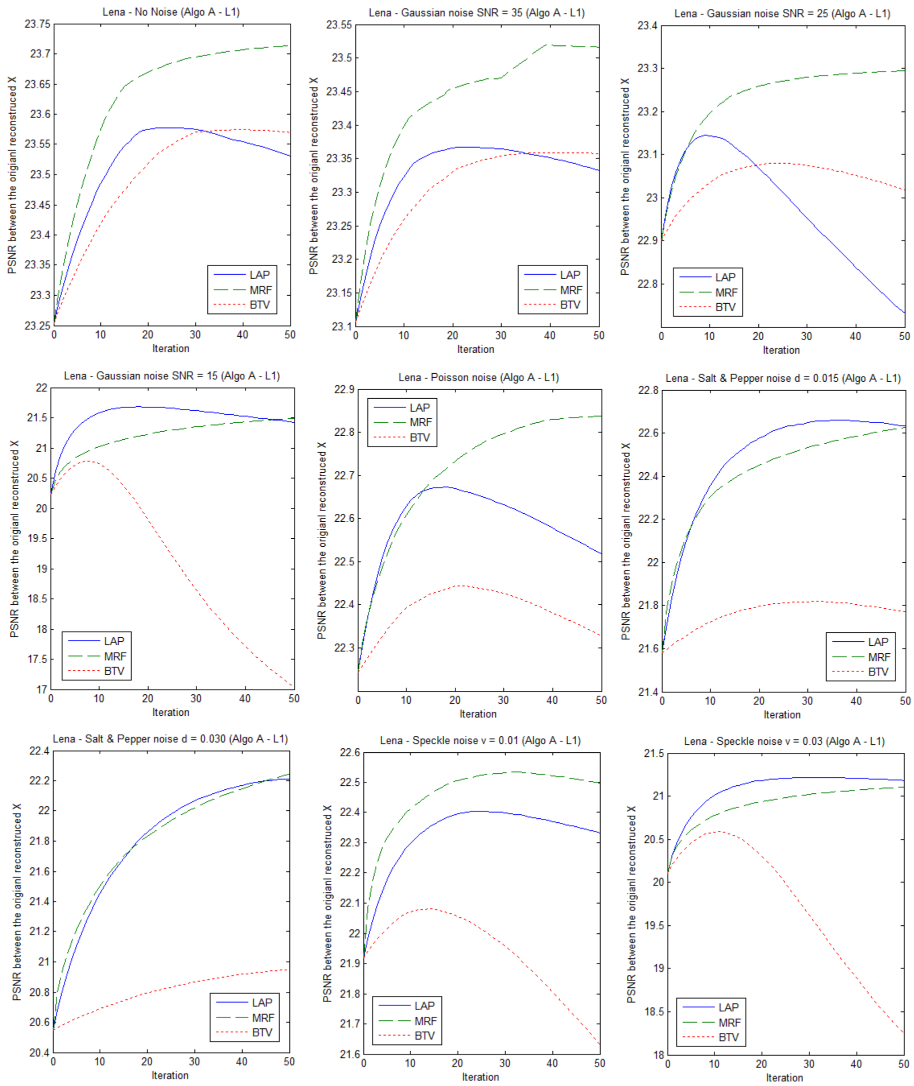


Figure 4.7: Regularization functions comparison (lena image) with the classical SRR algorithm (Algo A)

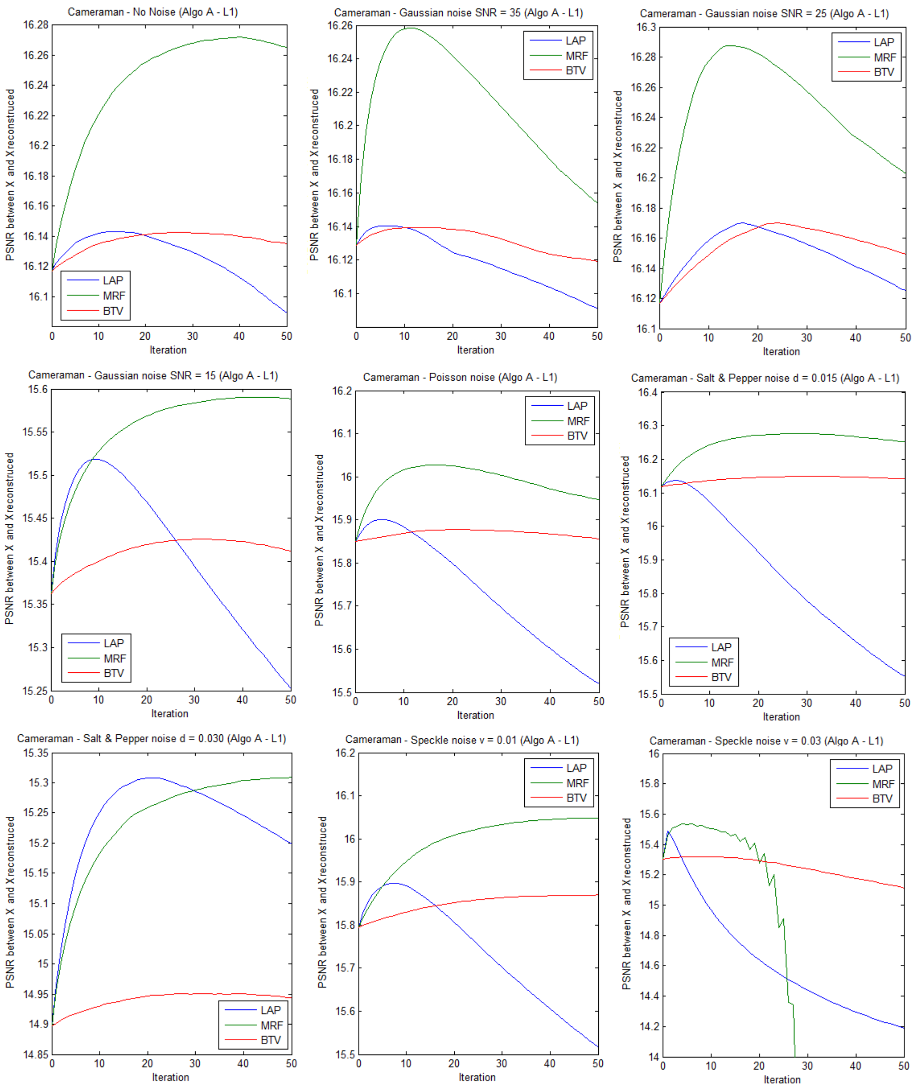


Figure 4.8: Regularization functions comparison (cameraman image) with the classical SRR algorithm (Algo A)

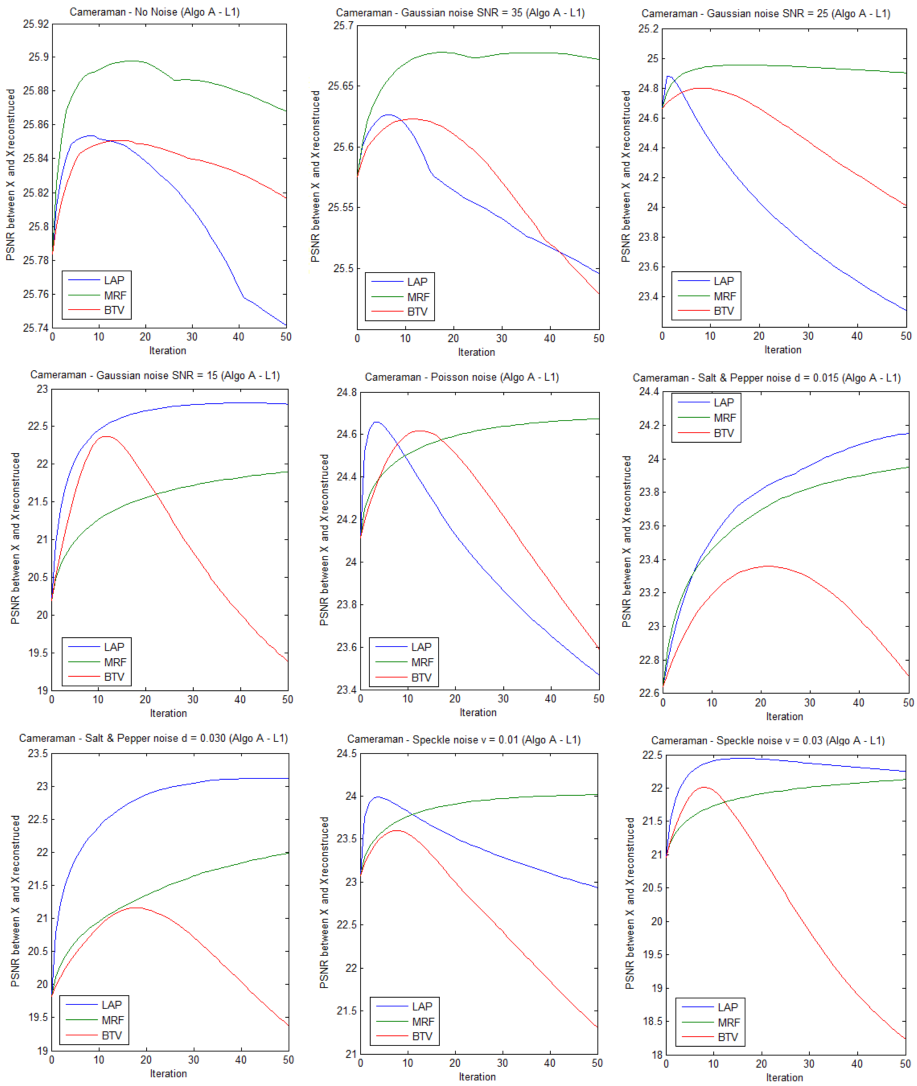


Figure 4.9: Regularization functions comparison (alumgrns image) with the classical SRR algorithm (Algo A)

Experimental analysis

This thesis will not mathematically analyses the best choice of the regularization function for every image case, but it will be just a short analysis in order to choose the appropriate regularization function for each case.
The results show that the best choice change with the kind of the image and the noise. It gives the impression that MRF is better than Laplacian for low noise power case.
The BTV results are really bad, and it should not be the case. For the alumgrns image (low texture, sharp edge), the BTV should be better than the others (at least for the no added noise case). The only explanation for it, is that the implementation in Matlab was not perfectly made.

4.2.4 Norm estimation

This section presents the comparative performance between L1 norm and L2 norm in the SRR framework.
As said before, L2 norm estimation is faster and more efficient, but is not to be used in case of a very noisy image. The following experiment are the SRR results (with Laplacian regularization) on lena, cameraman and alumgrns images.

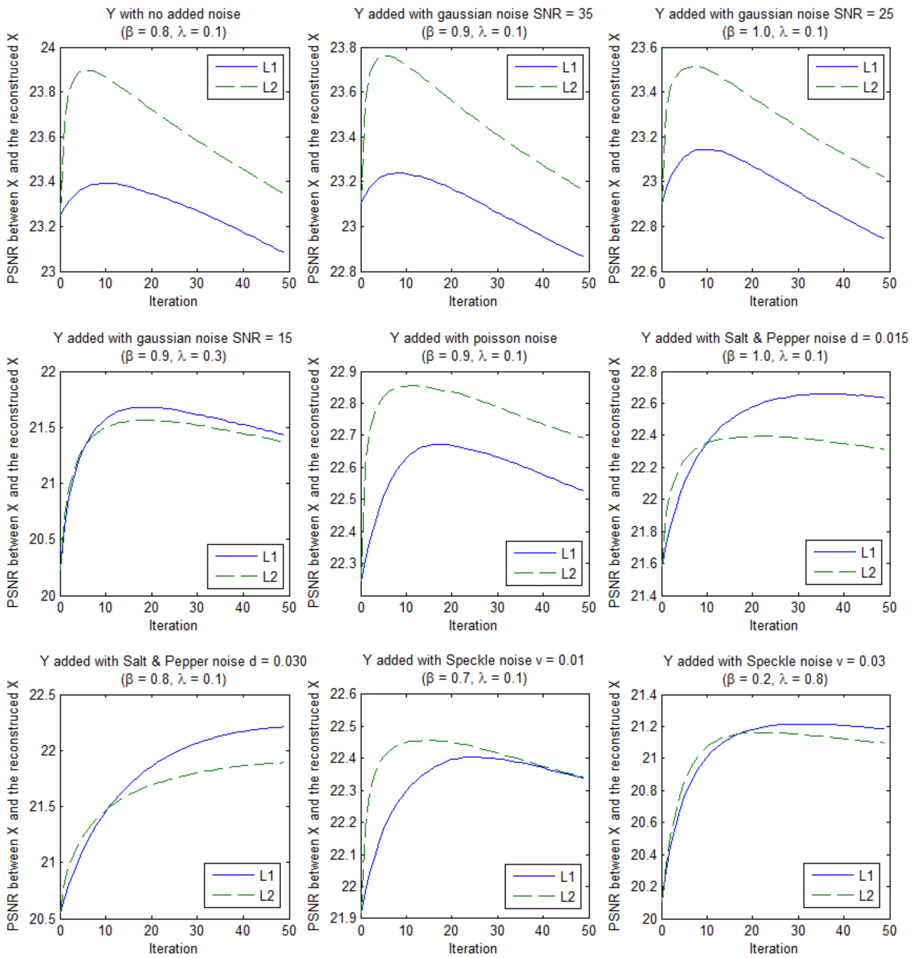


Figure 4.10: SRR (Laplacian reg.) for the lena image with L1 and L2 norm estimation

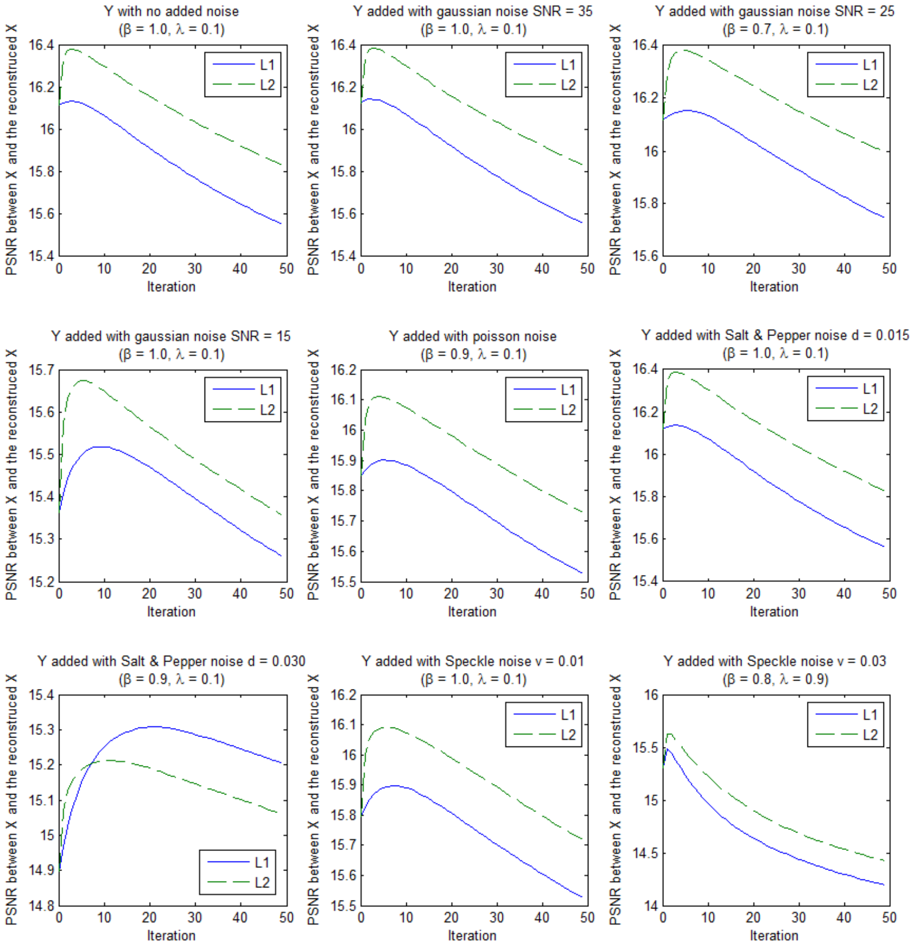


Figure 4.11: SRR (Laplacian reg.) for the camerman image with L1 and L2 norm estimation

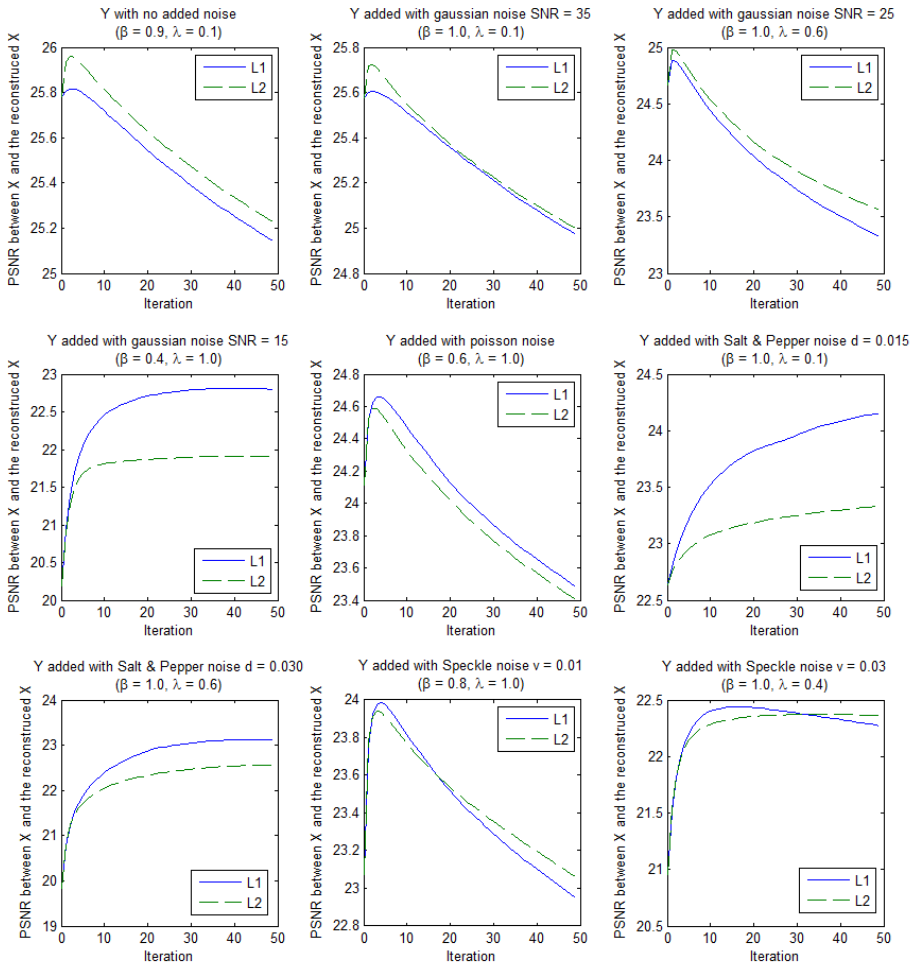


Figure 4.12: SRR (Laplacian reg.) for the alumgrns image with L1 and L2 norm estimation

Experimental analysis

The results are in accordance with the theory. For noiseless or low noised images, the L2 norm estimation gives better results. But in cases with high added noise, the L1 norm estimation is better than the L2 norm. For the Salt & Pepper noise cases, L1 is quite better. It is explained by the fact than Salt & Pepper noise is a high norm noise (modify pixels to 0 or 255 on a 8 bits image).

4.3 Experimental comparisons for multi-frame images (synthetic frames)

This experimental results have 2 tested images: Lena and foreman (frame 110^{th}). Each original image is used to create the 5 synthetic degraded images (frames set). Next, these 5 degraded images are used to reconstruct the SR image.

Lena frames set:

The original image is:

Figure 4.13: Original image (40x40 pixels) for the synthetic frames set No 1 (lena)

The chosen motions are the following ones (the motions are realized in the original size image, before downsampling):

Frame No	1	2	3	4	5
Motion (x) [pixel]	0	0	0.5	0.7	-1
Motion (y) [pixel]	0	1	1	0.5	-1

Table 4.6: Chosen motions for the synthetic frames set No 1 (lena)

The frames created will have the same added noise cases than at the begin of the chapter (Tab. 4.1).

The following figure shows the created frames sets:

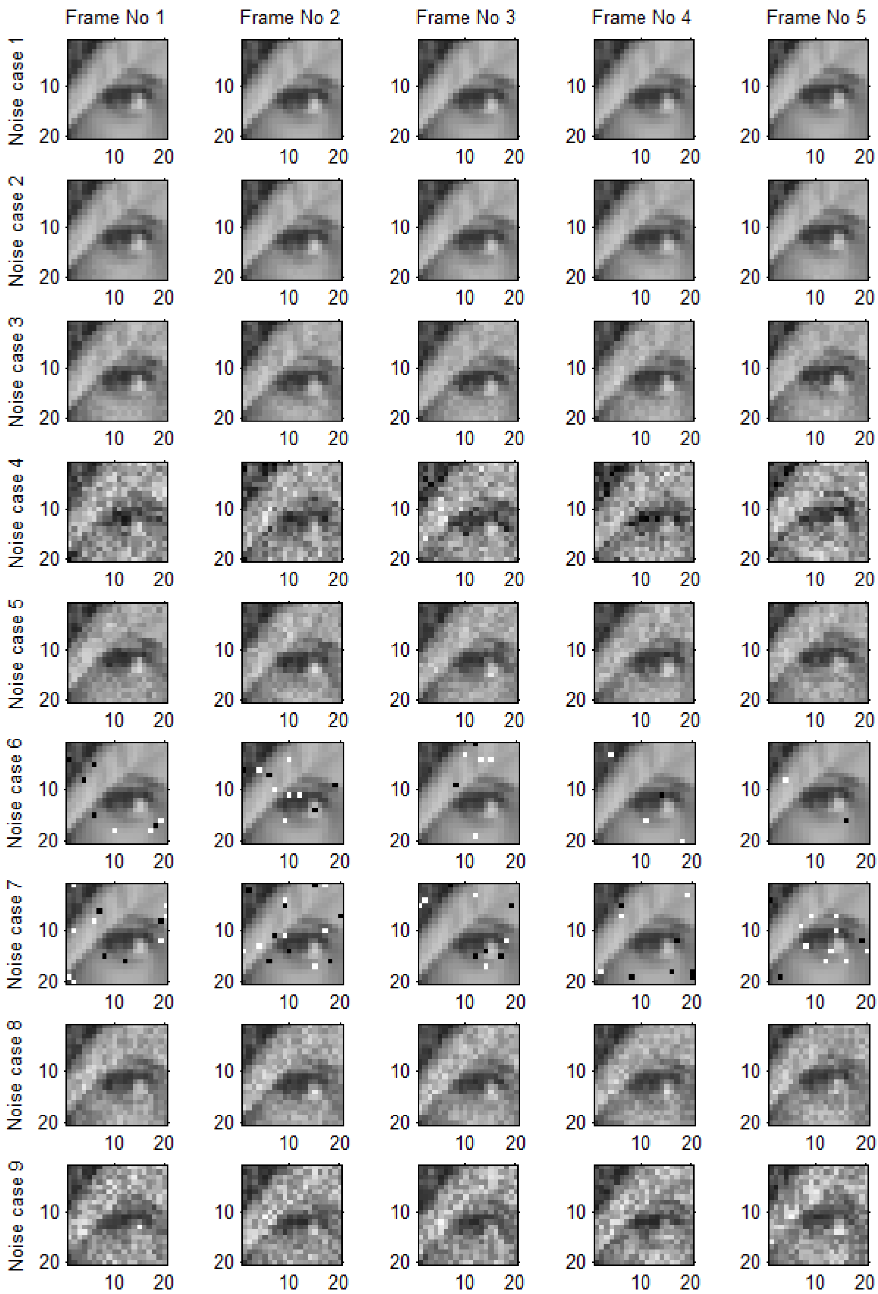


Figure 4.14: Synthetic frames sets No 1 (lena) with several added noise model at different power

Foreman frames set:

The original image is:

Figure 4.15: Original image (40x40 pixels) for the synthetic frames set No 2 (foreman)

The chosen motions are the following ones (the motions are realized in the original size image, before downsampling):

Frame No	1	2	3	4	5
Motion (x) [pixel]	0	0	0.5	0.7	-1.1
Motion (y) [pixel]	0	0.5	1.2	0.5	-1

Table 4.7: Chosen motions for the synthetic frames set No 2 (foreman)

The frames created will have the same added noise cases than at the begin of the chapter (Tab. 4.1).

The following figure shows the created frames sets:

Figure 4.16: Synthetic frames sets No 2 (foreman) with several added noise model at different power

4.3.1 Algorithms experimental results (in qualitative measurement)

Making simulations with different regularization terms have no real interest in this part of the paper. So, all the simulations will be made with the Laplacian regularization term to just compare the results among the algorithms.

The first simulation compute the super resolution reconstruction with every algorithm for the lena synthetic frames and every noise cases (using the rough parameters estimations of section 4.2.2, but for the parameters estimation who give $\lambda = 0$, it takes $\lambda = 0.1$ to use the regularization). The next figure shows the best results for every SRR:

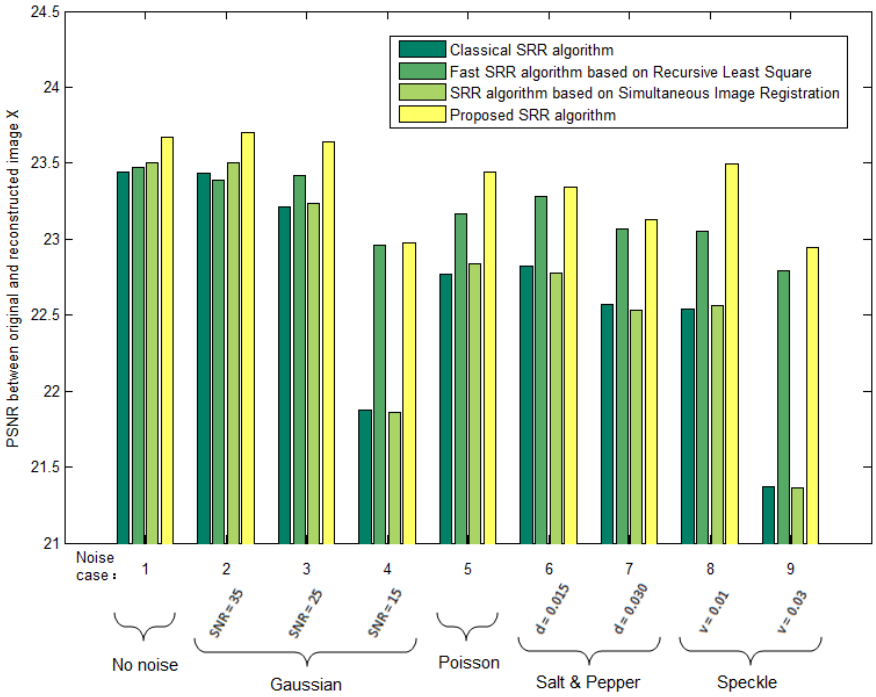


Figure 4.17: Results with the lena synthetic frames for all SRR algorithms (and the proposed one) with Laplacian regularization

PSNR between X and $\hat{X}$									
Noise case	1	2	3	4	5	6	7	8	9
Algo A	23.44	23.43	23.21	21.87	22.77	22.82	22.57	22.54	21.37
Algo B	23.47	23.39	23.42	22.96	23.16	23.28	23.07	23.06	22.79
Algo C	23.51	23.51	23.24	21.86	22.84	22.78	22.53	22.56	21.36
Algo D	23.67	23.70	23.64	22.97	23.44	23.34	23.13	23.50	22.95

Table 4.8: Results with the lena synthetic frames for all SRR algorithms (and the proposed one) with Laplacian regularization (A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

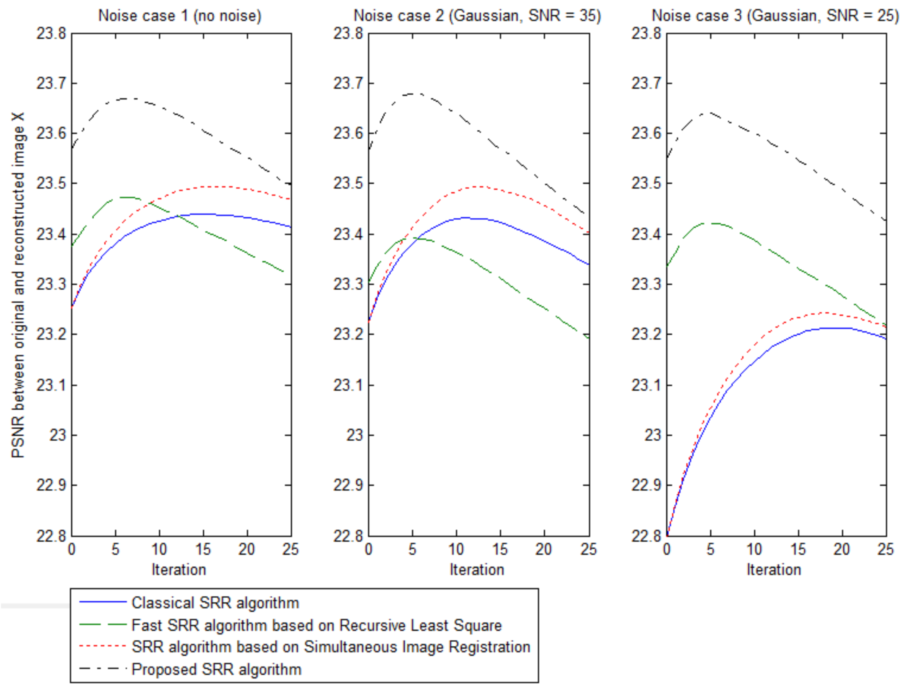


Figure 4.18: Results with the lena synthetic frames for all SRR algorithms (and the proposed one) with Laplacian regularization. Noise cases 1-3 in detail (evolution of the reconstructed images). Iteration 0 is the pre-process (interpolation or shift-and-add).

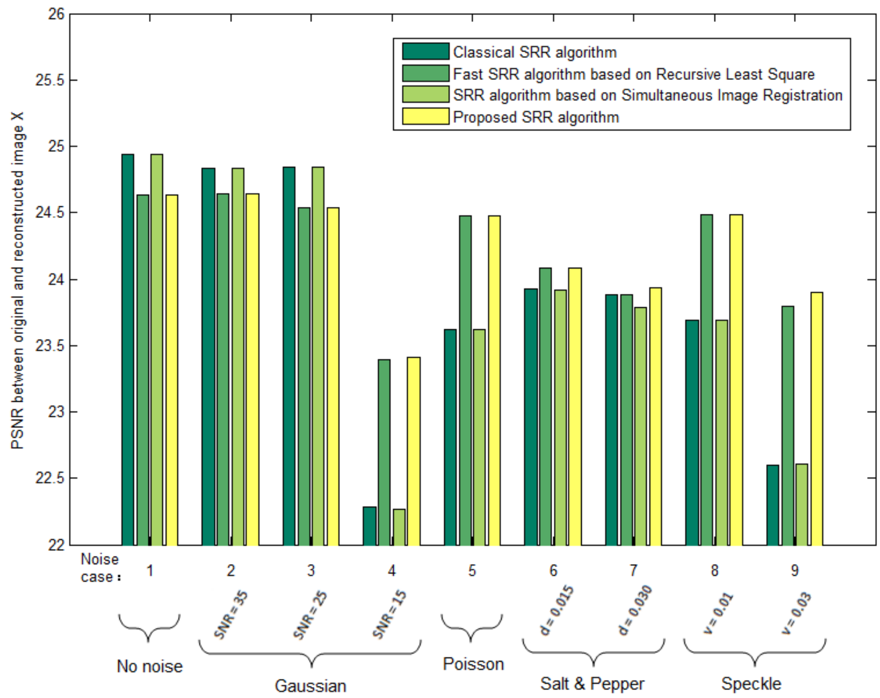


Figure 4.19: Results with the foreman synthetic frames for all SRR algorithms (and the proposed one) with Laplacian regularization

PSNR between X and $\hat{X}$									
Noise case	1	2	3	4	5	6	7	8	9
Algo A	24.94	24.84	24.85	22.29	23.62	23.93	23.88	23.69	22.60
Algo B	24.63	24.64	24.54	23.39	24.47	24.08	23.88	24.49	23.79
Algo C	24.94	24.84	24.85	22.26	23.62	23.91	23.78	23.69	22.61
Algo D	24.63	24.64	24.54	23.41	24.47	24.08	23.94	24.49	23.90

Table 4.9: Results with the foreman synthetic frames for all SRR algorithms (and the proposed one) with Laplacian regularization (A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

Experimental analysis

Lena synthetic frames:

At the start of the SRR algorithms, the "Classical SRR algorithm" and the "SRR algorithm based on Simultaneous Image Registration" give the same results. It is explained by the fact that the first reconstructed image is the same. After each iteration, the second one become better with added informations from the frames with better motion estimations.
The "Fast SRR algorithm based on Recursive Least Square" and the "proposed SRR algorithm" start already with better results. This is because these algorithm start with an image who is a fusion of all the frames (Shif-and-Add method). The better motion estimation of the "proposed SRR algorithm" gives directly a better start in the steepest descent.
It's easy to see that the results are better with the "proposed SRR algorithm". For high noised frames (noise cases No 4, 6, 7 and 9), the proposed algorithm is near to have the same performance than the "Fast SRR algorithm based on Recursive Least Square" but still better. It is the case because the new motion estimations are not really better with high noise.
The "SRR agorithm based on Simultaneous Image Registration" should be better for every case than the "Classical SRR algorithm". It is not the case with the Salt & Pepper noise. This is probably the case because Salt & Pepper is a high norm noise and the motion estimation with cross correlation might be not the best motion estimation algorithm for this kind of noise.
It is also to note that the number of iterations with the "Fast SRR algorithm based on Recursive Least Square" and the "proposed SRR algorithm" are much smaller than for the other. It should be because these algorithms start with all the frames informations and the steepest descent is used just for the deblurring and denoising. The other algorithms use also the steepest descent to add frames informations in the reconstructed images.

Foreman synthetic frames:

On this simulation (fig. 4.19), there are far fewer new motion estimation.
The "SRR algorithm based on Simultaneous Image Registration" is also worse than the "Classical SRR algorithm" in the Salt & Pepper noise cases (probably for the same reason).
But for the case with new motion estimation for the "proposed SRR algorithm", it is better than the "Fast SRR algorithm based on Recursive Least Square".
In most of the cases, the proposed SRR algorithm provide better results. This gives the impression that it is sometimes worse in cases with very little added noise (or no noise).

4.3.2 Algorithms experimental results (in subjective measurement)

The following figures show the results in subjective measurement of the previous section (qualitative measurement):

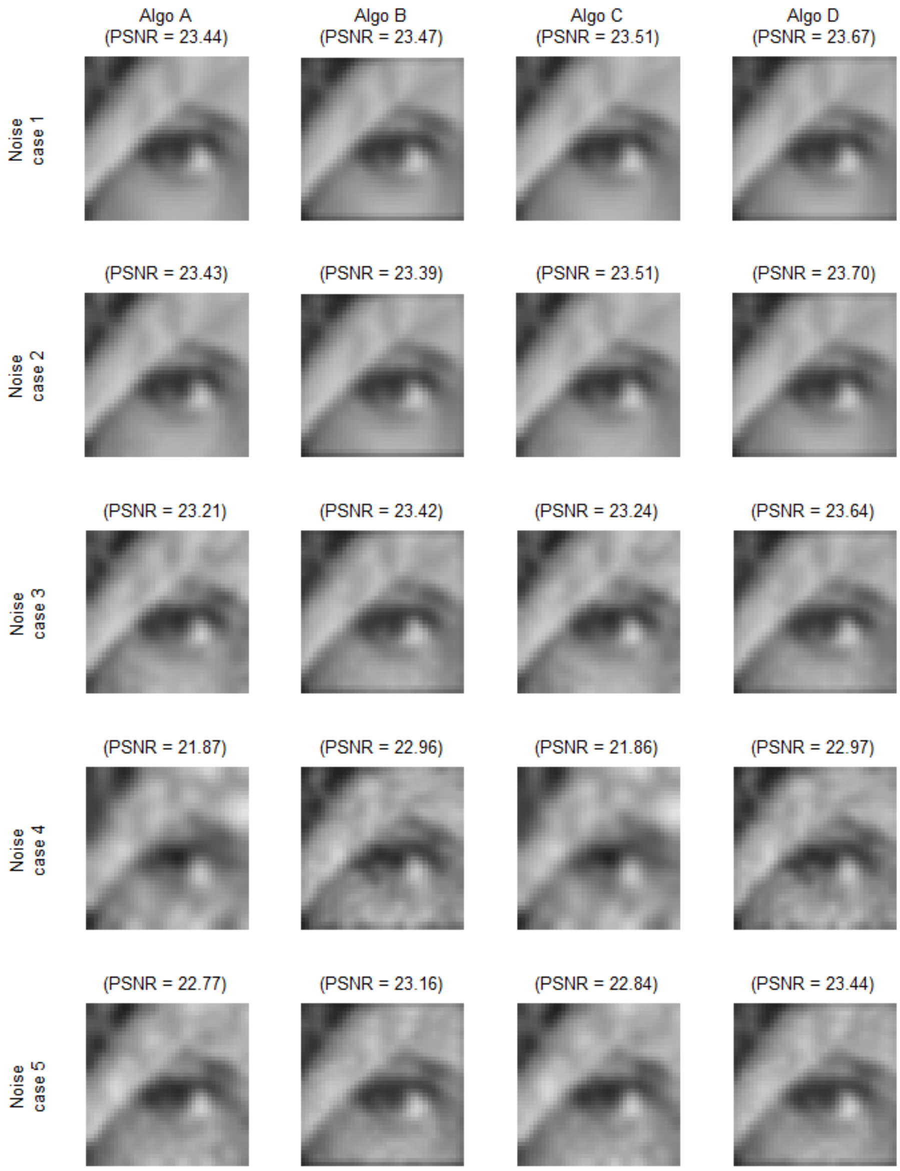


Figure 4.20: Algorithms experimental results (in subjective measurement). Lena synthetic frames, noise cases 1-5. A: "Classical SRR algorithm" (Laplacian and L1). B: "Fast SRR algorithm based on Recursive Least Square" (Laplacian and L1). C: "SRR algorithm based on Simultaneous Image Registration" (Laplacian and L1). D: "Proposed SRR algorithm" (Laplacian and L1).

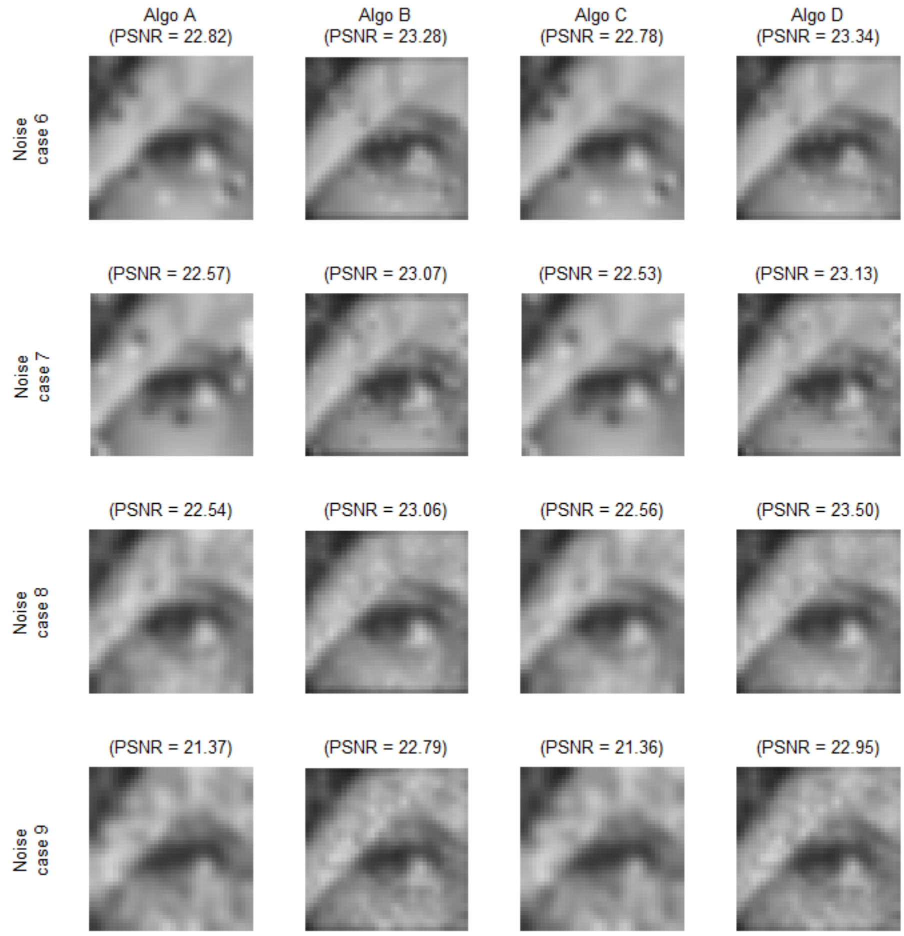


Figure 4.21: Algorithms experimental results (in subjective measurement). Lena synthetic frames, noise cases 6-9. A: "Classical SRR algorithm" (Laplacian and L1). B: "Fast SRR algorithm based on Recursive Least Square" (Laplacian and L1). C: "SRR algorithm based on Simultaneous Image Registration" (Laplacian and L1). D: "Proposed SRR algorithm" (Laplacian and L1).

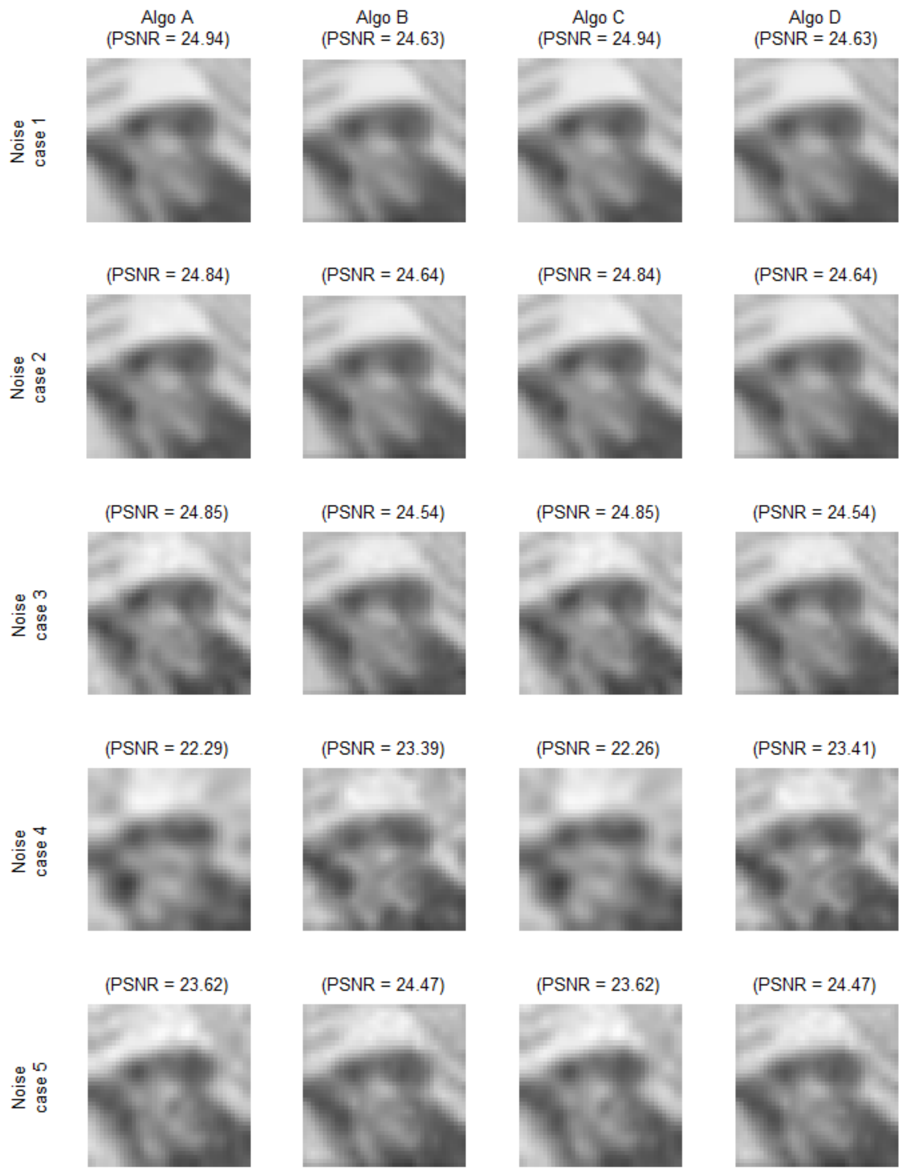


Figure 4.22: Algorithms experimental results (in subjective measurement). Foreman synthetic frames, noise cases 1-5. A: "Classical SRR algorithm" (Laplacian and L1). B: "Fast SRR algorithm based on Recursive Least Square" (Laplacian and L1). C: "SRR algorithm based on Simultaneous Image Registration" (Laplacian and L1). D: "Proposed SRR algorithm" (Laplacian and L1).

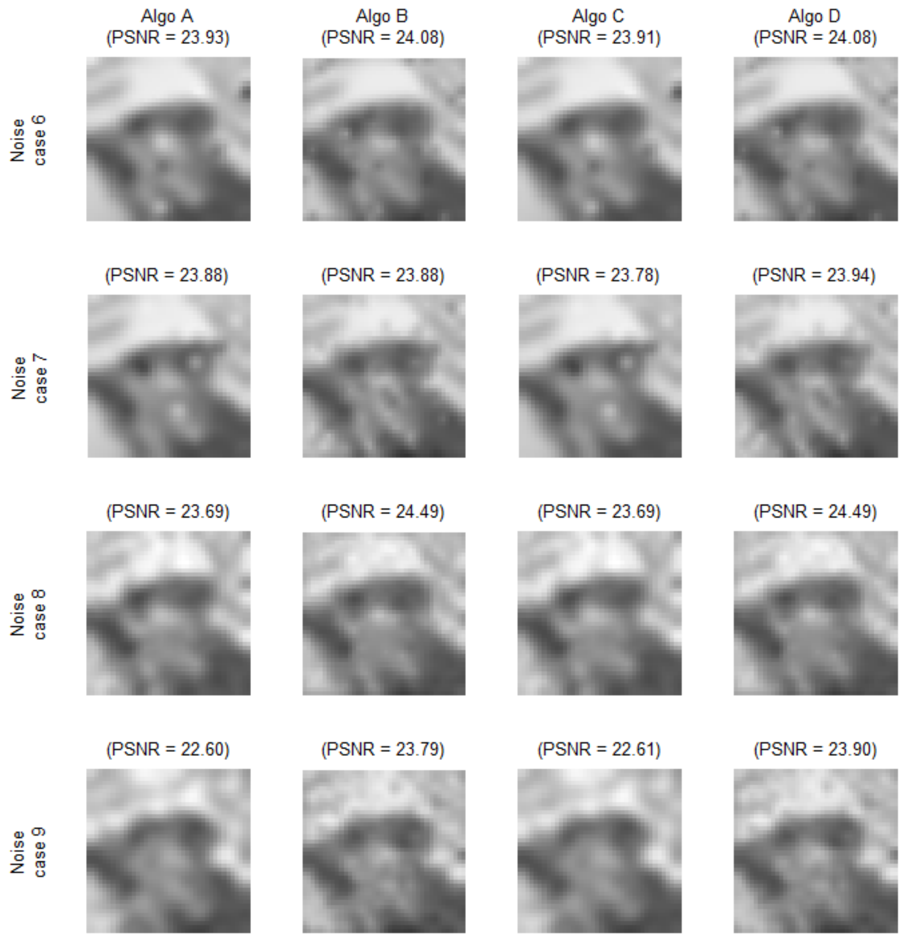


Figure 4.23: Algorithms experimental results (in subjective measurement). Foreman synthetic frames, noise cases 6-9. A: "Classical SRR algorithm" (Laplacian and L1). B: "Fast SRR algorithm based on Recursive Least Square" (Laplacian and L1). C: "SRR algorithm based on Simultaneous Image Registration" (Laplacian and L1). D: "Proposed SRR algorithm" (Laplacian and L1).

The next two figures present the reconstructed images with different algorithms and other SRR methods. The first simulation works on the synthetic frames with added Gaussian noise and the second with added Salt & Pepper noise (see appendix B for more informations about the Steering Kernel Regression).

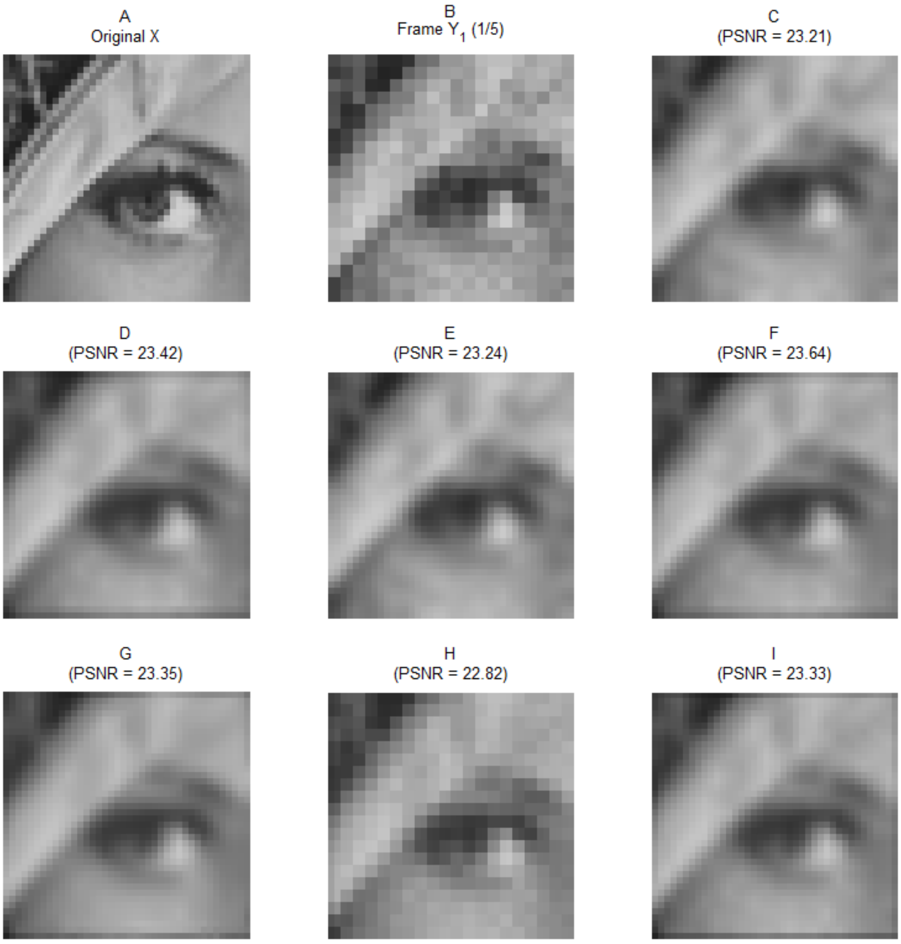


Figure 4.24: Reconstructed images examples with the reviewed algorithms and other SRR methods. Lena synthetic frames (noise case 3, Gaussian noise SNR = 25). A: The original X image. B: The first of the 5 frames. C: "Classical SRR algorithm" (Laplacian and L1). D: "Fast SRR algorithm based on Recursive Least Square" (Laplacian and L1). E: "SRR algorithm based on Simultaneous Image Registration" (Laplacian and L1). F: "Proposed SRR algorithm" (Laplacian and L1). G: Shift-and-Add method (Bicubic) + Iterative Steering Kernel Regression. H: Shift-and-Add method (Nearest). I: Shift-and-Add method (Bicubic).

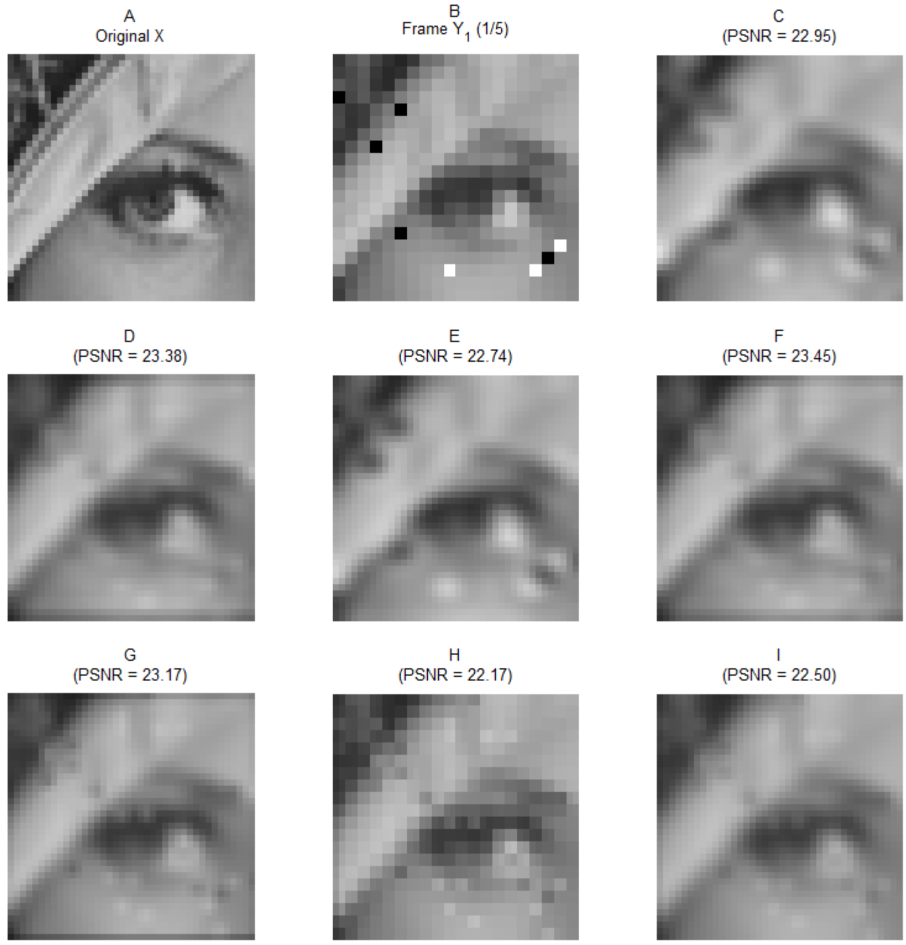


Figure 4.25: Reconstructed images examples with the reviewed algorithms and other SRR methods. Lena synthetic frames (noise case 6, Salt & Pepper noise d = 0.015). A: The original X image. B: The first of the 5 frames. C: "Classical SRR algorithm" (MRF and L1). D: "Fast SRR algorithm based on Recursive Least Square" (MRF and L1). E: "SRR algorithm based on Simultaneous Image Registration" (MRF and L1). F: "Proposed SRR algorithm" (MRF and L1). G: Shift-and-Add method (Bicubic) + Iterative Steering Kernel Regression. H: Shift-and-Add method (Nearest). I: Shift-and-Add method (Bilinear).

Experimental analysis

These two experiments demonstrate why the "Classical SRR algorithm" is so good for the cases without noise (or very little), or rather, why is it so bad with added noise (also the "SRR algorithm based on Simultaneous Image Registration"). The "Proposed SRR algorithm" uses the method of Shift-and-Add before the SRR using stochastic regularization. The noise in this case is mixed together, and with random noise, this tends to decrease it. Salt & Pepper noise example shows it good.
But the "Classical SRR algorithm" starts with a frame and adds informations of the other frames at each step of the steepest descent.
So by starting the "Fast SRR algorithm based on Recursive Least Square" and the "Proposed SRR algorithm" with a Shift-and-Add method, it makes them robust to noise.

4.3.3 Computing times

The next table shows the computing times (CT) until the best reconstructed image of the simulation on figure 4.17, the number of iterations (I) and de number of new motion estimations (NME) for the "SRR algorithm based on Simultaneous Image Registration" and the "proposed SRR algorithm":

Noise case		1	2	3	4	5	6	7	8	9
A	CT [s]	5.2	4.8	7.2	9.6	9.7	13.1	16.0	14.8	10.6
	I	15	14	21	30	30	41	50	46	33
B	CT [s]	2.1	1.9	2.1	1.5	1.8	2.1	2.3	3.0	1.2
	I	7	6	7	5	6	7	8	10	4
C	CT [s]	19.9	10.5	12.7	18.7	20.6	43.9	31.9	30.8	20.6
	I	17	15	19	29	32	34	50	48	32
	NME	2	1	1	1	1	3	1	1	1
D	CT [s]	4.9	6.3	3.6	4.5	4.2	4.7	5.3	7.1	2.4
	I	8	7	6	5	7	8	9	8	4
	NME	1	2	1	2	1	1	1	2	1

Table 4.10: Computing times with the lena synthetic frames for all algorithms (and the proposed one) with Laplacian regularization (A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

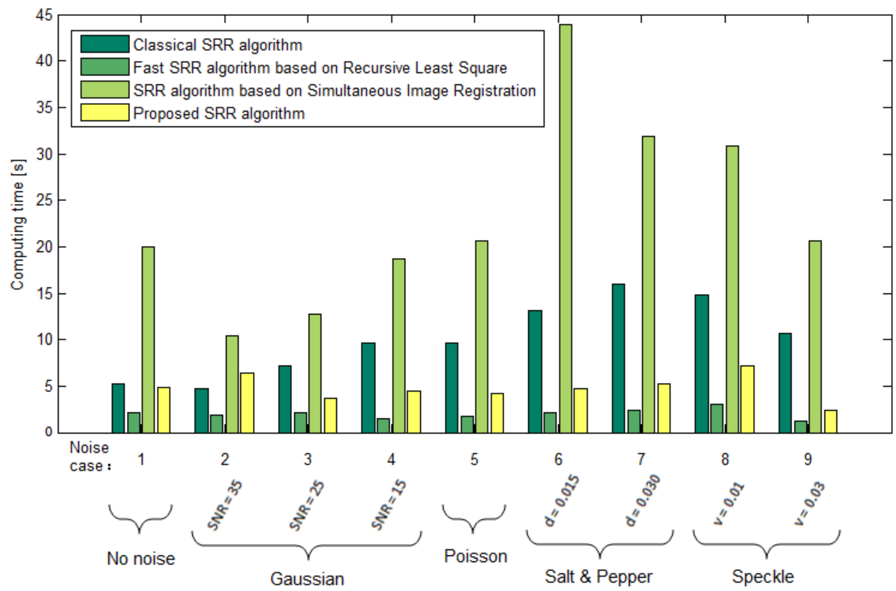


Figure 4.26: Computing times with the lena synthetic frames for all algorithms (and the proposed one) with Laplacian regularization

Experimental analysis

No surprise, the "Proposed SRR algorithm" is slower than the "Fast SRR algorithm based on Recursive Least Square". But even with the iterative motion estimation, he is most of the case faster than the "Classical SRR algorithm" (and for sure, faster than the "SRR algorithm based on Simultaneous Image Registration").
It is also to note that the new motion estimations are quite few (most of the time 1 or 2).

4.4 Experimental comparisons for multi-frame images (real frames)

For this experimental part, real frames will be used. So, it would not be possible to analyse the result with for example, the error with the original image. This analysis will be made just by visual quality of the output reconstructed images.

The results of two set of real frames will be analysed:

- A set of text frames, captured with an Olympus C-4000 camera [1]. These frames approximately follow the global translational motion model. The selected part is a 20x20 pixels zone (up left) of the real frames. The first frame will be the reference frame for the SRR.

- Foreman frames No 108-112 (the frame No 110, used in the experimental comparisons with synthetic frames, will be the reference frame for the SRR).

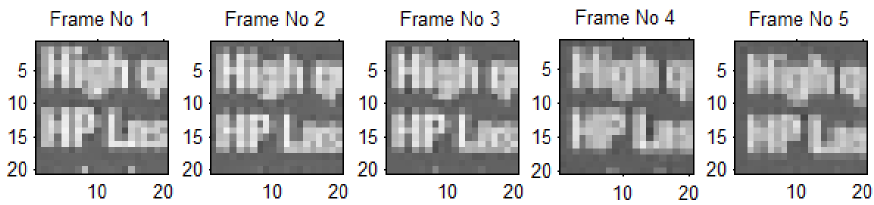


Figure 4.27: Real frames set No 1 (20x20 pixels)

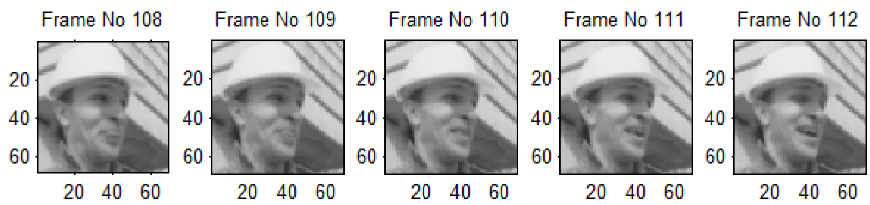


Figure 4.28: Real frames set No 2 (68x68 pixels)

[1]Source of the frames: Sina Farsiu,
`http://users.soe.ucsc.edu/~milanfar/software/sr-datasets.html`

4.4.1 The algorithm of iterative termination technique

Until this point, all the simulations compute a number of SRR iterations and take the best one. It is only possible because the real image $\boldsymbol{X}$ is available. Unfortunately, for real frames SRR, $\boldsymbol{X}$ is unknown. Thus for this experimental section and its subsection, measurement the criteria for determining the best result (or the best reconstructed image $\hat{X}$) have three methods as following:

- Visual analysis of the reconstructed images, but it is hard to determine.
- Training with approaching synthetics frames to find the estimate number of iterations.
- Analysis between the reconstructed images.

Analysis between the reconstructed images:

This method is choosed by this thesis because it can be really applicable. The next simulations give the PSNR and the RMSE between the reconstructed images given by the classical SRR algorithm.

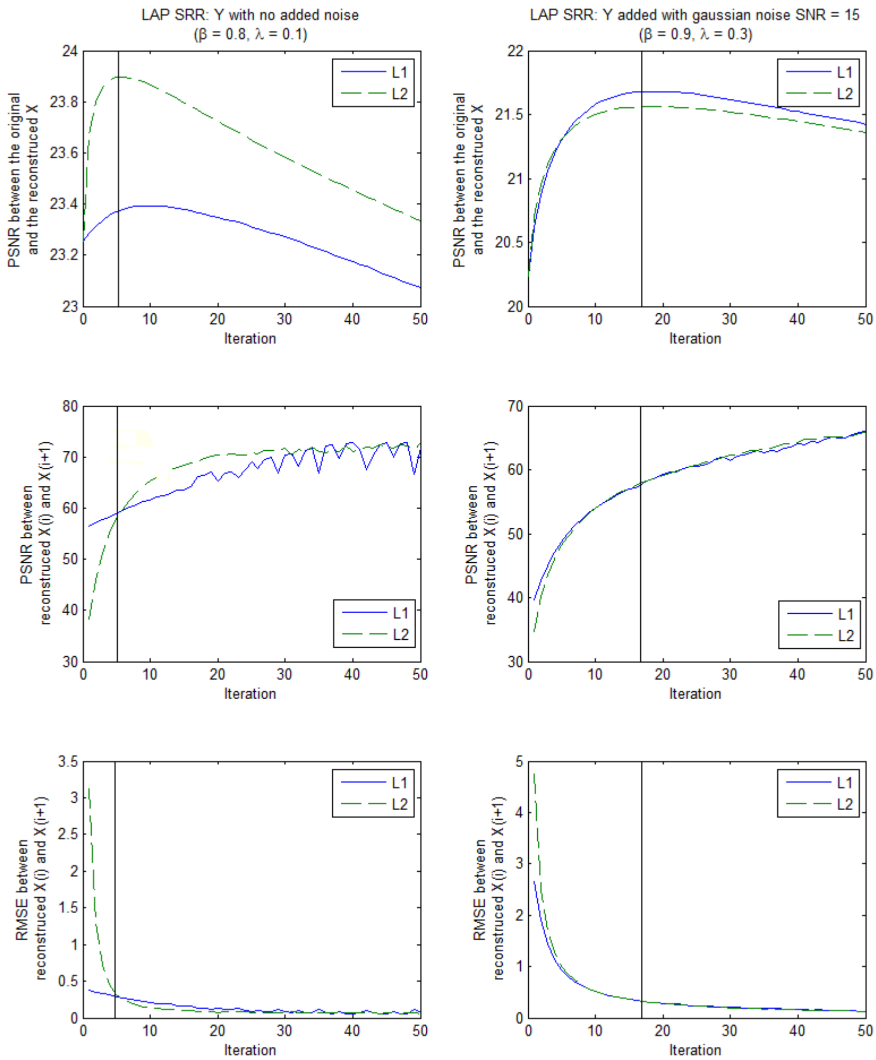


Figure 4.29: Analysis between the reconstructed images (lena image, noise case 1 & 4)

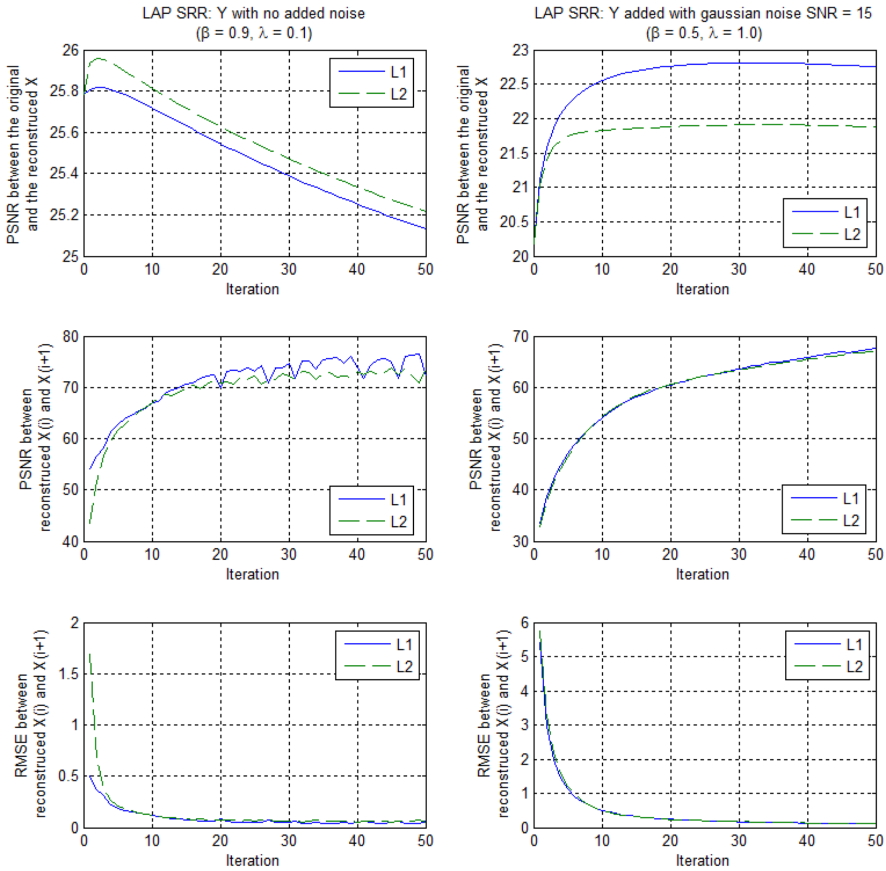


Figure 4.30: Analysis between the reconstructed images (alumgrns image, noise case 1 & 4)

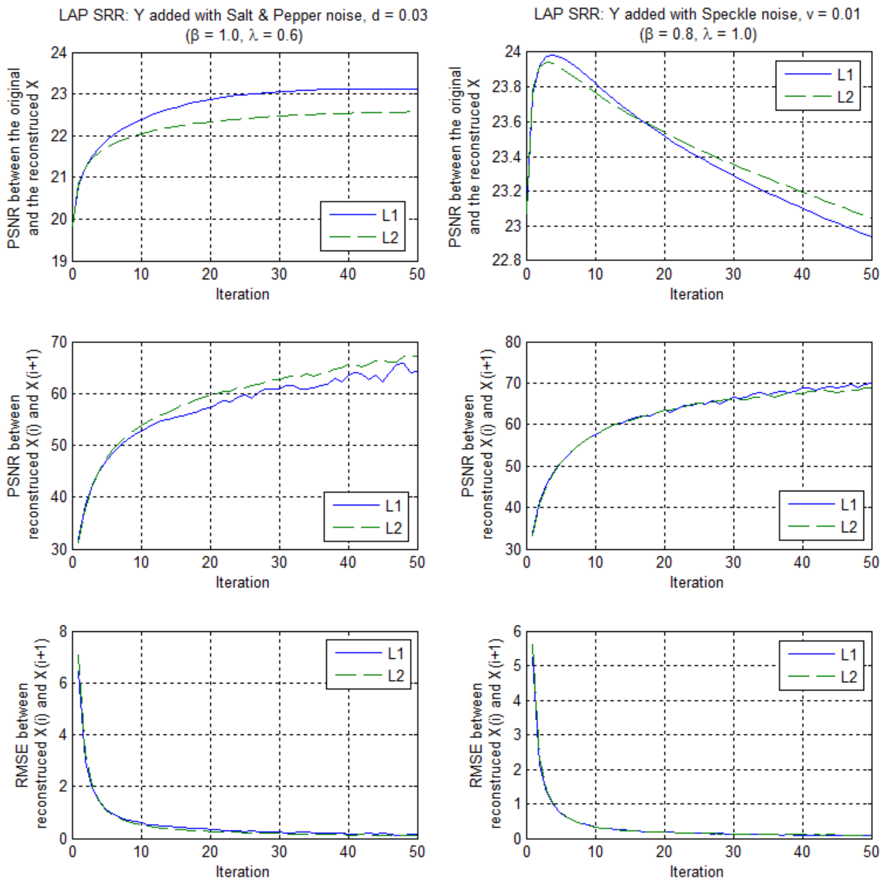


Figure 4.31: Analysis between the reconstructed images (alumgrns image, noise case 7 & 8)

Experimental analysis

Apparently it is possible to find an optimum reconstruction (or close to the one) by analysing the value of the PSNR or RMSE between the reconstructed images. It is the case even by using the L1 norm estimation or the L2.

For example, for the simulation on figure 4.29, a threshold of 0.3 in the RMSE looks good.

In the second simulation (alumgrns image, fig. 4.30), the thresholds seem to be the same (RMSE near 0.3, PSNR near 60). But this will not mean it will be the same with every case.

The last simulation (fig. 4.31) shows that the thresholding is still possible but not with the same threshold (for example, near 65 for PSNR in the Salt & Pepper noise case, and near 50 in the Speckle noise case). With RMSE, the thresholds change also (logic, they are mathematically related).

So it looks possible to find the best reconstructed image by analysing the difference between them. But for any cases, training of similar images to find the threshold (PSNR or RMSE) is necessary to have good results.
It is interesting to note that the thresholds seems to be the same (or close) with L1 and L2 norm estimation (for PSNR and RMSE).

4.4.2 Experimental comparisons on real frames set No 1

This comparison has results shown on Fig. 4.32 and 4.33. The first one shows the evolution of delta (RMSE between the reconstructed images) for all algorithms with the Laplacian regularization. The second figure shows some of the reconstructed images with different delta thresholds. The step β will be 1.0 and the regularization weight λ 0.1 for all the SRR.

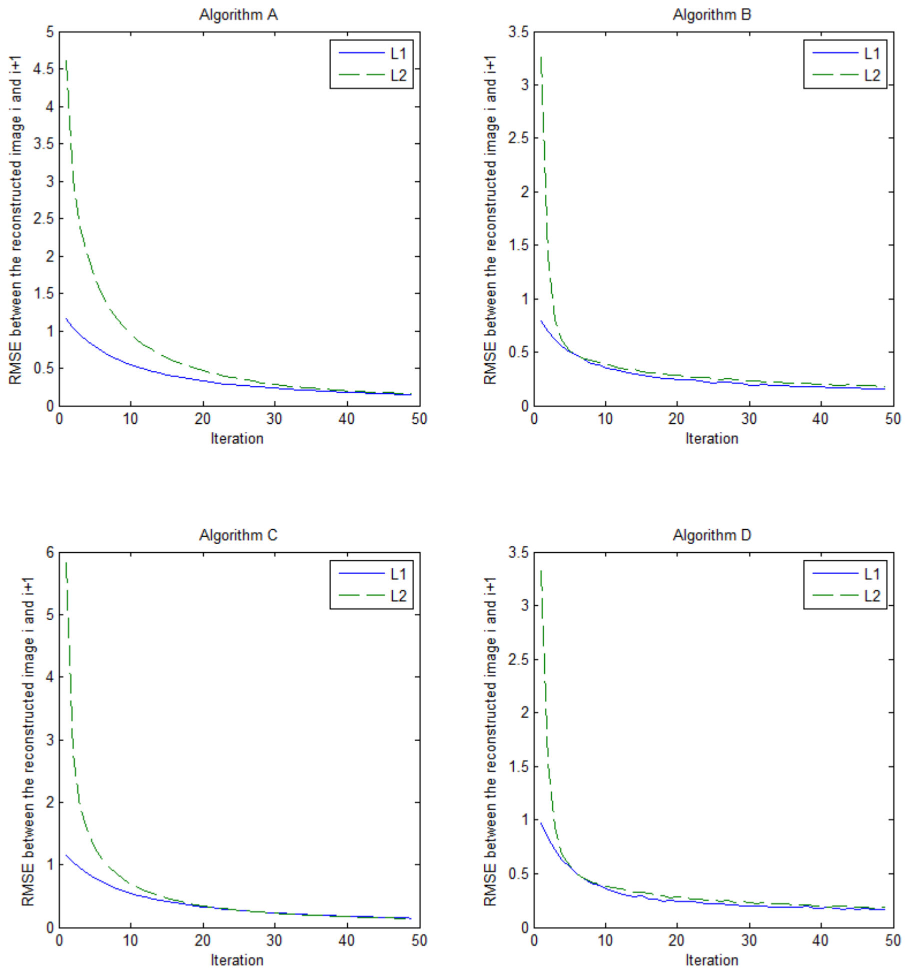


Figure 4.32: RMSE between the reconstructed images for the real frames set 1 using the Laplacian regularization (A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

Figure 4.33: Reconstructed images of the real frames set 1 (delta = RMSE between the reconstructed images, i = iteration, A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

Experimental analysis

As mentioned before, find the best reconstructed image by visual analysing is really difficult, if not impossible. No training was made before, so it is not possible to know at with delta (or iterations number) the algorithms should stop.
But even with that, the results show that the "Fast SRR algorithm based on Recursive Least Square" and the "Proposed SRR algorithm" are better than the others here.
The "SRR algorithm based on Simultaneous Image Registration" is visually better than the "Classical SRR algorithm", but to find the best between the "Fast SRR algorithm based on Recursive Least Square" and the "Proposed SRR algorithm" might be impossible in this case.

4.4.3 Experimental comparisons on real frames set No 2

The second real frames set have larger images (68x68 pixels). The processed image in vector format is than 136^2x1 = 18'496x1 (upsampling factor of 2). For example, the blur matrix would have a size of 18'496x18'496. The computation time would be too long and the computer memory to limited if the algorithm dealt directly with the frames. To reduce the computational time, the frames are fragmented to 20x20 pixels parts with an overlap of 8 pixels (40x40 pixels with an overlap of 16 pixels after the upsampling). The blur matrix is than just 1600x1600 large.

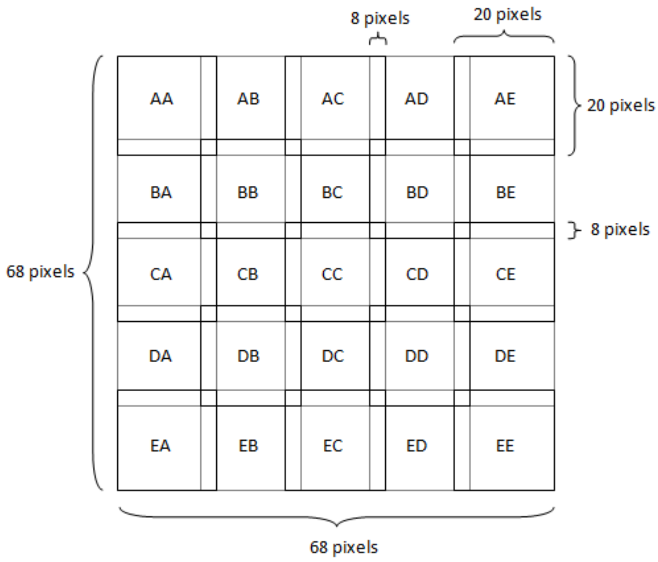


Figure 4.34: Separated parts of the frames set No 2

After super resolution reconstruction, the parts will be merged again to have the final reconstructed image (136x136 pixels). For the overlap, it will use the half width with one of the reconstructed part and the second half width with the next part.

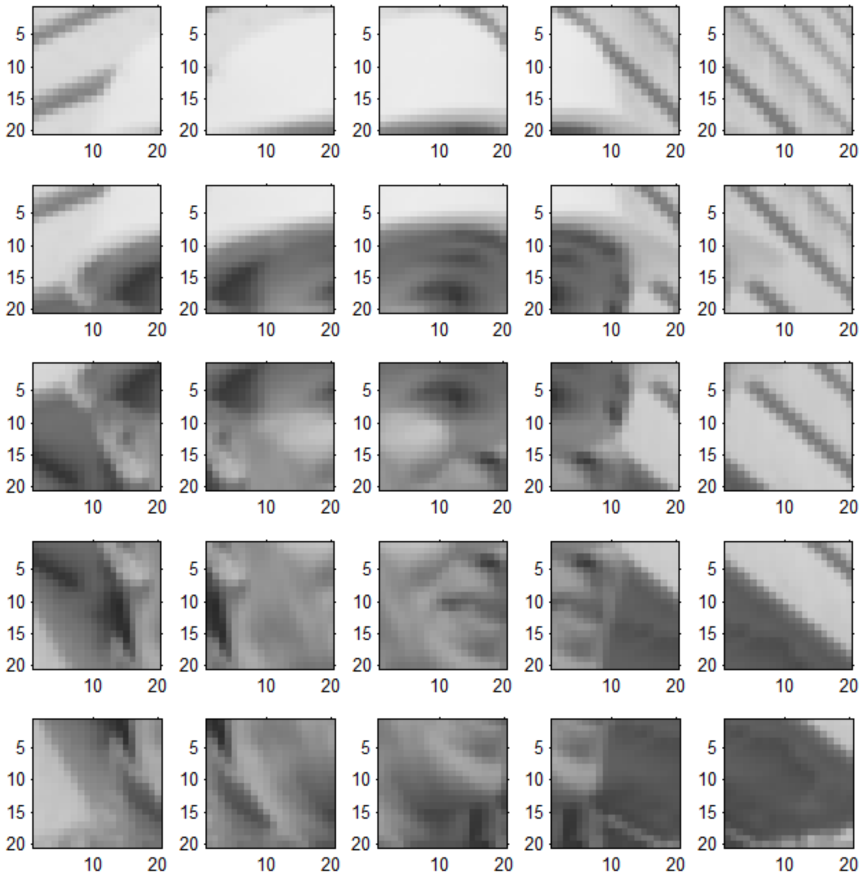


Figure 4.35: Separated parts of the frame No 110

Figure 4.36: Reconstructed images of the real frames set 2 (all algorithms) (A is for the "Classical SRR algorithm", B the "Fast SRR algorithm based on Recursive Least Square", C the "SRR algorithm based on Simultaneous Image Registration" and D the "Proposed SRR algorithm")

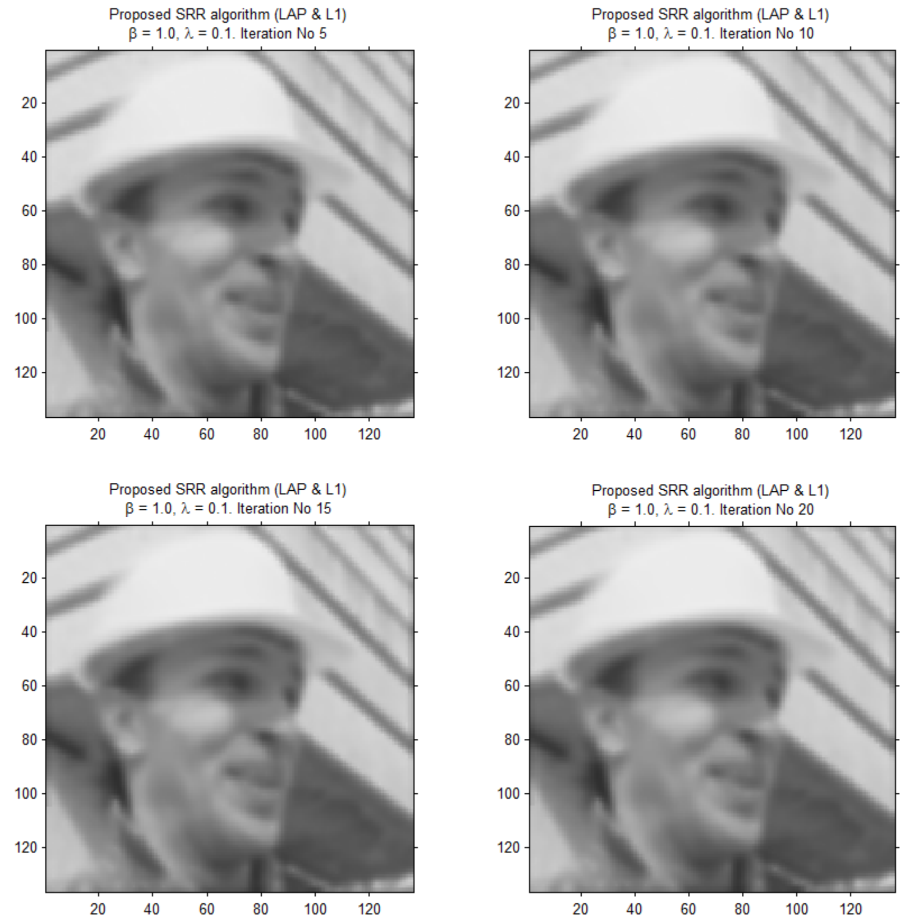


Figure 4.37: Reconstructed images of the real frames set 2 (proposed algorithm)

Experimental analysis

As for the real frames set No 1, it is hard to see any difference between the algorithms, and the different reconstructed images for the proposed algorithm. The only visible difference is the mouth between the "Classical SRR algorithm" and the "SRR algorithm based on Simultaneous Image Registration" against the "Fast SRR algorithm based on Recursive Least Square" and the "Proposed SRR algorithm". It is the case because the two first use the frame No 110 and add informations of the other frames in the steepest descent, and the two last algorithms use a frames fusion at the begin of the steepest descent method.

Chapter 5

Conclusion

This thesis present the Super Resolution Reconstruction and more specifically the Super Resolution Reconstruction using stochastic regularization approach. For this, four algorithms have been reviewed, and also tree types of regularization functions and two norm estimations.

This paper shows how Super Resolution Reconstruction using stochastic regularization approach works and why the choices of the parameters, the regularization function and the norm estimation are important.

But mainly, this work is about to introduce a "Proposed SRR algorithm" based on recursive least square estimation and simultaneous image registration and to test it.

The proposed algorithm is based on the "Fast SRR algorithm based on Recursive Least Square" and the "SRR algorithm based on Simultaneous Image Registration". So, several simulations were used to find the advantages or weak points inherited by these algorithms. This gives the following assumptions:
As the "Fast SRR algorithm based on Recursive Least Square", the proposed algorithm is quiet fast. Sure it is slower than the first because of the iterative motion estimation, but faster than the "Classical SRR algorithm" and the "SRR algorithm based on Simultaneous Image Registration".
For the iterative motion estimation on the "SRR algorithm based on Simultaneous Image Registration", and also implemented in the proposed algorithm, the simulations show that it is closer to real application by finding better motion estimations. For some experiments, the iterative motion estimation for the Salt & Pepper noise is some times not so good than the first one. It is probably because it uses a cross correlation motion estimation method. This algorithm might be not the best for high noised images registration. It should be interesting to use the algorithms with another registration algorithm.
Starting with a frames fusion, it makes the "Fast SRR algorithm based on Recursive Least Square" and the proposed algorithm more robust to noise than the "Classical SRR algorithm" and the "SRR algorithm based on Simultaneous Image Registration". The two first algorithms will start the steepest descent

with an image created from all the frames. The random noise on each frame will be smaller on the start image. But the start point of the steepest descent for the "Classical SRR algorithm" and the "SRR algorithm based on Simultaneous Image Registration" is the first input frame. So if there is noise on this image, it will be also on the first step of the reconstructed image.
The simulation could give the impression that the proposed algorithm is really better than the other ones. But it should be interesting to run for example the "SRR algorithm based on Simultaneous Image Registration" with the frames fusion image as start point for the steepest descent.

So the proposed SRR algorithm based on recursive least square estimation and simultaneous image registration is a fast algorithm, robust to noise and giving better reconstructed images tanks to the iterative motion estimation. This makes it particularly valuable for real practical use.

Directions for future work:

In addition to developing more the proposed SRR algorithm, it would be interesting to continue to develop the framework for the detection of the best reconstruction image without knowing the original image. This framework having already given good results (section 4.4.1).

Appendix A

SRR Toolbox for Matlab

For the need of the experimental comparisons, a toolbox for Matlab was created. It uses also functions of the Image Processing Toolbox of MathWorks, functions of the SRR toolbox by Vorapoj Patanavijit, functions of the SRR toolbox by Hiroyuki Takeda [15] and the function “dftregistration()” created by Manuel Guizar (Efficient subpixel image registration by cross-correlation) [6].

Rules for using the SRR toolbox

There is some points to take care to use the toolbox:

- The input images have to be square images (dimX = dimY)
- The upsampling factor has to be 0, 2, 4, 8, etc.
- The toolbox was created under Matlab v.7.7 but some used functions were created for older versions. The toolbox work well under v.7.7 but there is no guaranties that it also work under older versions (but it should).

The next two pages will give a list of the functions in the toolbox and a short explication of how to use it. For more informations about each function, see the help inside every m-file.

Functions names

SRR_AlgoA_Laplacian.m	**SRR_AlgoB_Laplacian.m**	**SRR_AlgoC_Laplacian.m**	**SRR_AlgoD_Laplacian.m**
Using: SRR_AlgoA_SDloop_RegMatGiven.m SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegMatGiven SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m	Using: SRR_AlgoA_SDloop_RegMatGiven.m SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegMatGiven SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m
SRR_AlgoA_MRF.m	**SRR_AlgoB_MRF.m**	**SRR_AlgoC_MRF.m**	**SRR_AlgoD_MRF.m**
Using: SRR_AlgoA_SDloop_RegMatGiven SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegMatGiven SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m	Using: SRR_AlgoA_SDloop_RegMatGiven SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegMatGiven SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m
SRR_AlgoA_BTV.m	**SRR_AlgoB_BTV.m**	**SRR_AlgoC_BTV.m**	**SRR_AlgoD_BTV.m**
Using: SRR_AlgoA_SDloop_RegBTV.m SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegBTV.m SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m	Using: SRR_AlgoA_SDloop_RegBTV.m SRR_AlgoA_FidelityTerm_WithoutMotion.m SRR_AlgoA_FidelityTerm_WithMotion.m	Using: SRR_AlgoB_SDloop_RegBTV.m SRR_AlgoB_FidelityTerm_WithoutMotion.m SRR_AlgoB_FidelityTerm_WithMotion.m
SRR_AlgoA_CKR.m	**SRR_AlgoB_CKR.m**	**SRR_AlgoC_CKR.m**	**SRR_AlgoD_CKR.m**
Using: ckr2L1_regular.m* ckr2L2_regular.m*	Using: ckr2L1_regular.m* ckr2L2_regular.m*	Using: ckr2L1_regular.m* ckr2L2_regular.m*	Using: ckr2L1_regular.m* ckr2L2_regular.m*
SRR_AlgoA_SKR.m	**SRR_AlgoB_SKR.m**	**SRR_AlgoC_SKR.m**	**SRR_AlgoD_SKR.m**
Using: skr2L1_regular.m* skr2L2_regular.m*	Using: skr2L1_regular.m* skr2L2_regular.m*	Using: skr2L1_regular.m* skr2L2_regular.m*	Using: skr2L1_regular.m* skr2L2_regular.m*

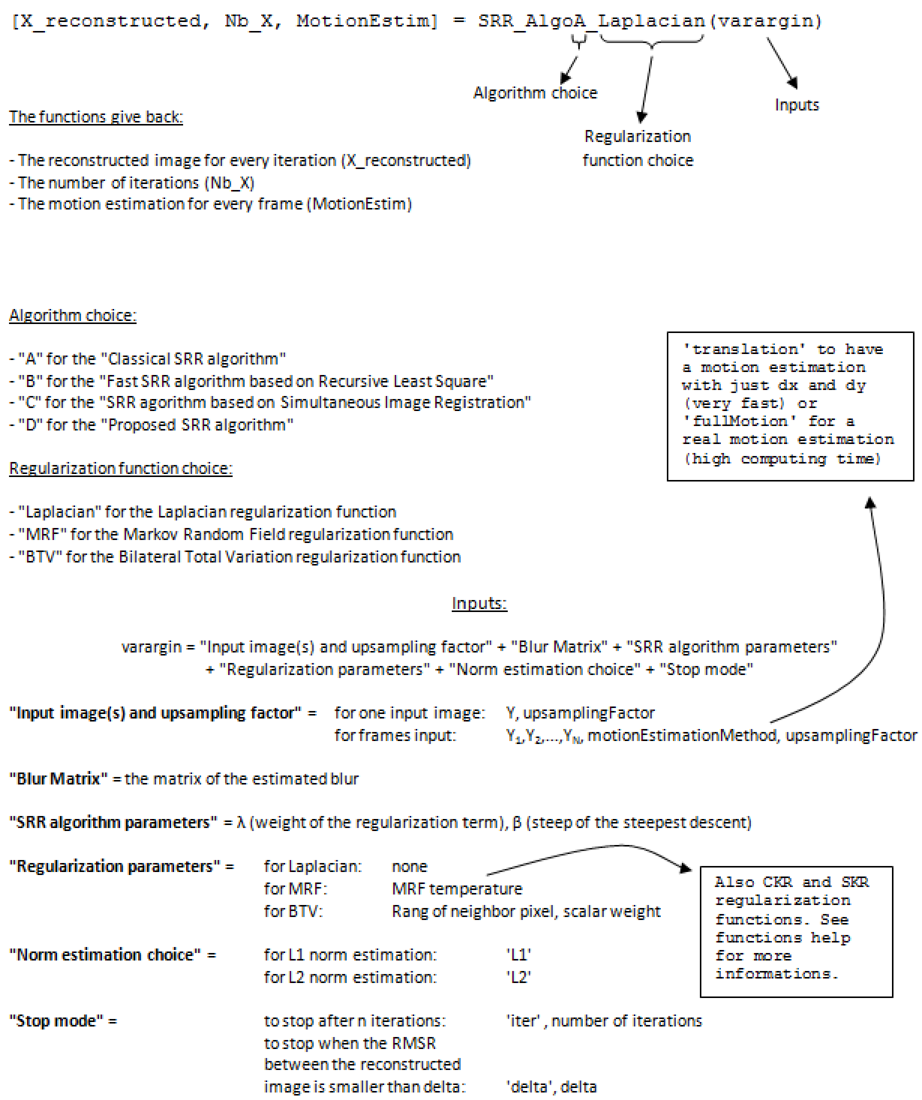
* = Modified function from the SRR toolbox by M.Takeda

And using:

Conversion_from_Matrix_to_Vector.m	
Conversion_from_Vector_to_Matrix.m	**And also:**
ImageDownSampling.m	"dftregistration.m",
ImageUpSampling.m	the SRR toolbox of M.Hiroyuki Takeda,
PSNR_2img.m	the SRR toolbox of professor Vorapoj Patanavijit
RMSE_2img.m	and the MathWorks's "Image processing toobox"

Figure 5.1: Functions of the SRR toolbox.

How to use the SRR toolbox



[X_reconstructed, Nb_X, MotionEstim] = SRR_AlgoA_Laplacian(varargin)

Algorithm choice

Regularization function choice

Inputs

The functions give back:

- The reconstructed image for every iteration (X_reconstructed)
- The number of iterations (Nb_X)
- The motion estimation for every frame (MotionEstim)

Algorithm choice:

- "A" for the "Classical SRR algorithm"
- "B" for the "Fast SRR algorithm based on Recursive Least Square"
- "C" for the "SRR algorithm based on Simultaneous Image Registration"
- "D" for the "Proposed SRR algorithm"

'translation' to have a motion estimation with just dx and dy (very fast) or 'fullMotion' for a real motion estimation (high computing time)

Regularization function choice:

- "Laplacian" for the Laplacian regularization function
- "MRF" for the Markov Random Field regularization function
- "BTV" for the Bilateral Total Variation regularization function

Inputs:

varargin = "Input image(s) and upsampling factor" + "Blur Matrix" + "SRR algorithm parameters" + "Regularization parameters" + "Norm estimation choice" + "Stop mode"

"Input image(s) and upsampling factor" = for one input image: Y, upsamplingFactor
for frames input: $Y_1, Y_2, \ldots, Y_N$, motionEstimationMethod, upsamplingFactor

"Blur Matrix" = the matrix of the estimated blur

"SRR algorithm parameters" = λ (weight of the regularization term), β (steep of the steepest descent)

"Regularization parameters" = for Laplacian: none
for MRF: MRF temperature
for BTV: Rang of neighbor pixel, scalar weight

Also CKR and SKR regularization functions. See functions help for more informations.

"Norm estimation choice" = for L1 norm estimation: 'L1'
for L2 norm estimation: 'L2'

"Stop mode" = to stop after n iterations: 'iter', number of iterations
to stop when the RMSR between the reconstructed image is smaller than delta: 'delta', delta

Figure 5.2: How to use the SRR toolbox.

Appendix B

Steering Kernel Regression

Steering kernel regression (or data-adapted kernel regression) works as an improvement of bilateral filtering.

Bilateral filter was introduced by Tomasi et al. [16]. It uses the concept of Gaussian smoothing by weighting it with the radiometric value of each pixel. So it reduces the noise while preserving edges [17].

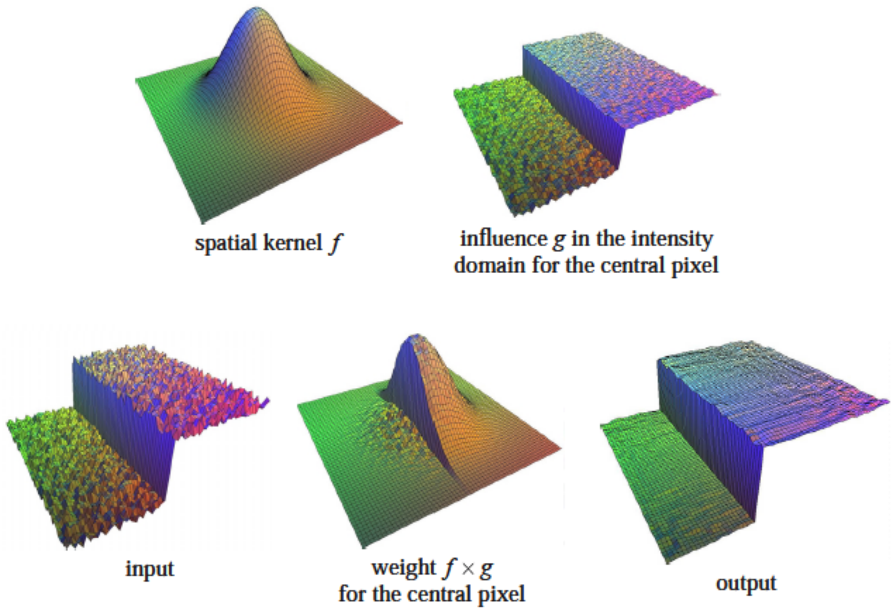


Figure 5.3: Bilateral filter operations (Image source: [18]).

Indeed, with great texture or an image with much noise makes badly the use of bilateral filtering. Data adapted kernel regression also uses a Gaussian kernel, but this core is modified in this case at each location by the texture of the image. If at any point of the image there is an edge, the Gaussian kernel will be stretched and will be rotated so that the filter preserves the edges (based on local gradient fields). In addition, the kernel size is changed according to the image (if it is an area with little information there will be greater than if it is an area with lots of texture) [19] [15] [20].

The data-adapted kernel at each point $\boldsymbol{K_i}$ is given by:

$$K_i(x_i - x) = \frac{\sqrt{det(C_i)}}{2\pi h^2 \mu_i^2} \cdot exp\left\{-\frac{(x_i - x)^T C_i (x_i - x)}{2h^2\mu_i^2}\right\} \tag{5.1}$$

where $\boldsymbol{C_i}$ is the covariance matrix based on the local values. This matrix can be decompose as follow:

$$C_i = \gamma_i U_{\Theta_i} \Lambda_i U_{\Theta_i}^T \tag{5.2}$$

and:

$$U_{\Theta_i} = \begin{bmatrix} cos\Theta_i & sin\Theta_i \\ -sin\Theta_i & cos\Theta_i \end{bmatrix} \tag{5.3}$$

$$\Lambda_i = \begin{bmatrix} \rho_i & 0 \\ 0 & \rho_i^{-1} \end{bmatrix} \tag{5.4}$$

where $\boldsymbol{U_{\Theta_i}}$ is the rotation matrix and $\boldsymbol{\Lambda_i}$ the elongation matrix. So the covariance matrix is given by three parameters: the scaling $\boldsymbol{\gamma_i}$, the rotation $\boldsymbol{\Theta_i}$ and the elongation $\boldsymbol{\rho_i}$ parameters. This terms are principally compute from the eigenvalues and eigenvectors of the local gradient fields.

Bibliography

[1] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito. Signal-processing based method for acquiring very high resolution image with multiple cameras and its theoretical analysis. *Proc. Inst. Elec. Eng.*, 140(1):19–25, Feb 1993.

[2] S.C. Park, M.K. Park, and M.G. KANG. Super-resolution image reconstruction: A technical overview. *IEEE Signal Processing Magazine*, 20:21–36, May 2003.

[3] Vorapoj Patanavijit. Super-resolution reconstruction and its future research direction. *AU Journal of Technology (AU J.T.), Assumption University*, pages 149–163, Jan 2009. `www.journal.au.edu`.

[4] Vorapoj Patanavijit. Mathematical analysis of stochastic regularization approach for super-resolution reconstruction. *AU Journal of Technology (AU J.T.), Assumption University*, pages 235–244, Apr 2009. `www.journal.au.edu`.

[5] C. E. Shannon. Communication in the presence of noise. *Proc. IEEE*, 86(2), Feb 1998. Reprint as classic paper. (Original: Proc. Institute of Radio Engineers, vol. 37, no.1, pp. 1021, Jan. 1949.).

[6] Manuel Guizar. Efficient subpixel image registration by cross-correlation. `http://www.mathworks.com/matlabcentral/fileexchange/authors/16644`, Jan 2008.

[7] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Trans. on Image Processing*, 13(10):1327–1344, Oct 2004.

[8] M. Elad and A. Feuer. Superresolution restoration of an image sequence: adaptive filtering approach. *IEEE Trans. on Image Processing*, 8(3):387–395, 1999.

[9] N. Nguyen, P. Milanfar, and G. Golub. A computationally efficient super-resolution image reconstruction algorithm. *IEEE Trans. on Image Processing*, 10:573–583, Apr 2001.

[10] Ross Kindermann and J. Laurie Snell. *Markov Random Fields and Their Applications.* AMS, 1980. Number of Pages: 142pp, ISBN: 0-8218-3381-2.

[11] Michael K. Ng, Huanfeng Shen, Edmund Y. Lam, and Liangpei Zhang. A total variation regularization based super-resolution regularization based

super-resolution. *EURASIP Journal on Advances in Signal Processing*, pages 1–17, 2007. Article ID 74585, ISSN: 16876172.

[12] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *Electrical and ELectronic Engineers in Israel, 2000. The 21st IEEE Convention of the*, pages 402–405, Apr 2000.

[13] M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur. *Image Processing, IEEE Transactions on*, 10:1187–1193, Aug 2001. Issue 8.

[14] Yu He, Kim-Hui Yap, Li Chen, and Lap-Pui Chau. A nonlinear least square technique for simultaneous image registration and super-resolution. *Image Processing, IEEE Transactions on*, 16:2830–2841, Nov 2007. Issue 11.

[15] Hiroyuki Takeda. Kernel regression for image processing and reconstruction. Master's thesis, University of California Santa Cruz, Mar 2006.

[16] C.Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Sixth International Conference on Computer Vision, New Delhi, India*, pages 839–846, Jan 1998.

[17] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11.

[18] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *Laboratory for Computer Science, Massachusetts Institute of Technology, ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, 2002.

[19] Hiroyuki Takeda. Kernel regression for image processing and reconstruction. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 16(2), Feb 2007.

[20] Hiroyuki Takeda. *Locally adaptive kernel regression methods for multidimensional signal processing.* PhD thesis, University of California Santa Cruz, Sep 2010.

Members of the jury:

Prof. Dr. Stephan Robert (HEIG-VD, Switzerland)
Prof. Dr. Hervé Dedieu (HEIG-VD, Switzerland)
Asst. Prof. Dr. Vorapoj Patanavijit (AU, Thailand)
Asst. Prof. Dr. Kittiphan Techakittiroj (AU, Thailand)

Bangkok
August 4, 2011

Stéphane Kirchner