Image Reconstruction and Restoration for Optical Flow-based Super Resolution

Master Thesis

Universitat Politècnica de Catalunya – ETSETB Barcelona

Technicolor - Hannover

Supervised by

Axel Kochale and Jordi Salvador from Technicolor

Javier Ruiz from UPC

Daniel Rivero Alfonso

18/04/2012
Acknowledgement

Technicolor and UPC of Barcelona (ETSETB) gave me this amazing opportunity. These words are for them, especially for Axel Kochale and Jordi Salvador (from Technicolor) and for Javier Ruiz and Ferran Marquès (from UPC). Without them this project could never have been developed and the experience I achieved in Hannover is thanks to all of them. Also I want to thank my university and all my professors who gave me the enough knowledge during the last five years.
Abstract

Super Resolution has been one of the main topics of research in Image Processing in the last twenty years. Different authors have presented a vast amount of different techniques and approaches. Nowadays reconstruction-based Super Resolution works effectively for synthetic data but practical problems appear in real scenarios with objects moving with arbitrary speeds and directions within the scene. With our work, we present a review of the state-of-the-art in reconstruction-based Super Resolution, we make contributions to the involved stages (registration, reconstruction and restoration) and finally, we introduce some initial ideas for future work leading to effective Super Resolution even in sequences with highly complex motion.
Table of contents

1. Introduction .......................................................................................................................... 11
   1.1. What is Super Resolution? ......................................................................................... 12
   1.2. Goals ............................................................................................................................ 14
   1.3. Thesis structure ........................................................................................................... 14
2. State of the art ...................................................................................................................... 15
   2.1. Registration .................................................................................................................. 15
   2.2. Reconstruction ............................................................................................................. 18
   2.3. Restoration .................................................................................................................. 21
3. Registration .......................................................................................................................... 25
   3.1. The necessity to register multi-frames ........................................................................ 25
   3.2. Motion estimation: Optical Flow ................................................................................ 27
       3.2.1. Global motion or local motion .............................................................................. 28
       3.2.2. Lucas-Kanade, Horn-Schunck and Classic Non-Local .......................................... 29
4. Reconstruction ...................................................................................................................... 37
   4.1. Warping ........................................................................................................................ 37
       4.1.1. First approach – nearest neighbour .................................................................... 39
       4.1.2. Final approach – bilinear .................................................................................... 40
   4.2. Inpainting ..................................................................................................................... 42
       4.2.1. Effect of the regularizer ....................................................................................... 42
5. Restoration: Deblurring ........................................................................................................ 49
   5.1. PSF estimation .............................................................................................................. 50
   5.2. Deblurring ..................................................................................................................... 54
6. Restoration: Denoising ......................................................................................................... 59
7. Experimental results .............................................................................................................. 65
   7.1. Qualitative results ......................................................................................................... 65
       7.1.1. Disk sequence ..................................................................................................... 66
       7.1.2. Book sequence ................................................................................................... 67
       7.1.3. Surveillance sequences ....................................................................................... 68
       7.1.4. Face sequence ..................................................................................................... 69
       7.1.5. Text sequence ..................................................................................................... 69
   7.2. Quantitative results ....................................................................................................... 71
       7.2.1. Butterfly .............................................................................................................. 71
7.2.2. Princeton..............................................................................73
7.3. Impact of each stage.................................................................75
8. Conclusions..............................................................77
   8.1. Achievements ..................................................................77
   8.2. Future work .................................................................78
9. Bibliography.............................................................................83
List of figures
Figure 1.1. Example of single-image SR.................................................................12
Figure 1.2. Outliers in single-image SR.................................................................13
Figure 1.3. Multi-frame SR ..................................................................................13
Figure 1.4. SR block ..............................................................................................13
Figure 2.1. Four steps of image registration .........................................................16
Figure 2.2. Reconstruction ....................................................................................18
Figure 2.3. Blurring example ................................................................................21
Figure 2.4. Image Noise ........................................................................................22
Figure 3.1. Registration block ..............................................................................25
Figure 3.2. Idea of registration .............................................................................26
Figure 3.3. Image registration example .................................................................26
Figure 3.4. Optical flow .........................................................................................28
Figure 3.5. Pyramidal Lucas-Kanade algorithm ..................................................30
Figure 3.6. Colour reference according for the Optical Flow result ...................32
Figure 3.7. Optical Flow ground truth of Teddy sequence ....................................32
Figure 3.8. Optical Flow result for the different algorithms ...............................33
Figure 3.9. References images ............................................................................34
Figure 3.10. Result of Optical Flow .....................................................................34
Figure 3.11. References images ...........................................................................34
Figure 3.12. Result of Optical Flow ....................................................................34
Figure 4.1. Reconstruction block .........................................................................37
Figure 4.2. LR grid and SR grid ..........................................................................38
Figure 4.3. Process to create the SR image .........................................................38
Figure 4.4. Problem of the nearest neighbour method .......................................40
Figure 4.5. Result obtained with the nearest approach and the bilinear approach ..41
Figure 4.6. Result obtained with TV and TK .......................................................46
Figure 4.7. Result obtained with the first derivative and the second derivative ....47
Figure 4.8. Convergence graph of TV

Figure 5.1. Deblurring block

Figure 5.2. Deblurring example

Figure 5.3. Sharp Edge Prediction

Figure 5.4. Canny edge detection

Figure 5.5. Profile of the sharp image

Figure 5.6. Original PSF and estimated PSF

Figure 5.7. Deblurring method applied in the Pepper image

Figure 5.8. Result for the deblurring process

Figure 6.1. Denoising block

Figure 6.2. Noisy image

Figure 6.3. Median filter process

Figure 6.4. A constant weight 3x3 filter mask

Figure 6.5. Test of denoising: noisy image

Figure 6.6. Test of denoising: results

Figure 7.1. Disk sequence

Figure 7.2. Results for the disk sequence

Figure 7.3. Book sequence

Figure 7.4. Results for the book sequence

Figure 7.5. Surveillance sequence

Figure 7.6. Results for the surveillance sequence

Figure 7.7. Face sequence

Figure 7.8. Results for the face sequence

Figure 7.9. Text sequence

Figure 7.10. Results for the text sequence

Figure 7.11 and Table 7.1. Butterfly test: LR image and their parameters

Figure 7.12. Butterfly test: Ground Truth
Figure 7.13. Butterfly test: results.................................................................72
Figure 7.14 and Table 7.3. Princeton test: LR image and their parameters........73
Figure 7.15. Princeton test: Ground Truth ..................................................74
Figure 7.16. Princeton test: results ...............................................................74
Figure 7.17. Histograms for each stage .........................................................76
Figure 8.1. Yoyo sequence ..........................................................79
Figure 8.2. Yoyo sequence: results for segmentation .................................80
Figure 8.3. Coloured background ...............................................................81
Figure 8.4. Over segmented LR image .......................................................81
Figure 8.5. Motion segmentation ...............................................................82
List of tables

Table 3.1...........................................................................................................................................35
Table 5.1.........................................................................................................................................57
Table 7.1.........................................................................................................................................68
Table 7.2.........................................................................................................................................70
Table 7.3.........................................................................................................................................70
Table 7.4.........................................................................................................................................72
1. Introduction

In this chapter we introduce the Super Resolution (SR) concept. This has been one of the main research topics in the field of image processing during the last twenty years. SR is an ill-posed problem, thus accepting an undefined number of plausible solutions. This thesis proposes a novel approach to solve this problem. Furthermore, we explain which are the goals based in our motivation.

New technology and consumers’ viewing habits are changing the face of TV. What was once thought of as a mere possibility is now a reality, as consumers want more choices and flexibility in their viewing experiences. The introduction of High Definition displays has led to advances in the field of SR and constant improvements are achieved, which provide new mechanisms for enriching visual contents and match these new display capabilities.

Digital photography has also made immense progress over the past 15 years, particularly in increasing resolution and improving low-light performance. But continuing progress in these aspects of performance is not possible, because cameras are now approaching the limits imposed by the basic nature of light. More pixels can only improve resolution significantly at wider apertures and if new lenses are up to the task. In addition, those physics limitations in image sensors introduce noise, which contaminates images. Thus, users need techniques to overcome those limitations and enjoy the TV of the future: more quality, more resolution and more interactivity. Beyond that, there is still a demand for increasing the resolution of images obtained using the current technology in lots of fields, like medicine, security, free-viewpoint video or others.

The main reason SR is ill-posed is because of the nature of the input sequences. These do not provide enough information to completely recover a High Resolution (HR) image. Furthermore, images captured by a camera are contaminated by additional noise and undesired blur. To face up those problems, we present here an approach to reconstruct and restore high-resolution video sequences out of low-resolution data.

Most current techniques perform properly with synthetic data, but, what happens when the input data are as complex as a real sequence from e.g. a football match? At the end of this thesis we present the outline of an approach to tackle the problem of processing video sequences with arbitrary motion. This would offer the user the possibility to interact with high quality. Hence, our work is motivated by some advances in a EU funded project called FascinatE requiring the improvement of the perceived quality of freely chosen views (free-viewpoint video) in a SR framework. FascinatE [41] is a European project involving eleven partners leading to new systems to allow end-users to interactively view and navigate around an ultra-high resolution video panorama showing a live event, with the accompanying audio automatically changing to match the selected view. The output will be adapted to the user’s particular kind of device, covering anything from a mobile handset to an immersive panoramic display.
1.1. What is Super Resolution?

SR is a term for referring to methods for upscaling video or images. Most SR techniques are based on the same idea: using information from several different images to create one upscaled image. Thus, algorithms try to extract information from every image in a sequence to reconstruct other frames.

A High Resolution (HR) image brings out details that would be blocked out in a Low Resolution (LR) image. More pixels in the same area imply a higher sampling frequency thereby offering more details.

Now that we have defined the concept, it is important to mention the two kinds of SR: single-image SR and multi-frame SR. We focus on the latter, but, firstly, we present a short introduction to the alternative former family of single-image SR methods (also known as Example-Based Super Resolution).

Single-image SR was first introduced by Freeman et al. in [1]. SR relates to image interpolation with two complementary ways for increasing resolution: sharpening by amplifying existing image details or estimating missing HR detail that is not present in the original image and cannot be visible by simple sharpening. In Figure 1.1 we show an example of single-image SR with two different methods: bicubic interpolation and their proposed one-pass SR interpolation.

![Figure 1.1 extracted from paper [1]. (a) Original image, (b) cubic interpolation, (c) one-pass SR interpolation](image)

In Figure 1.1 it is not noticeable, but single-images SR algorithms have a big problem with noise. When noise appears in the original image, it cannot be automatically removed in the Super Resolved output, because it is treated as a signal and it is amplified as well. In Figure 1.2 we can observe how JPEG compression artifacts are very visible in the SR images obtained with this approach.
This thesis focuses on the classical multi-image reconstruction-based SR, which obtains each output HR frame from multiple LR frames in a video sequence. A great variety of methods are published but the underlying model relating HR image and LR image is pretty standard. Most of the previous work assumes that the underlying motion has a simple parametric form, and/or that the blur kernel and noise levels are known. But in practice, the motion of objects and cameras can be arbitrary, the video may be contaminated with noise of unknown level, and motion and camera lens blur can lead to an unknown blur kernel. Therefore, we can estimate all those effects in three different steps that later will be explained: Registration, Reconstruction and Restoration.

In Figure 1.3, we sketch how multi-frame SR works: a SR image is obtained by fusing the information contained in a number of sub-pixel misaligned LR images. Finally, in Figure 1.4 we can observe the classical SR algorithm block diagram, separately involving the three steps mentioned above.

Set of LR images

![Set of LR images](image)

SR image

![SR image](image)
Beyond that, a question that naturally arises when working in SR is how to achieve maximum performance with minimum computational cost with a given imaging system and a given scene. That is, we want the best resolution and image quality without wasting resources. To reach this goal, we need to answer the following questions:

- Which is the maximum real increment in resolution we can achieve?
- How many LR images are needed to maximize the image resolution and quality? Is there a bound from which it makes no sense to add more LR images?

However, we will answer these questions in Chapter 8.

### 1.2. Goals

The main goal of this thesis is to propose a complete procedure to solve the problem of SR for video sequences. In our work we present various approaches for both the reconstruction and the restoration steps. The main goals of this thesis are:

- Implement an entire SR system for video sequences including the registration, reconstruction and restoration steps.
- Choose a robust algorithm for registration among state-of-the-art techniques.
- Define an effective strategy to reconstruct the HR image.
- Develop algorithms capable of improving the quality of the reconstructed image (restoration step).
- Detect elements of SR that could be improved leading to the processing of complex real video sequences.

### 1.3. Thesis structure

This thesis is organized in 8 chapters, including this Introduction. In Chapter 2, *State of the art*, we explain some references in SR. In Chapter 3, *Registration*, we explain how motion is estimated from multiple video frames. In Chapter 4, *Reconstruction*, we present our approach based on *Inpainting*. In Chapter 5, *Restoration*, we explain the last step of SR, which includes *Deblurring* and *Denoising* to compensate the blur and noise in LR images. In Chapter 6, we present our experimental results and some comparisons with state-of-the-art methods. Finally, in Chapter 7, we extract some conclusions and propose lines of future work.
2. State of the art

In the previous Chapter, we have introduced the SR concept. In this Chapter we review the most notable works in the literature related to all sub-problems of reconstruction-based SR on the spatial domain.

SR was first proposed by Tsai and Huang (1984) [2], who suggested a frequency domain approach to SR. This initial approach assumes a translational movement among the LR images and is based on Fourier shifting properties, aliasing and the assumption that the HR image is band-limited. In early works, frequency-based methods were the chosen approaches despite being restricted in the image observation model and the real scenarios, which they can handle. This is the reason why in current works, SR is spatial-based, which gives more flexibility to model several kinds of image degradations.

2.1. Registration

Registration is widely considered the most influential factor when considering sources of poor performance in super-resolution. Registration has to be extremely accurate for achieving real SR. Therefore; different techniques have been developed to tackle the registration problem. Registration compute the shifts (displacements of pixels) or the pixel motion estimation between the reference frame (image 1) and the subsequent frames (images 2, 3, ..., n). We show in Figure 2.1 an example of image registration. The first row shows two images: the reference one and a warped and distorted one, whilst the second row shows a set of salient points that are used to perform the matching (motion estimation) and the third row shows the image resampling and transformation.
Once registration has been discussed, we carry out a review of registration approaches in the literature. Vandewalle et al. proposed in [4] a frequency-based method to estimate the motion parameters. Only planar motion parallel to the image plane is allowed and the motion can be described as a function of three parameters: horizontal and vertical shifts and a planar rotation angle. It only uses low-frequency information (the part of the signal with the highest SNR). In 1988, Keren and Peleg [5] were the first to present the sub-pixel image registration method in super-resolution image reconstruction. The algorithm takes the most common rigid transformation model. The Keren image registration algorithm typically has three parameters where the following relation holds for the horizontal shift, the vertical shift and the rotation angle around the origin. In [6] Lucchese proposed a novel frequency domain algorithm for estimating planar roto-translations that exploits the zero-crossing lines of a suitably defined
difference of normalized Fourier Transform magnitudes. The proposed technique does not resort to polar coordinates to estimate the rotational angle and uses a phase correlation-type procedure to estimate the translational displacement. In [7] Marcel et al. propose a method for estimating camera displacements under the hypothesis that these can be represented as compositions of translations and rotations in the imaging plane. The key idea consists of representing the magnitudes of the Fourier transforms of the two images in polar coordinates to obtain two functions that differ in a translational displacement corresponding to the rotation angle.

One of the most popular registration methods for video sequences is Optical Flow (OF) developed by Bruce D. Lucas and Takeo Kanade [8]. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic OF equations for all the pixels in that neighbourhood by the least squares criterion. Later, the pyramidal approach was published to solve the OF requirement by computing the registration in different steps, with each taking a different window size (from low to high) [9]. Another remarkable OF method is the one employed by Horn and Schunck [10], which assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions showing more smoothness.

More recent is the comparison made by S. Baker et al. in Middlebury [11]. It contains four different methods of comparison (endpoint error, angular error, interpolation error and normalized interpolation error) with sixty-three kinds of registration. During our thesis we have not evaluated all of them but finally, we have used the Classic + Non-Local (CNL) method [12]. It improves the classical OF by formalizing the median filtering heuristic as an explicit objective function. While median filtering in a large neighbourhood has advantages, it also has problems. [12] proposes that for a given pixel, if it is known which other pixels in the area belong to the same surface, they can weight them more highly. The modification to the objective function is achieved by introducing a weight into the non-local term.

This thesis focuses on techniques based in OF (such as Lucas-Kanade (LK), Horn-Schunck (HS) and CNL that will be further discussed in Chapter 3). We reject other algorithms presented in the literature due to several limitations that prevented their use in this thesis. For instance, Marcel [5] only allows estimating pixel shifts but in SR, sub-pixel shifts are needed. Keren [5] and Vandewalle [4] are frequency-based and nowadays, spatial-based theory can handle and model complex motion. Finally, Lucchese’s algorithm [6] only works with very small and simple motion.
2.2. Reconstruction

Reconstruction uses the registration output information to create a HR image. In Figure 2.2 we show an example of HR image reconstruction from three different LR with blur and noise. This section overviews several techniques that have appeared in the literature, which deal with reconstruction problems.

Figure 2.2 extracted from paper[41]. Reconstruction for the Super Resolved image from three different LR frames

Irani and Peleg [13] proposed in 1990 an iterative algorithm to increase image resolution, together with a method for image registration with sub-pixel accuracy. The approach is based on the resemblance of the presented problem to the reconstruction of a 2-D object forms its 1-D projections in computer aired tomography (CAT) where images are reconstructed from their projections in many directions. Each LR pixel is a “projection” of a region in the scene whose size is determined by the imaging blur. The HR image is constructed using an approach similar to the back-projection method used in CAT. The Iterative Back Projection is similar to methods used to solve linear equations systems. In each iteration there is an estimation of \( Y^n \) of the HR image from a set of LR images:

\[
Y^{n+1} = d_k \cdot (Y^n(g_k(x,y)) \ast h) \quad (2.1)
\]

where \( d_k \) is the subsampling, \( h \) is the blur matrix, \( g_k(x,y) \) is the warping of the coordinates from the \( k \)th LR image to coordinates system of \( Y \) and \( \ast \) represents convolution.
Another approach is Projection Onto Convex Sets (POCS). This is a set theoretic method made by Stark and Oskou [14] in 1989 and then extended to include observation noise in Tekalp et al. [15] in 1992. POCS is based on defining convex sets of restrictions on a vectorial space with as many dimensions as pixels have the HR image X. The projection on a set is understood as an operator that relates any point in the space to the nearest point that belongs to the set. In each iteration, to obtain the point $Y^{n+1}$ from $Y^n$ we would use the formula

$$Y^{n+1} = P_m \cdot P_{m-1} \cdots P_1 \cdot Y^n \quad (2.2)$$

where $Pm$ is the projection operator onto a restriction set.

T.Q. Pham et al. [16] proposed a robust certainty and a structure-adaptive applicability function to the polynomial facet model and applied it to fusion of irregularly sampled data. The method is based on normalized convolution, in which the local signal is approximated through a projection onto a subspace spanned by a set of basic function.

But quite early [Cheeseman et al., 1996 [17]; Schultz and Stevenson, 1996 [18]], SR was defined as an estimation problem that could be solved in a Maximum Likelihood (ML) or Maximum a Posteriori (MAP) framework. Hence, in 2003 S. Farsiu et al. [19] proposed a fast and robust super-resolution algorithm using the $L_1$ norm (as the ML estimate of data in the presence of Laplacian noise), both for the regularization and the data fusion terms. Whereas the former is responsible for edge preservation, the latter seeks robustness with respect to motion error, blur, outliers, and other kinds of errors not explicitly modeled in the fused images. It also mathematically justifies a non-iterative data fusion algorithm using a median operation.

In 2000, William T. Freeman et al. presented a SR method called Learning Low-Level Vision [20]. It collects pairs of LR and HR image patches from a set of images as training. An input LR image is decomposed to overlapping patches on a grid, and the inference problem is to find the HR patches from the training database for each LR patch. They use the kd-tree algorithm, which has been used for real-time texture synthesis, to retrieve a set of HR, k-nearest neighbours for each LR patch. Using patches from a dictionary is an interesting technique that could be considered in our approach as a future work. Another algorithm that uses patches and dictionaries (learning and training data set) was presented by Jang et al. in [21]. It presents a novel approach toward single image super-resolution based on sparse representations (which is another common method in SR) in terms of coupled dictionaries jointly trained from HR and LR image patch pairs. They proposed a method for adaptively choosing the most relevant reconstruction neighbours based on sparse coding, avoiding over- or under-fitting of and producing superior results. However, sparse coding over a large sampled image patch database directly is too time consuming. Their approach is motivated by recent results in sparse signal representation, which suggest that the linear relationships among HR signals can be accurately recovered from their low-dimensional projections.

Filip Sroubek presented his Ph.D. in 2003 [22]. Techniques called blind deconvolution and Super Resolution remove the blur and increase the resolution, respectively. Current multiframe blind deconvolution techniques require no or very little prior information about the blurs and they are sufficiently robust to noise to provide satisfying results in most of the
real applications. The super resolution methods either assume that there is no blur or that it can be estimated by other means. His work proposes a unifying system that simultaneously estimates blurs and the original undistorted image, all in HR, without any prior knowledge of the blur or original image. He accomplishes this by formulating the problem as constrained least squares energy minimization with appropriate regularization terms, which guarantee close-to-perfect solution in the noiseless case.

As mentioned above, another common technique in SR is to use the sparse domain as Dong and Zhang in [23] presented. It proposes an adaptive sparse domain selection scheme for sparse representation. Apart from the sparsity regularization, other regularization terms can also be introduced to further increase the image restoration performance. In that paper, they propose to use the piecewise autoregressive models, which are pre-learned from the training dataset, to characterize the local image structures. On the other hand, considering the fact that there are often many repetitive image structures in an image, it introduces a non-local self-similarity constraint served as another regularization term, which is very helpful in preserving edge sharpness and suppressing noise. Mallat and Yu proposed in 2010 [24] a class of inverse problem estimators computed by adaptively mixing a family of linear estimators corresponding to different priors. Sparse mixing weights are calculated over blocks of coefficients in a frame providing a sparse signal representation. They minimize an $L1$ norm taking into account the signal regularity in each block. Adaptive directional image interpolations are computed over a wavelet frame.

It is not always necessary to apply SR to the whole image. In 2006 Xin Li [25] proposed a robust line geometry-based algorithm for registering zoom images. But this is beyond our purpose, which is working with multi-frames.

Most of the existing ML estimators assume that the noise in the LR observations is Gaussian distributed. However, for many real scenarios, the Laplacian distribution is more accurate to model the impulsive noise. Huihui Song et al. presented in [26] this method is based on a combination of both: Gaussian and Laplacian. It defines a Ratio factor, which is going to be the threshold between selecting the Laplacian and Gaussian distribution. Its value is about 0.706.

$L1$-norm (Gaussian): good performance of smoothing image.

$L1$-norm (Laplacian): can well preserve the details of image such as edges.

In order for a stable solution and for edge preservation the BTV [19] is used. It uses the scaled conjugate gradient algorithm to optimize the objective function and it proposes an adaptive convergence criterion by using the change of the ratio as an indicator, which is robust and very efficient. If the absolute value variance of the ratio is successively less than a fixed threshold more than four times, the convergence is assumed to be reached.
2.3. Restoration

Restoration tries to remove the image blur and the image noise. These techniques are called Deblurring and Denoising respectively. We explain how these processes work and comment briefly on some techniques used to compare our algorithm with.

2.3.1. Deblurring

The blur caused by camera motion is a serious problem in SR. As a rule, this problem occurs when low ambient light conditions prevent an imaging system from using sufficiently short exposure times. For example, when taking photographs by hand under poor lighting conditions, camera shakes lead to blur.

In Restoration literature, the term Power Spread Function (PSF) appears, which could be defined as a filter that models the image blur effect. This term refers exactly to the diffraction limit of finite-size lenses, a point source in the scene corresponding to an intensity blob in the imaging plane. Due to charge transport and sampling, the digitized image is further degraded. The combination of all degradations can be modeled by a system blur, which is a convolution of all PSFs along the imaging path. Two types of blur are present in all digital imaging systems: the optical blur which specifies the fine-resolving capability of the optics and the sensor integration blur caused by the accumulation of incoming light over a finite-size sensor element. Thus, image restoration tries to inverse this blurring degradation, but within the band limit of the imager (e.g., it enhances the spatial frequencies within the imager band). Deblurring allows the recovery of an image with a good resolution. Hence, the estimate or knowledge of the blurring function PSF is essential to the application of these algorithms, because knowing the PSF, the deblur process could be achieved perfectly.

Despite the fact that there has been a considerable effort in the image processing community in the last decades to find a reliable algorithm to remove the blurring in software, a method working for any arbitrary blur is not known. In Figure 2.3 we can observe an example of image deblurring, which in Chapter 5 will be studied in more detail.
2.3.2. Denoising

Image noise is the digital equivalent of film grain for analogue cameras. Alternatively, one can think of it as analogous to the subtle background hiss you may hear from your audio system at full volume. For digital images, this noise appears as random speckles on an otherwise smooth surface and can significantly degrade image quality. Noise increases with the sensitivity setting in the camera, length of the exposure, temperature, and even varies amongst different camera models. In addition, noise not only changes depending on exposure setting and camera model, but it can also vary within an individual image. For digital cameras, darker regions will contain more noise than the brighter regions; with film the inverse is true. It is also composed of two elements: fluctuations in colour and luminance. Colour or chroma noise is usually more unnatural in appearance and can render images unusable if not kept under control. The example below shows noise on what was originally a neutral grey patch. Along with the separate effects of chroma and luminance noise and its fluctuations can also vary in both their magnitude and spatial frequency, although spatial frequency is often a neglected characteristic. In Figure 2.4 we observe the two elements of the noise composition.

Figure 2.4. Image Noise

Once explained how noise affects images, we introduce two different denoising methods, which in Chapter 6 are used to compare with our technique, the Bilateral Filter (BF) and the Median Filter (MF) [28]. The BF was firstly presented by Tomasi and Manduchi in 1998 [27]. It is an edge preserving and noise reducing smoothing filter. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. This weight is based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance but also on the radiometric differences (differences in the range, e.g. colour intensity). This preserves sharp edges by systematically looping through each pixel and attributing weights to the adjacent pixels accordingly.
On the other hand, MF is a simple sliding-window spatial filter that replaces the center value in the window with the average of all the neighbouring pixel values including this center value itself. By doing this, it replaces pixels that are unrepresentative of their surroundings. It is implemented with a convolution mask, which provides a result that is a weighted sum of the values of a pixel and its neighbours. It is also a linear filter. The mask or kernel is a square. Often a 3×3 square kernel is used. If the coefficients of the mask sum up to one, then the average brightness of the image is not changed. If the coefficients sum to zero, the average brightness is lost, and it returns a dark image. The MF works on the shift-multiply-sum principle.

In this Chapter, we have presented some SR methods available in the literature. We have based our approach on known problems and their solutions, which are found on the literature. We have mentioned algorithms, which will be related to our method. Furthermore, in the following chapters some of them will be used to compare their results with our proposal outputs.


3. Registration

![Diagram](image)

Figure 3.1. Registration block. It is the first step in SR. From a set of LR images, it tries to compute the shifts within a sequence.

Registration is the process to identify how the information has moved within a sequence of images. The observations are LR images, which can contain aliasing, blur or noise or all of them together. These LR images are overlaying two or more images of the same scene taken at different times, from different viewpoints and/or by different sensors, which introduces differences between them. Furthermore, image registration is a crucial step in SR, thus, if its performance is not effective due to the extremely poor resolution of the observation or the complexity of the motion, errors appear and they are carried along all the SR process. In fact, artefacts caused by these registration errors are visually more noticeable than the blurring effect resulting from the interpolation of a single image.

This Chapter is structured as follows: Section 3.1 explains the necessity of registration; in Section 3.2 we describe three different Optical Flow-based algorithms and we conclude this section giving some notable comparisons.

3.1. The necessity to register multi-frames

We can formulate registration as the following equation. Given two images \( l_1(x,y) \) and \( l_2(x,y) \):

\[
l_1(x,y) = f \left( l_2(H(x,y)) \right) + n \quad (3.1)
\]

where \( f(I) \) is the function that maps the intensities, \( H(x,y) \) is the spatial transformation between the images and \( n \) is the noise. The main goal is to estimate \( H(x,y) \) and align the images. \( l_1(x,y) \) refers to the first image, which will be the SR one and is usually known as the reference. On the other hand, \( l_2(x,y) \) and so on, refers to the current or the consecutive image.
In Figure 3.2, the base of image registration is illustrated. Pixels belonging to the reference image have to be found in the current image. It is not an easy process due to the fact that the current image is also deformed, leading to the appearance of new artefacts like brightness or new objects such as the cup of glass. Thus, when registering an image a global translational transform is not enough, but rather a dense, per-pixel, warping transform.

![Figure 3.2 extracted from paper [29]. Idea of registration: find corresponding pixel points](image1)

In Figure 3.3 we can observe the difference between direct aligning of two images (before) and alignment plus warping the current image (warped). The higher the robustness of the registration method, the more problems it can tackle. Hence, when a global transform is applied (before) the result is not well registered because the current image (test) is smaller than the reference. This means, warping is necessary and in Chapter 4 is presented.

![Figure 3.3 extracted from [29]. Image registration example](image2)
In the following, before explaining the technique used for registering, we discuss briefly the main of its problem. As we have mentioned, the registration has continuously proved to be one of the central problems in computer vision but despite the fact that it is difficult to control, the main problems appear when multiple objects moving in different directions are captured by a video camera. Beyond this, in the case of a non-stationary camera\textsuperscript{1} these motions could be also contaminated. Hence, the method that can better tackle these problems is the OF, presented in Section 3.2.

To conclude this section, it is necessary to mention that a typical algorithm imposes regularization constraints on the spatial transformation and the intensity mapping to ensure that the problem has a unique solution. This term will be also evaluated in more detail in Chapter 4.

### 3.2. Motion estimation: Optical Flow

As seen in the previous chapter, most methods in the literature are algorithms based on the OF constraint, which encodes the apparent motion of all objects of a visual scene from one frame to the next. These algorithms use pixel motion estimation; in other words, compute the shift for each pixel of the image. However, it has to be mentioned that the pioneers of OF were Lucas and Kanade\textsuperscript{[8]}, which algorithm formulates the following equations. Consider \( I(x, y, t) \) as the reference frame, where \( x \) and \( y \) refer to pixel location and \( t \) to time location.

\[
I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad (3.2)
\]

We can rewrite the previous expression like the Displaced Frame Difference\textsuperscript{2} (DFD).

\[
DFD(\mathbf{r}, \mathbf{\bar{D}(r)}) = I(\mathbf{r}, t) - I(\mathbf{r} - \mathbf{\bar{D}(r)}, t - \Delta t) \quad (3.3)
\]

Where \( \mathbf{\bar{D}(r)} \) is denoted like the pixel displacement.

\[
\mathbf{\bar{D}(r)} = \mathbf{\bar{D}(x, y)} = \begin{pmatrix} d_x \\ d_y \end{pmatrix} \quad (3.4)
\]

\textsuperscript{1} The camera moves during the capture such as tracking, panning and zooming.
\textsuperscript{2} DFD refers to the differential picture if there is motion.
Most of the algorithms follow this mathematical formulation. However, we do not focus in more detail because the purpose of this thesis is not to solve the registration problem, but evaluate an efficient OF to use in reconstruction and restoration SR. Before concluding, in Figure 3.4 we depict how OF can be represented, although the most common is the 2D representation, which gives an intuitive view of how motion vectors move.

![Figure 3.4 extrated from [30]. Optical flow experience](image)

### 3.2.1. Global motion or local motion

In this section, the discussion of two different types of motion is evaluated: global or local. The motion field between two images can be described by a translation, an affine mapping or a projective mapping. In general, there are at least two parts involved in a video sequence: a background and an object. Let us explain briefly their meaning and importance in image registration.

Global motion estimation refers to the motion of the background caused mainly by the camera motion. This model can be usually applied to the entire frame, if there is lack of local motion. There are two main global motion estimation approaches: one consists of minimizing the prediction errors under a given set of motion parameters and then estimating global motion parameters. The other one consists of first finding pixel motion vectors and afterwards determining the global motion model by regression methods.

On the other hand, and as it could be deduced, local motion refers to individual and specific types of motion within the image, which behave in a different way to the global image. This means both types cannot be equally treated.

Video sequences may not only have global motion, but both global and local motion vectors if the scene contains moving objects. This is why registration plays a determinant role in SR. Thus, depending on the motion of the video sequence, algorithms can work better or worse and a unique registration method useful for both global and local motion does not exist. For instance, it is completely different to register a football match sequence than a sequence,
where only buildings appear (with motion parallel to the image plane). In the first case a local motion registration should be applied and, in the second case, a global one is enough.

Usually, when a global motion method is applied, there is only one output for each image, to be precise, one motion vector for the whole image. This reduces computational cost and unnecessary dense motion field estimation because afterwards, all of them shall have the same direction and module. On the other hand, when a local motion method is applied, its output is different. For each frame, there are as many motion vectors as image pixels. In spite of this, registration continues introducing artefacts and problems in SR.

In our case, we decided to work with pixel-wise registration because of the nature of our input sequences and despite the fact that the computational cost is higher with global motion methods; a dense OF estimation algorithm gives better outputs. We present some comparisons at the end of this chapter, whereby we demonstrate the previous sentence.

### 3.2.2. Lucas-Kanade, Horn-Schunck and Classic Non-Local

We have reviewed in Chapter 2 three relevant methods of OF: Lucas-Kanade, Horn-Schunck and the Classical Non-Local. In this section, we explain briefly the methods, their differences and the reason for our choice of the Classic Non-Local.

#### 3.2.2.1. Lucas-Kanade

We refer to Lucas and Kanade [8] as the creators of the OF concept. The LK method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighbourhood of the point p under consideration. Thus, the OF equation can be assumed to hold for all pixels within a window centered. Namely, the local image flow (velocity) vector \((V_x, V_y)\) must satisfy:

\[
I_x(q_1) \cdot V_x + I_y(q_1) \cdot V_y = -I_t(q_1)
\]

\[
I_x(q_2) \cdot V_x + I_y(q_2) \cdot V_y = -I_t(q_2)
\]

\[
\vdots
\]

\[
I_x(q_n) \cdot V_x + I_y(q_n) \cdot V_y = -I_t(q_n)
\]

(3.8)
where \( q_1, q_2, \ldots, q_n \) are the pixels inside the window centered at \( p \), and \( I_x(q_1), I_x(q_2), \ldots, I_x(q_n) \) are the partial derivatives of the image \( I \) with respect to position \( x \), \( y \) and time \( t \), evaluated at point \( q_i \) and the current time.

An improvement of this method is the pyramidal LK [31]. The main idea is to divide the image in levels as it can be observed in Figure 3.5. The algorithm is the same but it is applied step by step in each level in order to help disambiguating motion in large uniform areas.

![Figure 3.5 extracted from paper [31]. Three levels to apply pyramidal Lucas-Kanade algorithm](image)

### 3.2.2.2. Horn-Schunck

Another available algorithm is Horn-Schunck [10]. The HS method of estimating OF is a global variational method, which introduces a global constraint of smoothness to solve the aperture problem. It assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions showing more smoothness. The flow is formulated as follows:

\[
E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2)] \, dx \, dy \tag{3.9}
\]

where \( I_x, I_y \) and \( I_t \) are the derivatives of the image intensity values along the \( x \), \( y \) and time dimensions respectively, \( \mathbf{V} = [u(x,y), v(x,y)]^T \) is the OF vector, and the parameter \( \alpha \) is a regularization constant. Larger values of \( \alpha \) lead to a smoother flow.
3.2.2.3. Classic Non-Local

Finally, we introduce the last one we used: the classic and non-local registration (CNL) [12].

They define the classical OF.

\[
E(u, v) = \sum_{ij} \left\{ \rho_D \left( I_1(i, j) - I_2(i + u_{i,j}, j + v_{i,j}) \right) \\
+ \lambda \left[ \rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1}) \right] \right\}
\]

(3.8)

where \( u \) and \( v \) are the horizontal and vertical components of the OF field to be estimated from images \( I_1 \) and \( I_2 \), \( \lambda \) is a regularization parameter, and \( \rho_D \) and \( \rho_S \) are the data and spatial penalty functions.

Then, they show that the heuristic median filtering step in Classic-C (a variation using the convex Charbonnier Penalty which do not be discussed in this thesis) can now be viewed as energy minimization of a new objective with a non-local term. By formalizing the median filtering heuristic as an explicit objective function, it can be found ways to improve it. While median filtering in a large neighbourhood has advantages, it also has problems. A neighbourhood centred on a corner or the surround dominates thin structure and computing the median results in oversmoothing. Examining the non-local term suggests a solution. For a given pixel, if it is known which other pixels in the area belong to the same surface, they can be weighted more highly. The modification to the objective function is achieved by introducing a weight into the non-local term.

\[
\sum_{ij} \sum_{(i',j') \in N_{lj}} w_{ij,i',j'} \left( |\tilde{u}_{ij} - \tilde{u}_{i',j'}| + |\tilde{v}_{ij} - \tilde{v}_{i',j'}| \right)
\]

(3.9)

where \( w_{ij,i',j'} \) represents how likely pixel \( i',j' \) is to belong to the same surface as \( i,j \). They do not know \( w_{ij,i',j'} \), but they can approximate it like defining the weights according to their spatial distance, their colour-value distance and their occlusion state in a manner similar to that in Bilateral Filtering.
3.2.2.4. Comparison

In this section, we compare the three algorithms explained above. For the first experiment, we compute the registration given by those methods in some examples and for the second, we shift artificially the cameraman example to obtain statistical results.

First of all, we illustrate in Figure 3.6 the colour reference according for the OF result, used for the first experiment. As we observe, the red colour means that the motion goes to the right side; the dark blue to the up-left side; in other words, the more saturated, the more magnitude. Then, in Figure 3.7 we show the teddy sequence with its ground truth motion. It can be observed that in teddy frame, despite the fact the motion is parallel to the image plane, exist lots of different information such as the calendar, the teddy, the poster, the plants, etc. Thus, we consider than a pixel-wise registration method is better to estimate its motion vectors. Finally, in Figure 3.8 the OF result from LK, HS and CNL is illustrated.

![Figure 3.6](image1.png)

*Figure 3.6. Colour reference according for the OF result*

![Figure 3.7](image2.png)

*Figure 3.7 extracted from [11]. OF ground Truth of Teddy sequence*
In this case, we observe that the result obtained by LK is really poor compared with HS and CNL. The HS result gives a better output but it is still far from the one obtained by CNL, which if we compare with the ground truth, the differences are small.

Then, since the registration made by LK is worse than HS and CNL, in the following examples we use only the latter two methods. The sequences are taken from a traffic camera. In the first sequence, Figure 3.9, the background is more or less static (only the camera shift exist) and the trolley is moving to the right side. In the second one, Figure 3.11, the camera is also moving. Notice that for these examples only reference image 1 and image 2 are showed.
In these two examples we can observe the differences with HS and CNL when the sequence is not static and more complex motion appear (even the camera motion). As mentioned above, in Figure 3.9, the trolley is moving to the right side; thus, its colour should be red. Then, in Figure 3.10 is shown how CNL computes better the true motion of the trolley. For the other example, the grey car should be bright blue; meanwhile the black car, bright red, according to their motion. Then, in figure 3.12, we can notice the different motion estimation made by HS and CNL, observing than CNL almost estimate what it was expected. On the other hand, HS performs poorly and some parts of the sequence are evaluated without any motion vectors.
Finally, for the second experiment and in order to provide simple quantitative measures, we shifted artificially with known values the cameraman image to compute a global shift. Despite the fact a global shift is not considered for this thesis, this comparison using pixel-wise registration methods provides us additional results to decide which algorithm achieve the best results. Knowing the ground truth of each sub-pixel shift and then we could compute the error in each method and as HS and CNL gives a flow motion vector for each pixel, their mean is taken.

<table>
<thead>
<tr>
<th>GROUND TRUTH</th>
<th>LUCAS - KANADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE 0</td>
<td>X shift</td>
</tr>
<tr>
<td>IMAGE 1</td>
<td>0,5</td>
</tr>
<tr>
<td>IMAGE 2</td>
<td>0</td>
</tr>
<tr>
<td>IMAGE 3</td>
<td>0,5</td>
</tr>
<tr>
<td>Acc. Error</td>
<td>0,1026</td>
</tr>
<tr>
<td>MSE</td>
<td>0,0016</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GROUND TRUTH</th>
<th>HORN - SCHUNCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE 0</td>
<td>X shift</td>
</tr>
<tr>
<td>IMAGE 1</td>
<td>0,5</td>
</tr>
<tr>
<td>IMAGE 2</td>
<td>0</td>
</tr>
<tr>
<td>IMAGE 3</td>
<td>0,5</td>
</tr>
<tr>
<td>Acc. Error</td>
<td>0,9748</td>
</tr>
<tr>
<td>MSE</td>
<td>0,1577</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GROUND TRUTH</th>
<th>CLASSIC NON-LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE 0</td>
<td>X shift</td>
</tr>
<tr>
<td>IMAGE 1</td>
<td>0,5</td>
</tr>
<tr>
<td>IMAGE 2</td>
<td>0</td>
</tr>
<tr>
<td>IMAGE 3</td>
<td>0,5</td>
</tr>
<tr>
<td>Acc. Error</td>
<td>0,0464</td>
</tr>
<tr>
<td>MSE</td>
<td>0,0003</td>
</tr>
</tbody>
</table>

Table 3.1. Optical Flow comparison
We conclude that the CNL algorithm is working the best. Its error values (Accumulate Error in both directions and Mean Squared Error) are the smallest, even if the motion is global. Therefore, for our scope we decided to use the CNL registration algorithm of D. Sun, S. Roth, and M. Black [12], which gives us the highest results. Furthermore, we also based our decision on the Middlebury benchmark registration [11] where this algorithm has the seventh best performance⁴.

---

⁴ We have discarded the others six because they do not have available code and our purpose was not to implement a registration method.
4. Reconstruction

**Figure 4.1. Reconstruction block, which uses Registration output to combine it with the LR input frames, generating thus a HR image.**

In this chapter we introduce our proposed reconstruction algorithm for a set of LR images with known registration. Our scheme is based on, first, warping the neighbour frames with the registration results in order to compensate the motion. As a result, we obtain an initial approximation to the SR image with empty gaps. Then, in order to fill the gaps, we use an inpainting technique to complete the resulting SR image. Most importantly, we resort to a variational approach for this latter step. As it will be seen below, we consider different regularizers, such as Total Variation or Tikhonov, before choosing the best performing for obtaining the reconstructed SR image.

This chapter is organized as follows: Section 4.1 explains the motion compensation proposing two different approaches; Section 4.2 details how to enhance the warped image through the proposed regularized inpainting technique.

### 4.1. Warping

Since the registration of the input frames is known, it can be used to combine the several LR images by warping them to match the LR reference image. Image warping is the process of digitally manipulating an image such that any shapes portrayed, which have been significantly distorted in the image. Warping may be used for correcting image distortion as well as for motion compensation.

LR images are defined on a \([M \times N]\) grid denoting the location within the rectangular image domain in both directions, \(x\) and \(y\) (horizontal and vertical, respectively). Then, the entire SR image is computed on a \([m\cdot M \times m\cdot N]\) domain, where \(m\) is the magnification factor. In Figure 4.2 we show an example of a LR grid and its corresponding SR grid, defined on their domains.
Once the new location domain is defined, we denote $u$ and $v$ as the output of registration (also known as shift) for horizontal and vertical pixels respectively. The aim is to reconstruct the reference frame $t$. Pixels’ locations in each frame can be defined by $(x, y, t)$ and using the $u$ and $v$ shifts, the new position is found in the consecutive frames: $t+1$, $t+2$, ..., $t+n$. Following the notation cited above, it can be denoted as $(x+u, y+v, t+i)$. Thus, a sub-pixel location in the SR domain is obtained, which is the base for our two different approaches via warping.

But before introducing them, let us answer the following question: which are the ideal sub-pixels shifts? We gain extra information using several frames instead of using a single frame, due to the sub-pixel precision of the OF. Thus, without sub-pixels, SR is not possible. Theoretically, to obtain the ideal resolution by factor 2, half-pixel accuracy is needed, specifically, $(0,0)$, $(0.5,0)$, $(0,0.5)$ and $(0.5,0.5)$ in both directions, $(x, y)$, for the first four input frames, respectively. Thereby, the SR image takes the maximum information of each LR frame, filling as well, all the grid positions. But this hardly happens, so in Figure 4.3 we show the HR formation from a non-ideal LR grid pixels combination. Unfilled circles belong to pixels from reference image, grey circles from image 2, black circles from image 3 and triangles from image 4.

![Figure 4.2. LR grid and SR grid; magnification factor = 2](image)

![Figure 4.3. (Left) 4 input frames, and (Right) Combination of LR images to create the HR image](image)
4.1.1. First approach – nearest neighbour

The first approach could be referred to as a nearest neighbour method. First, we round the sub-pixel displacement given by \((u\cdot m, v\cdot m)\) and, then, we round the new sub-pixel position in the HR domain given by \((x\cdot m + u\cdot m, y\cdot m + v\cdot m)\). In both cases, only one value is taken. The more detailed descriptions of these two different ways to warp the LR image in order to reconstruct the SR image are:

- **Rounding shift**
  1. Multiply the output of the registration step by the magnification factor.
     - Vertical shift: \(u \cdot m\)
     - Horizontal shift: \(v \cdot m\)
  2. Round them to make them integer values.
     - Vertical shift: \(\text{round} (u \cdot m)\)
     - Horizontal shift: \(\text{round} (v \cdot m)\)
  3. Find the new position in the SR grid.
     - Vertical shift: \(x' = x \cdot m + \text{round} (u \cdot m)\)
     - Horizontal shift: \(y' = y \cdot m + \text{round} (v \cdot m)\)

- **Rounding warped position**
  1. Multiply the output of the registration step by the magnification factor.
     - Vertical shift: \(u \cdot m\)
     - Horizontal shift: \(v \cdot m\)
  2. Find the new position in the SR grid.
     - Vertical shift: \(x' = x \cdot m + u \cdot m\)
     - Horizontal shift: \(y' = y \cdot m + v \cdot m\)
  3. Round them to make them integer values.
     - Vertical shift: \(\text{round} (x')\)
     - Horizontal shift: \(\text{round} (y')\)

The performance of both cases is poor in practice because of the uneven distribution of sub-pixel displacements illustrated in Figure 4.3. As a result, in almost all the cases, there were still lots of empty pixel positions in the HR image due to the rounding process, which loses information. In other words, the accuracy obtained by the sub-pixel displacements is misused. This has a second repercussion on the posterior inpainting method, which will require a higher time to convergence (larger amount of empty cells in the HR grid) and will be less accurate due to the sparser amount of information. In Figure 4.4 the problem mentioned above can be observed. The black circle and the blue square are the two first LR frames (using a magnification factor 2), which have a displacement less than 0.5 pixels. Then, the yellow circle and the green square are the others two LR images, which have a shift less than 1 pixel and higher than 0.5 pixel. When rounding the two first ones, in the SR image, the position becomes 1 in \(y\) direction, and for the others, the position becomes 2 in \(y\) direction. This means, that
frames 1 and 2 behave like the same shifts and they are not contributing with extra information. The same happens with images 3 and 4. Thus, for the four input frames, only two are contributing and the problem becomes higher, which ones are the rights ones? In the next section, we tackle this inconvenience.

4.1.2. Final approach – bilinear

To avoid the problems mentioned in the previous approach, we approximate the SR image filling the pixel positions in a bilinear manner. The technique is similar but once the rounding warped position⁴ is computed, the initial SR pixel value is weighted as follows:

\[
SR(x', y') = SR(x', y') + (1 - d_u) \cdot (1 - d_v) \cdot LR_{t+1}(x, y)
\] (4.1)

\[
SR(x' + 1, y') = SR(x' + 1, y') + d_u \cdot (1 - d_v) \cdot LR_{t+1}(x, y)
\] (4.2)

\[
SR(x', y' + 1) = SR(x', y' + 1) + (1 - d_u) \cdot d_v \cdot LR_{t+1}(x, y)
\] (4.3)

\[
SR(x' + 1, y' + 1) = SR(x', +1y' + 1) + d_u \cdot d_v \cdot LR_{t+1}(x, y)
\] (4.4)

where \(d_u\) is the vertical distance to the pixel \((x', y')\) , \(d_v\) is the horizontal distance to \((x', y')\) and \(LR_{t+1}(x, y)\) refers to the current frame evaluated, where the shifts \(u\) and \(v\) have been computed.

According to these equations, the contribution to each of the four closest pixels is modulated by bilinear weights regarding their proximity to the warped and scaled position. After accumulating the contributions from all frames, the resulting SR grid is normalized by the accumulated weights. The resulting SR image obtains fewer holes with closer original values of the frames.

---

⁴ The entire part of the pixel position is taken.
Before starting with the inpainting method, we compare both approaches for the book sequence\(^5\) [34] with a magnification factor 4 and 16 input frames. We can observe in Figure 4.4 the effect explained above, where the first approach gives more holes than the second one.

\(^5\) In Chapter 7 is evaluated in more detail.
4.2. Inpainting

In this section we describe the technique used for filling the SR image gaps after upsampling and warping the LR frames. In general, the higher the magnification factor, the higher the importance of inpainting.

Image inpainting refers to the process of filling-in missing data or damaged parts in a designated region of the visual input. Inpainting could be solved by different manners, and considering regularization for picking a stable solution is very useful. Also, regularization can help the algorithm to remove artifacts from the final result and converge faster if an iterative model is followed. In the next section, we present the impact of regularization in SR Reconstruction-based models.

4.2.1. Effect of the regularizer

The regularization term compensates the missing measurement information with some general prior information about the desirable SR solution, and is usually implemented as a penalty factor in the generalized minimization cost function. It takes an important role in inpainting like in SR as well. In our work we used two different types of regularization: Total Variation and Tikhonov. We proceed to introduce them briefly.

4.2.1.1. Total Variation

The Total Variation (TV) norm is a gradient penalty function. The TV criterion penalizes the total amount of change in the image as measured by the $L_1$ norm of the magnitude of the gradient $A(X) = ||\nabla X||_1$ where $\nabla$ is a gradient operator that can be approximated by Laplacian operators and $X$ is the matrix to regularize, in our case, the image. The $L_1$ norm in the TV criterion favors sparse gradients, preserving steep local gradients while encouraging local smoothness. Farsiu et al. generalized the notation of TV and proposed the so-called bilateral TV (BTV) for robust regularization.
4.2.1.2. Tikhonov

One of the most widely referenced regularization cost functions is the Tikhonov (TK) cost function $Y_T(X) = \| \Gamma X \|^2$ where $\Gamma$ is usually a high pass operator such as derivative, Laplacian, or even identity matrix and $X$ is the matrix to regularize, in our case, the image. The intuition behind this regularization method is to limit the total energy of the image (when is the identity matrix) or forcing spatial smoothness (for derivative or Laplacian choices of $\Gamma$). As the noisy and edge pixels both contain high-frequency energy, they will be removed in the regularization process and the resulting denoised image will not contain sharp edges.

Once TV and TK have been presented, we define the mathematical formulation to solve these constraints. Firstly, to solve TV norm-1 regularization, an iterative model is followed. The cost function can be written as:

\[
L = R(Y) = \sum (| \nabla_u M(Y) | + | \nabla_v M(Y) |) \tag{4.5}
\]

where $Y$ is the SR image, $M$ is the mask applied which lets modify only the empty pixel positions (with ones and zeros), $\nabla_u$ is the gradient in horizontal direction and $\nabla_v$ is the gradient in vertical direction. In the first iteration the first $Y$ is the output of the previous section step. We solve the cost function using the steepest descent gradient:

\[
Y = Y_{bef} - \gamma \nabla L \tag{4.6}
\]

where $Y$ is the SR image, $Y_{bef}$ is the previous SR image computed, $\gamma$ is the updating parameter and $\nabla L$ is the gradient of the cost function (4.5). First of all, let us redefine this equation.

\[
L = \sum (| MD_u Y(x, y) | + | MD_v Y(x, y) |) \tag{4.7}
\]

Then, if equation (4.7) is derivated, the gradient is:

\[
\nabla L = M \cdot (D_u^T \cdot \text{sign}(D_u \cdot Y(x, y)) + D_v^T \cdot \text{sign}(D_v \cdot Y(x, y))) \tag{4.8}
\]

where $M$ is the mask of size $(m*\text{original}_X, m*\text{original}_Y)$, $m$ is the magnification factor,
\((D_u^T \cdot \text{sign}(D_u \cdot Y(x, y))) + D_v^T \cdot \text{sign}(D_v \cdot Y(x, y)))\) is a vector of size \((m^2 \times \text{original}_X \times \text{original}_Y)\) and \(Y(x, y)\) is the SR image as a vector. Finally, the matrices \(D_u\) and \(D_v\) are the forward-differences derivative operators. Their correspondence with the gradient operator is observed in equation (5.3), horizontal, and (5.4), vertical, respectively.

\[
D_u x = \text{vec}(\nabla_u x) \quad (4.9)
\]

\[
D_v x = \text{vec}(\nabla_v x) \quad (4.10)
\]

where \(x\) refers to a random matrix.

We assure with this method that the function will always converge to the optimal solution. To improve the performance, we select an updating parameter variable, which means that, at each iteration, its value will be different. We use an exponential step-down based on the current iteration value.

Furthermore, to make the most of pixel information, we defined the second derivative, which uses a larger number of coefficients leading to achieve an inpainting more effective filling the gaps. However, not always the second derivative could be used, specifically, in small images, where undesired artifacts in their contours appear. The first derivative has coefficients different from zero in all the rows of the image. Thereby, it is giving information for all the pixels. Whereas, the second derivative matrix contains rows and columns with zeros (at least the first and the last), introducing problems in the contours where some gaps are not filled.

Taking up again the mathematical formulation, the second derivative is applied using the same method. However, the gradient takes another shape where matrix \(D\) has the same coefficients values for both directions.

\[
D = \begin{bmatrix}
1 & 1 & 1 \\
1 & -8 & 1 \\
1 & 1 & 1
\end{bmatrix}
\]

\[
\nabla L = M \cdot 2 \cdot D^T \cdot D \cdot Y(x, y) \quad (4.11)
\]

where \(D\) is a sparse convolution matrix. Finally, equations (4.11) or (4.8) are used then to solve equation (4.6).

On the other hand, referring to TK, the cost function, if an iterative model is followed, is now:

\[
L = R(Y) = \| \nabla_u M(Y) \|^2 + \| \nabla_v M(Y) \|^2 \quad (4.12)
\]
To make the notation easier, sparse matrices and image as a vector are used:

\[ L = \|MD_u Y(x, y)\|^2 + \|MD_v Y(x, y)\|^2 \]  \hspace{1cm} (4.13)

where \( Y(x, y) \) is image \( Y \) defined as a vector.

If equation (4.13) is developed:

\[ L = 2 Y(x, y)^T D_u^T M^T MD_u Y(x, y) + 2 Y(x, y)^T D_v^T M^T MD_v Y(x, y) \]  \hspace{1cm} (4.14)

Then, after derivating equation (4.14), we obtain:

\[ \frac{\partial L}{\partial y} = \nabla L = 2D_u^T M^T MD_u Y(x, y) + 2D_v^T M^T MD_v Y(x, y) \]  \hspace{1cm} (4.15)

As \( M \) is a sparse matrix of ones and zeros (the diagonal contains all the ones), following the property of equation (4.16) we can obtain the final solution for the iterative TK regularization, equation (4.17).

\[ M^T M = M^T = M \]  \hspace{1cm} (4.16)

\[ \nabla L = M \cdot 2 \cdot (D_u^T \cdot D_u + D_v^T \cdot D_v) \cdot Y(x, y) \]  \hspace{1cm} (4.17)

Finally, to obtain the resulting inpainted image, equation (4.6) is used.

To conclude, we show two examples\(^6\) about what we have developed in this Chapter. In Figure 4.6 is observed a Super Resolved image using, on the one hand, an iterative TV regularization and, on the other, the Least Square TK.

---

\(^6\) The book and the disk sequence both available in [32].
In this case, when the number of input frames (16 images) is elevated, the warping process takes an important role, so the impact of regularization is lower. We can observe in Figure 4.6 that the differences are hardly noticeable. Hence, with enough input information TK also gives a good. As a remark, the parameter $\gamma$ was initially 0.004 in the case of TV and 0.01 in TK, both with a magnification factor 4.

However, when using TK regularization least squares method is used rather than the iterative model. The method of least squares assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared (least square error) from a given set of data. The main advantage is that is faster than the iterative model and the solution converge to the same. Otherwise, the main problem involves the selection of the constant parameter$^7$, which a wrong choice makes the process not convergent due to the noise level distortion.

$^7$ $\lambda$, parameter of equation (4.18).
We do not proceed to explain how inpainting is solved using TK and least squares. In this case, all the pixels of the image can be modified and not only those that are empty after the warping process, but only the error in those positions is measured. The purpose of introducing the data term is to tackle the noise problem in this level. Why do not use this technique? Because the method that removes well the noise\(^8\) is not lineal (TV) and it can introduces undesired problem when deblurring. However, the cost function that has to be minimized is:

\[
L = \|M(X - Y)\|^2 + \lambda(\|\nabla_u Y\|^2 + \|\nabla_v Y\|^2)
\]  

(4.18)

where \(X\) is the initial SR image (the one obtained after the warping process) and \(Y\) is the desired SR inpainted image.

Finally, in Figure 4.7 we illustrate the resulting image using the first or the second derivate with TV regularization. The input frames used were 16 and the magnification factor was 4.

\[\text{Figure 4.7. (Left) First derivative, (Right) Second derivative}\]

In the example above we used a small \(\gamma\) 0.004 and a maximum of 500 iterations. Using the second derivative we obtain more contrast in the resulting image because the coefficients used are more (as explained above with the differences between first and second derivatives), thus, we can compensate the truth brightness with more accuracy. In Figure 4.8 we show the convergence curve obtained. As \(\gamma\) was updated at each iteration to converge faster, we note that 500 iterations were not needed to find the optimal solution. After 100 iterations, the variation is almost zero.

\[\text{Figure 4.8. Convergence graph of TV}\]

---

\(^8\) Explained in Chapter 6.
5. Restoration: Deblurring

The third step of our proposed SR system is restoration, which is divided in two different parts: deblurring and denoising. Both have been explained in Chapter 2 and, in this chapter, we focus on the deblurring technique.

As a reminder, the term PSF could be defined as a filter that models the image blur effect. This term refers exactly to the diffraction limit of finite-size lenses, a point source in the scene corresponding to an intensity blob in the imaging plane. This blur affects the image resolution, which is crucial in SR; however, knowing the real PSF, the image can be exactly recovered. Thus, and as the PSF is not known, we can estimate it in a proper way, which leads the inspiration to focus on deblurring.

But first of all, to illustrate how blur affects an image, in Figure 5.2 we show the original peppers image\(^9\) and its blurred version with a Gaussian PSF with a high variance (\(\sigma^2\)) and size of 11x11.

\[\text{Figure 5.2. (Left) Original peppers image, (Right) its blurred version with a Gaussian PSF with size 11x11 and } \sigma^2 = 10\]

\(^9\) Available in Matlab library.
This chapter is organized as follows: in Section 5.1 we explain a different approach to estimate the PSF based on [35]; then, in Section 5.2 we describe the technique followed for removing the image blur; and, finally, at the end of this chapter, we perform some experimental results.

5.1. PSF estimation

PSF plays an important role in image processing. It has been demonstrated that estimating or knowing the true PSF of an image can be used to increase its resolution. Hence, we focus on improving an algorithm capable of estimating a PSF from a blurred image.

Our proposed algorithm for the PSF estimation is based on the work of Yoshi, Szeliski and Kriegman [35]. We have chosen their work because the achievements they got were suitable for our purposes. Their algorithm estimates non-parametric and spatially varying blur functions (e.g. PSF) at sub-pixel resolution from a single image. It handles blur due to defocus, slight camera motion, and inherent aspects of the imaging system. It operates by predicting a “sharp” version of a blurry input image and uses the two images to solve for a PSF. However, we introduce a different approach in the sharp image estimation (computing the edge profile), which is explained in the next section, Sharp Image Estimation. In order to finish the method, and using the estimated sharp image obtained, in section PSF Estimation we explain the last part of the technique.

But before explaining in the following sections in more detail the technique, the main idea is illustrated in Figure 5.3. We proceed to comment it very briefly: departing from the blurred image, the contour points are computed and then, using the gradient direction, an estimated sharp image is obtained to estimate the final PSF. As it can be deduced, the key is in the sharp image, which makes the estimation of the PSF feasible.
Image Reconstruction and Restoration for Optical Flow-based Super Resolution

Figure 5.3 extracted from [35]: Sharp Edge Prediction. A blurry image (top left) and the 1D profile normal to an edge (top right, blue line). They predict a sharp edge (top right, dashed line) by propagating the max and min values along the edge profile. The algorithm uses predicted and observed values to solve for a PSF. Only observed pixels within a radius R are used. Bottom left – Predicted pixels are blue and valid observed pixels are green. Bottom right – The predicted values.

Sharp Image Estimation

The blurring process is formulated as an invertible linear system, which models the blurry image as the convolution of a sharp image with the imaging system’s PSF. Thus, if we know the original sharp image, recovering the PSF is straightforward. In our algorithm, we define a different approach to estimate the sharp image, which in our experiments appears to be more robust when processing images with very low resolution, consisting in:

1. Canny edge detection [36]. First of all, the edge points of the blurred image have to be computed. Following the peppers example, and despite the fact (1) is applied in the blurred version, in Figure 5.4 the resulting image after applying (1) on the original image is observed, in order to show how (1) works.

Figure 5.4. Canny edge detection.
2. **Edge transversal local binarization** (to the maximum and minimum values at each side of the edge). In other words, with a window size of $R$ pixels, the maximum and the minimum are searched in the edge profile created around the contour point.

3. **Elimination of the transition band.** This reflects the fact that, in heavily downsampled images, a single pixel covers a large area around a contour point.

Once the edge points are found using (1), technique (2) and (3) are applied (only in those points obtained). It is based on the gradient method, which gives us the direction of the greatest rate of increase of the scalar field, and whose magnitude is that rate of increase. We use, then, the gradient to create a profile edge as in [35], which involves in the right part, the maximum direction (along the positive direction of the gradient), and in the left part, the minimum (along the negative direction of the gradient). In Figure 5.5 we illustrate the edge profile obtained by the original authors and by our approach.

![Figure 5.5. Original paper profile extracted from [35] (left) and our proposed edge profile (right).](image)

Once Figure 5.5 is illustrated, we proceed to explain in more detail all the process involving (1), (2) and (3) to make the process understandable for the reader. Starting from the contour point (the green one), the maximum along the $\left\lfloor \frac{R-1}{2} \right\rfloor$ right pixels of the edge profile is searched. Once it is found, all those pixels in the right side are assigned with this value. For the minimum, situated in the left side of the contour point, the same process is followed. But there is still a point with an unsigned value, the edge point. To decide if it belongs to the maximum or minimum, we compute the distance between both possibilities, giving it the lowest rate. This means that, for the $R$ pixels of the edge profile, one side will have one more pixel than the other.

As a remark, a high choice of the $R$ value achieves a poor consistency in the sharp image. In general, it should take the value related with the PSF that wants to be estimated, e.g. when a PSF size $7 \times 7$ is desired, $R$ should be 3. Hence, we can parameterize the window size as $R = \frac{K_s}{2} - 1$ where $K_s$ is the PSF size.
**PSF Estimation**

Once the sharp estimated image is obtained, we proceed to compute the PSF. We formulate the estimation using a Bayesian framework solved using a maximum a posteriori (MAP) technique [37]. In MAP estimation, one tries to find the most likely estimate for the PSF $K$ given the sharp image $I$ and the observed blurred image $B$, using the known image formation model.

In this part, we follow the same notation for the cost function that in [35] is minimized, but solving it in a different way:

$$L = \frac{\|M(B) - M(I*K)\|^2}{\sigma^2} + \lambda \gamma \|\nabla K\|^2$$  \hspace{1cm} (5.1)

where $M$ is masking the function such that this term is only evaluated for “known” pixels in $B$, i.e., those pixels that result from the convolution of $K$ with properly estimated pixels $I$ (estimated sharp image), which form a band around each edge point. $\lambda$ controls the weight of the smoothness penalty, and $\gamma = (2R + 1)^2$ normalizes for the PSF area. Since the PSF should sum to one (as PSF is energy conserving) the individual values decrease with increased $R$. This factor is needed to keep the relative magnitude of PSF gradient values on par with the data term values regardless of PSF size. And finally, $\sigma^2$ is the noise level.

[35] is based on the non-negative linear least squares using a projective gradient Newton’s method. Our proposal is to solve the problem using only the least squares method. This let us solve the equation in a really fast and effective way for our purpose. Departing from equation (5.1), to make the notation easier, we use convolution sparse matrices, vectorized image matrices and differential operators defined also as sparse matrices. Thus, the new equation takes the following shape:

$$L = \|MB(x, y) - MK\|^2 + \lambda(\|DuK\|^2 + \|DvK\|^2)$$  \hspace{1cm} (5.2)

where $K$ is the estimated PSF, $I$ is the convolution matrix form of the vectorized estimated sharp image (containing as many warps of the image (columns) as elements has $K$), $M$ is the sparse matrix of the mask, $B(x, y)$ is the blurred image as a vector and $\lambda$ is the regularization parameter (notice that $\sigma^2$ and $\gamma$ have not been considered due to the fact they are others parameters that weight the cost function, so using only $\lambda$ is enough). Finally, the matrices $Du$ and $Dv$ are the forward-differences derivative operators, both $Du$ and $Dv$ are defined in the domain $[k_s \times k_s, k_s \times k_s]$, where $k_s$ is the size of the PSF, specifically the size of $K$. Their correspondence with the gradient operator is observed in equation (5.3), horizontal, and (5.4), vertical, respectively.

$$Du x = vec(\nabla_u x)$$  \hspace{1cm} (5.3)

$$Dv x = vec(\nabla_v x)$$  \hspace{1cm} (5.4)

where $x$ refers to a random matrix.
We also have mentioned that \( I \) is the convolution matrix form of the vectorized estimated sharp image. Let us explain it with an easy example:

\[
Y(x, y) = S_x X(x, y)
\]  

(5.5)

where \( Y(x, y) \) is the output image as a vector, \( S_x \) is the sparse convolution matrix and \( Y(x, y) \) is the input image as a vector. Equation (5.5) is the equivalence of the convolution operator, \( i.e., Y = S * X \), where \( S \) is the filter that convolves image \( X \). Thus, \( S_x \) contains all the coefficients needed to compute the convolution.

Then, developing equation (5.2) we obtain the final cost function that has to be minimized.

\[
L = B(x, y)^T M^T M B(x, y) + K^T I^T M^T M I K - 2B(x, y)^T M^T M I K + \lambda (K^T D_u^T D_u K + K^T D_v^T D_v K)
\]

(5.6)

Thus, as equation (5.6) is convex, a global minimum exists, which can be found applying the derivative of \( K \) on it and make it equal to zero.

\[
\frac{\delta L}{\delta K} = 0 + 2I^T M^T M I \bar{R} - 2I^T M^T M B(x, y) + 2 \lambda (D_u^T D_u + D_v^T D_v) \bar{R} = 0
\]

(5.7)

\( M \) is the matrix mask, which contains ones in those pixels’ positions that result from the convolution of \( K \) with properly estimated pixels \( I \). It is defined in the domain \([U \times V, U \times V]\), where \( U \) defines the horizontal size of \( B \) and \( V \) the vertical size, in order to coincide with the size of vector \( B(x, y) \). As \( M \) is a sparse matrix of ones and zeros (the diagonal contains all the ones), following the property of equation (5.8)\(^{10}\) we can obtain the final solution for the PSF estimation, equation (5.9).

\[
I^T M^T M I = I^T M M I = I^T M^2 I = I^T M I
\]

(5.8)

\[
\bar{R} = [I^T M I + \lambda (D_u^T D_u + D_v^T D_v)]^{-1} I^T M B(x, y)
\]

(5.9)

### 5.2. Deblurring

In the previous section, the process to estimate the PSF has been described. Then, the resulting PSF has to be applied to remove the image blur for improving the resolution of the SR image. Thus, we formulate the cost function (5.10) and we solve it by following a similar technique used in the inpainting TK method, but using least squares. Although both cases of regularization (L1-norm and L2-norm) are considered, for deblurring the L2-norm behaves better than L1-norm. Thus, we only describe the TK solution.

\[^{10}\text{M is a diagonal matrix of ones and } M^2 = M\]
\[ L = \|B - X \ast K\|_2^2 + \lambda R(X) \]  

(5.10)

where \(B\) is the blurred image, \(X\) is the desired deblurred image, \(K\) is the estimated PSF, \(R(X)\) is the regularization method, in this case \(TK\), and \(\lambda\) is the regularization parameter.

Then, departing from (5.10), using convolution sparse matrices, we can redefine it as follows:

\[ L = \|B(x, y) - K_s X\|_2^2 + \lambda (\|D_u X\|_2^2 + \|D_v X\|_2^2) \]  

(5.11)

where \(X\) is desired deblurred image, \(K_s\) is the estimated PSF redefined as a sparse convolution matrix, \(\lambda\) is the regularization parameter and \(B(x, y)\) is the blurred image as a vector. The matrices \(D_u\) and \(D_v\) are the forward-differences derivative operators, both \(D_u\) and \(D_v\) are defined in the domain \([k_s \times k_s, k_s \times k_s]\), where \(k_s\) is the size of the PSF, specifically the size of \(K\). Their correspondence with the gradient operator is observed in equation (5.3), horizontal, and (5.4), vertical, respectively.

Notice that \(K_s\) has the same shape of matrix \(S_s\) of equation (5.5), but in this case, with the values of the estimated PSF, \(\hat{R}\). So in order to continue with the solving process, equation (5.11) is redefined as:

\[ L = B(x, y)^T B(x, y) + X^T K_s^T K_s X - 2X^T K_s^T B(x, y) + 2 \lambda (X^T D_u^T D_u X + X^T D_v^T D_v X) \]  

(5.12)

Then, in order to minimize equation (5.12):

\[ \frac{\delta L}{\delta K} = 0 + 2K_s^T X - 2B(x, y)^T K_s + 2 \lambda (D_u^T D_u + D_v^T D_v) \hat{R} \]  

(5.13)

Finally, sorting and isolating \(\hat{R}\) from equation (5.13), we obtain the final desired deblurred image:

\[ \hat{R} = [K_s^T K_s + \lambda (D_u^T D_u + D_v^T D_v)]^{-1} K_s^T B(x, y) \]  

(5.14)

As mentioned above, deblurring L2-norm regularizer behaves better than L1-norm because, in this process, the regularization does not play an important role as inpainting or denoising. So, the impact of \(TK\) is less in the final solution. Hence, the result is only smoothed and a numeric stability with a small contribution of regularization is given.

Finally, and to conclude this chapter, we demonstrate below that recovering the original image is possible when the PSF is well estimated. We use the image example used above, \textit{peppers}, in a simpler case. It is blurred with a Gaussian PSF (size = 5x5 and \(\sigma^2 = 1.4\)), showed in Figure 5.5 with its correspondence PSF estimated by this method. On the one hand, to deblur the image,
we use the estimated PSF, and on the other, to compare with, we deblur the image using the original PSF. Our approach results can be observed in Figure 5.6 jointly with the original, its blurred version, the estimated sharp image and the deblurred one. The comparison of the results when deblurring with the estimated or the original PSF are illustrated in Figure 5.7. Then, in Table 5.1 statistical values are showed in order to compare the blurred version with its deblurred images.

![Figure 5.6. Original PSF (left) and estimated PSF (right)](image)

![Figure 5.7. Original peppers (Top-Left) and its blurred version (Top-Right), (Bottom-Left) Sharp estimated image, (Bottom-Right) Deblurred with the estimated PSF](image)
In this chapter it has been explained that with an estimated PSF, the desired deblurred image can be recovered. Thus, the technique followed has been described. To conclude, gaining resolution through the deblurring process is feasible. Usually, when SR, the LR sequence contains undesired blur and the PSF is not known, leading to a blind deconvolution to recover the sequence without blur. Hence, the proposed method tackles this problem and achieves great results when the PSF is well estimated. The demonstration can be clearly observed in Figure 5.8 and in Table 5.1: when the original PSF is used, the desired deblurred image is completely recovered; but when the PSF is estimated, the results are high enough to improve in resolution, which leads to increase the SR system performance.

Table 5.1. Statistical results: Peak Signal to Noise Ratio and Mean Structural Similarity of the image

<table>
<thead>
<tr>
<th></th>
<th>Blurred</th>
<th>Deblurred Estimated PSF</th>
<th>Deblurred Original PSF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>31.09 dB</td>
<td>38.21 dB</td>
<td>43.65 dB</td>
</tr>
<tr>
<td>MSSIM</td>
<td>0.8659</td>
<td>0.9721</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 5.8. (Left) Deblurred with the estimated PSF, (Right) Deblurred with the original PSF.
6. Restoration: Denoising

In this chapter we explain in more detail the denoising technique used to remove image noise. In addition, at the end of it, we show some comparisons between Bilateral Filter, Median Filter and our approach.

Denoising is the process that removes the image noise, which is a random variation of brightness or colour information in images. Since the noise is likely amplified at high frequencies resulting in undesired artefacts on the output image, denoising process also plays an important role in SR, despite the fact is lower than the deblurring impact. In Figure 6.2 we illustrate this effect of how noise affects an image.

For denoising, the same technique used to solve inpainting and deblurring is applied. Despite the fact the method has already been explained in Chapter 4, we define the following equations for TV regularization, which is the one considered. Furthermore, and on the contrary of deblurring, in this process the contribution of the regularizer must be bigger and TV impacts in a higher way than TK. Thus, the iterated model to solve the cost function (6.1) is followed.

\[ L = ||X_{in} - X_{out}||_1^2 + \lambda R(X) \]  \hspace{1cm} (6.1)

where \( X_{in} \) is the noised image and \( X_{out} \) is the desired denoised image.
\[ X_{\text{out}} = X_{\text{bef}} - \gamma \nabla L \]  \hspace{1cm} (6.2)

where \( X_{\text{out}} \) is the desired denoised image, \( X_{\text{bef}} \) is the previous denoised image computed, \( \gamma \) is the updating parameter and \( \nabla L \) is the gradient of the cost function.

\[
\nabla L_{TV} = 2[X_{\text{out}} - X_{\text{in}}] + \lambda [D_u \text{sign}(D_u X_{\text{out}}(x, y)) + D_v \text{sign}(D_v X_{\text{out}}(x, y))] \]  \hspace{1cm} (6.3)

where \( X_{\text{out}} \) is desired deblurred image, \( X_{\text{in}} \) is the noised image, \( \lambda \) is the regularization parameter, \( D_u \) is the sparse matrix of the gradient in \( u \) direction, \( D_v \) is the sparse matrix of the gradient in \( v \) direction and \( X_{\text{out}}(x, y) \) is the desired denoised image as a vector. Notice that in the first iteration, \( X_{\text{out}} \) has the same value as \( X_{\text{in}} \).

Our regularization choice has been TV and we proceed to justify it. In this case, \( L_1 \)-norm regularizer behaves better than \( L_2 \)-norm. Despite the fact that TV is computationally more complex than TK, the main interest for denoising is to preserve edges while removing the image noise, and TV achieves this purpose. In addition, smoothness is not assumed and let the regularizer work without losing image contrast. On the other hand, TK does not preserve edges and the result is smoothed. Even though TK can also remove the noise, the resulting denoised image is poor in resolution and differences between TV and TK are big enough to consider \( L_1 \)-norm regularizers. Its unique advantage here is that computationally is no complex, but the price for losing quality on the image is too high.

Once the method has been detailed and the equations described, in order to obtain qualitative results, we compare our method with other traditional techniques: bilateral filter and median filter, which both have been introduced briefly in Chapter 2, and we demonstrate that our method removes the noise in an effective way. But first of all, we remind briefly both methods.

**Bilateral Filter**

Bilateral Filter [27] requires Windows Size (W), which is related to the spatial Gaussian and based on the property of the Gaussian distribution (it should be around 2 or 3 times the standard deviation of the Gaussian), Sigma Spatial (\( \sigma_d \)) and Sigma Intensity (\( \sigma_f \)) as parameters and mathematically, at a pixel location \( x \), the output of Bilateral Filter could be computed as:

\[
I(x) = \frac{1}{W} \sum e^{-\frac{|x-y|^2}{2\sigma_d^2}} \cdot e^{-\frac{|f(y)-f(x)|^2}{2\sigma_f^2}} \cdot I(y) \]  \hspace{1cm} (6.5)
**Median Filter**

The Median Filter [28] acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels. Median filter is a spatial filtering operation, so it uses a 2-D mask that is applied to each pixel in the input image. To apply the mask means to centre it in a pixel, evaluating the covered pixel brightness’s and determining which brightness value is the median value. Figure 6.4 presents the concept of spatial filtering based on a 3x3 mask, where I is the input image and O is the output image.

![Figure 6.3 extracted from [44]. Median filter process](image)

Placing the brightness’s in ascending order and selecting the centre value determine the median value. The obtained median value will be the value for that pixel in the output image. Figure 6.5 shows an example of the median filter application, as in this case, habitually a 3x3 median filter is used.

![Figure 6.4 extracted from [44]. A constant weight 3x3 filter mask](image)
In Figure 6.5 we can observe the cameraman, the example used, with a random noise introduced. Then, in Figure 6.6 the results obtained with median and bilateral filter and our approach are observed.

Figure 6.5. Noisy image

Figure 6.6. Denoised images with different parameters for each method. Column 1: TV Regularization with $\gamma = 1$ (Top), $\gamma = 5$ (Middle) and $\gamma = 10$ (Bottom). Column 2: Median Filter with Window Size (WS) = 2 (Top), WS = 3 (Middle), and WS = 5 (Bottom). Column 3: Bilateral Filter with WS = 7 and $\sigma_{\text{spatial}} = 3.5$ and $\sigma_{\text{intensity}} = 0.5$ (Top), WS = 5 and $\sigma_{\text{spatial}} = 2.5$ and $\sigma_{\text{intensity}} = 0.1$ (Middle) and WS = 3 and $\sigma_{\text{spatial}} = 1.5$ and $\sigma_{\text{intensity}} = 0.01$ (Bottom)
We can observe in Figure 6.6 that our approach achieves the best results and removes almost all the noise, giving the highest performance. With bilateral filter we need to control three different parameters and to obtain some high performances, all of them should have an accurate value. On the other hand, median filter only needs one parameter like our proposed methods, so it is also easy and fast to use. However, its results are the lowest ones.

We used different values for the parameters in order to compare their impact. The most differences are noticed in bilateral filter but, on the other hand, in our approach, varying the value does not achieve big differences between the results. In conclusion, our estimated technique is capable to denoise in an efficient way, achieving important performances.
7. Experimental results

In this chapter, we evaluate and discuss, using Matlab, some results obtained with our proposed SR system in order to compare them with algorithms from SR literature. The chosen methods are the bicubic interpolation and the Farsiu method [19]. We have discarded other techniques because, due to the nature of our proposed approach, Farsiu is the more related.

We have made qualitative and quantitative comparisons. For the first one, the data sets tested have been gathered through the past several years in SR literature, and in Multi-Dimensional Signal Processing Research Group as well, where Farsiu belongs. For the second one, an original SR image has been downsampled adding blur and noise. Thus, the ground truth of the SR image is known in order to compare quantitatively the results using the following methods: Peak to Signal Noise Ratio (PSNR), Mean Absolute Difference (MAD), Mean Square Error (MSE) and Structural Similarity Image (SSIM). Finally, and to conclude this chapter, we evaluate our SR process step by step, contributing with the results and the histograms obtained after each block of Figure 1.4.

This chapter is distributed as follows: Section 7.1 describes and displays the qualitative results; Section 7.2 illustrates the quantitative ones; and in Section 7.3 the impact of each stage of our SR approach is evaluated.

7.1. Qualitative results

In this section, we apply our proposed SR system with a magnification factor 4 in order to compare with the two methods mentioned above showing qualitative results. The input frames used in each experiment are 16; however, we do not show all the sequence for each example. We will notice that in the experiments Farsiu iterative is mentioned. Let us explain the reason, Farsiu works well with magnification factor 2, but when a higher magnification factor is desired, an iterative process should be applied. This means that, firstly, for each of the sixteen images, the Farsiu method with factor 2 has to be computed. Once this is done, the Farsiu method with factor 2 has to be applied again to obtain the final SR image. This has to be done because of the problem of BTV regularization mentioned in the previous chapters. Thereby, Farsiu technique with magnification factor 4 could be achieved.

11 Farsiu results were obtained using the Matlab GUI publicly available in their website.
12 The original source can be found in [34].
7.1.1. Disk sequence

The following images are the first 4 images of the sequence and the original resolution is 57x49 pixels. It is a static image sequence but with small shifts of the camera. In some images, as we can observe in the first one or in the third one, appear some transcendental blur.

Figure 7.1. Disk sequence. 4 first images of the sequence

Figure 7.2. (Up-left) Bicubic interpolation x4, (Up-Right) Our approach x4, (Down) Farsiu x4 iterative

We note in this sequence that our algorithm obtains better results than the other ones. Even that the image is still a bit blurred, if we observe the text, we can read clearly “Colour makes the difference”. In the bicubic one, it is impossible to read anything. Notice that the reconstructed first LR image is one of the worst of the sequence. Note that some of the borders of the image are missing in the Farsiu algorithm, which is a known drawback of the algorithm.
7.1.2. Book sequence

The original resolution is 91x121 pixels. This sequence is similar to the previous one, the books are static but the camera has small shifts leading to introduce some new information to super resolve the reference image.

Figure 7.3. One image of the book sequence

Figure 7.4. (Up) Bicubic interpolation x4, (Middle) Our approach x4, (Down) Farsiu x4 iterative
In this sequence we can observe that Farsiu algorithm introduces some noise and outliers in the change of book edges or even in the letters “French”. Bicubic makes a not so bad SR image but with our algorithm we remove almost all the noise and if we observe the name “French”, as said before, the difference is big compared with the others methods.

### 7.1.3. Surveillance sequences

The original resolution is 288x352 pixels. The sequence is still but the camera shifts are more notable that the ones used before.

![Figure 7.5. One image of the surveillance sequence](image)

In this sequence we can observe small differences as the one described before, Book sequence. We can observe that Farsiu does not give a bad result as in the previous sequence. However, some noise appears in the black lines (on the top) or some “dirtiness” in the big white circle but if we look to the numbers, they are easier to read than with our algorithm.
7.1.4. Face sequence

The original resolution is 31x32 pixels. This one is a bit different from the others. Apart from the camera shifts, the man is moving and so, his mouth or his eyes too.

![Image](image)

**Figure 7.7. One image of the face sequence**

![Figure 7.8. (Left) Bicubic interpolation x4, (Middle) Our approach x4, (Right) Farsiu x4 iterative](image)

This sequence is more related with a real scenario, where appears a person and his motion is more difficult to estimate. This is the reason why the results are not as good as the other ones. Despite of the kind of sequence, our algorithm still behaves better than bicubic and Farsiu. Farsiu introduces lot of noise and distortion in the image. Finally, the quality of the bicubic image is acceptable, but the noise is still there and it is not removed.

7.1.5. Text sequence

The following images are the first 4 images of the sequences. The original resolution is 57x49 pixels and this sequence contains camera shifts, noise and blur. It has been selected to understand how the algorithms behave with images, which contains text to be read.

![Figure 7.9. Text sequence. First 4 images of the sequence](image)
As we can observe, in the sequence above, the Farsiu algorithm obtains a better result than ours because of his regularization, Bilateral Total Variation (BTV), which works better with images with high contrast in small spaces, like the letters (very close) and the background. For the other sequences, where the grey values are more distributed, Farsiu algorithm is not able to recover a good image as we can see in the results, where some outliers appear.
7.2. Quantitative results

In this section, we compare the algorithms described above with qualitative values. For this, we use an original high defined image, then, we downsample it getting 16 images with different shifts and added blur and noise. We use the magnification factor 4 to compare qualitatively the images too. The statistics we used were:

- Mean Square Error
- Structural Image Similarity
- Mean Absolute Difference
- Peak Signal to Noise Ratio.

7.2.1. Butterfly

Figure 7.11 and Table 7.1. (Top) LR image and the values for the blur and the noise added, (Middle) Sequences, magnification factor and registration used, (Bottom) Computational Time and Parameters used for our approach

In Figure 7.11 we can observe one of the sixteen LR images created with all the values from Table 7.1. Furthermore, the elapsed time and the parameters (such as the regularization constant or the number of iterations) used in each stage are showed in order to specify the conditions of the experiment.
Figure 7.12. Ground Truth

Figure 7.13. (Top) Bicubic interpolation, (Middle) Our approach, (Bottom) Farsiu iterative
Image Reconstruction and Restoration for Optical Flow-based Super Resolution

<table>
<thead>
<tr>
<th></th>
<th>Bicubic Resize</th>
<th>Farsiu algorithm</th>
<th>Farsiu Iterative</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>+18.09 dB</td>
<td>+16.44 dB</td>
<td>+18.41 dB</td>
<td>+22.47 dB</td>
</tr>
<tr>
<td>Average Abs. Dif.</td>
<td>0.0801</td>
<td>0.0811</td>
<td>0.0683</td>
<td>0.0469</td>
</tr>
<tr>
<td>MSE</td>
<td>0.0155</td>
<td>0.0227</td>
<td>0.0144</td>
<td>0.0057</td>
</tr>
<tr>
<td>SSIM</td>
<td>0.4481</td>
<td>0.5626</td>
<td>0.5622</td>
<td>0.7432</td>
</tr>
</tbody>
</table>

Table 7.2. Statistical comparisons between the three algorithms

In the butterfly example, we observe in Figure 7.13 that our algorithm preserves better the contours of the wings, even inside of them, where there are lot of details. If we look in the Farsiu result, it is smoothed and it is like an artificial image (like a watercolour). Finally, in Table 7.2 the statistical results are showed and our algorithm achieves the best performance followed by the bicubic resize.

7.2.2. Princeton

![LR image](image)

### Added blur and Added noise

| Values       | Gaussian, K = 5, s = 0.83 | Gaussian: 5e-3 |

<table>
<thead>
<tr>
<th>Number of sequences</th>
<th>Starting frame</th>
<th>Factor</th>
<th>Registration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>16</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Time</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Values</th>
<th>Derivative inpainting (250 iter)</th>
<th>PSF Estimation</th>
<th>Deblurring (Tikhonov)</th>
<th>Denoising (TV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2^5</td>
<td>K = 5</td>
<td>0.09</td>
<td>10</td>
</tr>
<tr>
<td>Time</td>
<td>87.46 s</td>
<td>34.37 s</td>
<td>19.13 s</td>
<td>29.29 s</td>
</tr>
</tbody>
</table>

Figure 7.14 and Table 7.3. (Up) LR image and the values for the blur and the noise added, (Middle) Sequences, magnification factor and registration used, (Down) Computational Time and Parameters used for our approach
In Figure 7.14 we can observe one of the sixteen LR images created with all the values from Table 7.3. Furthermore, the elapsed time and the parameters (such as the regularization constant or the number of iterations) used in each stage are showed in order to specify the conditions of the experiment.

![Ground Truth](image)

**Figure 7.15. Ground Truth**

![Bicubic interpolation](image)

![Our approach](image)

![Farsiu iterative](image)

**Figure 7.16. (Top) Bicubic interpolation, (Middle) Our approach, (Bottom) Farsiu iterative**
In the Princeton image, we introduced more noise to prove how the algorithms deal with it. With the bicubic result, as observed in Figure 7.16, the achievements are the worst ones and Farsiu is not capable to remove all the noise. However, as in example showed in Figure 7.10, the letters (“Hamilton Jewelers”) obtained by Farsiu are more readable than with our algorithm because of the regularization used. Our algorithm is capable to remove the noise but losing some contrast. In Table 7.4 the statistical results are showed and our algorithm achieves the best performance.

We have demonstrated in these two examples that our algorithm behaves better than the other methods evaluated: bicubic, Farsiu (applying directly magnification factor 4) and Farsiu iterative (applying firstly, magnification factor 2 and then, again factor 2, as explained before). For all the statistical results we have considerer, the values for us are the best ones. In addition, we can observe in the images that we almost recover the Ground Truth.

### 7.3. Impact of each stage

In this section we show the normalized histograms for the Princeton sequence evaluated at each stage of our proposed SR system. We compare them with the histogram from the original image. The main purpose of this section is to illustrate how each stage affects to the final resolution of the image and to identify which one gives the lowest performance.

In Figure 7.17 we illustrate these histograms we have just mentioned to make the comparison easier.
Figure 7.17. (Top-Left) GT, (Top-Right) Our approach, (Bottom-Left) After Inpainting, (Bottom-Right) After Deblurring

We observe that the ground truth histogram and the one obtained with our approach are very similar. But if we focus in each part, we notice that after Inpainting, the brightness and the contrast is distributed with the same shape but the peaks in the [0, 0.1] margin are smoothed and they disappear. However, after deblurring, we recover those peaks. But after denoising, we observe that we lose again those peaks and near 0 and 1, we lose some contrast too. This is happening because the noise level is really high and to eliminate it, some contrast and brightness is lost.

Finally, to conclude this Chapter, we talk about the computational cost. As we have observed in Tables 7.1 and 7.3, Farsiu and bicubic are faster than our algorithm, which is not now prepared to work in real time. However, in the next chapter we propose its reduction as a future work.
8. Conclusions

In Chapter 1 we have commented that new technology and consumers' viewing habits are changing the face of TV. What was once thought of as a mere possibility is now a reality, as consumers want more choices and flexibility in their viewing experiences. The introduction of High Definition displays has led to advances in the field of SR and constant improvements are achieved, which provide new mechanisms for enriching visual contents and match these new display capabilities.

Hence, in this thesis, we have presented a novel system for SR reconstruction and restoration. We have presented a modular variational approach for solving the subtasks of reconstruction and restoration in reconstruction-based SR. This, combined with contemporary state-of-the-art optical-flow estimation, allows us to obtain deblurred and denoised SR frames in video sequences showing spatial aliasing.

The main advantage of proceeding in a modular manner is that we are able to introduce a suitable regularizer for each substage. This allows us to provide closed-form solutions to some of the tasks, thus providing analytical optimal solutions when employing the right formulation (e.g. $L_2$ data fidelity term with Tikhonov regularization).

This chapter is organized as follows: in Section 8.1 we present the achievements obtained; and in Section 8.2, we introduce the beneficent for segmenting in order to avoid the presence of complex motion.

8.1. Achievements

In Chapter 1, after introducing the SR concept, we proposed the goals for our work. In this thesis we have developed an entire SR system in order to cover those points. But, firstly, let us recall the goals before explaining if they have been covered:

- Implement an entire SR system for video sequences including the registration, reconstruction and restoration steps.
- Choose a robust algorithm for registration among state-of-the-art techniques.
- Define an effective strategy to reconstruct the HR image.
- Develop algorithms capable of improving the quality of the reconstructed image (restoration step).
- Detect elements of SR that could be improved leading to the processing of complex real video sequences.
Once we have cited the goals, we proceed to focus in each point. In our work we have presented, through seven different chapters, our approach. Thus, and following with the same bullets, we explain if this thesis has covered every single goal:

- A whole SR process has been presented, including the three steps involved in it: registration, reconstruction and restoration.
- The literature regarding registration has been studied in detail, in order to compare and thus choose the most effective method for our SR approach.
- An inpainting method based on variational formulations has been presented to tackle the reconstruction in SR, obtaining effective results when the motion is not complex.
- In order to deal with restoration and to increase the quality of the reconstructed image, two novel approaches - deblurring and denoising - have been proposed. PSF estimation helps to improve the deblurring results when compared to typical assumptions on the shape of PSF, whereas, we have based our approach in [8] introducing a new formulation for the edge profile to obtain a robust sharp image. Hence, encouraging results have been achieved. Furthermore, following the same variational approach followed in inpainting and deblurring, we have proved that our technique achieves higher levels of noise removals in comparison with typical denoising algorithms.
- Motion within real scenarios presents the greatest problems for SR. It has been found that complex motion, e.g. that of players in a football match, presents more difficulties in registration. Due to these problems of registration, reconstruction and restoration become harder to perform. In the next section, we present some new approaches to deal with these local motion problems.

In general, we consider that our work has succeeded in the improvement of SR departing from a LR video sequence. However, the results showed in Chapter 7 confirm that there is still ways to improve in SR. Thus, in Section 8.2 we explain how our proposed SR system can be extended.

## 8.2. Future work

Our work has possible extensions for the future. We present several ideas that we started to work on, despite the fact that they were beyond the scope of this thesis. First of all, the computational cost should be reduced both to work with real time video sequences and to obtain the results in a faster way without the loss of resolution. The algorithms we developed were not optimized because that was not within the scope of the thesis, but as we have seen, the results are good enough to improve the techniques used and to apply them to other application in real time settings.

We have discussed the problem of registering images for video sequences in real scenarios such as a football match. In this case, there is a lot of motion, including that of the players, the crowd, the hoarding, all of which are moving in a random manner. Current algorithms cannot
estimate their motion properly and in fact, without a good registration, SR is difficult to achieve. Hence, we consider whether introducing segmentation in the image will improve the results, both in registration and in the reconstruction steps. Segmentation, in image processing, is the process of partitioning an image into multiple sets of pixels with similar features such as colour, motion, intensity, histogram, shape, etc. The main goal is to simplify the image but in SR, the goal is to separate features with similar motion. Imagine a football match: there is a football stadium, a crowd, twenty-two players, a referee and a ball. The sequence is captured by a High Definition Television (HDTV) camera. If we try to apply SR without pre-processing the sequence, it is highly probable that the registration step will not be able to estimate the motion of all the elements of the sequence, and therefore, the final SR result will not be adequate. This is due to the fact that the motion of the players is completely different to that of the stadium. Thus, the idea is to separate the background, which could involve the stadium and the crowd (small motion) and the players. Applying segmentation, we can separate both and then register each of them separately.

We propose to consider a simple approach to improve the registration in sequences with very complex motions. We separate (segment) the image into objects with high motion and objects with low or no motion. Then, apply SR (registration, reconstruction and restoration) only to the part with less motion. Finally, add the segmented motion feature in the HR image with bicubic interpolation. As we see in figure 8.1, the result is promising. This method will allow the possibility of perceptual increase in the quality of the final SR image, without the need for any registration of complex motions within the sequence.

![Figure 8.1. First (left) and Second (right) LR images from the video sequence](image-url)
We notice that the background gain in resolution. If we achieve to separate the motion parts from the background, SR is possible with the available registration methods as we mentioned before. Before introducing the next step, let us show in Figure 8.3 the gain of resolution in the background.
As it has mentioned before, the next step is to segment the video sequence automatically. As a first approach, we super segmented the current frame with the tool EDISON available on [38] and based on [39] and [40]. In figure 8.3 we can observe the result obtained.

Then, we use another tool [41] to segment the kid playing with the yoyo. It needs two different markers, one for the background and one for the motion, which wants to be segmented. To avoid do it manually, we created the markers using the K-means process applying it in the over segmented image. In figure 8.4 we show the first result obtained using [41] and K-Means. But this is only a small experiment to demonstrate that segmenting the motion SR could be achieved in real video sequences.
Finally, and to conclude this thesis, we have suggested in Chapter 1 two questions, which we are now prepared to answer:

- Which is the maximum real increment in resolution we can achieve?
  - In synthetic scenarios a magnification factor higher than 4 is not necessary because the resolution gained beyond this factor is unnoticeable. The same happens when a real scenario is captured to apply SR on it, where the limit of the magnification factor is 2.

- How many LR images are needed to maximize the image resolution and quality? Is there a bound from which it makes no sense to add more LR images?
  - The ideal case is to use $m^2$ input frames, where $m$ is the magnification factor. But this rarely happens; it depends on the SR algorithm used. Ones are more prepared to tackle SR problems with fewer frames than others. The fewest the input is, the most complex SR system should be. Despite the fact there is not a fixed number of input frames, it is desired to use as fewer images as possible.
9. Bibliography


[34] - http://www.soe.ucsc.edu/~milanfar/software/sr-datasets.htm


[38] - http://coewww.rutgers.edu/riul/research/code/EDISON/


[41] - www.fascinate-project.eu