CHALMERS

Methods and algorithms for image fusion and super resolution

Master of Science Thesis

ANDERS ÖHMAN

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden, 2009
Report No. EX011 2009

Methods and algorithms for image fusion and super resolution

Author:
Anders Öhman

Supervisor and examiner:
Irene Gu

March 23, 2009

Abstract

This project is focused on image fusion of images of different focus depth and fusing for super resolution. The aim is to study these concepts and provide simulations and evaluations on various implementations. When performing multi focal fusion the images are decomposed by wavelets to obtain high frequency coefficients which is used to determine which parts of the input images that makes it into the fused image. The same technique is tested on images of different modality. Super resolution utilizes measurements of subpixel shifts between several low resolution images of the same scene to create a fused image of higher resolution by interpolation and image enhancement. The thesis describes a modular approach for super resolution where registration, interpolation and blind deconvolution is treated as separate modules. Tests are performed for different images, choices of modules and input parameters.

List of abbreviations

CT	Computerized tomography
HR	High Resolution
LR	Low Resolution
MBD	Multiframe Blind Deconvolution
MR/MRI	Magnetic Resonance (Imaging)
PSNR	Peak Signal to Noise Ratio, evaluation
SNR	Signal to Noise Ratio
SR	Super Resolution

Acknowledgments

The author wishes to give credit to Filip Šroubek and Jan Flusser who provided the Multiframe Blind Deconvolution used as the basis that the Super Resolution within this project is built around. Also 2D spline interpolation program provided by Tiesheng Wang PhD student at Chalmers university of technology.

Contents

Chapter 1

Introduction

Here the purpose of this thesis will be presented, the terms image fusion and super resolution will be explained, what usage they might have with a few examples. In this thesis, several methods for image fusing and super-resolution has been studied, the first study is related to fuse two characteristically different (one is focused on foreground, another on the background) images taken from the same scene that automatically takes the relevant information from each image and puts them together to yield a fused image that is both focused on foreground and background. The second study concerns using several nearly identical images taken from the same scene, however, contained small shift changes. These images are used for obtaining an image of higher resolution. Methods have been investigated for each of those cases, together with computer simulations. The aim in both cases is to aid in visual assessment of the images.

1.1 Fusion from multi focused or multisensor images

The purpose of image fusing is to obtain an enhanced image containing the information present in multiple images, the algorithm could be as simple as averaging both images together or more sophisticated as one that extracts the relevant information from each image and construct one new from that. The goal is to develop an algorithm that fuses two images with as good result as possible. The images have to be of the same scene but there has to be some differences such as different focal depth, modality, light spectrum or time. The key is to determine which information that is relatively more important at one specific location in one of the images compared to the other. Without specifying what is important information like adding templates to look for

one can make an assumption. Lines and sharp contours contain important information, thus favoring high frequency image content is a possible solution.

Examples A very simple example is when the user has got two pictures taken on a person or an object, in one of the images the person is in focus and the background isn't while the other image is the opposite. It is desirable that an image that is fully focused. This is possible because the images together have sharp areas in all places. Using some kind of image processing one could cut and paste the sharp foreground object onto the background of the other picture, or he could use an algorithm like this.

Another example is when a doctor is examining two images, one obtained by Magnetic Resonance Imaging (MRI) and one by Computer Tomography (CT). Both images contain relevant information, the MRI works by aligning hydrogen atoms present in water in the body tissues and by altering the magnetic field causing the nuclei of the hydrogen to emit a rotating magnetic field of its own which is detectable, creating an image showing various tissue types. The CT works like an x-ray producing images depending on the density of the tissue. Instead of showing the images side by side and make comparisons there might be desirable to have only one image.

1.2 Super resolution from a sequence of low resolution images

Resolution defines how densely sampled an image is, the higher amount of pixels per unit length allows more details to be visible in the image. Several technical difficulties follow the aim of obtaining good high resolution images. A typical way of making the sensor detect finer detail is to decrease the pixel sizes thus increasing their number on a specified area. However, the amount of light allowed on each pixel is decreased in the same manner. By doing the sensor more sensitive makes it more prone to noise. The method called super resolution is to create a high resolution image from several low resolution counterparts. Using the fact that there might be shifts smaller than one pixel between the low resolution images depicting the same scene, they can be put together resulting in a picture of high resolution. A common way of obtaining the images is by using adjoining video frames. If the user already has several low resolution images of a motif and wants a high resolution image super resolution is a possible choice if it was hard obtaining the images. It's also more expensive manufacturing devices capable of taking high-resolution pictures directly than having a low cost device taking several pictures and by

mathematical means create the high resolution image. In medical imaging, where the amount of radiation affects what resolution is possible it is possible to lessen the dose the patient is exposed to.

Chapter 2

Overview of theories and related work

In this chapter the problems are formulated based on what's found in the literature. The aim is that when the reader reaches the end of this chapter he will have a sufficient understanding of the purpose of the methods and experiments in this thesis.

2.1 Fusing different focus

The main source of information for this part of the thesis is found in the image fusion articles written by Petrovic [2] and Piella [3]. Image fusion is a process of analyzing the input images to determine what content to keep for a composite image. Its goal is to reduce the load for a human observer by decreasing the number of images that he must browse through, in some cases simultaneously. Common for many image fusion tools is to start with some kind of multi resolution decomposition to split up the image based on its frequency content. The higher frequencies are known as the detail coefficients and are used as the main source of information when determining if a location in the image contains information to keep. The low frequency parts or approximation is then used for creating more levels of detail coefficients and approximations, generally speaking the frequency spectrum is split in an upper and lower part and the lower part is once again split until the level wished for is reached. The concept of splitting up the frequency spectrum is shown in figure 2.1 In this thesis wavelets are chosen as the method for doing the decomposition, here three detail coefficients are received; horizontal, vertical and diagonal.

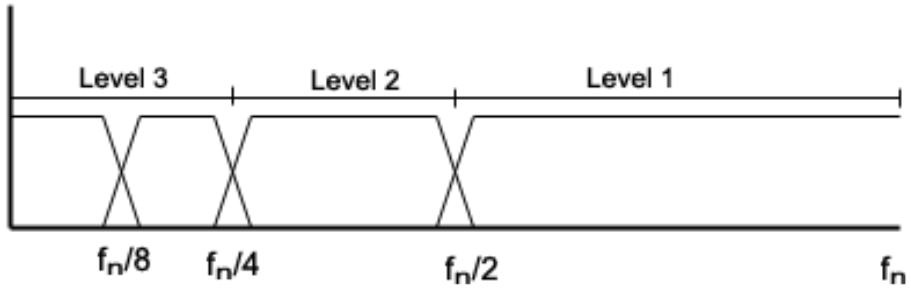


Figure 2.1: Example of a frequency band decomposed by wavelets

2.1.1 Addressed problems

With regard for the aim of image fusion there are a certain number of rules that have to apply to the fused images. Firstly, all information that is necessary for decision making should be present in the composite image and thus the algorithm must carefully choose the content depending on what the observer needs. This might favor a specialized program instead of a general purpose image fusion program. Next is that the algorithm itself is not allowed to create content of its own or change the present content.

2.1.2 Methods of fusion

Fusion of images can be performed on many different levels. As defined by [2] the most straight forward fusion is pixel level or sensor level fusion. It's when the algorithm uses the pixel value only to make decisions about how the fusion is to be carried out. Next level is feature level, where features in the images are extracted by simple means such as segmentation, morphological operations or similar automatic processes. The increased understanding of this level is that certain pixels are part of the same group. On the top there's symbol level fusion there classification of said features are made. The report by Petrovic [2] focuses on the first level namely pixel level fusion. Without leaving the field of pixels there are still many possibilities to choose from when obtaining the data. Picking all the data from a 3×3 neighborhood is one such option; it allows a higher robustness in the case that it represses salt and pepper like patterns in the fused image. He also examines the concepts

of horizontal and vertical integration. In horizontal integration all the detail components of the same level in the wavelet decomposition are used together to form weights at one location of the image. In vertical integration, the detail components from different decomposition levels acts as parent and child and are used together at a specific location when calculating weights. Piella writes about region based fusion and introduces a concept of matching; the match measure compares the similarity between the images to further decide if the algorithm chooses either one of the images or an average. This thesis will cover pixel and feature fusion as well as it incorporates horizontal integration.

2.2 Super resolution

A scene could be described as a continuous two dimensional function viewed by an aperture of any kind. When digitalizing the scene into an image the signal is sampled, the more densely sampled the signal is, the better resolution and image quality is obtained, that is if the sensor is sensitive and selective enough to capture the scene as it is without distorting effects. A common drawback from decreasing the pixel size for better resolution is that the amount of light also decreases for each pixel generating shot noise that damages the image quality. In addition light bends when passing the lens and is truncated creating ripples around edges in images. The thesis is based on several sources, the most significant being [5] as a demo of the program the report is based upon is used as it is in a step of the process. Other significant sources are [6] and [7] defining the general structure of the algorithm.

2.2.1 Mathematical formulation of low resolution images

A crucial part of image restoration and enhancement is to understand what affects the image and causes the resolution to decrease. These processes are fairly well understood and its definitions are widely available in the literature. The terminology chosen is partly taken from [5] and [6]. Let x be the original high resolution image of size $L_1N_1 * L_2N_2$. The image is a continuous two dimensional signal sampled at a high frequency. D is the down sampling of factor L in x and y dimension independently thus yielding several LR image y_k of size $N_1 * N_2$ by using only every Lth sample from the HR image. If all pixels are used there will be L^2 low resolution images. M_k is warping of the image, it can contain translation, rotation, scaling and skewing. Both D and

M are regarded as operators affecting the image. B_k is the blur and warping expressed as a function convolved with the image. N_k is noise for each LR image respectively and as it does not rely on the pixel intensities of the image it's is regarded as an additive term. The idea behind this formulation is that y_k is the only known variable and by using several images, hence the index k where x is the same between them it is possible to remove all the degradations.

$$y_k = D(B_k * x(M_k)) + n_k \tag{2.1}$$

Downsampling The HR image is as stated ideally sampled at Nyquist frequency. The sampling at high resolution could be described as a fine grid covering the image here each cell in the grid represents one pixel at a certain gray level. When downsampling, a coarser grid is placed upon this fine grid, the cells in this new grid would each cover more than one cell in the fine grid thus losing the fidelity within the cell as it can only contain one gray level. The gray level is simulated as being the level of one of the pixels $x = 1, \cdots, L_1$, $y = 1, \cdots, L_2$ denoted as phase. If the motion is regarded to be global i.e. no local changes in the picture the phase is set to the same when creating every LR pixel. The principle is shown in Figure 1 where a HR image is down sampled by a factor of 2. In a simulated scenario it is possible to create an ordered set of LR images containing all the information at phases suited for easy reconstruction.

Blur In this category all kinds of blur is included. First the sensor point spread function (PSF) which is caused by the aperture cutting off the light waves. There might be local blur caused by moving objects within the image or objects that are out of focus, there may also be global blur caused by movement from the sensor itself during the time the picture was taken. In this project only global blur will be taken into account, a local blur can be regarded as global if the image is cropped down to only show an area affected by the same type of blur. When not dealing with simulated LR images taken from a single HR image there might be different blurs depending on what frame one are looking.

Warping This effect describes all kinds of geometric distortions happening between the LR frames and the original HR image. Possible warping operations are translation, rotation, scaling and skewing. As in the case of blur, warping effects can be classified as either global or local, where global warping acts uniformly over the whole image and local only affects a part of the

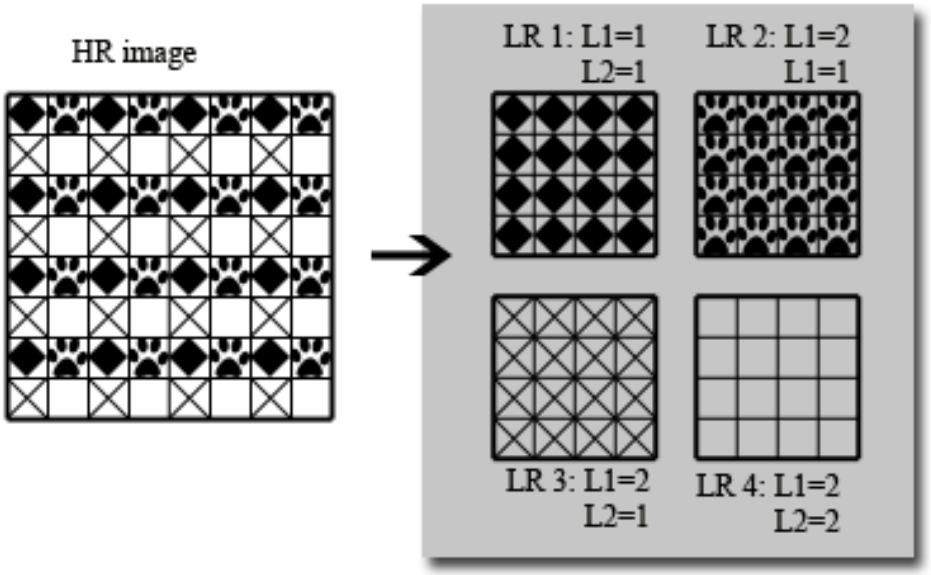


Figure 2.2: An example of downsampling, the pixels of one image are evenly divided between two images each in two dimensions

image. There are no limits of how small or large changes due to warping the image is exposed to as long its still possible to recognize features belonging to the scene described in the desired HR image.

Noise The noise is a function of random values that is added to the image, it can be visualized as a grainy layer covering the image. For the case of Gaussian noise used in these simulations the random values are following a Gaussian distribution, previously mentioned shot noise follows Poisson distribution. Its relative signal power regarding to the image's defined in decibels. Although noise handling is combined with the deconvolution still it's important to know it's characteristics since it's used as an input parameter for the blind deconvolution algorithm.

2.2.2 Super resolution strategies

How the super resolution is preformed differ greatly between the different articles studied. Methods have been developed both for one starting image as well as several [12], focusing on the approach using several images it is possible to divide the process into three steps; registration to align the images, interpolation to increase the resolution and blind deconvolution to correct

the blur. As stated in [6] the approaches either carry out all three processes simultaneously or one by one. Šroubek and Flusser aims to carry out all processes simultaneously in [5] while [7] have a clear distinction between the processes. One interesting question is how much information is needed to make an acceptable super resolution image according to [5] the minimum amount is L^2 that is the sum of pixels in all the low resolution images is equal to or greater than the super resolution image. The other example encountered on Wikipedia [12] used 9 images for a super resolution enhancement of 2. From these observations the maximum number of images used is set to L^2.

Chapter 3

Methods studied in this thesis

This chapter is about what methods that were studied during this thesis, how they work and how they were implemented in the simulations. The chapter consists of two sections, one about multi focal fusion and one about super resolution fusion. The methods are of varying extent ranging from a simple summation to long algorithms, they share the common characteristic that they drives the fusion process forward by either providing inputs or using those inputs in the images.

3.1 Fusion of multi focal images

Assuming there is no relative warping present between the images and that the images are of equal sizes. Available are two images (A and B) captured from the same scene in such a way that a specific pixel in any image shows the same small part of the motif but in a different way. The fusing algorithm follows the general scheme outlined in the previous chapter, an outline can be seen in figure 3.1 naming the intermediates. The steps corresponding to the numbers in said figure are found in the list below:

1. Wavelet decomposition
2. Watershed segmentation (Only used in region-based fusion)
3. Merging of segments (Only used in region-based fusion)
4. Weighting
5. Discrimination
6. Fusion

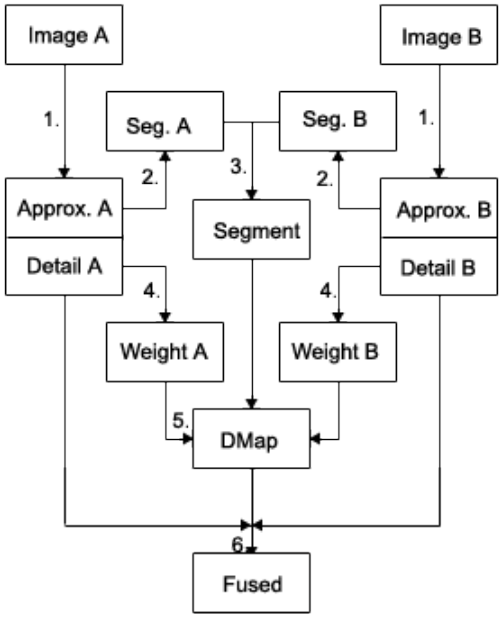


Figure 3.1: Diagram over the image fusion process

3.1.1 Image decomposition by wavelets

Since well focused image part contains sharp edges (with large magnitudes) in the high frequency band, sub band filters using discrete wavelet transforms are employed, followed by examining the edges or magnitudes in the high frequency band. Feature detection using gradient magnitudes and similar techniques might discover the sharp edges but fail when it comes to smoother and coarser details. Likewise the Fourier transform may present the frequency content in an image in a good way but there is nothing in it that describes the actual location of the features in the image. The idea behind the multi resolution processing is to make a decomposition the signal is split in two according to frequency content resulting in one high pass component and one low pass component. The high frequency component is regarded as the details of the image and the low frequency component the approximation. The result could be interpreted as a pyramid because the size of each new level is half of the previous. The low pass signal could then be decomposed another time and so on. One such type of multi resolution decomposition is the wavelet transform.

The information on wavelet is found in the textbook "Digital image processing" [1] and on Wikipedia [10]. The wavelet transform works like Fourier

transform in the case that the signal is convolved with a function, but instead of using a constant function like the sine waves used in Fourier transforms the convolution is carried out with a bi orthogonal function called a wavelet. The wavelet acts as a filter as well as a window function. For the case of a time dependent signal, starting at its smallest size the highest frequencies are extracted with high time precision. As the window grows the time resolution is worse but the frequency resolution is improved due to the smaller interval that is split. The Wavelet function consists of a finite asymmetric waveform, in its basic form it is referred to a mother wavelet. The time and frequency dependence is implemented by scaling and translating the mother wavelet, the scaling j is binary and the translation m,n is by integers. This is defined as the scaling function. In its original form j is 1 and m,n is 0. The signal is convolved with the low- and the high- pass wavelet function creating the high- and low frequency components which then are down sampled to half their size. The wavelet is scaled and the process can start again for next level.

The method used in this project is the 2D discrete wavelet transform. Assuming the transform kernel is separable, it is performed by first wavelet transform the image in the y direction by treating each column as a signal. The second dimension is implemented by applying the transform another time, this time x dimension by transforming the rows of the previously created high and low frequency components. This ends up in having four components for further processing. The approximation (I_i^a) consists of the low frequency component in both directions, the horizontal (W_j^h) and vertical (W_j^v) details consist of either high pass or low pass in one dimension and its counterpart in the other. The diagonal (W_j^d) component consists of high frequency components of both dimensions.

$$\begin{aligned} \phi_{j,m,n}(x,y) &= 2^{j/2}\phi_{1,0,0}(2^j x - m, 2^j y - n) \\ \psi^i_{j,m,n}(x,y) &= 2^{j/2}\psi_{1,0,0}(2^j x - m, 2^j y - n) \quad i = \{h,v,d\} \end{aligned} \tag{3.1}$$

$$W_\phi(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{n=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x,y)\phi_{j,m,n}(x,y) \tag{3.2}$$

$$W^i_\psi(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{n=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x,y)\psi^i_{j,m,n}(x,y) \quad i = \{h,v,d\} \tag{3.3}$$

A common way of display the transform components is to do a mosaic containing the approximation in the upper left corner horizontal details to it's right, vertical details at the bottom left and diagonal details in the bottom right. It will also be clear for the viewer that their combined size is equal

(a) (b)

Figure 3.2: (a)The cameraman sample image from MATLAB. (b) Decomposition by wavelets in one level.

to the size of the image before decomposition. An example of how wavelet decomposition of images may look like can be seen in figure 3.2. Even though the developed application can support several levels of decomposition it never performs more than one level in the scope of this thesis. The reason for that is that artifacts are generated when the reconstruction is made and those artifacts increase for each level. Furthermore, in the case of multi focal fusion it is possible to classify parts of the image as in or out of focus just by using the first level.

The inverse transform seen in (3.4) is a summation of the four components of the level above. It reconstructs the original image from the components.

$$\begin{aligned} f(x,y) = & \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\phi(j,m,n)\phi_{j,m,n}(x,y) \\ & + \frac{1}{\sqrt{MN}} \sum_{i=\{h,v,d\}} \sum_m \sum_n W_\psi^i(j,m,n)\psi_{j,m,n}^i(x,y) \end{aligned} \tag{3.4}$$

3.1.2 Image segmentation using a marker controlled watershed algorithm

To be able to utilize region based fusion it is necessary to segment the image into regions. The method of choice was Marker controlled watershed segmentation found in the MATLAB documentation. Even though it's made

primarily to segment images containing several bright objects separated by darker valleys, it was considered sufficient for this task.

It is carried out by using a series of morphological operations to set the markers for foreground and background and then doing watershed segmentation on the marked image. First find the edges using a sobel gradient filter. The foreground markers are created using an opening by reconstruct/closing by reconstruct operations to remove much of the textures that may interfere with the regions. Next define the foreground markers by picking out the local maximums of every region. Apply a closing and a dilation operation to even up the markers. The background markers are computed by firstly applying a threshold to the foreground image to force a space between fore and background. As a binary image now is available the euclidean distance is computed to the non-zero foreground markers. A watershed transform makes the background markers appear as ridge lines. Imposing regional minimums to the gradient image and then using it for watersheding.

As the segmentation is made on each of the input images they have to be merged into one segmentation. It is performed by a simple algorithm that for each pixel in the image, checks what region each of the segmentations assigns it to and for each unique combination assigns it to a region in the composite segmentation accordingly.

3.1.3 Decision map

The decision map is a matrix of the same dimensions as the images that is to be fused, it could be described as a guide that tells the fusion algorithm from which one of the input images each pixel will be taken from. It is computed from the detail components of the decomposed images which are recalculated as weights. Depending on what rules that are set the values of the decision map are calculated from the weights. Two distinct versions of the decision map could be made, the one that for each pixel, only takes information from one of the images and the one that makes combinations of the pixel values in all combinations. Between this could be intermediates, for example: If there is a significant difference between the images in that certain position, act like the first and only choose the most significant, else way choose to make a average or weighted average.

Weights The weight of a specific object in an image is derived from the detail components. In its simplest form only one pixel is taken into account. It could be expanded to a neighborhood without much effort so it is still seen as pixel-wise fusion. Going further into region based fusion [3], whole areas of the same general gray level separated by edges are treated together. When

doing region-wise fusion all pixels are assigned to regions obtained by some means, for instance watershed used in this. In this project a simple square grid and watershed segmentation were tested. The weight consists of the energy of chosen detail coefficients at the location x, y. In equation (3.5), the horizontal and vertical components (I_i^h and I_i^v) are chosen from image i (i is either A or B) to calculate its weights. For region based fusion the weight is the same for the whole region, seen in eq. (3.6), there is a summation over all pixels in the region $\mathcal{R}$ divided by the area size represented by the number of pixels here denoted $|\mathcal{R}|$ obtaining an average, this average is assigned to all pixels in the same region.

$$w_i(x, y) = (I_i^h)^2 + (I_i^v)^2 \tag{3.5}$$

$$w_i(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{n \in \mathcal{R}} w_i(n) \tag{3.6}$$

Discrimination When the weight for a specific pixel or region defined in both images is obtained they are to be compared against each other. Depending on the application either one or both objects are of interest, thus two discrimination criterias one where $w_A(x)$ and $w_B(x)$ are proportionally represented in the map at x and one where the one has the largest weight is solely represented.

As the algorithm shouldn't change any properties of the image such as brightness it is important that $w_A(x) + w_B(x) = 1$. The decision map d_m is created either by (3.7) or (3.8) depending on which criterion used.

$$d_m(x, y) = \frac{w_A(x, y)}{w_A(x, y) + w_B(x, y)} \tag{3.7}$$

$$d_m(x, y) = \begin{cases} 1 & \text{if} \quad w_A(x, y) > w_B(x, y) \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

3.1.4 Fusion

The fusion of the images uses the map made in the previous step. It is performed using the images in their decomposed state. For each wavelet decomposition component (j = A,H,V,D) the fusion is made according to (3.9). As the decision map is made to correspond to image A at x it also applies that $1 - d_m(x)$ corresponds to the contribution from B. After implementation of 3.9 on the decomposed images, the result is a decomposed fusion image

$f_i(x, y)$, the inverse wavelet transform are performed on them to yielding the fused image.

$$f^j(x, y) = d_m(x, y)I_A^j(x, y) + (1 - d_m(x, y))I_B^j(x, y) \tag{3.9}$$

3.2 Super resolution methods

As the aim of the project is to obtain a super resolution image from several low resolution images taken from a same scene, the problem formulation could be made simpler by assuming that the only warping present is translation in the sub pixel domain because instead of making the computations more demanding it is an absolute necessity for super resolution. The problem is simplified further by assuming that the only blur present is a point spread function that is equal for all images.

As can be realized from looking at the mathematical formulation of the degraded image there are a number of degradations to be handled and they all could be regarded as unknown. When enhancing the image by removing degradation it first has to be identified and then removed. When handling downsampling, blur and noise together it does not necessarily need to be more complex but less intuitive.

When splitting the process up, registration is always carried out first, this is for two main reasons. Firstly the interpolation is totally dependent of the data from the registration to work. Then there could also be matter of selecting images suited for any of the following steps. In the deconvolution step the result may be really poor if the HR image x differs between the frames. There should also be possible to put either interpolation or Multiframe Blind Convolution next, in the case of putting MBD first of those two, the intermediate images has to be re, registered due to previously stated dependence of registration data in the interpolation. In figure 3.3 the process structure is shown as it is carried out in this project, the order of methods is registration, interpolation and multiframe blind deconvolution. The number of input images to use depends on what the user regards as sufficient. A fact is that both interpolation and multiframe blind deconvolution requires several images and results in one. In figure 3.4 the decline in images is shown until the user has got one image when the process is finished. As the order of applying interpolation and multiflame deconvolution is assumed interchangeable they are here referred to as step 1 and 2 depending on which one is executed first. From the figure it's apparent that depending on how many images is used in the second step, the first step is to be run as many times as there should be images. The number could be expressed as $N_{images} = \sum_{l=1}^{L} M_l$ where N

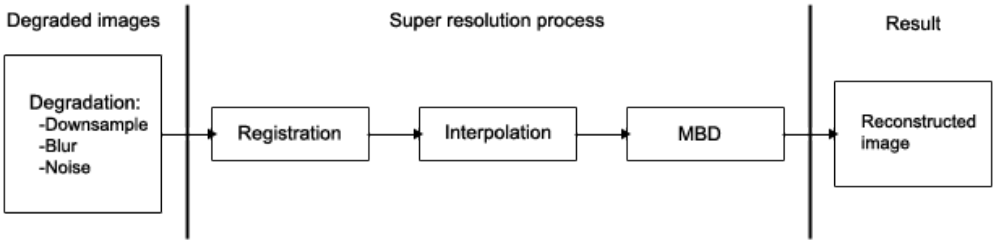


Figure 3.3: Illustrates the work flow of the super resolution process, in this project registration, interpolation and blind deconvolution are treated separately.

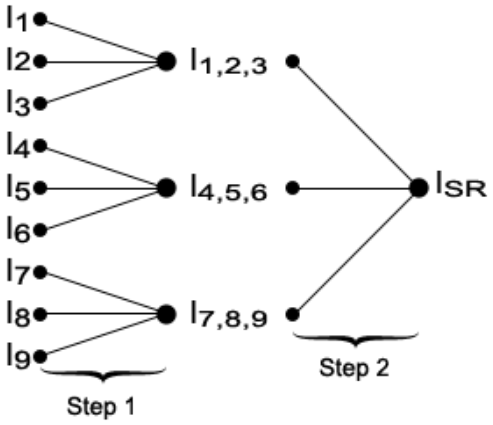


Figure 3.4: Illustrates how the number of images available is changing as the process goes on.

is the total number of images, L is the number used by the second step and M_i is the number for each repetition of the first step.

3.2.1 Registration

To perform super resolution it is crucial for further processing, to perform super resolution one must know the warping parameters to either correct them or exploit them. In this implementation and with warping restricted to subpixel shift the output from registration is used in further steps. The registration algorithm is divided into two parts, the first part performing cross correlation on images of increased size to obtain a rough estimate and the other part uses the result from the first to fit a polynomial which minimizer refines the output of the cross correlation.

Pixel-level registration Cross-correlation is normally used to compare two signals [a..b]. The sources for this section is [7] and [13]. The cross correlation works by comparing two sets of data while shifting one of them by u in the x-dimension and v in the y-dimension as in equation (3.10), the resulting squared difference is stored in matrix $d^2_{f_1,f_i}(u,v)$ and it is possible to find a minimizer at the coordinates u, v where the images are as similar as possible. It could be expanded as shown in equation (3.11) where f_1 is one of the low resolution frames used as reference and f_i is the LR frame which shift is measured.

$$d^2_{f_1,f_i}(u,v) = \sum_{x,y}[f_1(x,y) - f_2(x-u, y-v)]^2 \tag{3.10}$$

$$d^2_{f_1,f_i}(u,v) = \sum_{x,y}[f_1^2(x,y) - 2f_1(x,y)f_i(x-u,y-v) + f_i^2(x-u,y-v)] \tag{3.11}$$

As $\sum_{x,y} f_1^2(x,y)$ does not depend on u, v and $\sum_{x,y} f_i^2(x+u, y+v)$ can be regarded as constant if the shift is circular, what remains is (3.12) which is the same convolution of an image with one that is reversed, using the relation $\mathcal{F}\{f(-x,-y)\}^* = F(u,v)$, $F(u,v)$ is the complex conjugate of the fourier transform of $f(-x,-y)$.

$$d_{f_1,f_i}(u,v) = \sum_{x,y}[f_1(x,y)f_i(x-u,y-v)] \tag{3.12}$$

$$d_{f_1,f_i}(u,v) = \mathcal{F}^{-1}(\mathcal{F}(f_1)\mathcal{F}^*(f_i)) \tag{3.13}$$

In the spatial domain the computation would require $M^2 * N^2$ calculations for two images of size M*N. By applying the fast Fourier transform when correlating the images in the Fourier domain the number of calculations $30M^2 \log_2(M)$ calculations. If m is 128 the FFT approach would cost 19000 calculations when the spatial approach costs over 200 millions.

Examining $d^2_{f_1,f_i}(u,v)$ it is apparent that it will have a maximum at the u and v where the images are the most similar. That is because when looking at equation (3.11), if the quadratic terms are considered constant the remaining term which is negative in the equation minimizes it when it's as large as possible.

Registration within subpixel domain Using cross correlation each position in the resulting matrix represents a shift by a whole pixel in a direction described by u and v. To move into the subpixel domain there are several

approaches possible. The first one is to increase the number of values that u and v can take by a factor of A thus obtaining a $D^2_{f_1,f_i}(u,v)$ where the position of the maximum corresponds to u/A and v/A pixels. This can be achieved either by zero-padding the image or increasing the size using a simple interpolation algorithm.

Another approach is to examining the neighboring values, if the true maximum is situated between two points the neighbors with the shortest distance to that maximum should have a value higher than the more distant neighbors. It is then possible to fit a second order polynomial to the points in the neighborhood and use the maximum as the subpixel shift.

The approach used in this project uses a combination of both approaches. The aim is that any faulty approximations from the cross correlation is corrected by the polynomial fitting. The images are increased to ten times the size in both dimensions before the cross correlation is preformed upon them. The result is a matrix containing a peak where the best fit is found. A neighborhood around it's maximum is then taken out for the next step where a column and a row passing the maximum is used as data points to fit two curves, one for the x-dimension and one for the y dimension. The maximums of those curves are calculated and the point is chosen to be the real point of best fit. The output will be in the range of [-1 1], that is the largest possible correction is 1/10 of a pixel.

3.2.2 Interpolation

Interpolation works like a reverse downsampling by piecing together several low resolution images to one of a higher resolution. It is done by using the warping between the frames to compute the exact location of all pixels, using that information it is possible to increase the scale and transfer the pixel values to a larger image. Once again, there were a variety of approaches to choose for the interpolation, this step consists of coordinate conversion and the actual interpolation. As the warping is assumed only to consist of sub pixel shifts in our model, the coordinate conversion is fairly straight forward. This is because there is no need do determine which regions on the frames to use during the interpolation.

A high resolution grid of desired size (M, N) is created as a matrix containing zeros hence after named HR- or target- grid $I^{HR}(x,y)$. This grid is later to be filled with the interpolated values. The high resolution grid uses the coordinates $x = 1 \dots M$ and $y = 1 \dots N$ measured in pixels from the upper left corner of the image. Using the number of elements in the input frame i $I_i^{LR}(x,y)$ of size (m_i, n_i) with the coordinates $x_i = 1 \dots m_i$ and $y_i = 1 \dots n_i$ as well as the HR grid to compute a scaling factor in x and y

directions respectively gives the following equation.

$$S_i = \begin{cases} M/m_i & \text{In } x \text{ dimension} \\ N/n_i & \text{In } y \text{ dimension} \end{cases} \tag{3.14}$$

As can be seen the scaling factor can vary between the frames. Coordinates are computed as distance in x- and y-dimension from the upper left corner at both scales. Taking the coordinates from the LR images as well as their corresponding subpixel shift (x_i^s, y_i^s) which is the same for each pixel in one frame and inserting them into an equation together with the scaling factors returns the coordinates of all the pixels in the HR scale resulting in a new set of coordinates for the low resolution frames $(\hat{x}_i, \hat{y}_i)$ at high resolution scale, (3.15) is applied on all $x_i = 1 \dots m_i$ and $y_i = 1 \dots n_i$.

$$\begin{aligned} \hat{x}_i &= S_i(x_i + x_i^s) \\ \hat{y}_i &= S_i(y_i + y_i^s) \end{aligned} \tag{3.15}$$

Now it's a matter of filling the HR grid with the pixels taken from the LR images. In the ideal case with perfectly aligned shifts and sufficient LR images to fill all zeros in the target grid with unique data the process is trivial. In most cases some of the frames containing information at certain shifts is missing and the LR pixels may be situated at any coordinate not restricted to be directly over an HR pixel but rather in between them. The process of choosing gives a number of options all leading to different results.

(A) Use the nearest neighbor pixels A simple method that chooses the LR pixel by computing the distance between HR and LR pixels, it then chooses the nearest and uses it for each site on the HR grid. The purpose of this method is to test the previous step so that the coordinate conversions are correct and it can't be used for more sophisticated extensions when densely placed values are severely penalized in comparison to sparsely.

(B) Inverse distance weighting Instead of choosing just one sample there may be beneficial to use a number of samples, especially as it prevents large areas with the same value if there is only one data point available at one location. This approach is based on the formulations in [7] and [8]. It also lets LR pixels situated at a distance slightly larger than the nearest to get some influence. The samples are chosen either by an area or as the N closest LR pixels. Both approaches have their advantages and problems. As the values is scattered choosing an area might return nothing at all as well as a tremendous amount of pixels for weighting. However it guarantees that all LR pixels used in the averaging are near to the HR site.

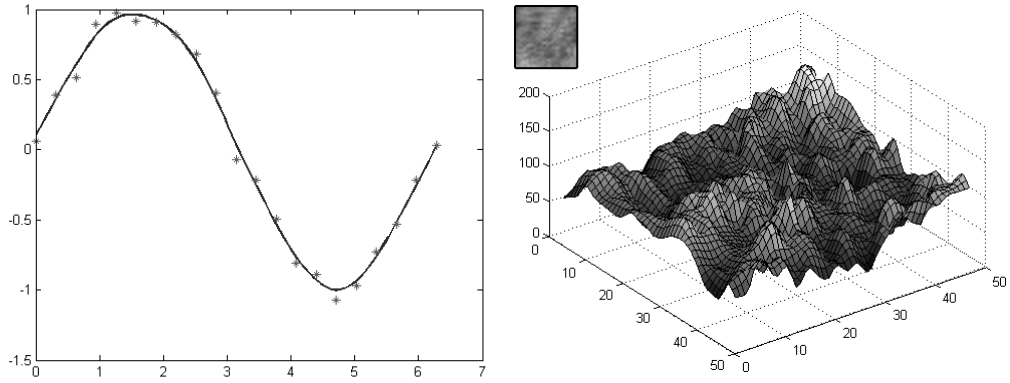
Neither choosing the N closest pixel values nor choosing pixel values depending on an area did seem flexible enough so an alternative approach was used. In this project the method of choice is area based with a condition that there must be at least one sample in the area, else the search area is increased and a new search for pixels is carried out until they are found. The weighting is carried out using the equations (3.16) and (3.17) and the pixels found in the previous steps. Pixels: n $= 1 \dots N$. The weight is computed by inverting the Euclidean distance between the value at its coordinates and the site on the target grid to a power of two. The exponent 2 is chosen when recommended by Shepard in [8] with the motivation that the results are empirically satisfying as well as computationally very simple. The distance in x and y dimensions are computed using the coordinates from 3.15, $\Delta X_n = \|x_n - x'_n\|$ and $\Delta Y_n = \|y_n - y'_n\|$.

$$HR(x,y) = \frac{\sum_{i=1}^{N} w_n * LR_n}{\sum_{i=1}^{N} w_n} \tag{3.16}$$

$$w_n = \frac{1}{(\Delta X_n + \Delta Y_n)^2} \tag{3.17}$$

(C) Spline interpolation An alternative method implemented was the spline interpolation. A spline is a piecewise polynomial that given a set of values and data points, is constructed to while being a continuous function minimize the sum of distances to the chosen data points in a similar way like linear regression is carried out. Examples of splines can be seen in figure 3.5 where in (a) shows the spline generated from data points on a sinus curve with added noise and (b) shows a spline surface based on a small portion of the images used in this project.

Moving into two dimensions the spline interpolation creates a surface. Even though the shift is regarded as global the control points is to be regarded as shattered as it is impossible to fit a grid so that it exactly covers all the sample pixels without having any empty cells. It is apparent when only looking at an area of $L \times L$ anywhere in the image, the scattering is due to even in the case of ordered subpixel shifts between the frames the registration errors are always present, therefore it is safe to assume that the data points always are scattered. In MATLAB the recommended method for scattered points is the thin plate smoothing spline as referred to in [14]. Its name refers to the analogy of bending a thin sheet of plate so that at every data site $(\hat{x}, \hat{y})$ it passes $z(\hat{x}, \hat{y})$ as close as possible without having to make dents in the plate. Mathematically the spline is a minimizer of the energy function combined with the smoothing function that is the space integral of the squared second



(a) (b)

Figure 3.5: (a) An example of spline interpolation, here a sinusoidal curve with a small randomizer is used for data points. (b) Into two dimension it generates a surface, an image rendered from same data embedded in the upper left corner.

derivatives. In the energy function, y_i is the data value at location x_i and $f(x_i)$ is the polynomial that defines the surface.

$$E_{tps} = \arg\min_{f} \sum_{i=1}^{K} \|y_i - f(x_i)\| + \lambda \iint \left[\frac{d^2 f}{dx^2} + \frac{2d^2 f}{dxy} + \frac{d^2 f}{dy^2} \right]^2 dxdy. \quad (3.18)$$

λ is the smoothing parameter, it varies between 0 and 1 and the interpolation behavior varies from linear interpolation at $\lambda = 0$ to thin plate interpolation at $\lambda = 1$. The function "tpaps.m" in MATLAB carries out the mathematics in (3.18). When supplied with a number of data sites in 2 dimensions and their corresponding values it returns a 2 dimensional function. As it is computationally complex in the case that there is a linear equation systems with as many equations and unknowns that there is data sites so the method is limited to around 1000 samples. Even the small test images used over 20000 samples and the problem had to be solved by dividing the area into squares that would contain around 576 (24^2) data points, depending of how densely the samples are placed the size of the areas differ. An overlap was forced upon the square pattern to avoid lines in the output image where the squares meet as well as to avoid having an area at the edge of the image not differentiable by the chosen areas.

The function is evaluated at all places of the HR grid and thus yielding the interpolated image. A special matrix keeps track of where two or more

squares are intersecting and tells the algorithm to average the values at those locations.

3.2.3 Blind deconvolution

This chapter treats the process of estimating the blur and simultaneously eliminating it from the image. As the process of eliminating blur in the presence of noise contains more unknowns than it has equations. By treating it as a minimization problem and solve it in an iterative process as well as imposing constrains to the equation makes the solution desirable. This part of the algorithm uses a program named Multiframe Blind Deconvolution GUI by Šroubek and Flusser and the theory presented here is to be regarded as a summary of their articles [4] and [5].

The key is to minimize the error between the received low resolution frames and the theoretical high resolution image with the mathematical degradations affecting it. By alternating fixation of either the blur or the HR image while minimizing the SR image should converge to the HR image as the proposed blur converges to the real. To make equation (3.19) more specific, a number of regularization terms are added.

$$E(x, B) = \sum_{K} \|DB_k x - y_k\|^2 + \alpha Q(x) + \beta_1 R(b) + \beta_2 Q(b) \tag{3.19}$$

Regularization terms Image regularization term penalizes high frequency information as noise is generally of a high frequency. This will end up in a trade off between preserving edges and remove noise. The term itself consists of $Q(x) = x^T Lx$ where L is a positive semidefinite block trigonal matrix where the values are taken from the gradient of image x. Blur regularization term R(b) binds the PSF approximations and prevents them from moving freely. The term consist of $R(b) = \|\mathcal{N}b\|^2 = b^T\mathcal{N}^T\mathcal{N}h$. $\mathcal{N}$ is the nullity of the blur constructed from the low resolution frames. It is based on the assumption that any two correct blurs b_i, b_j satisfy $\|y_i * b_j - y_j * b_i\| = 0$ [5].

Alternating minimization Equation (3.19) has got the nice property of having only quadratic terms showing convex characteristics (not necessarily strictly convex) and thus being differentiable over both f and B in a way that a minimizer could be found. Starting with an initial b^0 in (3.20) to find a minimizer for x. Working similar to the steepest descent method, the newly computed x^m is then used with (3.21) to find b^{m+1} and the alternation continues for a user specified amount of times.

$$x^m = \arg\min_x F(x, b^m)$$
$$= (\sum_{k=1}^{K} B_k^T D^T D B_k + \alpha L) f = \sum_{k=1}^{K} B_k^T D^T y_k \quad (3.20)$$

$$b^{m+1} = \arg\min_b F(x^m, b)$$
$$= ([I_K \otimes X^T D^T D X] + \beta_1 \mathcal{N}^T \mathcal{N} + \beta_2 L) h = [I_k \otimes F^T D^T y_k] \quad (3.21)$$

With the relations $X = C_b^v(x)$, $B = C_x^v(b)$ where v denotes valid convolution. Examining the equations it is apparent that they are of the type $Ax = B$ where the minimizer to x is obtained by the least squares approximation. As many possible x could minimize this problem the regularization terms guides the minimization towards one that is desired. In MATLAB the equations are solved by computing all parts individually, the right part(B), each y_k is convolved with B^T and D^T for and summation over k takes place for equation (3.20) and for (3.21) g is convolved with F^T and D^T and the results are put in a diagonal matrix. The left side (A) is calculated in a similar fashion but the regularization terms are added after the summation. With A and B defined the Multiframe blind Deconvolution uses a function minimizer to find x and b. In the case of (3.20) the built in function minimizer "pcg.m" (Preconditioned Conjugate Gradients Method) is used, for (3.21) which needs certain constrains namely the upper and lower bounds of the PSF for preservation of image brightness, the bounds are set to $b\langle 0, 1\rangle^{B^2}$. Due to the smaller size of b compared to x it is possible to use the more computationally demanding constrained function minimizer "fmincon.m".

3.2.4 Adjustable parameters when using MBD software

The software does not work without the user submitting the correct values for the parameters. The weights α and β_1 depends on the signal to noise ratios if the inputs. According to the instructions for the mbdgui α should be 1×10^3 for a SNR of 30dB 1×10^2 for 20dB and so on. β_1 varies with α but 10 or 100 times less depending on if the SR application is active or not. As the demo with SR disabled is used β_2 may be fixed at 0.01α. If the noise is underestimated the algorithm starts amplifying it instead and if its overestimated the algorithm will start repressing finer textural details. A wrongly estimated PSF size causes alias-like patterns in the image. In figure

3.6, the interface of the MBD software is shown. The parameters are listed below for easier overview.

1. α depends on signal to noise ratio of the image and is recommended to increase by a factor of 10 for every 10 dB.

2. β_1 also depends on signal to noise ratio and is in this implementation $\alpha \times 0.01$.

3. β_2 is not used in this thesis and remains at 0 according to the recommendations.

4. PSF size is a matrix that the point spread function should fit into. There is no upper border but the computional complexity makes the software to run slower.

Figure 3.6: The MBD software interface, from the program described by Šroubek and Flusser in [5], the notations are a bit different in the software, α = u_lambda, β_1 = h_mu and β_2= h_lambda

Chapter 4

Experiments, Results and Evaluation

In this chapter, the various experiments are presented, that motivation there are to perform them, how they were performed, with what parameter setup and what methods tested. The resulting images are shown in the figures and where it's possible, tabular values over compared values. The results are discussed for each experiment, how the result depends on the parameter settings and some possible reasons for the outcome.

4.1 Test images used for fusion

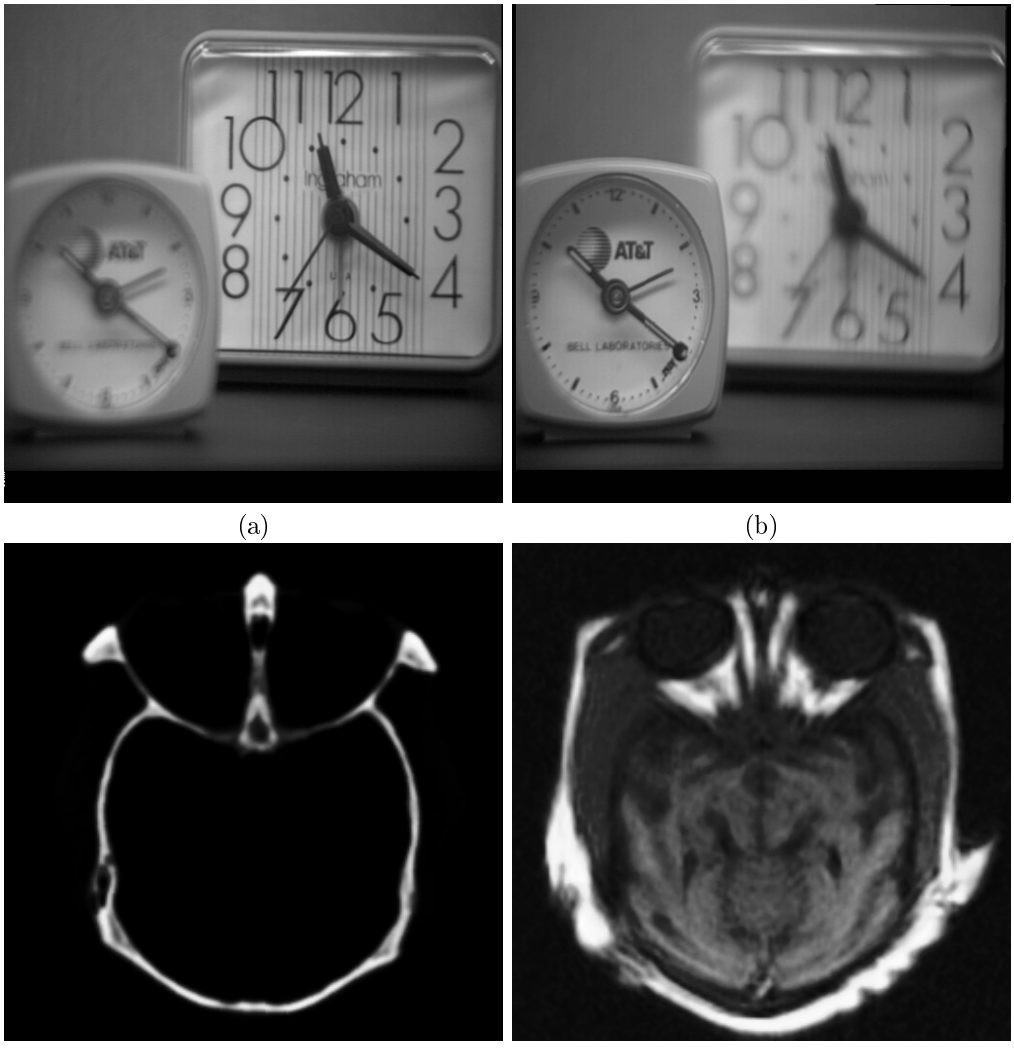
The first set of images primary used to test the algorithm is two clocks. In each image one clock is out of focus and the other in focus, the same is applied to the desk which they stand upon and the wall behind them. It is a typical case where only one of the images is of interest at a specific location. The images are of a horizontal slice from a head obtained with different medical imaging techniques, one CT image showing bone structure and one MR image showing the different tissues. In this case there may be an interest to have both images present in a specific location. Furthermore, there are no great similarities between the images like the case with the clocks; it has to be assumed that they are aligned. The test images for the fusion algorithm is shown in figure 4.1, upper row consist of the images of different focus and lower row consist of images with different modality.

4.2 Test images used for super resolution

Images used in testing were with few exceptions artificially made as described in section 2.2.1. The first test image, picturing the face of a canadian lynx chosen for the very detailed texture of the fur is shown in Figure 4.2(a). The 16 low resolution synthetic images, down sampled by 4 both along the x and the y directions, are created from the HR image in Figure 4.2(b) using the following equation also seen in section 2.2.1. Their position corresponds to the subpixel shift, starting from the upper left with no shift each frame shifts by 0.25 pixels in either x or y direction. The image is filtered by a gaussian blur of size 5 and a standard deviation of 0.5, then the downsampling occur, lastly, gaussian noise with a signal to noise ratio of 30dB is added.

$$y_k = D(B_k * x(M_k)) + n_k \tag{4.1}$$

To verify that the algorithm produces similar results even when restoring other images than the "lynx", another image was treated in the same way. This image depicting a man contains areas of fairly constant brightness and instead of a detailed texture it contains many straight and curved lines and changes between bright and dark, the image is referred to as "Ueshiba". It is shown in figure 4.3. The third test image is obtained as a dataset of 16 low resolution frames and is thus not created in the same manner as the previous images, it depicts a television test card and is displayed in figure 4.4. Like the other images even this one is artificially made with corresponding shifts.



(a) (b)

(c) (d)

Figure 4.1: The original images used to test the fusion algorithm. (a) Left clock is in focus (b) Right clock is in focus. (c) CT image of brain slice. (d) The corresponding MR image.

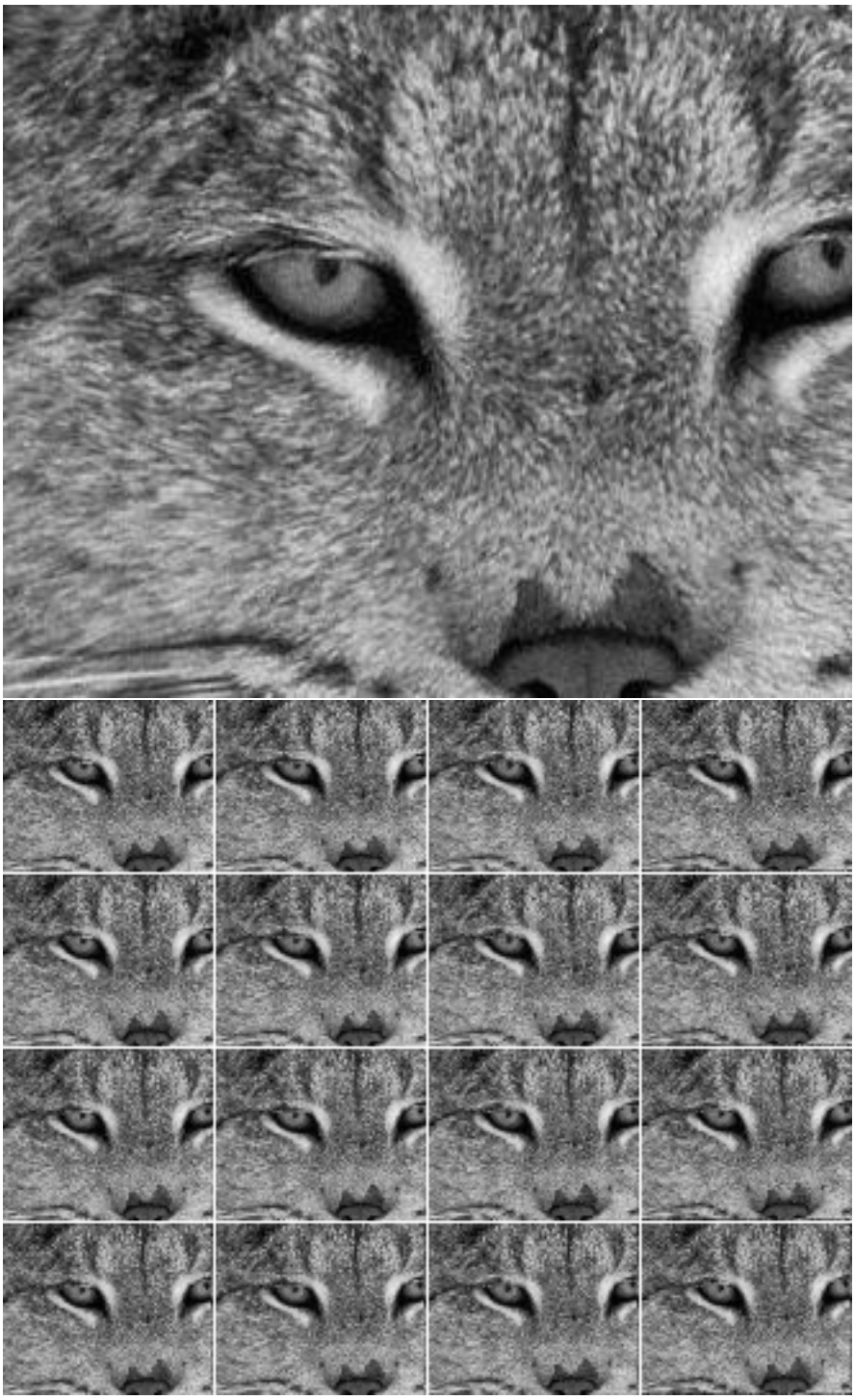


Figure 4.2: (a) The original image of size 267×327 pixels, (b) 16 low resolution images downsampled by a factor of 4 in each direction, convolved with a gaussian PSF of size 5×5 and with additive noise with a SNR of 30dB.

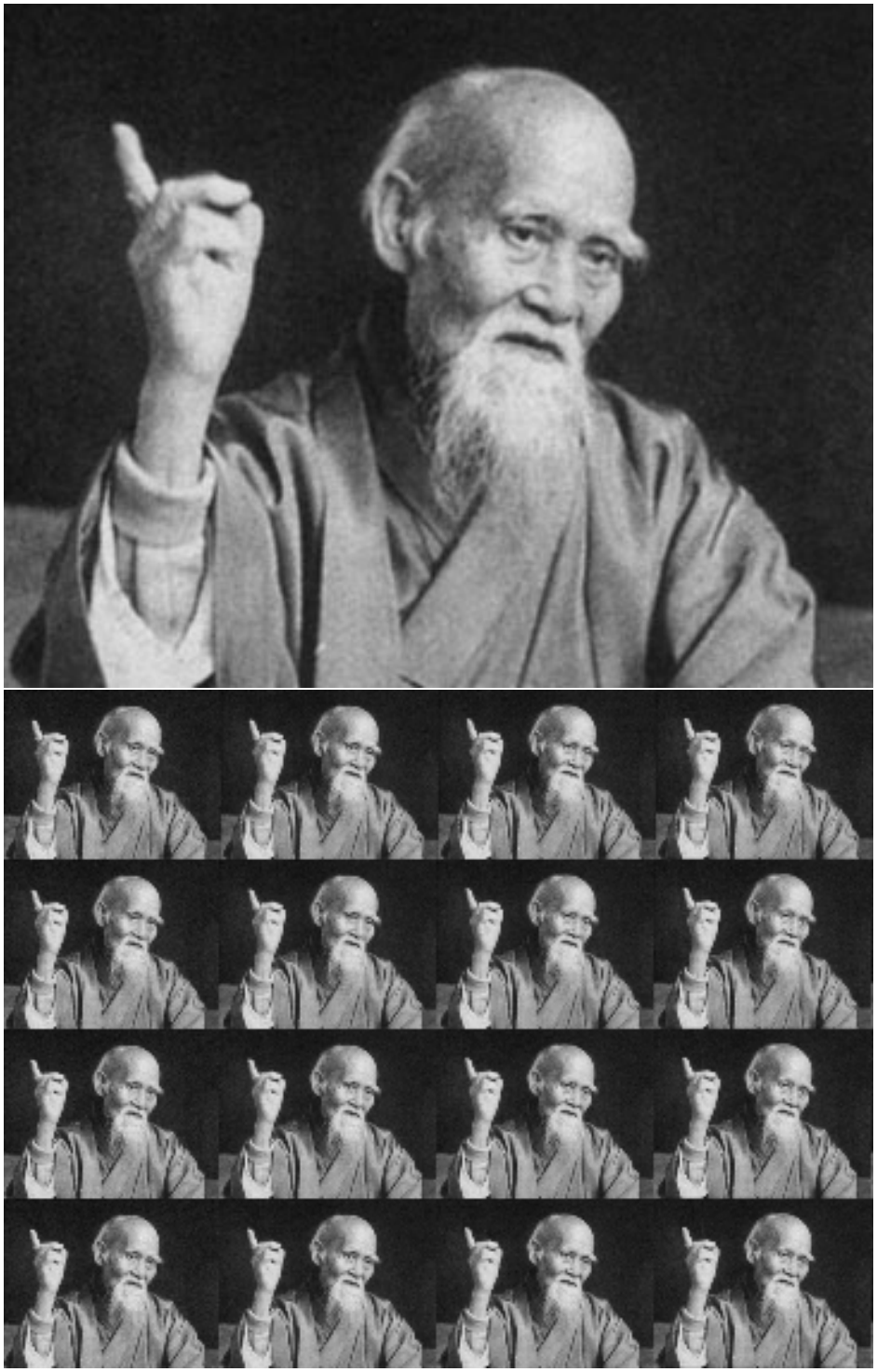


Figure 4.3: (a) The original image of size 284×361 pixels, (b) 16 low resolution images downsampled by a factor of 4 in each direction, convolved with a gaussian PSF of size 5×5 and with additive noise with a SNR of 30dB.

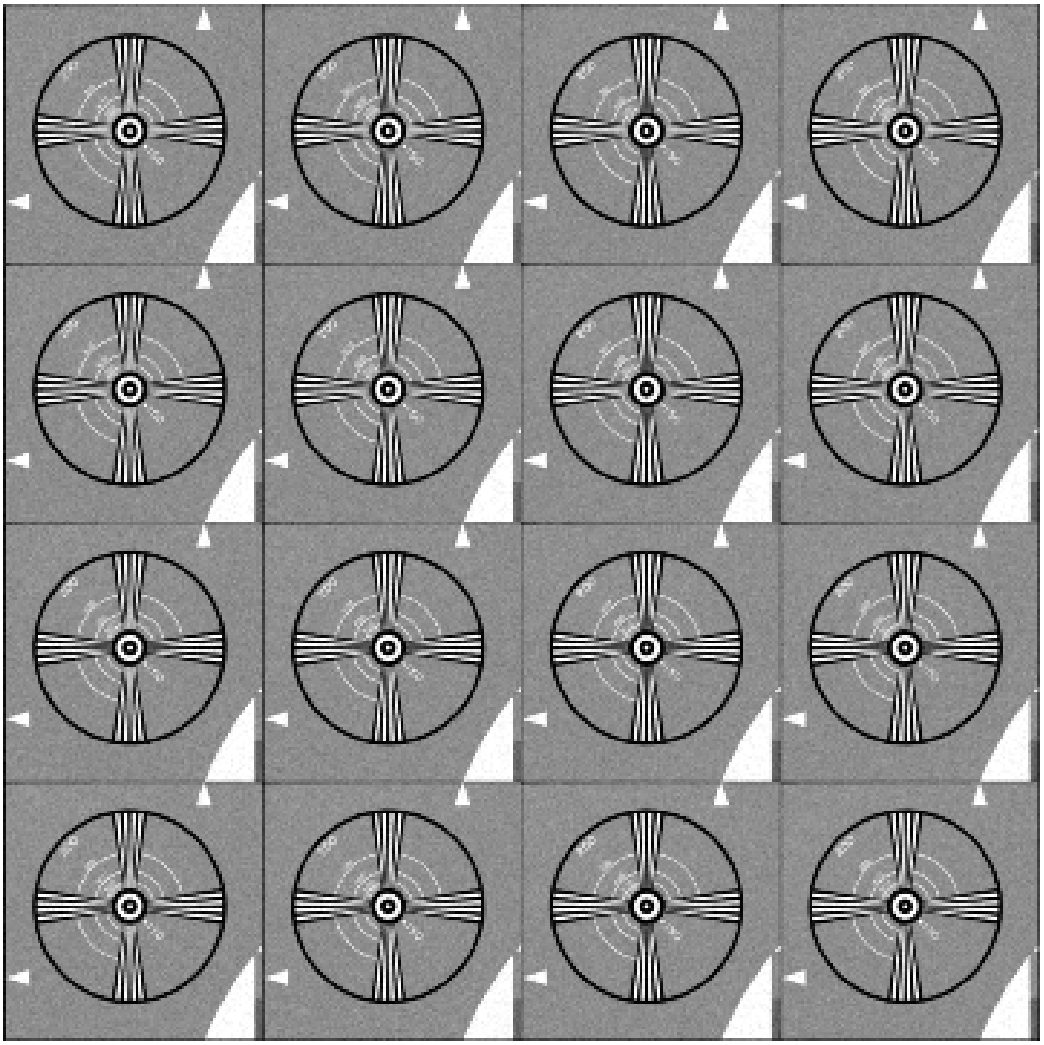


Figure 4.4: 16 low resolution images of size 90×90 pixels.

4.3 Objective criterion used for evaluation

When answering the question of how good the results are, there are many factors to consider. First of all, what is a good result? It could be assessed subjectively just by looking at the fused images and by that determine the outcome, the images the spectator feels most appealing is the better ones. For many cases this is the only possible evaluation method. In the case of image super resolution with simulated images there is a possibility to make an objective measurement by computing the peak signal-to-noise ratio to determine how similar the SR image is to the original HR image. It is done by computing the root mean squared error between the images and comparing that to the signal strength which is chosen to be the highest value from the compared images, for an 8 bit image it is 255.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|HR(i,j) - SR(i,j)\|^2 \tag{4.2}$$

$$PSNR = 10 \log_{10} \left(\frac{max_I^2}{MSE} \right) \tag{4.3}$$

4.4 Experiments

In this section, the various experiments are shown, the order of experiments are chosen to be as similar to the order where the concepts are encountered in the report.

4.4.1 Fusion

Beginning with fusion of two images of different focus, in this first experiment, the region based approach with choose maximum criterion is used, in figure 4.5, all the intermediates can be shown. The upper row consists of the segmentations of each of the input images, the lower image to the left shows the composite segments and the lower right image shows the decision map. The result of the fusion is shown in figure 4.6.

Same images are fused in a pixel wise manner, here; results for both cases of maximum selection and weighted average are presented. In figure 4.7 the results are shown. (a) and (b) are the decision maps for maximum selection and weighted averaging, (c) and (d) are their corresponding results. Figure 4.8 contains the brain image fused pixel wise. The order of images is the same as the figure before. It becomes apparent that a certain amount of randomness is present in the images fused in pixel level. It is not certain that the

pixels neighboring each other are chosen because the relative high frequency between image A and B content may vary greatly over small distances. There may actually be favorable in some cases. There are no results for applying the region based fusion on the medical images because the algorithm couldn't segment the images in a satisfying way.

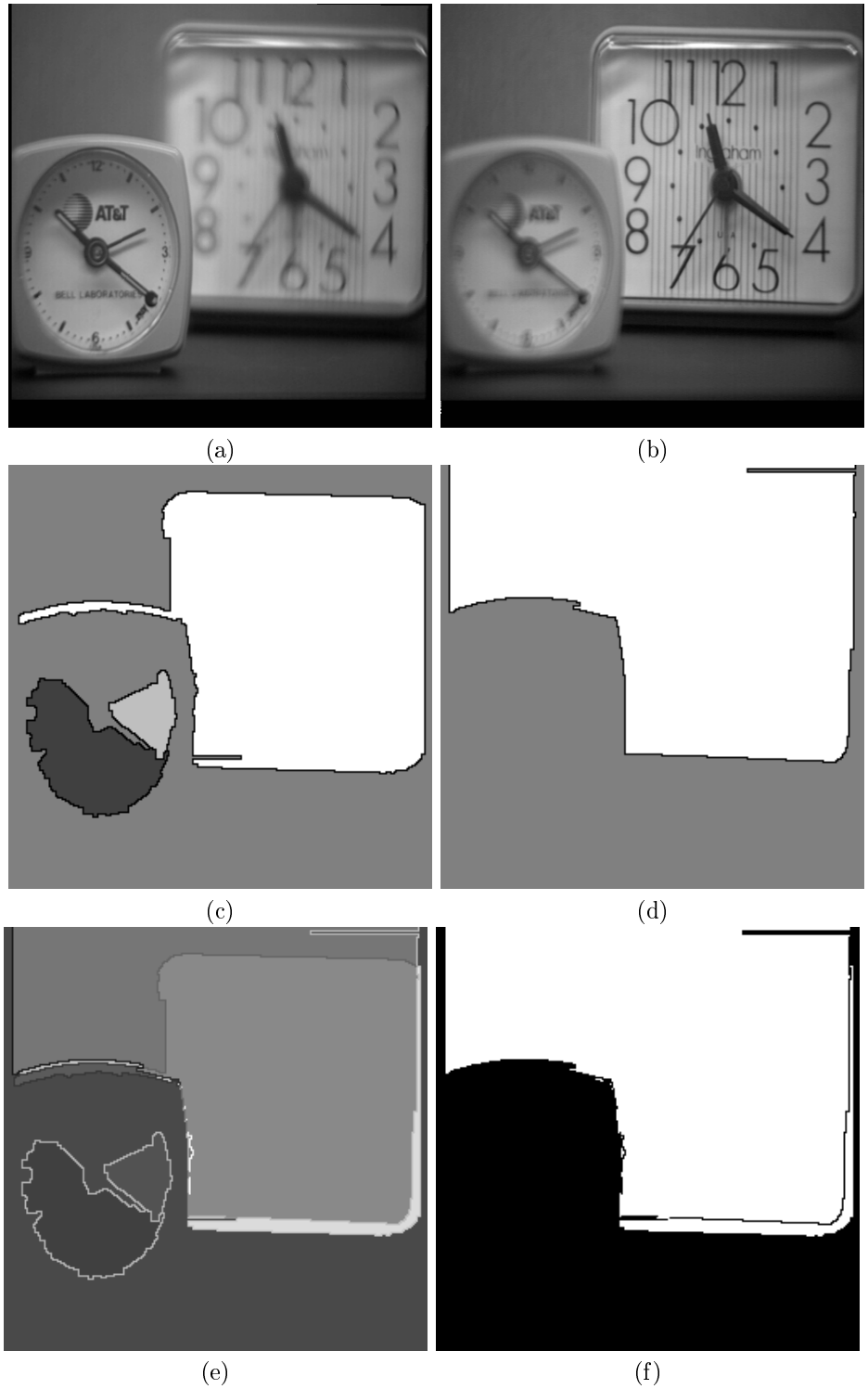
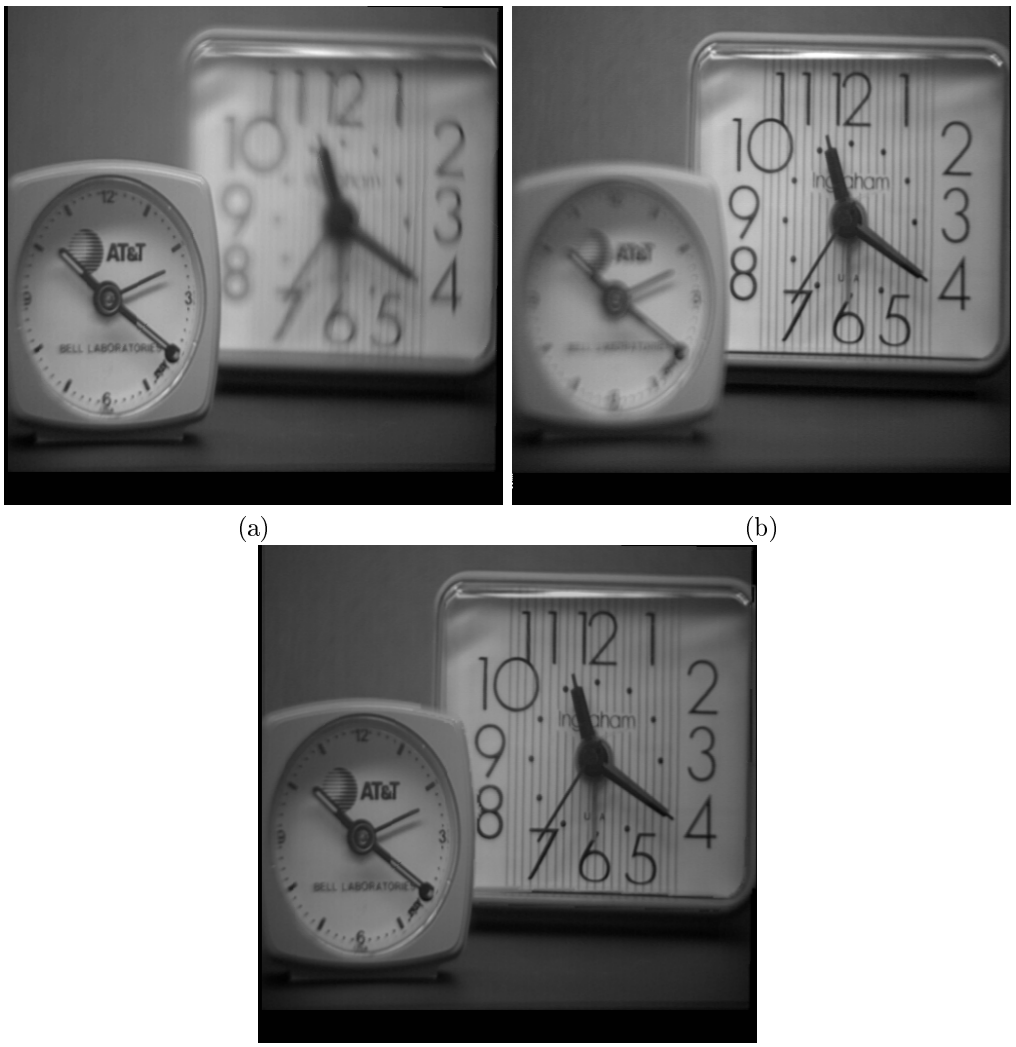


Figure 4.5: Creation of region based decision map as described in section 2.1.2 (a-b) original images, (a)left clock in focus, (b) right clock in focus. (c-d) segmentations of (a) and (b). (e) merged segmentations. (f) decision map.



(a) (b)

(c)

Figure 4.6: The final result for region based image fusion.

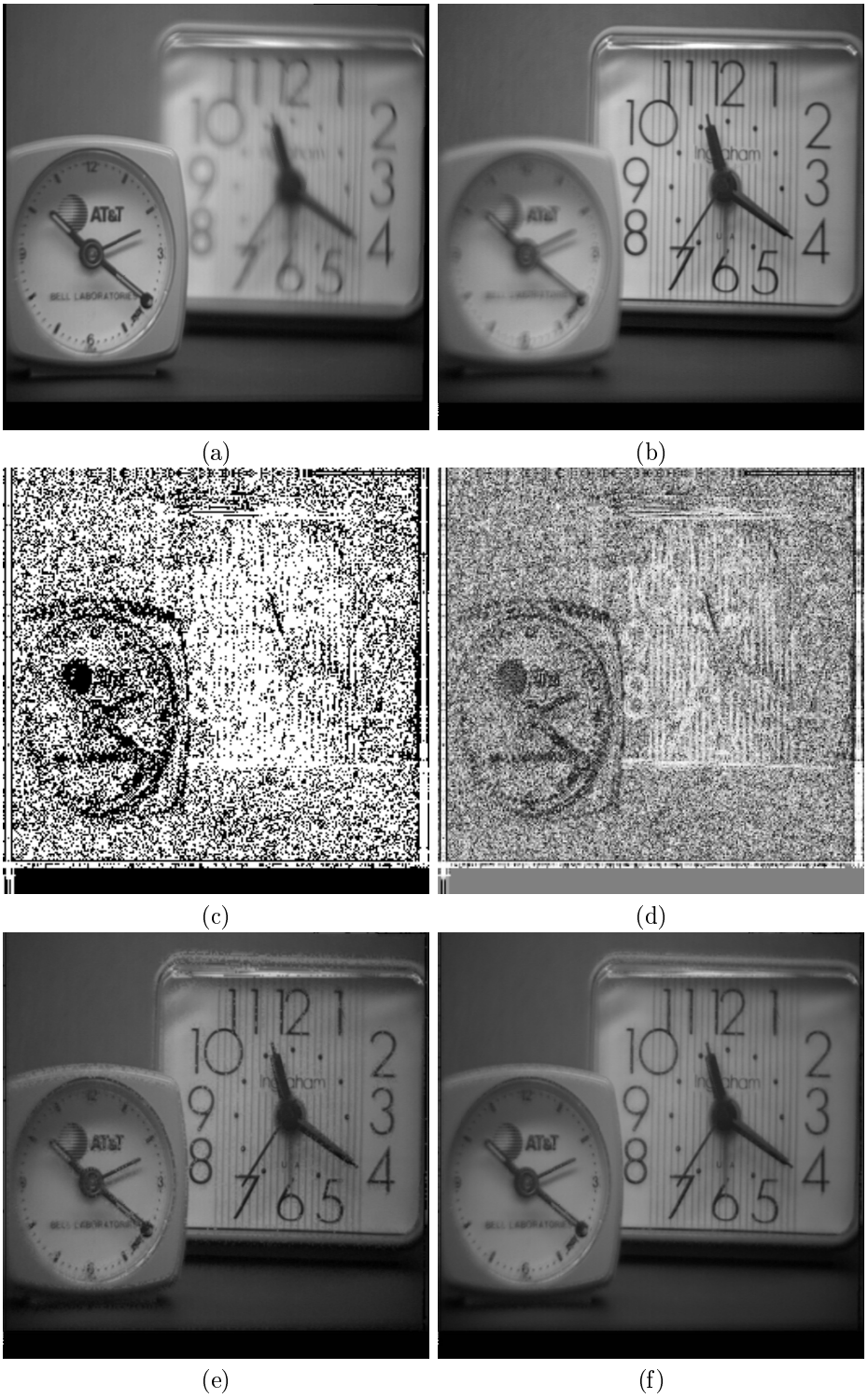


Figure 4.7: For pixel based fusion of the multi focused images, as described in section 2.1.2, (a-b) the original images, the decision maps are shown as a binary image (c) for choose maximum and a grayscale image (d) for weighted average. Image (e) shows the fused image based on the binary decision map (c) and and image (f) shows the fused image based on grayscale decision map (d).

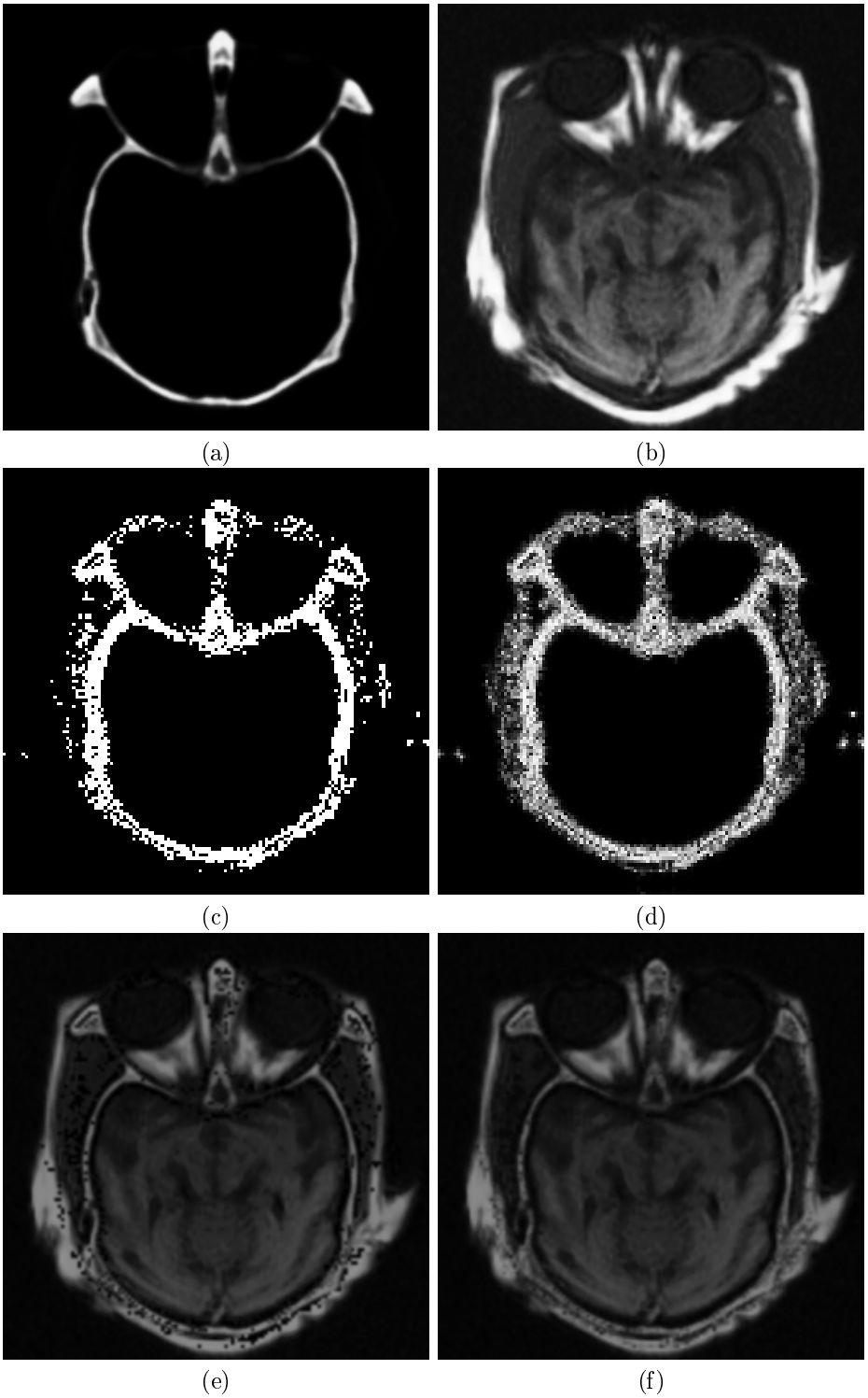


Figure 4.8: For pixel based fusion of the multi focused images, as described in section 2.1.2, (a-b) the original images, the decision maps are shown as a binary image (c) for choose maximum and a grayscale image (d) for weighted average. Image (e) shows the fused image based on the binary decision map (c) and and image (f) shows the fused image based on grayscale decision map (d).

4.4.2 Evaluation of the proposed SR algorithms

There are many different tests to be performed when evaluating the SR algorithm. All three images have to be tested, then there are many different setups for the algorithm to test. There are many aims when testing the methods, what is the optimal setup if there is any and how does it vary depending on the input, what are the limitations of this approach for super resolution and how flexible and robust is it. The registration will be tested to see how near the true shifts its estimates are. When testing the interpolation it is of interest to see how it performs depending on which shifts the input images has, spline interpolation is regarded as the default method and will be used if nothing other is mentioned. When incorporating the Multiframe Blind Deconvolution the test has got two purposes, first to see how the whole algorithm work and second to see which parameter settings gives the best results. By assuming that the registration already has been performed with perfect result it is possible to see the effect of registration errors on the final results. As both spline and inverse distance interpolation is available the parameters were set to use the inverse distance interpolation. The test done are listed below:

1. Evaluation of the whole algorithm with different parameters to the MBD
2. Evaluation of the subpixel registration
3. Evaluation of the interpolation for different shift combinations
4. Evaluation of the algorithm using prior knowledge of the subpixel shifts between frames
5. Evaluating the inverse distance weighting interpolation
6. Evaluation of super resolution algorithm with missing data

Evaluation of the whole algorithm with different parameters to the MBD In this experiment the whole chain of methods as shown in figure reffig:process-diagram is involved for making the SR images, for interpolation, method (C) in section 3.2.2 spline interpolation was chosen. Mainly two variables were tested: PSF size, α, and β_1 that is coupled with alpha. The recommended input values for this particular PSF and noise were used as base for the tests but it was of interest to see if deviating from those numbers would give better output. The same values were used in most tests involving the multiframe blind deconvolution step for comparisons. As the noise of

the simulated images had a noise SNR of 30, alpha is set to 110^3 and the PSF size is set to 5. As seen in figure 4.9 using the recommended value for a certain noise intensity and the PSF size used when creating the degraded images gave a much lower Peak Signal to Noise Ratio and it's also apparent by looking at the images in figure 4.9. One can assume that what's work for the Multiframe Blind Deconvolution application as a standalone program might not work when the circumstances are changed like using it as a part of a stepwise SR algorithm like this. The results for the "Ueshiba" image seen in figure 4.10 follows the same pattern, likewise the EIA image seen in figure 4.11. It can be debated why the larger estimations of PSF and noise gave better results. Examining how the images have been treated during the process, there should be two possible explanations. The degradation might be faulty and thus result in images degraded with different parameters than the chosen. The more plausible explanation is that while interpolating the images to desired resolution using splines an additional smoothing is being made. By increasing the estimated PSF size further could give a hint on the magnitude of this effect. The increments were made in steps by 2 until the estimated PSF was set to 15. For easier comparison the best SR results for the three images are displayed together with the original images in figure 4.13, 4.14 and 4.15. A closer examination of the images from figure 4.15 is available in figure 4.16.

Evaluation of the registration It is of interest to see how close to the true shifts the registration estimates are as well as to see how the other parts of the program works. The results from the registration of all images tested in this project are present as this is such an important test, it is also of interest to see if the registration error is proportional with errors in the images after the whole process.

As the registration algorithm compares the relative shift between the first supplied image with itself (as a self test) and all the others, what's available is a N*2 vector where N is the number of images. Additionally, the registration algorithm return the coordinates from the cross correlation step and the correction from the polynomial fitting step as separate variables so that it is possible to evaluate if the second step is helpful.

The output tested in this section will be the output from the registration with or without the correction by polynomial fitting and the purpose is to see how severe the errors are and if the polynomial fitting produces any improvement.

As the images are simulated it is possible to store the subpixel shifts for each low resolution frame frame. It allows comparison of the computed

Image and registration	Error mean	Error std
Lynx (figure 4.2) cross correlation	0.0625	0.061
Lynx (figure 4.2) polynomial fitting	0.0648	0.0464
Ueshiba (figure 4.3)cross correlation	0.125	0.146
Ueshiba (figure 4.3)polynomial fitting	0.104	0.1098
Eia (figure 4.4)cross correlation	0.0750	0.0762
Eia (figure 4.4)polynomial fitting	0.0706	0.0582

Table 4.1: Registration errors.

shifts with the true ones. The errors are computed for each image and a mean error and an error standard deviation is calculated. The number off errors of different sizes are tabulated in figure 4.17 and the mean error and error standard deviations could be found in table 4.4.2. For reference, an error of 0.25 is equivalent of a false classification of an image as if it would be its neighbor. The polynomial fitting gives different effects on all the images. Judging from the mean error the "lynx" image gets a worse registration than without the polynomial addition. The "Ueshiba" image which has the biggest error of the images tested gets the biggest improvement and the EIA image gets a little improvement. The remarkable effect is the one of the error standard deviation. All errors are moved towards the mean error, where the cross correlation is putting out a value close to the true value the polynomial fitting makes a little bit worse, but most significally where the registration is as it's worst the polynomial fitting works best. For a better overview the errors are sorted by size and plotted in figure 4.17 where the blue bars represents the errors from cross correlation only and the red bars are the refined values by polynomial fitting. As can be seen, in all cases the number of greatly miss registered values has decreased but the total number of errors has increased.

Evaluation of the interpolation for different shift combinations as described in section 3.2 In a real situation it is not possible to have 16 images each with an increasing shift of 0.25 so that all phases are covered. A way to simulate that there are differences between the data is to pick images from the dataset in different patterns and see how the algorithm works. This test is both to see how sensitive the algorithm is to the shifts of the input images and a test to see if it is possible to exploit those characteristics, the best performing combination in this test is the one that will be used in the following experiments. This test will only incorporate the "Lynx" image. Under the assumption of a high number of images available, there is possible

to choose images depending on their subpixel shift at ones own pleasure it might be possible to optimize the selection. The resulting images and peak signal to noise ratios for the "lynx" image shown in figure 4.18 for various combination of images using registration and interpolation but excluding blind deconvolution. The small matrix seen in the top left corner of each image tells which frames used for making that particular image using the same relative order as in Figure 4.2. The reason for doing this experiment is to examine how much which images are available affects the outcome of the SR algorithm. From the images it is apparent that the visual result differ even though there are only small differences in PSNR between the results. Image 4.18 (a) "Evenly spaced" shows a slightly better value and is also perceived as somewhat better than the rest so that setup is used in the following experiments. The effects of blurring are clearly visible in all images as the blind deconvolution has not been applied. For a more close examination the original image and the image from figure 4.18 (a) is zoomed in and displayed in figure 4.19.

Evaluation of the algorithm using prior knowledge of the subpixel shifts between frames It could also be of interest to see how the interpolation and MBD performs with perfectly registered images. In this case perfectly means using the same shifts that the simulated images were created with. The purpose of this experiment was to test how the algorithm would work without registration errors. The results from the experiment are shown in figure 4.20 for the "lynx" image, figure 4.21 for the "Ueshiba" image and figure 4.22 for the EIA image. Judging from the PSNR values there was no big improvement which tells that the registration errors aren't any devastationing flaw. However, by looking at the "Ueshiba" and EIA images it is apparent that the visual appearance is somewhat damaged by the registration errors as a periodic artifact is induced. Even in the "lynx" image it is possible to spot the artifact but it is generally more visible in images with prominent edges.

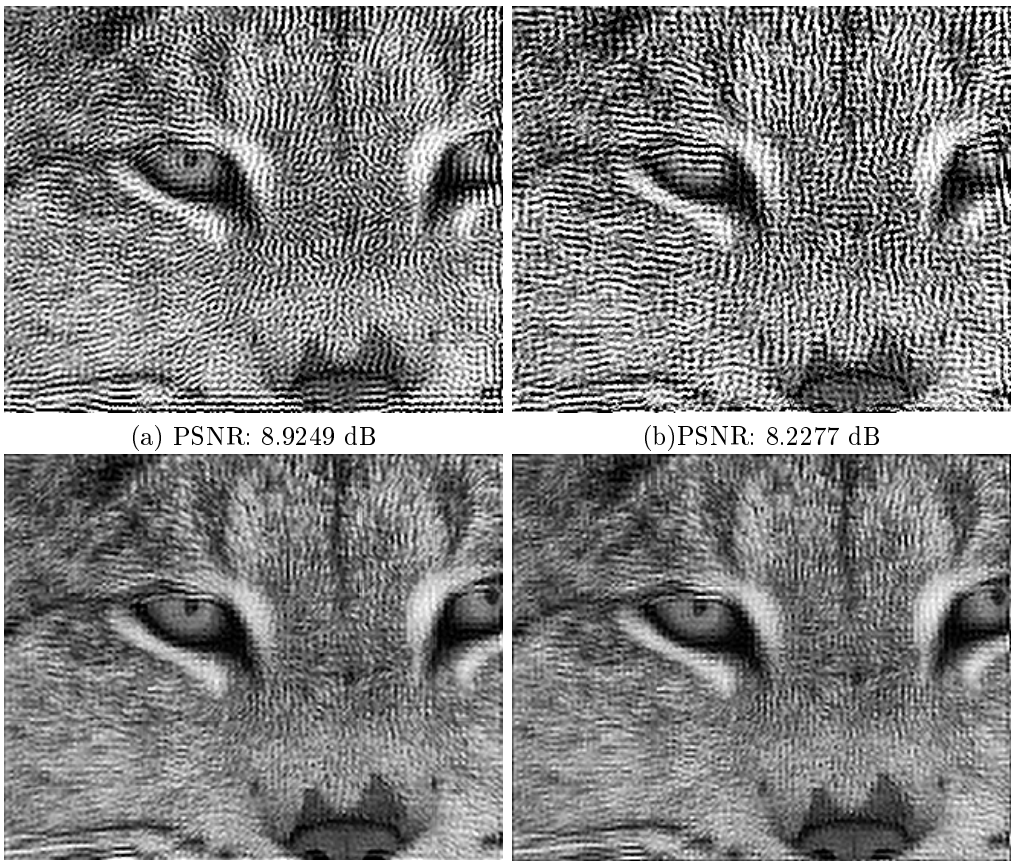
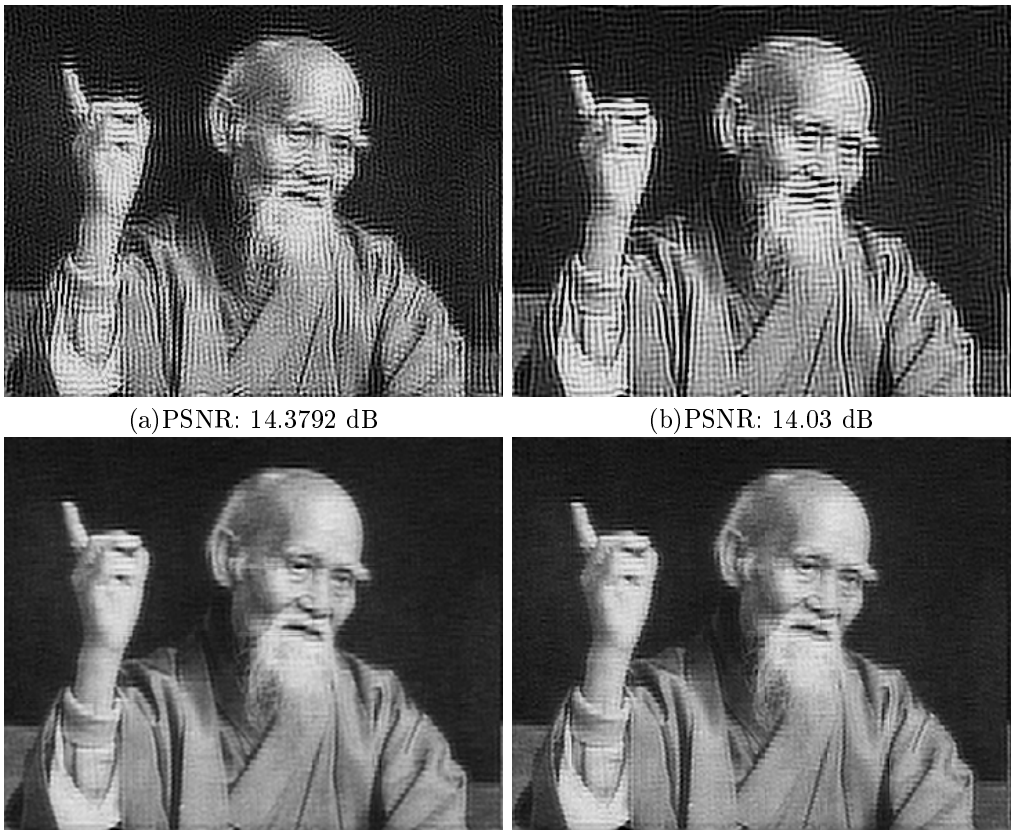


Figure 4.9: Results for experiments using the modules registration, spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution on the "lynx" image from figure 4.2, 16 low resolution images were used in the experiment. The parameter for MBD, described in section 3.2.4 were varied for the images (a-d). (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5. (e) illustrates the what modules used in this experiment.



(a)PSNR: 14.3792 dB (b)PSNR: 14.03 dB

(c)PSNR: 19.6963 dB (d)PSNR: 19.7394 dB

Figure 4.10: Results for experiments using the modules registration, spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution on the "Ueshiba" image from figure 4.3, 16 low resolution images were used in the experiment. The parameter for MBD, described in section 3.2.4 were varied for the images (a-d). (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5

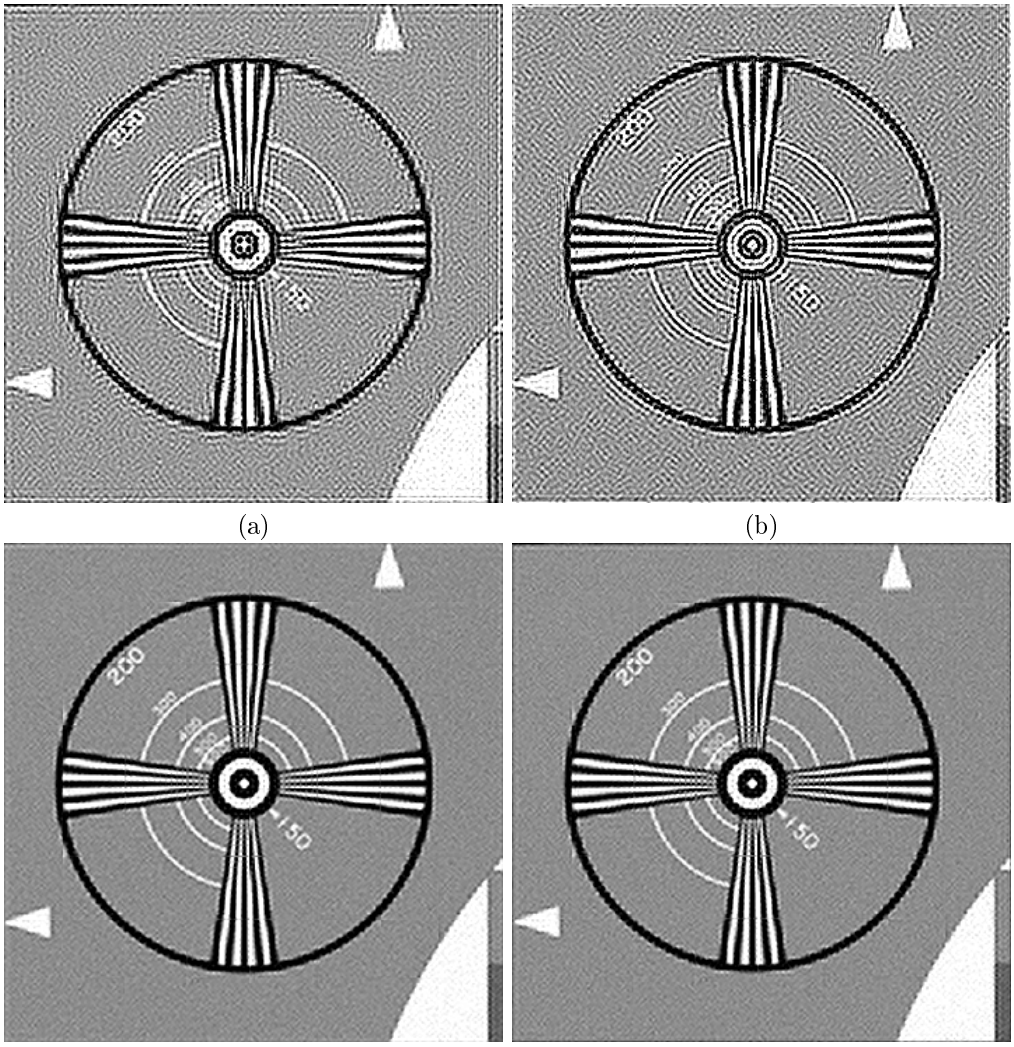


Figure 4.11: Results for experiments using the modules registration, spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution on the "lynx" image from figure 4.4, 16 low resolution images were used in the experiment. The parameter for MBD, described in section 3.2.4 were varied for the images (a-d). (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5

(a)PSNR: 15.8402 dB (b)PSNR: 15.5577 dB

(d)PSNR: 16.7629 dB (d)PSNR: 15.5244 dB

Figure 4.12: Further increment of the estimated PSF on the "lynx" image from figure 4.2, 16 low resolution is used in the experiment, α is 100 and PSF size is varied over images (a-d), sizes are 9, 11, 13, and 15 pixels respectively

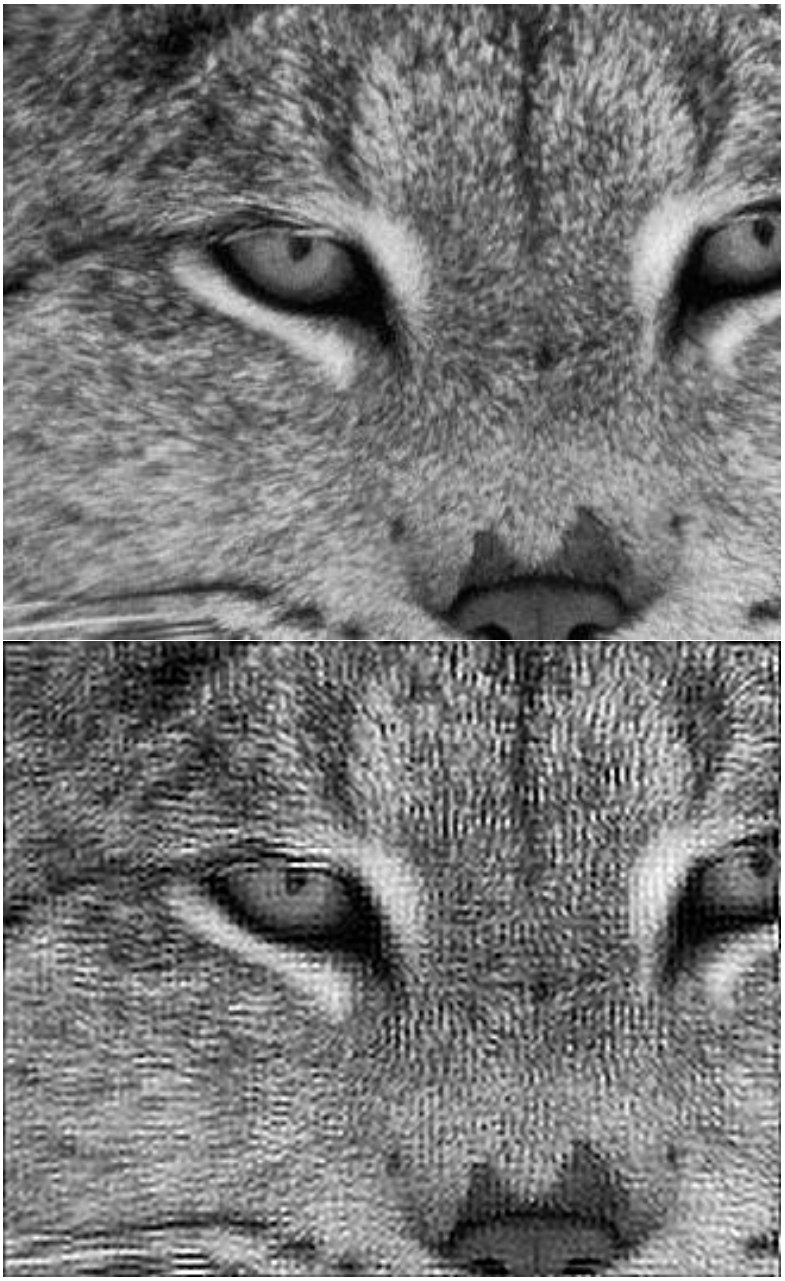


Figure 4.13: Comparing the (a) original image and (b) the super resolution image from figure 4.9(d)

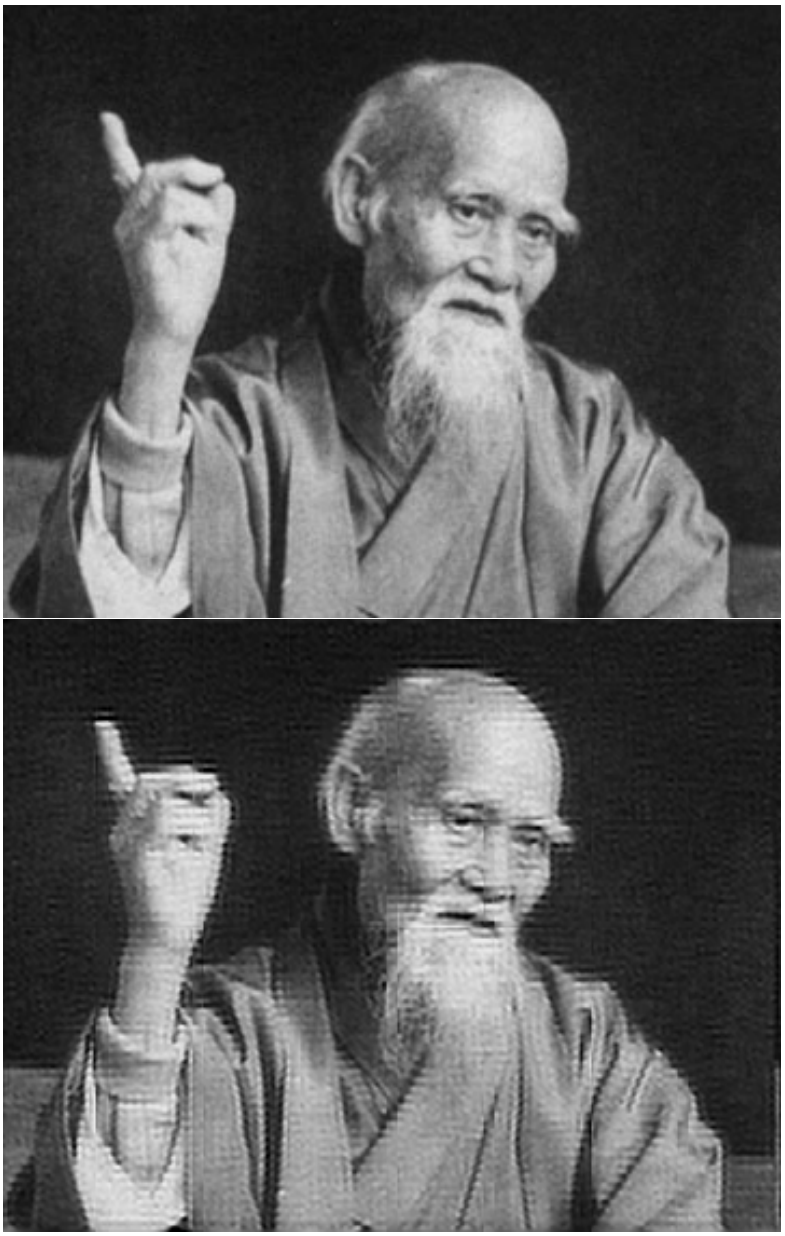


Figure 4.14: Comparing the (a) original image and (b) the super resolution image from figure 4.10(d)

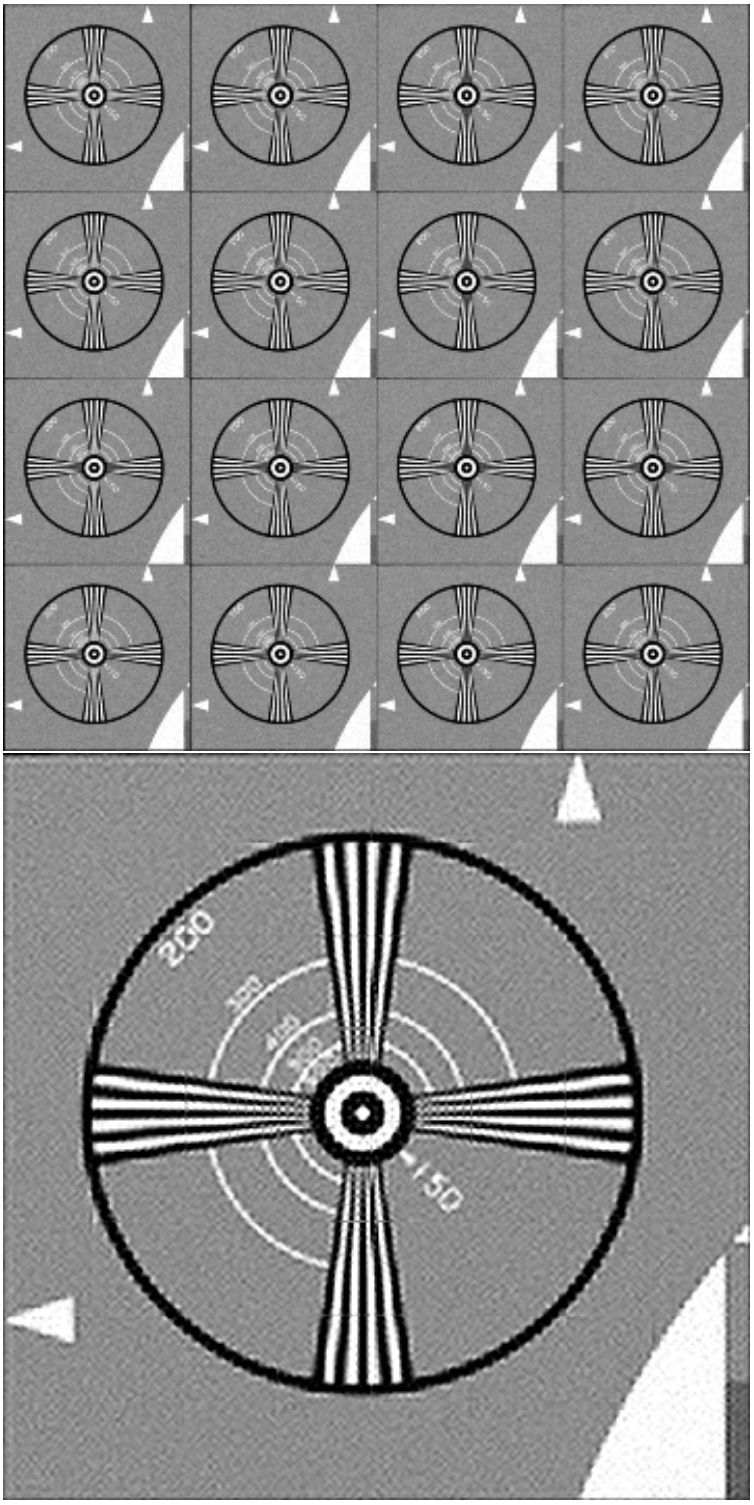
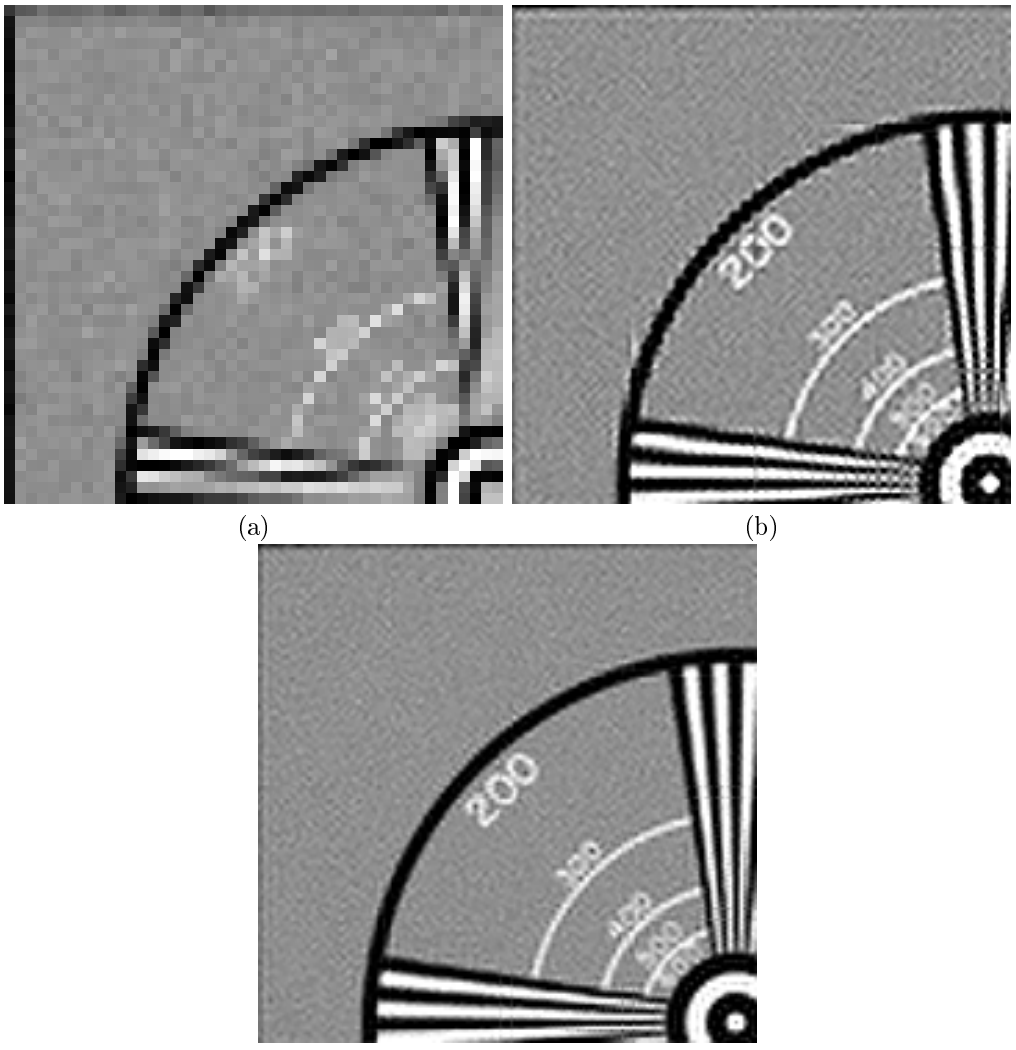


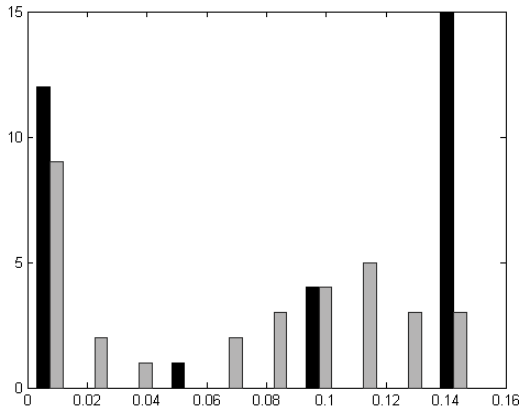
Figure 4.15: Comparing the (a) original image and (b) the super resolution image from figure 4.11(d)



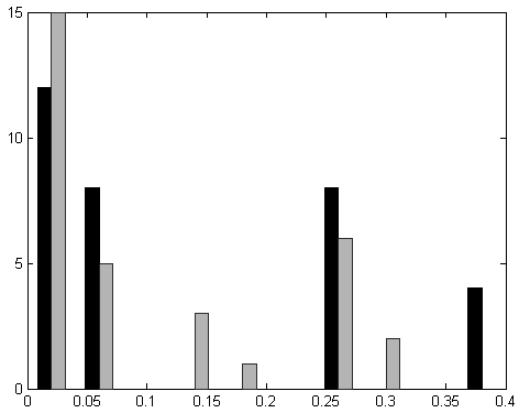
(a) (b)

(c)

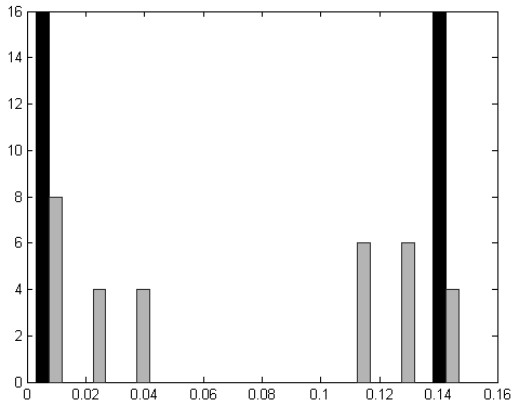
Figure 4.16: A closer examination at the images from figure 4.15 (a) original image, (b) the super resolution image from figure 4.11(d) and (c) super resolution with pre-registered LR images from figure 4.20(d)



(a)

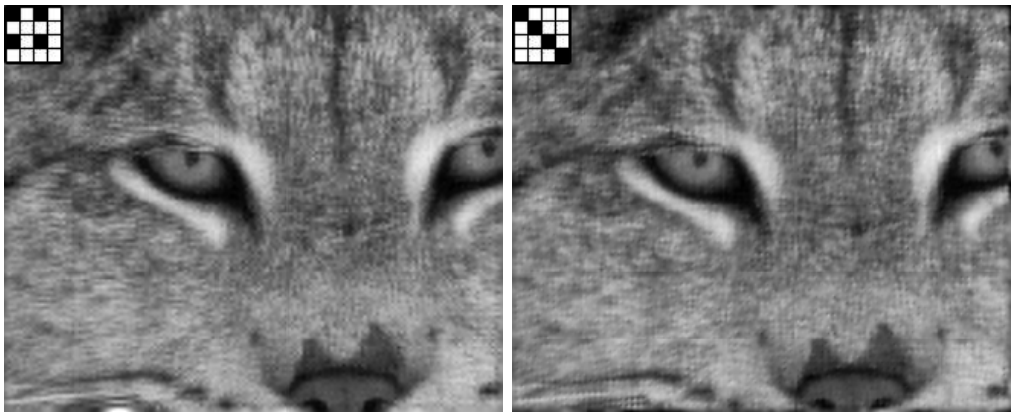


(b)



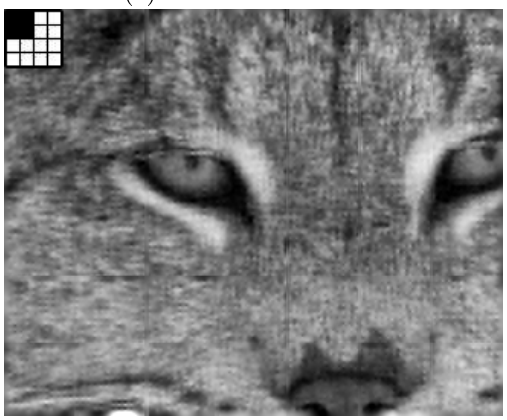
(c)

Figure 4.17: Graph over the error sizes for images (a) "Lynx" (figure 4.2), (b) "Ueshiba" (figure 4.3), (c) EIA (figure 4.4). Black bars indicate results for cross correlation only and gray bars represents cross correlation coupled with polynomial fitting.



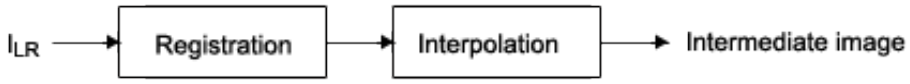
(a) PSNR: 28.7773 dB

(b) PSNR: 28.7148 dB



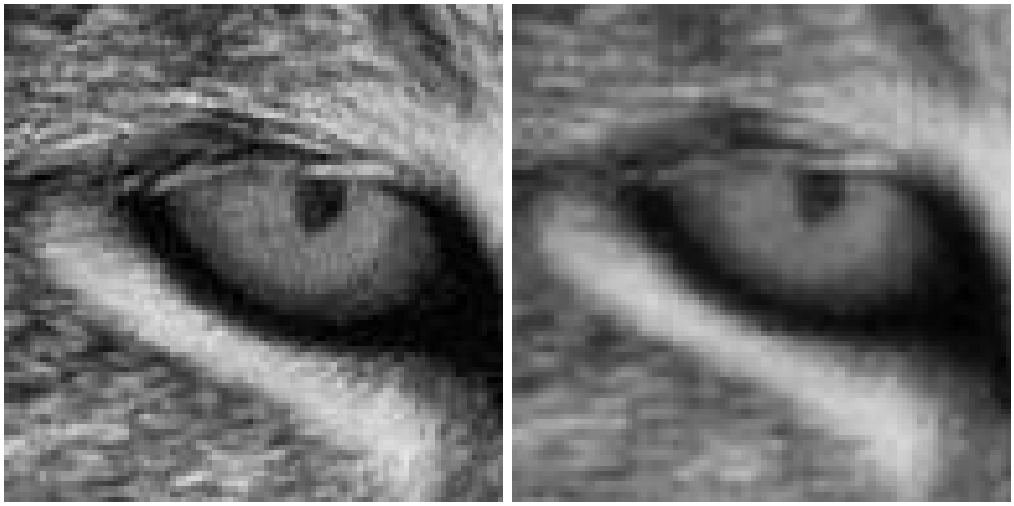
(c) PSNR: 28.6177 dB

(d) PSNR: 28.7144 dB



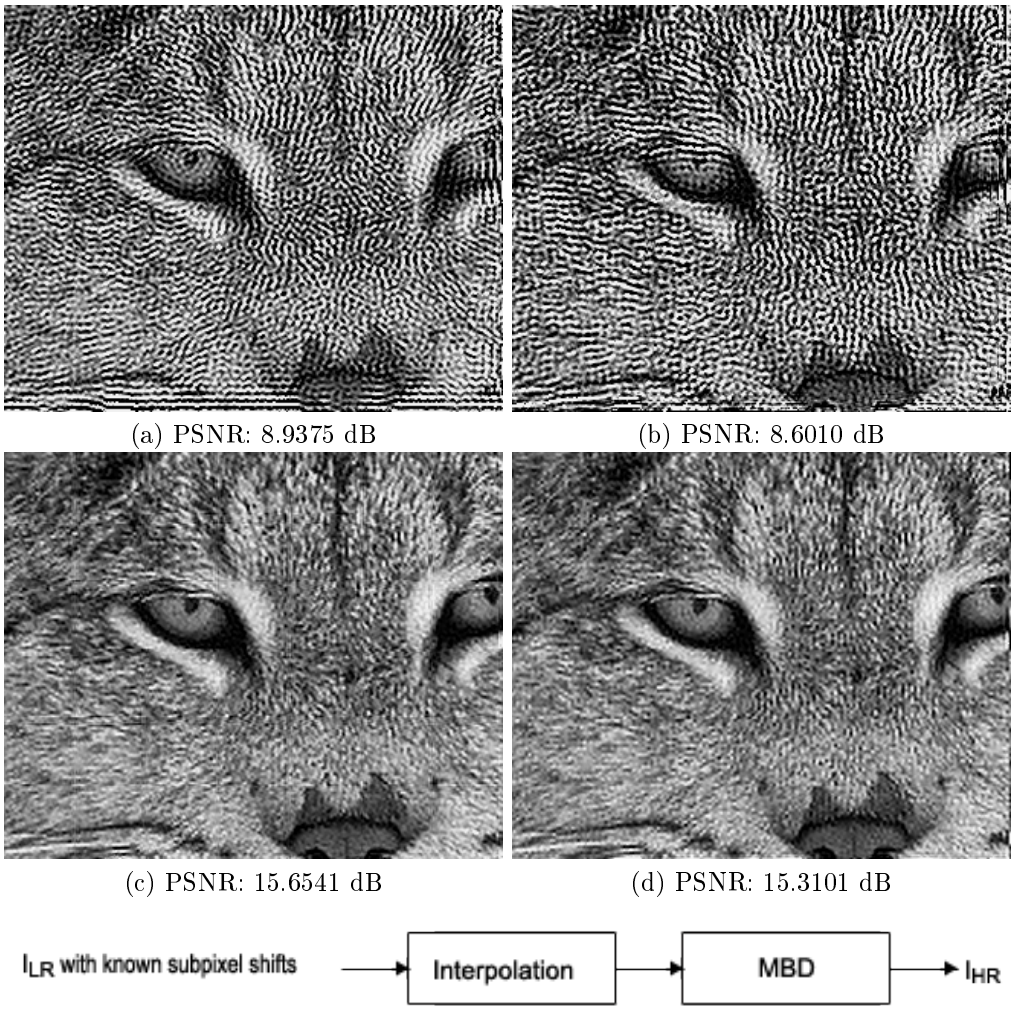
(e)

Figure 4.18: Results for experiment involving registration and spline interpolation for combinations of images with different shifts. The diagrams in the corners explains which frames are used from array of low resolution images in figure 4.2. (e) illustrates the what modules used in this experiment.



(a) (b)

Figure 4.19: An area is zoomed in and enlarged for better comparison. (a) The original image from figure 4.2. (b) Zoomed image from figure 4.18 (a)



(a) PSNR: 8.9375 dB

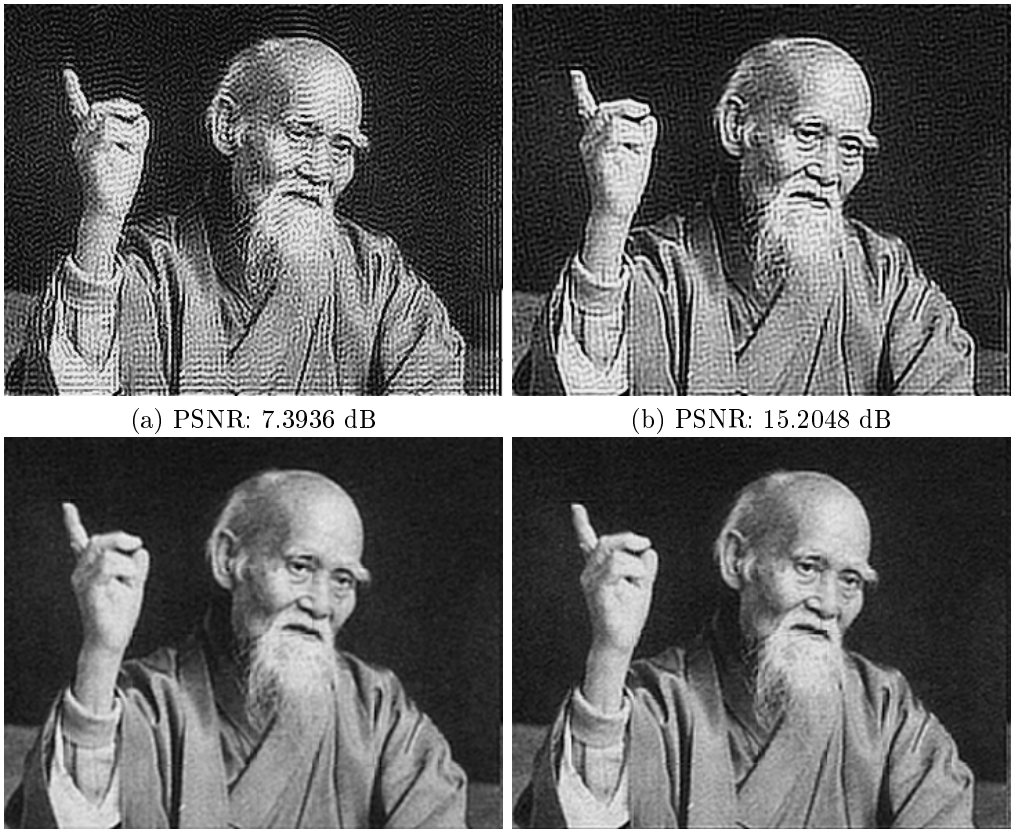
(b) PSNR: 8.6010 dB

(c) PSNR: 15.6541 dB

(d) PSNR: 15.3101 dB

(e)

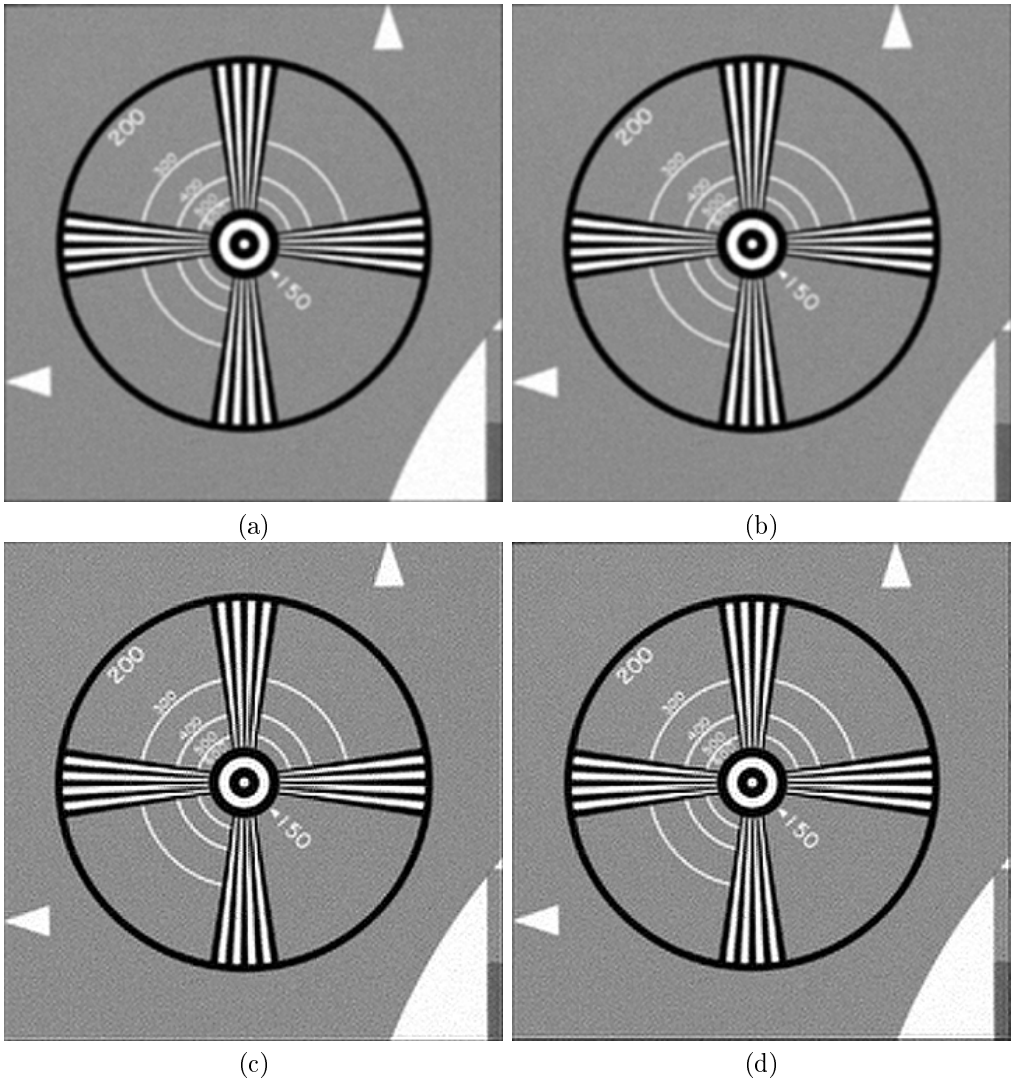
Figure 4.20: Experiment involving prior knowledge of image shifts for spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution for the "lynx" image (figure 4.2), 16 low resolution images and MBD parameters described in section 3.2.4 are varied between images. (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5. (e) illustrates the what modules used in this experiment.



(a) PSNR: 7.3936 dB (b) PSNR: 15.2048 dB

(c) PSNR: 19.2362 dB (d) PSNR: 19.4753 dB

Figure 4.21: Experiment involving prior knowledge of image shifts for spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution for the "Ueshiba" image (figure 4.3), 16 low resolution images and MBD parameters described in section 3.2.4 are varied between images. (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5



(a) (b)

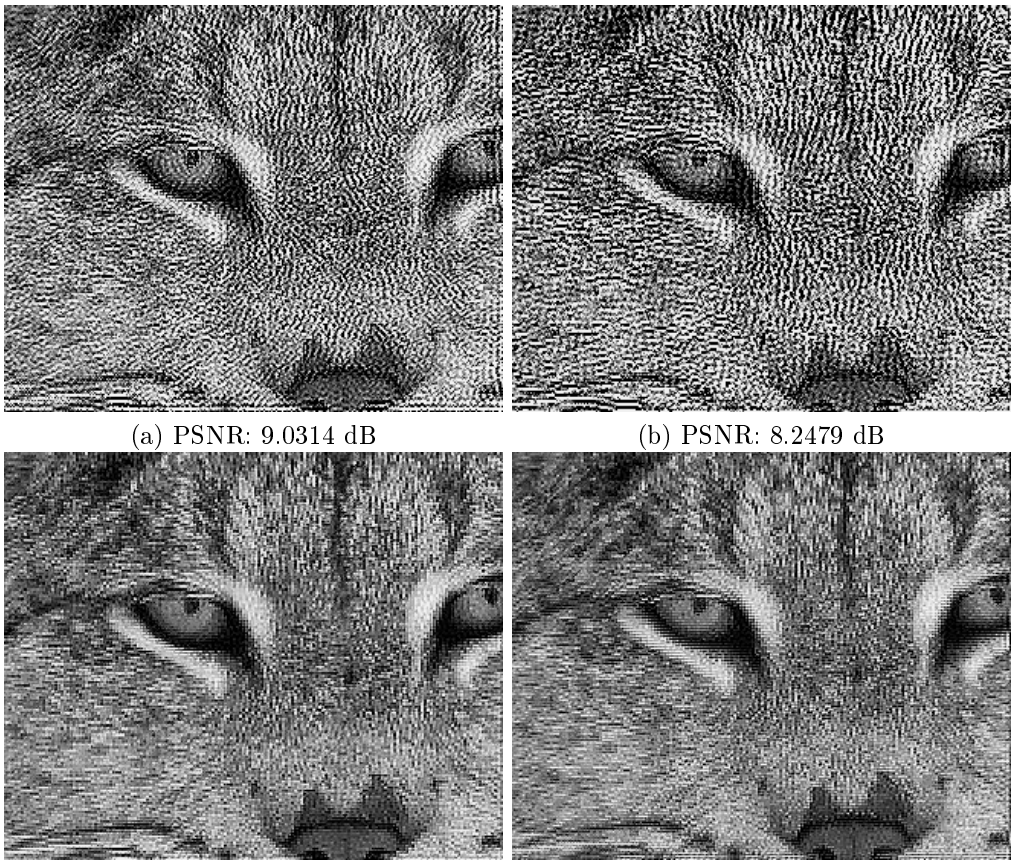
(c) (d)

Figure 4.22: Experiment involving prior knowledge of image shifts for spline interpolation (section 3.2.2, method (C)) and multiframe blind deconvolution for the EIA image (figure 4.4), 16 low resolution images and MBD parameters described in section 3.2.4 are varied between images. (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5

Method	Lynx	Ueshiba	EIA
InvDist	60.1058s	69.7120s	89.8685s
Spline	36.9493s	42.7582s	58.2883s

Table 4.2: Mean running time for the spline and the inverse distance interpolations.

Evaluating inverse distance weighting interpolation The inverse distance weighting (see: section 3.2.2 (B)) were originally written as the method to be used in this project but as the spline interpolation became more interesting this method became the alternative backup. This method represents a mathematically easy-to-follow approach. The experiment setup was to use the whole algorithm (registration, inverse distance interpolation and blind deconvolution) with the same parameter setup as in the same experiment with spline interpolation and compare the results from both tests. The images and the corresponding peak signal to noise ratios can be seen in figure 4.23. When comparing the Signal to noise ratio the inverse distance weighting setup didn't have the high maximum as the spline interpolation at image c and d, but when looking at images a and b the difference was little and image a was even better. It is notable that the visual appearance if these images are grainier than using the spline interpolation, a reason should be that spline interpolation contains the smoothing parameter which could be both positive and negative. Another interesting comparison is the computation time for both methods, the test setup is by executing the interpolation method for all three images on the same machine with same processes running in the background and comparing how fast the algorithm is executed when increasing the size by 4 times the original size, mean is taken from four runs, the times in seconds are tabulated in 4.2 and as can be seen, the spline interpolation works a lot faster. For better comparison between spline and inverse distance interpolation, a small area has been scaled up from the original image (figure 4.2), image (d) in figure 4.23 and the corresponding image in figure 4.9, they are shown in figure 4.24.



(a) PSNR: 9.0314 dB (b) PSNR: 8.2479 dB

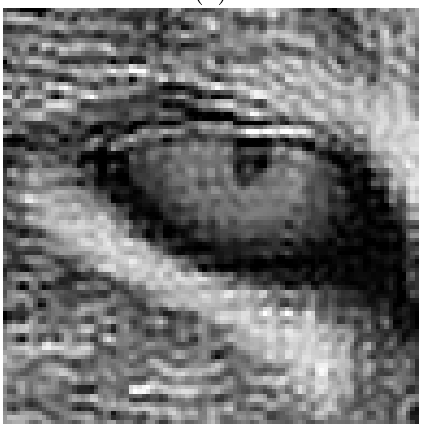
(c) PSNR: 13.2147 dB (d) PSNR: 12.7496 dB

Figure 4.23: Experiment involving registration, inverse distance interpolation (section 3.2.2, method (B)) and multiframe blind deconvolution for the "lynx" image (figure 4.2), 16 low resolution images and MBD parameters described in section 3.2.4 are varied between images. (a) α=1000, psf size = 5, (b) α=1000, psf size = 7, (c) α=100, psf size = 7, (d) α=100, psf size = 5

Discussion: Reslts for the two interpolation approaches As seen in detail in figure 4.24, the results differ between spline interpolation and inverse distance interpolation. Compared to splines which impose a certain smoothness in the image the inverse distance interpolation has a more diverse spread of pixel intensities. This can be a good or a bad thing depending on what the user wants the program to do. The slight unevenness noticed when looking at the image from the inverse distance interpolation in figure 4.24(b) is most likely caused by a registration error and not by the interpolation itself. Inverse distance interpolation is probably the most accurate method of the two but in the same time less robust and more unforgiving.

(a)

(b)



(c)

Figure 4.24: Comparision between spline and inverse distance based interpolation, the figure contains zoomed in versions of (a) the original image (figure 4.2(a)), (b) inverse distance interpolation (figure 4.23(d)) and (c) spline interpolation (figure 4.9(c))

Evaluation of super resolution algorithm with missing data There might be cases where there aren't enough images to fill all the possible shifts. In this simulation 4/16 images are removed, either one from each group before doing interpolation or one of the groups is removed altogether. The experiment setup is as follows: registration, interpolation and blind deconvolution is all used, alpha is set to 110^2 and PSF size is set to 5. The images and their PSNR are displayed in 4.25. The groups are ordered in the same manner as in figure 4.18 (a) where the shifts are even between the frames. When compared with each other it is apparent that removing one group gave a better result than removing one from each group. This might have two possible explanations, either is it caused by the choice of images, the images removed had shifts that were even leaving a fairly constant pattern comparing to removing one from each group there the pattern became more random. The other explanation might be that the interpolation is more sensitive to losing data than the MBD. When comparing the better of them to 4.9 (c) there is a really small difference.

(a) PSNR: 13.6139 dB (b) PSNR: 15.2498 dB

Figure 4.25: Experiment involving registration, spline interpolation, multiframe blind deconvolution with parameters α=100 and PSF size = 7. 12 images used instead of 16. As described in figure 3.4 the input frames may be removed either one for each intermediate image or one intermediate may be removed. (a) One frame removed for each intermediate image, (b) one intermediate is removed.

Chapter 5

Conclusions and further work

In this paper, a general approach for image fusion has been presented, approaches to obtain super resolution have been discussed and some methods have been investigated further. This work follows what more or less could be called standard procedure within image fusion and super resolution. For multi focal fusion studies were made on pixel-level and region level and the formulation were extended to multi modal fusion. For Super Resolution, a combination of readily available techniques resulted in a procedure capable of increasing the resolution of images. Many tests were performed to evaluate different preferences and cases. Many parts show promising results, especially that the algorithm wasn't sensitive to missing input frames. Even though comparable results in the PSNR measurement were received, it was possible to spot differences between the resulting images and clearly possible to pick out images more visually appealing than others. This observation was most apparent when comparing the preregistered images with the images registered by the algorithm. The spline interpolation is an interesting approach that works fast with good results. The drawback is that it introduces extra smoothing to the image that has to be understood and eliminated, the same applies to the segmentation effects. Compared to inverse distance interpolation it wasn't really as accurate but should be more able to suppress registration errors. What may prove be a limitation is the fact that both interpolation and blind deconvolution requires multiple inputs to produce one output image. Replacing one of those methods with one that only requires one input would eliminate the need to carefully combine the images in the first step.

5.1 Further work

Due to the boundaries set for this project a lot of assumptions were made simplifying the problem, in both multi focal fusion and super resolution fusion preregistered images or images only containing certain warping aspects were tested, it is a fact that the algorithms are designed only to handle these types of images. In many cases such images are not readily available and thus to make the algorithm practical an extended registration method has to be developed and changes has to be made to the interpolation method. The inclusion of affine transforms would be a possible solution.

When applying the procedure for multi focal fusion on multimodal images only the most basic techniques did work, a more specialized approach for multimodal fusion should be needed, one way is to refine the decision map after it's created to remove randomness like lone pixels.

It would be of interest to see if there is possible to implement the spline interpolation without the technical workarounds used within this project as they might contain liabilities like unwanted segmentation of the image. During development it was tested but even with a much smaller number of samples than what is used in the application the interpolation still kept going after 20 minutes and compared to around one minute when interpolating small parts of the images at the time it was decided that they were inevitable.

Another interesting topic is that if it is possible to develop an AI that from the registration inputs can choose the most suited frames as a complement to a smart mathematical formulation for super resolution. As providing images to the algorithm require the most effort as long as the programming is done doing super resolution with the fewest possible images is preferable, likewise, more data to process without gain is something that should be avoided.

Bibliography

[1] Rafael C Gonzales, Richard Woods, 2002, *Digital Image Processing, 2nd edition*

[2] Vladimir Petrovic, *M*ultisensor pixel-level image fusion PhD thesis, University of Manchester, UK, February 2001

[3] Gemma Piella *a general framwork for multi resolution image processing: From pixels to regions* Technical report PNA-R0211, ISSN 1386-3711, CWI, Amsterdam, The Netherlands, May 31, 2002

[4] Filip Šroubek and Jan Flusser, 2003, *Multichannel Blind Iterative Image Restoration* IEEE Trans. Image Processing, 12(9):1094Ű1106

[5] Filip Šroubek and Jan Flusser, 2007, *Multiframe blind deconvolution coupled with frame registration and resolution enhancement*

[6] Sung Cheol Park, Min Kyu Park, Moon Gi Kang, 2003, *Super resoluion image reconstruction, a technical overview* Signal Processing Magazine, IEEE, Volume 20, Issue 3, May 2003 Page(s): 21 - 36

[7] Nils Karlsson, Selman Jabo, Anders Hildeman, *Fusing images for super resolution* BSc Thesis, Chalmers University of Technology, Gothenburg, Sweden 2008

[8] Donald Shepard, 1968, *A two-dimensional interpolation function for irregularly-spaced data*

[9] V. Argyriou and T. Vlachos, *A study of sub-pixel motion estimation using phase correlation* University of Surrey Guildford GU2 7XH, United Kingdom 2006

[10] "Discrete wavelet transform", Wikipedia, the Free encyklopedia, 2008, <http://en.wikipedia.org/wiki/Discrete_wavelet_transform>

[11] “Peak signal to noise ratio”, Wikipedia, the free encyklopedia, 2008, <http://en.wikipedia.org/wiki/PSNR>

[12] “Super Resolution”, Wikipedia, the free encyklopedia, 2008, <http://en.wikipedia.org/wiki/Super-resolution>

[13] “Fast Normalized Cross-Correlation” <http://www.idiom.com/ zilla/Work/nvisionInterface/nip.html >

[14] “MATLAB: Online help for function tpaps.m - Release 2007b” <http://www.mathworks.com >