HYBRID MULTIPLE-IMAGE SUPER-RESOLUTION SYSTEM USING INTERPOLATION, LEARNING AND RECONSTRUCTION

by

Aran Shaker

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

6 February, 2018

SUMMARY

HYBRID MULTIPLE-IMAGE SUPER-RESOLUTION SYSTEM USING INTERPOLATION, LEARNING AND RECONSTRUCTION

by

Aran Shaker

Supervisor(s):	Dr. H. C. Myburgh
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Super-Resolution, Interpolation, Learning, Reconstruction

High-resolution images are a fundamental requirement of modern imaging applications. However, sensor hardware can only go so far in its ability to capture high density images of a scene. Post-processing algorithms, such as the super-resolution technique, can be used to enhance image quality after capture, thus overcoming the limits of sensor hardware. The super-resolution technique can be approached with different methods, including interpolation, learning and reconstruction. Each of these is best suited for different regions of a scene. Interpolation performs well on smooth areas of an image, while learning is better at enhancing edges and reconstruction will be more appropriate for a heavily textured region. This research investigates the question of combining the strengths of these methods, in order to provide more accurate high-resolution approximations of a scene.

The system performs feature detection in order to determine the prominent features of the input images. Depending on the presence of different features, the hybrid system will choose which super-resolution algorithms to use in the construction of the final output image. The interpolation algorithm's fast computation speed and simplicity made it the perfect candidate for the processing of the simplest identified feature - smooth regions. The learning algorithm's greater complexity and ability to enhance high frequency information efficiently allowed it to be used in cases where many

edges were present in an image. The reconstruction algorithm's ability to deal with high levels of noise and blur in an image made it useful when dealing with features that contained windows of rapidly changing pixel intensity. Feature detection made the combination of these algorithms possible and by analysing different algorithms' behaviour when faced with different features, different combinations of algorithms were able to be applied appropriately to maximise performance and accuracy, while minimising computational complexity.

The super-resolution image reconstruction technique is capable of overcoming the inherent limitations of an imaging system and can improve the resolving quality of low-resolution images. Different methods and approaches to this technique have different practical and theoretical applications. The advancement of knowledge in this field will make it possible for image processing researchers to not only further investigate the use of multiple images in high-resolution imaging systems, but also make imaging technology affordable in a technologically advancing world.

LIST OF ABBREVIATIONS

LR	Low-Resolution
HR	High-Resolution
SR	Super-Resolution
CCD	Charged-Coupled Device
CMOS	Complementary Metal-Oxide-Semiconductor
CFT	Continuous Fourier Transform
DFT	Discrete Fourier Transform
PSF	Point Spread Function
IBP	Iterative Back Projection
POCS	Projection Onto Convex Sets
MAP	Maximum A posteriori Probability
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

High-resolution images are a fundamental requirement of modern imaging applications. However, sensor hardware can only go so far in its ability to capture high density images of a scene. Post-processing algorithms, such as the super-resolution technique, can be used to enhance image quality after capture, thus overcoming the limits of sensor hardware. The super-resolution technique can be approached with different methods, including interpolation, learning and reconstruction. Each of these is best suited for different regions of a scene. Interpolation performs well on smooth areas of an image, while learning is better at enhancing edges and reconstruction will be more appropriate for a heavily textured region. This research investigates the question of combining the strengths of these methods, in order to provide more accurate high-resolution approximations of a scene.

1.1.2 Research gap

The respective super-resolution techniques are capable of improving resolving quality on their own. However, the combination of these approaches and their application to specific image features will enable the system to utilise the strong aspects of each of these techniques. This will also result in the optimisation of the computational load of enhancing the resolving quality of various image areas.

Figure 1.1 shows an image with distinct image features. Given any one of the super-resolution approaches, they would successfully enhance the resolving quality of the image. However, each one of them would perform optimally on a specific image feature, rendering the processing of other features done, futile.

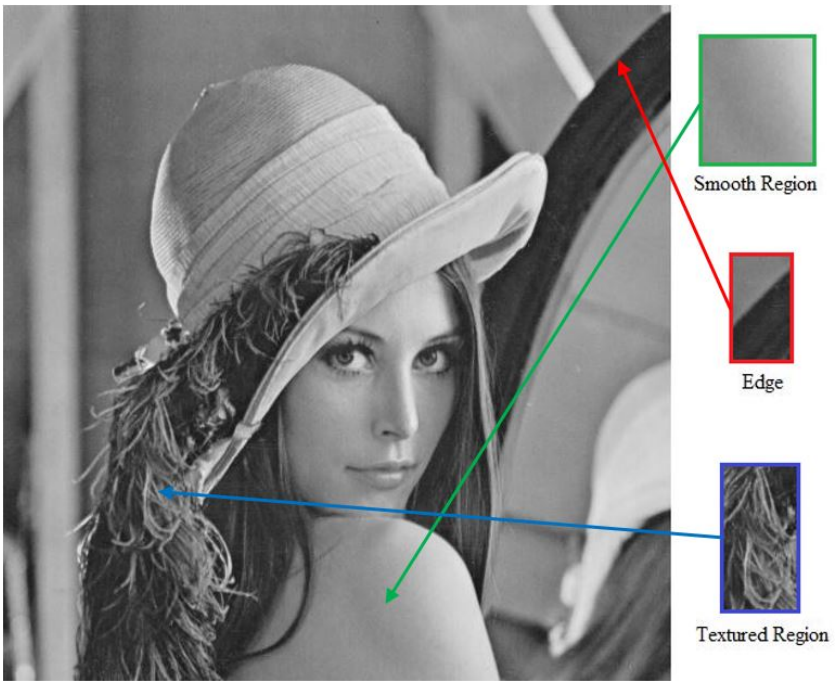

Figure 1.1. Image with distinct features

A hybrid system, in which each super-resolution approach is performed on specific image features, will enable the optimal improvement in resolving quality of images with multiple image features, while ensuring that computation time is not wasted on sub-optimal processing. The new super-resolution system will improve image quality in many applications where high-resolution images are required, such as medical forensics, satellite technology and computer vision.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The objective of this research is to combine different super-resolution algorithms in order to reconstruct high-resolution approximations of a scene more accurately, thus taking advantage of the strengths of these algorithms, overcoming limitations of sensor hardware and reducing the cost of imaging systems.

The research questions are as follows:

- What are the limitations of current super-resolution algorithms in terms of accuracy and complexity?
- How can existing super-resolution methods be combined to provide more accurate high-resolution approximations?
- Which features of an image can be enhanced best by different super-resolution algorithms?
- Which algorithms from the three super-resolution categories (interpolation, learning and reconstruction) should be used when creating a hybrid algorithm?
- Is the increase in accuracy of a hybrid super-resolution algorithm worth the increase in computation complexity?

1.3 APPROACH

In order to achieve the objectives of the super-resolution system, the following approach needs to be followed:

1. Investigate and implement interpolation based methods of super-resolution.
2. Investigate and implement learning based methods of super-resolution.
3. Investigate and implement reconstruction methods of super-resolution.
4. Investigate feature detection algorithms to identify different image areas.
5. Combine and apply super-resolution methods to the different image features.

6. Optimise and reduce computational load.

1.4 RESEARCH GOALS

The goals of this research are to thoroughly investigate and implement different existing super-resolution algorithms to obtain an understanding of each of their strengths and limitations. This knowledge will then be applied to identify the best way in which these algorithms can be combined to provide more accurate high-resolution approximations of a scene. The system will also need to be optimised to reduce overall computational load and speed up execution time. Thus, different combinations of algorithms will be explored to investigate the optimal selection of algorithms for appropriate features. This research will provide insight into the use of super-resolution image reconstruction techniques in image processing applications. It provides a novel post-processing algorithm by combining various super-resolution approaches, which can be used in conjunction with currently existing sensor hardware in order to enhance image quality, thus overcoming hardware limitations. The research will have a valuable impact on applications of image processing such as computer vision, surveillance technology, medical forensics and satellite imaging.

1.5 RESEARCH CONTRIBUTION

This research is intended for publication and has been submitted to IEEE Access for peer review under the title: *Hybrid Multiple-Image Super-Resolution System using Interpolation, Learning and Reconstruction by A. Shaker and H.C. Myburgh.*

1.6 OVERVIEW OF STUDY

The study accomplishes its goals by performing a number of research steps explained below:

- Literature Study. A literature study is conducted on the current limitations of sensor hardware and an investigation into different super-resolution algorithms.

- Methods. Development of a hybrid super-resolution system, which is able to characterise different image features and to apply appropriate algorithms to regions of a scene, is developed.

- Results. The results of the developed system are presented and the performance of various algorithms is measured and compared. The systems performance on large sets of data is analysed and optimised for minimal computational load and complexity.

- Discussion. The results obtained are discussed and analysed. Strengths and limitations are explored to understand design choices and potential improvements.

- Conclusion. Closing remarks and recommendations for future work.

CHAPTER 2 LITERATURE STUDY

In this chapter, a literature study on high-resolution imaging systems is presented. Furthermore, existing super-resolution image reconstruction algorithms for high-resolution applications are discussed and compared. Finally, a brief investigation into the use of feature detection for the super-resolution technique is performed.

2.1 HIGH-RESOLUTION IMAGING SYSTEMS

Image processing technology has experienced rapid development in recent years, due to the growing demand for high-resolution image information, subsequently causing the advancement of both hardware and software in the image processing field. High-resolution images provide more information and clarity to an observer of a scene in various applications such as medical imaging, satellite technology, surveillance systems and computer vision [1]. High-resolution images can be obtained by using image data, captured by multiple low-resolution observations of the same scene [2]. This technique is known as super-resolution image reconstruction.

High-resolution images have a high pixel density, therefore providing more detail for use in imaging applications [1]. Because of this, high-resolution images are a fundamental requirement for the functionality of modern imaging systems. Examples of such applications include: doctors using high-resolution medical images to make diagnoses more accurately, the improvement of computer vision pattern recognition performance and the enhancement of satellite technology's ability to distinguish similar objects by using high-resolution satellite images [2].

Since the late 20th century, CCD (charged-coupled device) and CMOS (complementary metal-oxide-semiconductor) image sensors have been the most widely used sensor technology for capturing digital

images [3]. There is no doubt concerning the suitability of these sensors for most digital imaging applications. However, sensor technology has become increasingly expensive [4]. Applications which require higher-resolution cameras or camcorders have become more expensive. The need for the increase in current resolution levels is apparent in many applications; for instance, scientists require a very high-resolution level close to that of an analogue 35 mm film, which has no visible artifacts during magnification [5].

The most straightforward solution to increasing spatial resolution in an image is to increase the number of pixels within a sensor, thus reducing pixel size. This solution is, unfortunately, not feasible since a smaller pixel size also means that the pixel cannot capture as much light from the scene [5]. This generates shot noise, which severely degrades image quality. We can therefore deduce that a limitation of pixel size reduction exists, before suffering the effects of shot noise. Current image sensor technology has almost reached the optimally limited pixel size estimated at 40 μm^2 for a 0.35 µm CMOS sensor [5] and we must, therefore, search for alternative solutions to further enhance spatial resolution.

Another approach is to increase the chip size, which leads to an increase in capacitance [6]. However, large capacitance vastly decreases charge transfer rates, producing blurred images, thus rendering this approach ineffective. A larger storage capacity is an inevitable requirement for the storage of higher resolution images, as well as the need for higher bandwidth requirements for the transmission and reception of high-resolution images [6]. Along with these approaches, a crucial concern also exists for the high cost of high precision optic technology used in many commercial applications for high-resolution imaging.

In order to overcome the limitations of the sensors and optics manufacturing technology, a new approach towards the enhancement of spatial resolution must be adopted. Many studies investigate the use of several images to increase the performance of an imaging system. One example combines pictures of a static scene taken from one camera with varying exposure times to create images with increased dynamic range [7]. Another method stitches together pictures taken from one position with abutting fields of view to create high-resolution mosaics [8]. Super-resolution is a digital signal processing method which is able to approximate a high-resolution image by using data captured by multiple low-resolution observations [9]. It is a post-processing technique and can therefore be used in conjunction with currently available hardware, resulting in a much more cost-effective system.

2.2 SUPER-RESOLUTION IMAGE RECONSTRUCTION

Currently, super-resolution image reconstruction can be divided into three major categories [10], namely: interpolation, learning and reconstruction based approaches. The interpolation based approach is one in which multiple images are used to estimate missing pixel data within a region of support. The learning based approach acquires correlational prior knowledge between low and high-resolution image patches through a learning process, extracts high frequency information based on local features from training images and then adds the high frequency detail for low-resolution images, in order to guide high-resolution image reconstruction. The reconstruction based approach models observed low-resolution images through the acquisition process and constructs a high-resolution image, using a sequence of low-resolution images obtained from the same scene.

2.2.1 Interpolation based approaches

The interpolation based approach for super-resolution was first proposed by Tsai and Huang [9]. The first interpolation based methods used a frequency domain approach, which utilised under-sampled low-resolution images to reconstruct high-resolution images [9]. The super-resolution approach was expanded to use recursive algorithms for the reconstruction of super-resolved images from noisy and blurred observations [9]. These and other methods can be categorised into two major classes of interpolation based approaches: The first category is super-resolution using motion cue, a technique that uses sub-pixel shifts and relative motion between low-resolution observations in approximating a high-resolution image. The second category is motion-free super-resolution, which uses differing zoom, blur and photometric cues to obtain super-resolution approximations.

2.2.1.1 Non-uniform interpolation

Digital image capture produces discrete representations of continuous scenes. This discretion in both space and intensity can be described as a sampling process, which creates aliasing, as well as a loss of information at frequencies above the Nyquist rate [11]. Non-uniform interpolation is a motion cue technique, which uses a collection of closely related lower resolution images in order to construct a higher resolution image, reducing the effects caused by aliasing and information loss. Each low-resolution image has aliased higher frequency information of the scene differently and therefore

it becomes possible to reverse some of the aliasing and reconstruct higher frequencies in the image with information from multiple images. Non-uniform interpolation can be described by three steps: registration, reconstruction and restoration [11].

The registration step can be described as a process of determining relative translations between the observed images and is required to determine the offsets between the images with near pixel accuracy. This could be conducted in a number of ways: by the contents of the images alone or with prior knowledge of the observed images. The reconstruction step involves the placement of registered images onto a high-resolution grid by combining pixels from all the lower resolution images to form a composite image. Due to the global translational motion between the observed images, the newly formed merged image has a semi-uniform structure, instead of being a sampled perfectly uniform rectangular grid. Figure 2.1 shows an example of this rectangular grid.

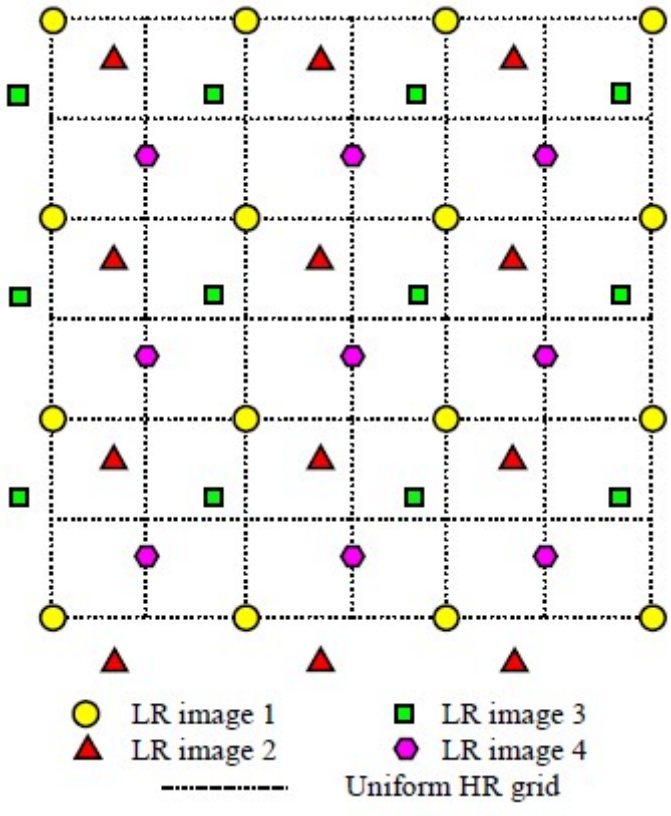

Figure 2.1. Composite image on a semi-uniform grid. Adapted from [12], ©2011 IEEE

Reconstruction requires that missing pixel data be approximated, using neighbourhood information. There are several approaches to consider, which will be able to execute the reconstruction process. Possible algorithms include: nearest neighbour interpolation, where the value of the closest pixel is

adopted; linear interpolation, where the average pixel value of the linear area surrounding a pixel is calculated and cubic interpolation, where the average pixel value of the cubic area surrounding a pixel is calculated. The restoration step is described as a de-blurring procedure which can restore high frequency data, which has been suppressed by the initial low-resolution image acquisition process.

2.2.1.2 Interpolation in the frequency domain

The frequency domain approach uses aliasing in low-resolution observations to reconstruct a high-resolution image. The global translational motion between the low-resolution observations is used to define a relationship between the low-resolution images and the corresponding high-resolution image. The approach is founded on three main principles: Firstly, the assumption that the original high-resolution image is band-limited [13]; secondly, the use of the Fourier transform shifting property [13] and thirdly, the aliasing relationship between the Continuous Fourier Transform (CFT) of the original high-resolution image and the Discrete Fourier transform (DFT) of the observed low-resolution images [13]. Based on these principles, it is possible to formulate a system equation relating the aliased DFT coefficients of the low-resolution images to a sample of the CFT of an unknown high-resolution image [13].

The frequency domain approach is theoretically simple and offers minimal computational load. This was further established with the introduction of a Discrete Cosine Transformation (DCT) instead of the traditional DFT [12], reducing memory requirements and computational costs. This approach can be executed in parallel, further reducing computation time [12].

2.2.1.3 Interpolation using zoom cue

Interpolation using zoom cue requires the low-resolution input images to be obtained with varying degrees of zoom in order to reconstruct high-resolution images. Figure 2.2 shows an example of how the observed low-resolution input images are related to the high-resolution approximation.

In Figure 2.2, Y_k represents the low-resolution input images and X represents the high-resolution image. The observed images must all have the same dimensions and prior knowledge of the zoom factors

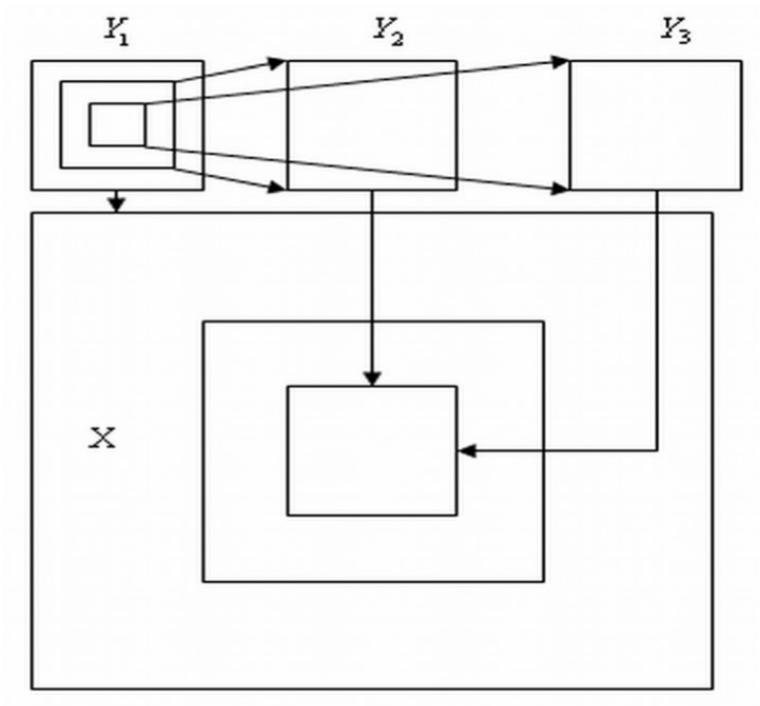

Figure 2.2. Block schematic of super-resolution using zoom cue. Adapted from [12], ©2011 IEEE

between observations must be known. To obtain a high-resolution image from a specific region, the image with the least zoom corresponding to the scene, in this case Y_3, is up-sampled by a factor equal to the product of the zoom factors of the remaining images. The rest of the input images are modeled as decimated and noisy versions of the high-resolution image.

Equation (2.1) represents this mathematically, y representing the decimated and noisy image, x being the high-resolution image and D, the decimation matrix, which changes size depending on the zoom factor of the image.

$$y_k = D_k x + n_k \tag{2.1}$$

2.2.1.4 Interpolation using blur cue

There are two main causes of blur in an image: The first is the presence of a sensor's Point Spread Function (PSF) at the sensor surface and the second is due to the relative motion between the camera

and the scene when capturing an image. The blur cue approach assumes that the blurring matrix for each low-resolution observation is different. Therefore, blur can be used as a cue for super-resolution [12]. The low-resolution images are blurred by a varying, known linear space invariant blurring kernel. These blurred images are then used to perform super-resolution.

2.2.2 Learning based approaches

The learning based approach uses prior knowledge as a basis for super-resolution. It acquires relational data from a database of training sample sets consisting of low and high-resolution image pairs. It then produces new high frequency details to construct a high-resolution approximation. It is able to do this without increasing the input image sample size, thus allowing better performance under the circumstance of large amplification factors [14], making it the ideal option for large amounts of pixel information.

The system learns the missing high-resolution geometric properties of test images and finds the best fit patch for corresponding low-resolution patches [15]. A database of low-resolution images and their high-resolution counterparts are given to the system to learn about the relationship between them. This relationship is used to obtain an initial high-resolution estimate of an input image. A number of tools are used to optimise and select the best fit patch: a data fitting term, which is an estimation of the aliasing derived from the high-resolution estimate [15]; geometrical properties of the high-resolution estimate [11] and the high-resolution estimate itself. The output super-resolution image is a collection of best fit patches. This process is described in Figure 2.3.

There are a number of different learning based approaches, which train data and decide on the best corresponding high-resolution patch, using varying methods.

2.2.2.1 Markov network

The Markov network based approach is an iterative approach, which is used to model the spatial relationship of an image by using probability. This approach divides an image into small blocks, known as patches, which are mapped to a node on a Markov network. Through learning, the system generates two matrices: A transition probability matrix for adjacent high-resolution image blocks [16] and a

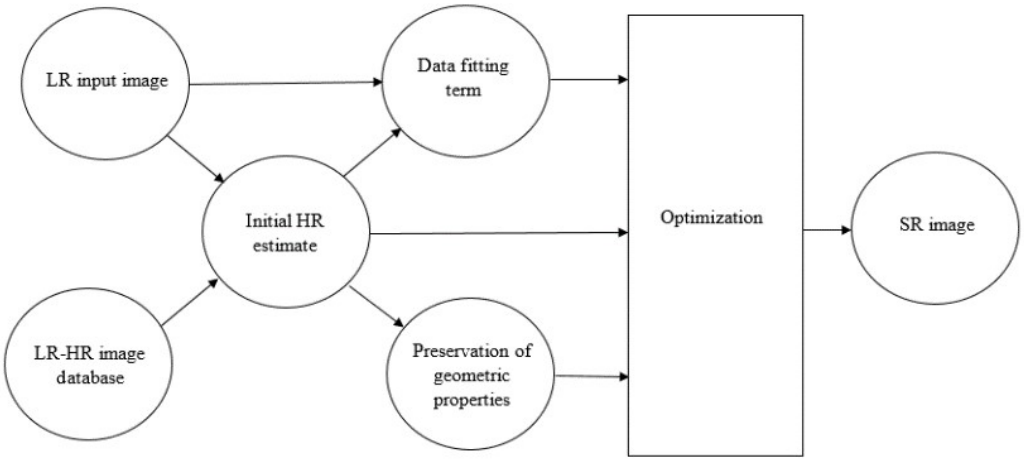

Figure 2.3. Schematic representation of learning based super-resolution. Adapted from [12], ©2011 IEEE

transition probability matrix between high and low-resolution image blocks [16]. Figure 2.4 depicts the Markov network model in relation to high and low-resolution image blocks.

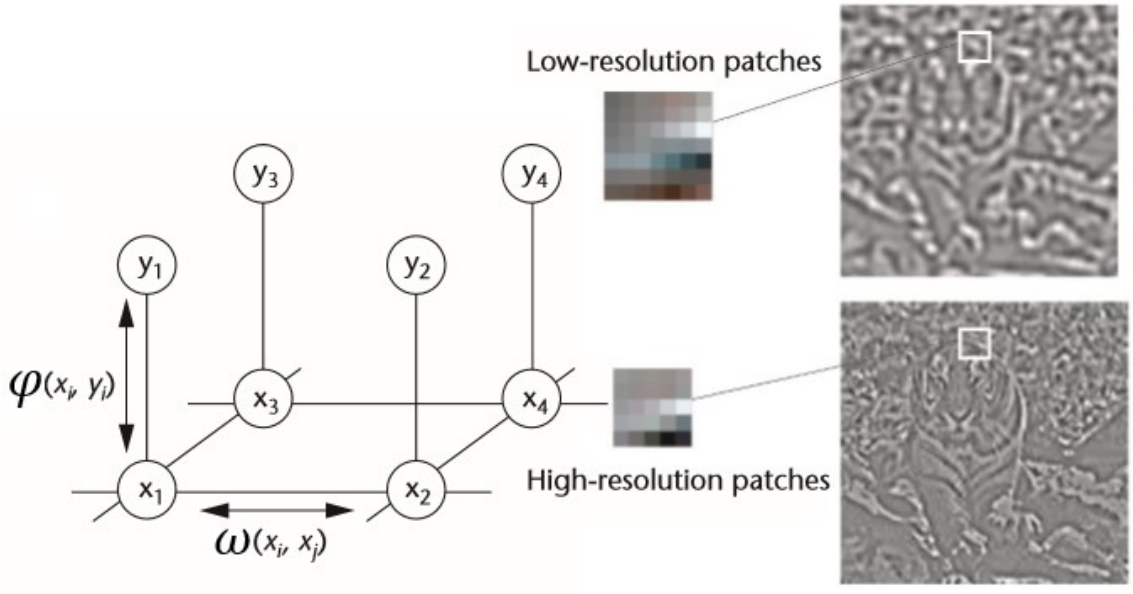

Figure 2.4. Markov network model. Adapted from [17], ©2002 IEEE

The Markov network based approach can be further enhanced by organising image block databases to improve matching efficiency [14] and using major counter priori knowledge of the scene to improve image quality [16].

2.2.2.2 Neighbourhood embedding

The neighbourhood embedding approach assumes that the high and low-resolution image blocks could compose similar local geometric structures within a feature space. A number of nearest neighbour representations in the low-resolution structure are located for each block in a test sample database and then the estimation of the image blocks in the higher resolution structure is produced by using weighted coefficients [18].

Under ideal circumstances, each high-resolution image block relates to a corresponding low-resolution image block [19] and correlation connections among its neighbourhood image blocks are maintained [20]. This ensures that the algorithm is accurate and local features are preserved in the reconstructed image.

2.2.2.3 Sparse representation

The sparse representation of image blocks can be used to realise super-resolution. This is done by randomly selecting a number of image blocks from a group of high-resolution images, which are used to form an over-complete dictionary. A linear programming method is then used to obtain sparse representations of each test image block in the over-complete dictionary. The sparse coefficients are then used to reconstruct the high-resolution image [21].

2.2.2.4 Face hallucination

The face hallucination approach was developed exclusively for facial images. Using this method, the face image feature space is chosen from the horizontal and vertical derivations of the Gaussian and the Laplacian pyramid of the face image [22]. Each low-resolution pixel's characteristics are used as an index to search for example samples in a sample repository. These samples correspond to high-resolution information, which is used to reconstruct the high-resolution image. Because the reconstruction process is carried out on each pixel, it results in an eight times amplification effect, similar to interpolation methods. However, this causes it to lose some facial global constraints, like symmetry and brightness uniformity.

2.2.3 Reconstruction based approaches

Reconstruction based approaches for super-resolution were introduced to estimate sub-pixel displacement and noise variance in low-resolution images, allowing high-resolution images to be estimated through expectation maximums [23]. Input low-resolution images suffer from warping effects, blurring and sub-sampling. The reconstruction based approach focuses on creating an observation model relating to the original high-resolution scene and the observed low-resolution images. This relationship can be described by equation (2.2).

$$y_k = SB_kW_kx + n_k \tag{2.2}$$

y_k is the kth low-resolution input image; S represents the sub-sampling matrix; B_k is the blur matrix associated with the kth input image, due to the movement of the camera lens or the presence of fast moving objects within the scene; W_k represents the warping matrix, including the global or translational motion, as well as rotation information, and n_k represents a noise vector. In the framework of reconstructing an image based on the observation model, a number of reconstruction based approaches exist.

2.2.3.1 Iterative back projection estimation

The iterative back projection (IBP) estimation approach is a process in which a high-resolution image is estimated, using the given low-resolution observation images as an initial solution. A simulated low-resolution image is then created according to the observation model, using the new estimation as the high-resolution image. This process is derived from equation (2.2) and is described in equation (2.3).

$$y^0 = SB_kW_kx^0 + n_k \tag{2.3}$$

y^0 is the simulated low-resolution image and x^0 is the first high-resolution estimation. The IBP estimation approach then states that for the simulated high-resolution image x^0 to be equivalent to

the original high-resolution scene x, using the formula described in equation (2.3), the simulated low-resolution image y^0 must equal the original low-resolution observation y [24]. If y^0 and y are not equivalent, then the high-resolution estimation must be improved further. This is done by projecting the error between y^0 and y reversely onto the high-resolution estimate to correct it. This correction is described in equation (2.4).

$$x^k = x^{k-1} + H^{BP}(y - y^{k-1}) \tag{2.4}$$

x^k is the kth revised high-resolution estimation and H^{BP} is a back projection kernel. This process is repeated iteratively until the error between the simulated low-resolution image and the original low-resolution observation is minimised to a satisfactory level. The IBP estimation approach has a very low computational load and a strong robustness to noise [25]. It can also be combined with interpolation approaches to further improve the initial high-resolution estimation and reduce the time it takes to converge on a solution [26].

2.2.3.2 Projection onto convex sets estimation

The projection onto convex sets (POCS) estimation approach has the ability to enhance detail and texture information of an image, as well as greatly reducing the influence of the error caused by motion information. It uses the theory that the solution space for the super-resolution reconstruction technique has multiple constraints characterised by the ideal properties of an image, namely: positive definiteness, energy bounds, observation identity and local smoothness [27]. Each of these properties can be represented by constraint conditions defined by convex sets of a vector space. Thus, the solution space for the super-resolution image reconstruction technique is the intersection space of this convex constraint set [27].

The POCS estimation approach is an iterative technique in which an initial solution x^0 is estimated. Each constraint, mentioned above, is defined as a convex set projection operator, P_k. The next revised estimation is the product of the previous estimation and the constraint sets [28], as described in equation (2.5).

$$x^{k+1} = P_k ... P_3 P_2 P_1 x^k \tag{2.5}$$

2.2.3.3 Maximum a posteriori probability estimation

The Maximum a posteriori probability (MAP) estimation approach treats the high-resolution image and the observed low-resolution input images as two different stochastic processes [29]. The super-resolution reconstruction problem is regarded as a statistical estimation problem which can be solved by estimating a high-resolution image, using the maximum a posteriori probability of the low-resolution image sequence. The MAP estimator of an ideal high-resolution image x is acquired by maximising the a posteriori probability density function of a low-resolution image sequence $y_k(k = 1,2,n)$, described in equation (2.6).

$$P(x|y_1, y_2, ..., y_n) \tag{2.6}$$

2.2.3.4 Regularisation

In general, the super-resolution reconstruction technique can be described as an ill-posed problem. This is due to the insufficient availability of low-resolution observations and ill-conditioned blur operators inherent to sensor hardware [30]. However, procedures known as regularisation can be adopted to stabilise the inversion of such problems. This approach imposes a priori knowledge on the solution in order to achieve super-resolution. Two different approaches can be used, namely; deterministic and stochastic regularisation.

When using regularisation methods, the observation model discussed earlier is estimated with parameters from the registration process. The deterministic approach solves the inverse problem of the observation model by using prior knowledge of the solution and constrained least squares regularisation. Stochastic regularisation uses the a posteriori probability density function of the original image and estimates a high-resolution approximation, using Bayesian estimation [30].

2.3 COMPARISON OF SUPER-RESOLUTION ALGORITHMS

2.3.1 Interpolation based approaches

Each of the interpolation based approaches have their strengths and weaknesses. Non-uniform interpolation works very well when neighbouring pixel density differences are low. This makes it very easy to approximate missing pixel information in a region. However, edges are not very well-managed by this approach, resulting in a lack of fine detail in the high-resolution estimation.

The frequency domain approach is the simplest to implement and has the lowest computational requirements of all the approaches. However, it lacks the ability to accurately correlate global translational data, which could result in the degradation of the high-resolution approximation.

Super-resolution using zoom cue allows for images to be captured from the same perspective, eliminating the need for global translational data. However, the up-scaling process used during interpolation has the potential to degrade parts of the image resulting in a lack of fine detail. Interpolation approaches in general are not good at handling large amplification factors, exponentially increasing computational load when faced with much larger images.

Table 2.1 shows a comparison of all the interpolation algorithms with relation to their average peak signal to noise ratio (PSNR) and structural similarity (SSIM) to a set of original high-resolution scenes, containing multiple features. A more detailed examination of these metrics is presented in the performance metrics section of Chapter 4.

Table 2.1. Comparison of interpolation algorithms. Adapted from [31] ©2013 IEEE

Algorithm Name	Computation Time (s)	PSNR	SSIM
Non-uniform interpolation	8.57	22.40	0.83
Frequency domain	3.56	20.90	0.77
Zoom cue	6.50	21.54	0.78
Blur cue	4.53	20.33	0.73

2.3.2 Learning based approaches

Each of the learning based approaches discussed have their strengths and weaknesses. The Markov network model is capable of obtaining abundant high frequency information in comparison to other approaches, while performing better under magnification conditions. However, the training sample data information is very large and therefore susceptible to noise.

The neighbourhood embedding method, on the other hand, uses a much smaller training sample size, reducing its vulnerability to noise. However, its shortcoming lies in the difficulty of choosing an appropriate neighbourhood block size.

The sparse representation approach overcomes this problem, but suffers when the over complete dictionary chooses inappropriate image blocks. Random selection of an over complete dictionary can achieve satisfactory results when focusing on a specific domain of images, but when images are generalised, random selection can have negative effects on the reconstruction process.

Face hallucination is very successful in reconstructing facial images due to prior knowledge of facial parameters. However, it struggles when faced with distinct differences such as different species, facial expressions and wider age distributions.

Table 2.2 shows a comparison of all the discussed learning algorithms with relation to their average peak signal to noise ratio (PSNR) and structural similarity (SSIM) to a set of original high-resolution scenes containing multiple features. Values for face hallucination are obtained from images with only faces.

Table 2.2. Comparison of learning algorithms. Adapted from [32] ©2014 IEEE

Algorithm Name	Computation Time (s)	PSNR	SSIM
Markov network	17.45	27.28	0.82
Neighbourhood embedding	27.41	26.27	0.71
Sparse representation	41.78	26.54	0.75
Face hallucination	18.85	28.32	0.85

2.3.3 Reconstruction based approaches

Each of the reconstruction based approaches discussed have their strengths and weaknesses. The IBP estimation approach is computationally simple and easy to realise. It performs the best in regions where there are significant pixel density differences within a neighbourhood region. The choice of the back projection kernel has a major influence on the outcome of the problem. Because of the ill-posed nature of the problem, the solution to the IBP estimation approach is not unique. This makes it difficult to use a priori constraints and thus makes it harder to find a suitable back projection kernel.

The POCS estimation approach's strength lies with its flexibility when faced with complex spatial observation models. However, it requires heavy computational loads during the projection process. It has the ability to use non-linear constraints during the reconstruction process, thus preserving high frequency details of the image. The POCS approach relies heavily on motion estimation obtained during the registration process, thereby making it unsuitable for reducing heavy noise in an image.

Unlike POCS and IBP, the MAP estimation approach is able to produce a unique solution by introducing prior regularity constraints. The MAP approach is very good at removing noise in an image due to its high convergence stability. However, its ability to preserve edges in an image is much weaker than other methods, which could cause a loss of detail within the high-resolution estimation.

When using the regularisation approach, the quality of the reconstructed image is dependent on the design of the spatial geometry model. This means that it does not need to rely on statistical hypotheses or noise information of the original scene. However, increasing amplification coefficients make it difficult to obtain accurate sub-pixel translational motion information and can thus severely degrade the high-resolution estimation.

Table 2.3 shows a comparison of all the discussed reconstruction algorithms with relation to their average peak signal to noise ratio (PSNR) and structural similarity (SSIM) to a set of original high-resolution scenes, containing multiple features.

Table 2.3. Comparison of reconstruction algorithms. Adapted from [31] ©2013 IEEE

Algorithm Name	Computation Time (s)	PSNR	SSIM
IBP	28.82	27.68	0.81
POCS	42.86	27.55	0.80
MAP	60.79	26.98	0.80
Regularisation	29.19	25.80	0.79

2.4 FEATURE DETECTION FOR SUPER-RESOLUTION

Humans are able to acquire, process, understand and interpret information from a scene just by looking at it. The challenge emerges when a machine is required to acquire and understand this information in order to perform a certain task without the need for human intervention. It is therefore necessary to investigate areas of image processing which can help accomplish this task. One of the most important aspects of image processing is feature detection. Feature detection is the process of identifying certain areas of an image by defining feature types and by making local decisions concerning the boundaries of those features [33].

Feature detection is an actively advancing field of image processing and has many important applications in machine learning and pattern recognition. It can further be applied in image classification techniques such as feature extraction, feature encoding and classifier learning [34]. Given an input image, various image features can be extracted to identify features such as colour, illumination, gradient and intensity, depending on the application. Image feature detection is used in many areas of computer vision. Medical professions make use of feature detection for more educated diagnoses, robotics technology requires the identification of objects in the surroundings to make decisions and satellite surveillance systems use feature detection to accurately identify areas of importance [35].

A basic image feature, described by pixel intensity, can be broken down into three distinct categories. The first is a collection of pixel intensities which represents the boundaries of objects [36]. These regions experience a quick change in pixel intensity from one side of the region to the other. The variability of these intensities is not entirely random, since the boundaries of objects are often expected and can be estimated. These are typically referred to as the edges of an image. The second type is

a collection of pixel intensities which represent the areas of an image with little to no variability in pixel intensity [37]. These regions share similar gradient and colour values and can be classified as smooth regions. The third type is a collection of pixels with rapidly changing pixel intensities, colors or gradients [38]. The variability of the intensities of these regions are unpredictable and occur within a localised area. These regions are classified as textures since they often represent surfaces of objects with different colours or gradients.

Super-resolution technology can also use image feature detection to improve algorithm complexity and resolving quality in images. The three categories of super-resolution algorithms perform better on certain areas of an image [9]: interpolation approaches surpass other approaches when implemented on smooth areas of an image; learning based approaches have the greatest effect on the edges of an image; and reconstruction based approaches are the most beneficial when used on textured regions of an image. Feature detection can be used to identify these regions and apply the appropriate super-resolution technique to ensure that maximum resolving quality is achieved with minimal computational load.

2.5 CONCLUSION

In this chapter, a study of existing super-resolution approaches is performed in the context of understanding the need to overcome the limitations of high-resolution imaging systems. Furthermore, each algorithm in the different categories of approaches is compared to understand the advantages and disadvantages of each. Finally, the concept of feature detection is discussed in the context of its use in super-resolution algorithms to create higher resolution approximations.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

In this chapter the methods for designing and implementing a hybrid multiple-image super-resolution system are discussed. This involves an investigation into the super-resolution algorithms to be used in the development of the system. Furthermore, an exploration into methods with which to combine the algorithms into a hybrid system is discussed. This also requires additional investigation of feature detection algorithms.

3.2 INTERPOLATION ALGORITHM

When choosing which interpolation algorithm to use for the hybrid system, it is necessary to consider complexity and time constraints, as well as overall performance. Referring to table 2.1, the non-uniform interpolation method provides the best structural similarity to the original image. This increase in accuracy is well worth the additional complexity of the algorithm. Unfortunately, the non-uniform interpolation algorithm does not manage edges very well, resulting in a lack of fine detail in the high-resolution estimation. However, in the hybrid system, the interpolation algorithm will focus on enhancing resolving quality in smooth regions of an image (areas where the average pixel intensity does not fluctuate rapidly) and will allow the learning algorithm to handle high frequency edge information.

3.2.1 Registration

The basic premise for increasing spatial resolution with interpolation algorithms is the availability of multiple low-resolution images obtained from the same scene. However, the images obtained need to have a sub-pixel shift between them, which ensures that each image contains new information that can be used to construct a high-resolution image [9]. The low-resolution images must represent different perspectives of the same scene and must be sub-sampled, otherwise the images will all contain the same information. Figure 3.1 describes a sub-pixel shift between multiple images.

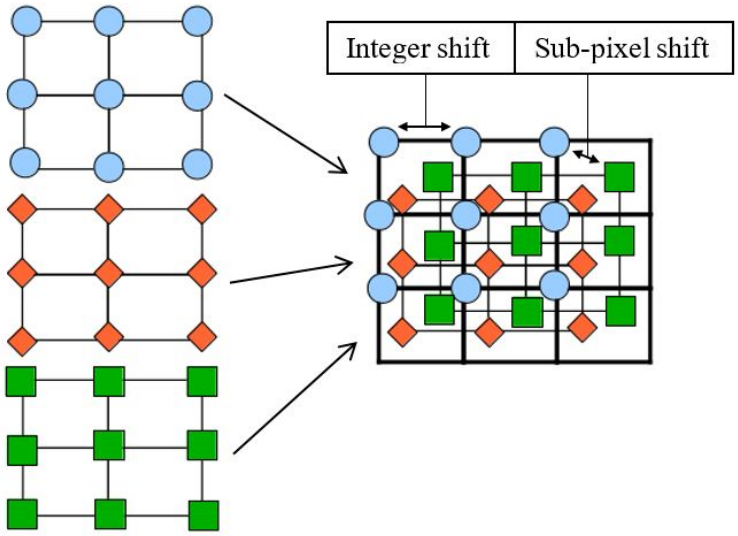

Figure 3.1. Low-resolution images with sub-pixel shifts

Once the desired number of low-resolution observations has been obtained, they need to be registered. Image registration can be described as a process in which multiple images are aligned geometrically [39]. One image is denoted as the reference image while the additional images are regarded as sensed images.

Image registration techniques can be categorised into three main groups, based on the different methods with which low-resolution observations are captured [40]: either from different points of view, at different times or by different sensors. In the hybrid system, multiple-image super-resolution will be simulated by acquiring sensed images from a reference image. This process is described in figure 3.2 and explained by equation (3.1).

In equation (3.1), S_k is the kth sensed image, R is the reference low-resolution observation and r_k is the translational displacement parameter associated with the kth sensed image.

$$(S_k^x; S_k^x) = (R^x; R^y) + (r_k^x; r_k^x) \tag{3.1}$$

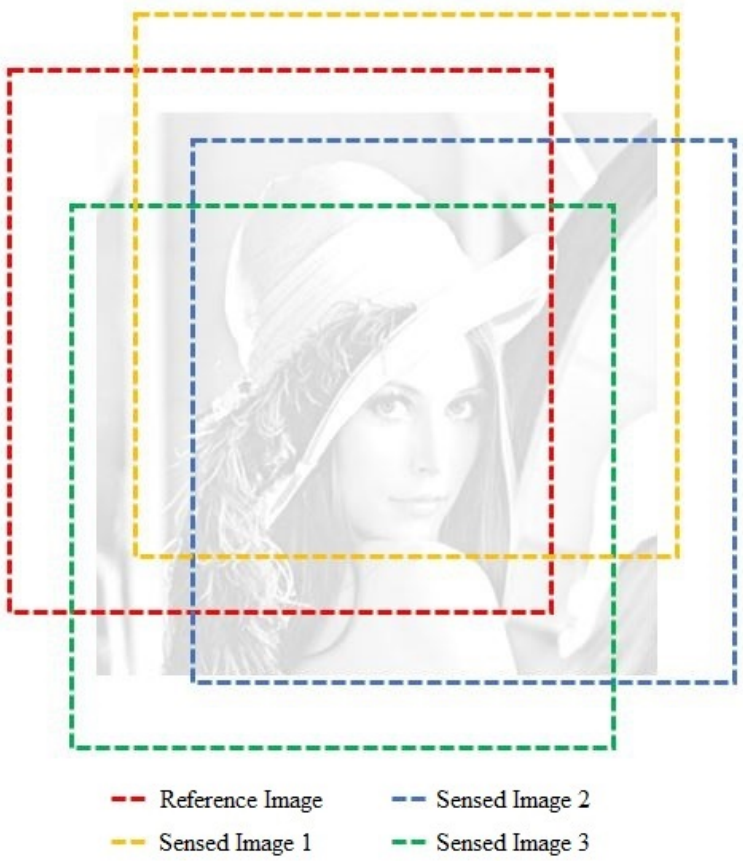

Figure 3.2. Image registration

The registered images are then placed on a semi-uniform grid to form a composite image. Consider n observed images of size $s_x s_y$. In order to fit all the necessary information from these images onto the new high-resolution grid, a single pixel in the high-resolution grid must represent n by n pixels of the observed images. Thus the size of the new composite grid will be $n^2 s_x s_y$. Figure 3.3 illustrates an array of four sensed images, which will place its pixels onto the high-resolution grid with the necessary sub-pixel shifts. The resolution of the composite grid is described in equation (3.2) in the case where n images are registered.

$$HR_{res} = n^2 LR_{res} \tag{3.2}$$

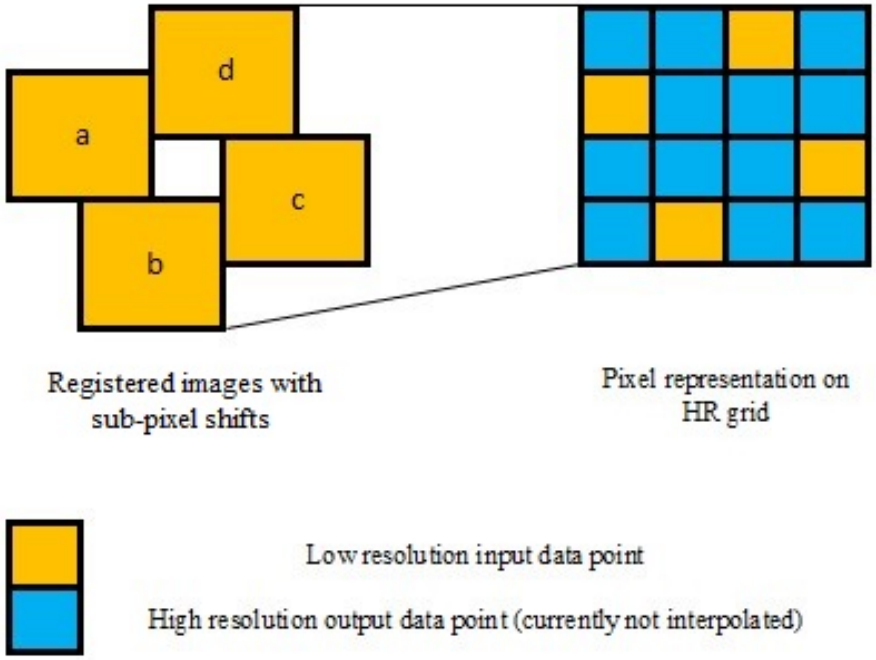

Figure 3.3. Registration pattern for four sensed images

3.2.2 Interpolation models

Once the registered observations have been combined into a composite image, an interpolation model will be responsible for constructing new data points given the range of the discrete set of known data points. Table 3.1 describes and compares different interpolation models.

The spline and bi-cubic models perform similarly at lower degree polynomials. Therefore, using the more intuitive bi-cubic model is more justified than implementing the more resource dependant spline interpolation for the purposes of the hybrid system. However, restricting the model to lower polynomials may result in a lack of fine detail in the high-resolution approximation. This problem can be solved by incorporating a weighted nearest neighbour approach, which uses a model in which the value assigned to the interpolant is the average of weighted surrounding data points [41]. This weight is determined by the distance from the interpolant. Combining the weighted nearest neighbour method with polynomial interpolation, will produce an efficient interpolation model in which the surrounding area will be defined by a third order polynomial.

Table 3.1. Comparison of interpolation models

Criteria	Nearest Neighbour	Linear	Polynomial	Spline
Complexity	This is the simplest interpolation model.	Still relatively simple.	This model is much more complex.	This model builds on the polynomial model, adding more complexity.
Description	Assigns the value of a data point to the value of its nearest neighbour.	Assigns the value of a data point as the average of the two closest data points.	A generalisation of the linear model, which replaces the interpolant with a polynomial of a higher degree.	Uses a low degree polynomial in each interval and chooses polynomial pieces such that they fit smoothly.
Accuracy	Very low accuracy.	Low accuracy.	Accuracy depends on the degree of the interpolant. Generally, the third degree is the most efficient before experiencing diminishing returns.	Incurs slightly higher accuracy than the polynomial model and performs better than very high degree polynomials.
Speed	Fast and easy to implement.	Fast and easy to implement.	Relatively fast, intuitive and easy to implement.	Highest computational load, difficult to implement.

3.2.2.1 Continuous function approximation

Digital images are samples of a continuous scene and are characteristically defined by a discrete set of data points on a uniform grid. To increase the accuracy of a function's ability to represent the continuous scene, it becomes necessary to reconstruct the original continuous function. This process is

known as interpolation. Before interpolation can be performed, it must be assumed that the image is band-limited and smooth in order to derive a solution.

Given a band-limited continuous function representing the observed scene $g(x)$, $x \in \mathbb{R}$, $f_I(p)$ can be defined as one of the discrete observed input images sampled on a set of discrete points P, described in equation (3.3).

$$f_I(p) = g(p) \tag{3.3}$$

The role of interpolation then is to find an estimate of the scene, $\tilde{g}(x)$ from the observed image, $f_I(p)$. The interpolation function, denoted by H_p with associated input point p, can be applied as a filtering function using spatial convolution. The estimated continuous scene can then be described by a linear combination of input values, shown by equation (3.4).

$$\tilde{g}(x) = \sum_p f_I(p) H_p(x-p), \quad p \in P \tag{3.4}$$

When re-sampling a digital image, it is not necessary to reconstruct the entire continuous surface. Only the areas which contain the values at the new sample locations are used for re-sampling. The newly constructed output image, denoted by $f_O(q)$, is sampled at new locations $q \in Q$ and can be described by equation (3.5).

$$f_O(q) = \tilde{g}(q) \tag{3.5}$$

Understanding that the estimated continuous scene can be described by a linear combination of input values, equation (3.5) translates to equation (3.6).

$$f_O(q) = \sum_p f_I(p) H_p(q-p), \quad q \in Q, \quad p \in P \tag{3.6}$$

3.2.2.2 Weighted region of support

During the interpolation process, not all input values from the observed images are used to interpolate each desired point. Instead, a small neighbourhood of input points is used to interpolate the data at the desired point. This is known as the region of support, a set of points within a predetermined distance from the desired output point. This region differs, depending on the degree of the polynomial used, as seen in figure 3.4.

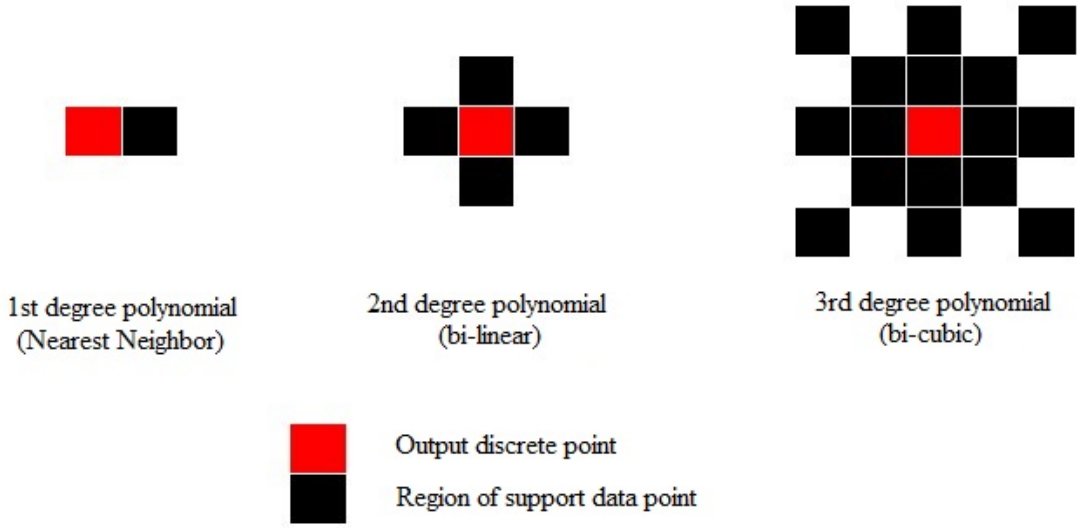

Figure 3.4. Regions of support

Equation (3.6) can now be simplified to only use the region of support to calculate the value at the desired point, where the predetermined distance from the desired point is represented by w. This is shown in equation (3.7).

$$f_O(q) = \sum_p f_I(p) H_p(q-p), \quad q \in Q, \quad p \in P_{q,w} \tag{3.7}$$

It is also necessary to consider how far away neighbouring data values are from the interpolant and adjust their effect on the interpolant accordingly. For each point on the composite image grid that must be filled, all points in the region of support are located and given weights inversely proportional to their distance from the desired point on the grid. This process is illustrated in figure 3.5. The values of each weight are calculated using equation (3.8).

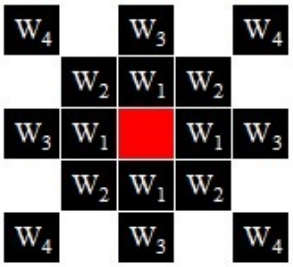

Figure 3.5. Weighted region of support

$$W_n = \frac{1}{\sqrt{(x_n - x_o)^2 + (y_n - y_o)^2}} \tag{3.8}$$

o and *n*, denoting the desired output point and the known data point respectively. The weighted data points are then averaged to form the interpolant at the desired point.

3.3 LEARNING ALGORITHM

When choosing which learning algorithm to use for the hybrid system, it is necessary to consider complexity and time constraints, as well as overall performance. The learning algorithm must also be able to accurately reconstruct high frequency edge information in the high-resolution approximation. Referring to table 2.2, the Markov network learning algorithm provides the best structural similarity to the original image. This increase in accuracy is made more appealing by the lower complexity of the algorithm when compared to others. Unfortunately, the Markov network learning algorithm is susceptible to noise, resulting in the appearance of undesirable artifacts in the high-resolution estimation. However, in the hybrid system, the learning algorithm will focus on enhancing the resolving quality of the edges of an image (areas which divide regions with different pixel densities) and will allow the reconstruction algorithm to handle high levels of noise.

3.3.1 Training set generation

Learning based super-resolution algorithms begin with the generation of a training set. This is done by collecting high-resolution images of the same desired output resolution, then degrading each image in a manner corresponding to the degradation that the input images would have experienced. The images

are blurred and sub-sampled to create a low-resolution image, containing a fraction of the number of original pixels in each dimension [42]. The newly created low-resolution images are enlarged, using a nearest neighbour interpolation, generating images of the desired number of pixels that lacks high-resolution detail. To develop the training set, the differences between the nearest neighbour interpolation and the original high-resolution image need to be stored.

The next step in the process is to store a high-resolution patch corresponding to every possible low-resolution patch. A high-resolution patch is typically made up of a 5 by 5 block of pixels, while a low-resolution patch is a 7 by 7 block of pixels [17]. Considering the size of these image blocks, this is a large amount of data to store and so before this is done, each image is pre-processed to remove variability and make the training set as generally applicable as possible. The most important information when predicting extra details is the highest spatial frequency components of the low-resolution images. Therefore, lower frequency components can be filtered out so that storage space for example patches is not wasted on low frequency data. The relationship between high-resolution and low-resolution image patches is essentially independent from contrast information of the local image [43] and so local contrast normalisation can be applied during the training process and contrast can be reapplied when the final high-resolution image is being reconstructed.

Figure 3.6 shows an example of low-pass image blocks and their corresponding high-resolution image blocks, while figure 3.7 describes the steps involved in the training set generation.

Figure 3.6. Training set image blocks. (a) the low-resolution 7 by 7 image blocks and (b) the corresponding high-resolution 5 by 5 image blocks.

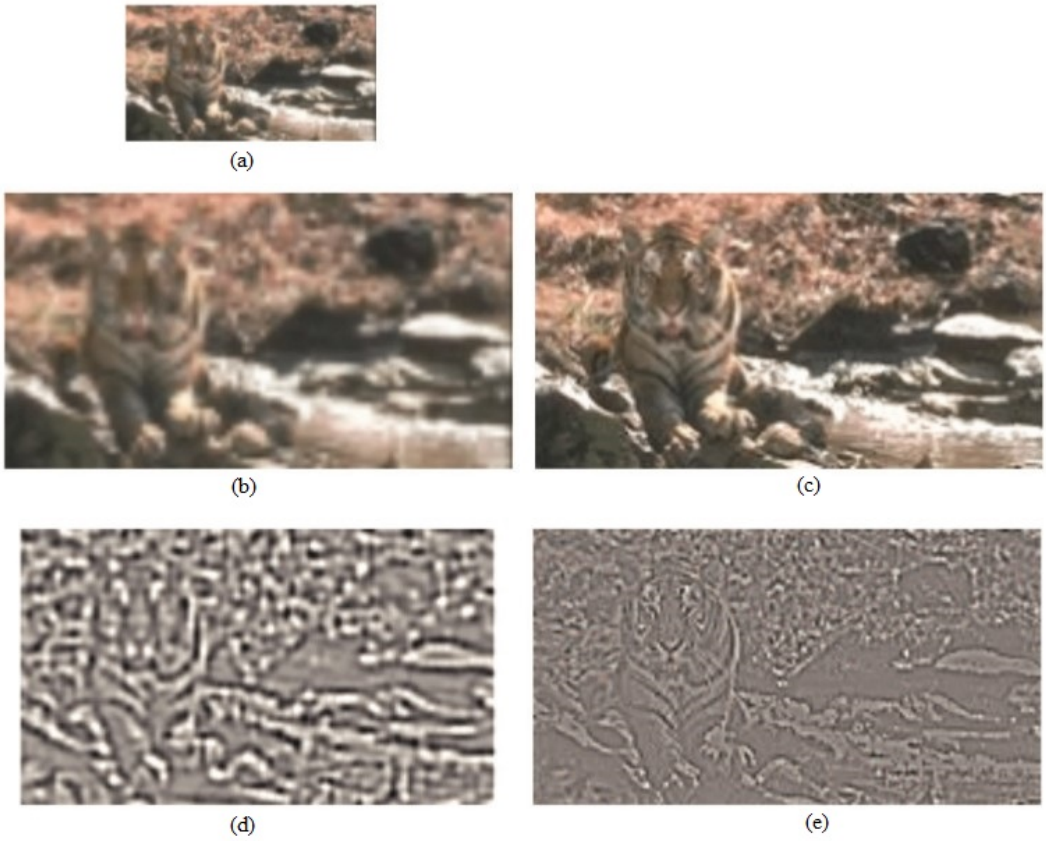

Figure 3.7. Training set generation. (c) is the original high-resolution image. (a) is the down-sampled low-resolution image. (b) is the nearest neighbour interpolation of the low-resolution image. (d) and (e) are the high-pass filtered and contrast normalized versions of (b) and (c) respectively.

3.3.2 The Markov network

The Markov network algorithm has many uses in image processing and can be applied to model the spatial relationships between patches [44]. Figure 3.8 below describes the Markov network model.

The circles in figure 3.8 represent network nodes. The lines indicate statistical dependencies between nodes. Observation nodes, denoted by y, represent the low-resolution image patches. The 16 closest examples to each input patch are selected as the different states of the hidden nodes x, which need to be estimated. In this network, the probability of any given choice of a high-resolution patch for each node is proportional to the product of all sets of compatibility matrices ω, relating the possible states

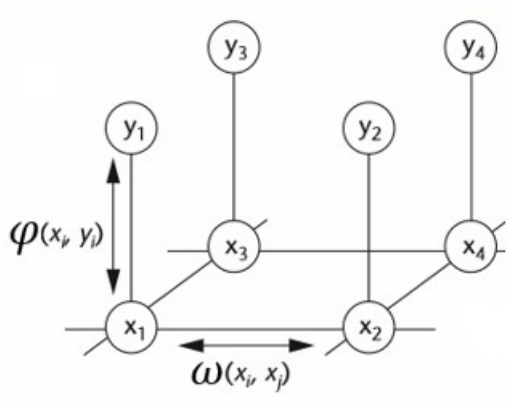

Figure 3.8. Markov network model

of each pair of neighbouring hidden nodes and vectors φ relating each observation to the underlying hidden states. This process can best be described by equation (3.9).

$$P(x|y) = \frac{1}{Z}\prod_{ij}\omega_{ij}(x_i,x_j)\prod_{i}\varphi_i(x_i,y_i) \tag{3.9}$$

y is the input low-resolution image, x is the high-resolution image being estimated, Z is a normalisation parameter, x_i and x_j are locally adjacent image blocks within the high-resolution image and y_i is the low-resolution image block corresponding to x_i.

The input images nodes are sampled so that the high-resolution patches overlap each other by at least one pixel. In the overlap region, the pixel values of compatible neighbouring patches should theoretically correspond. The sum of squared differences between patch candidates x_i and x_j in their overlap regions is measured at nodes i and j, thus creating the compatibility matrix between nodes i and j, described in equation (3.10).

$$\omega_{ij}(x_i,x_j) = \exp\left(\frac{d_{ij}(x_i,x_j)}{2\sigma^2}\right) \tag{3.10}$$

σ is a noise parameter and $d_{ij}(x_i,x_j)$ is the sum of squared differences between patch candidates x_i and x_j. The same process is used on differences between the observed low-resolution image patch y_i and the candidate low-resolution patch found in the training set x_i, to specify the Markov network

compatibility function $\varphi_{ij}(x_i, y)$ [15]. The chosen optimal high-resolution patch at each node is the collection that maximises the Markov network model's probability.

3.3.3 One pass algorithm

Finding an exact solution using the Markov network model takes a heavy computational load and a very large amount of time. The one pass algorithm is an approximate solution of the Markov network approach, which uses the same local relationship information. It is able to achieve results similar to the iterative solution and with much faster convergence stability [17]. High-resolution patch compatibility is only computed for neighbouring high-resolution patches that have already been selected. Typically, these are the patches above and to the left of the input patch [17]. If training data is structured in an appropriate way, the local low-resolution image data can be matched and high-resolution patch candidates can be selected in only a single operation.

The one pass algorithm generates the missing high frequency content of an image as a sequence of predictions from local image information. The input image is sub-divided into low frequency patches and then traversed level by level. At each step, a high frequency patch is selected from the training set by a nearest neighbour search, based on the local low frequency details and adjacent high frequency patches that were previously determined [42]. Similar to the interpolation algorithm discussed earlier, the sub-sampled image is enlarged to the size of the desired high-resolution image, then missing pixel information is estimated, using the data from the training set and the local image data.

3.3.3.1 Prediction

Given the high frequency data of the input image, the learning algorithm predicts the frequencies missing from the enlarged image. Therefore, the output of the algorithm is the sum of its input and the predicted high frequency information. High frequency data for an *N*x*N* pixel image is done row by row and each prediction is based on two requirements:

1. The high frequency patch that is selected, should come from a location in the training images that has a similar low frequency appearance.

2. The high frequency prediction should correspond with the edges of the overlapping pixels of its nearest neighbour, in order to ensure that predictions are compatible with neighbouring patches.

The first requirement is met by extracting an MxM low frequency patch from the input image, then by attempting to find a match in the training set, which is made up of low frequency and high frequency patch pairs. The high frequency data, which was previously predicted, is also used to select the best patch pair. The second requirement is met by overlapping potential predictions at the borders of the estimated patch. The relationship between these two requirements can cause the prediction time of the algorithm to take longer than desired and so a weighing factor is introduced to adjust the relative importance between matching low frequency patches and matching the neighbouring high frequency patch borders.

3.3.3.2 Search algorithm

Matches are searched for by using a tree based approximate nearest neighbour search. The tree is built by recursively splitting the training set in the direction of higher variation. A balanced tree is maintained by dividing the training data in half at each step. A best first search provides an appropriate speed and quality trade-off. Even though this approach may search through more branches, finding a better match is worth the time and computational resources.

3.4 RECONSTRUCTION ALGORITHM

When choosing which reconstruction algorithm to use for the hybrid system, it is necessary to consider complexity and time constraints, as well as overall performance. The reconstruction algorithm must also be able to accurately reconstruct high-resolution approximations from observations with large amounts of noise. Referring to table 2.3, IBP estimation provides the best structural similarity to the original image. This increase in accuracy is made more appealing by the lower complexity of the algorithm when compared to others. In the hybrid system, the reconstruction algorithm will focus on enhancing the resolving quality of textured regions of an image (areas where pixel densities change rapidly within a small neighbourhood).

3.4.1 The observation model

As discussed earlier, the first step to comprehensively analyse the super-resolution problem using the reconstruction approach, is to formulate an observation model to relate the original high-resolution image to the captured low-resolution images [9]. The original high-resolution image is denoted as x, the ideal image which has not yet suffered from degradation and which has been sampled at or above the Nyquist rate from a scene which is assumed to be band limited. Down-sampling factors can be represented by the parameters L_x and L_y respectively for the horizontal and vertical directions. The size of the original high-resolution image can, therefore, be represented as L_xN_x x L_yN_y and the observed low-resolution images are all of size N_x x N_y. It is assumed that x remains constant during the acquisition process and k low-resolution images are captured. The low-resolution images result from the high-resolution image x, suffering from degradation effects such as blurring, warping, sub-sampling and noise operators. Let W_k represent a warp matrix, B_k represent a blur matrix, S represent a sub-sampling matrix and n_k represent a noise vector, which all degrade the kth low-resolution image y_k. We can, therefore, develop an observation model represented by equation (3.11).

$$y_k = SB_kW_kx + n_k \tag{3.11}$$

Therefore, the reconstruction algorithm must remove these degradation effects by estimating sub-pixel displacement and noise variance in low-resolution images through expectation maximums, in order to acquire the desired high-resolution output image [23].

3.4.2 IBP estimation

Figure 3.9 shows the back-bone behind the IBP estimation approach. The algorithm begins with the input low-resolution observation. Interpolation algorithms are applied to the input image to generate an initial estimation of the high-resolution image. The high-resolution estimation is then down-sampled and degraded, in order to generate a simulated low-resolution image. This process can be described by equation (3.12).

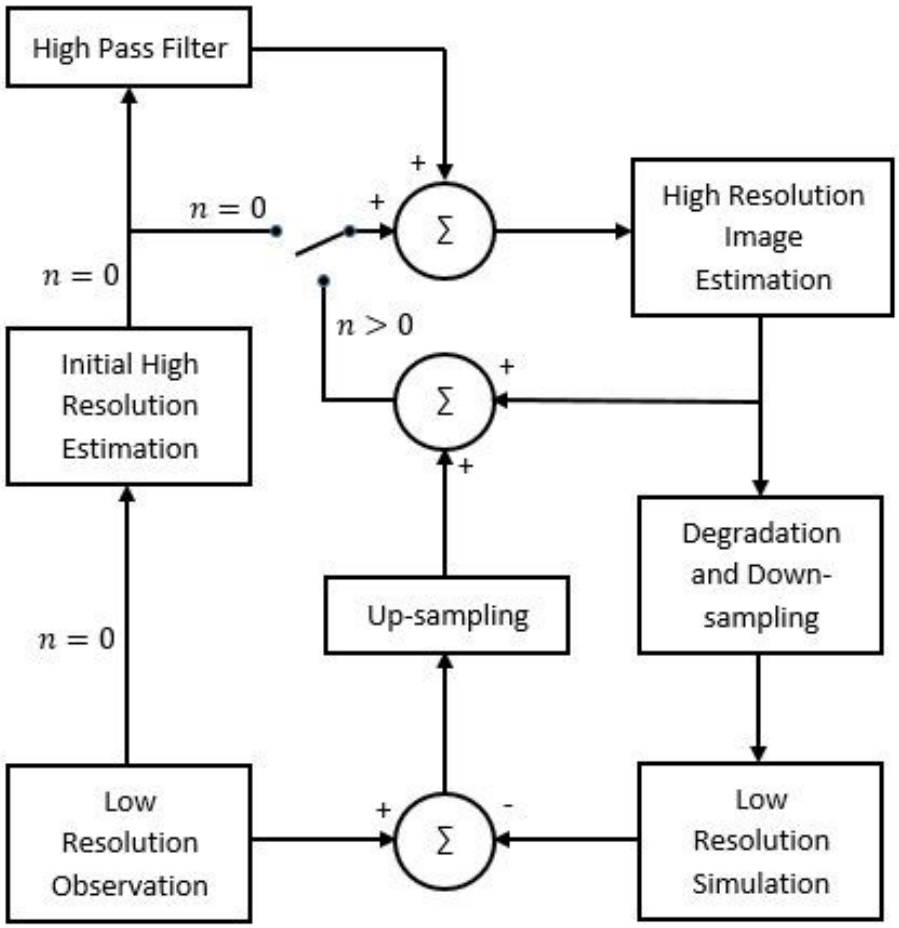

Figure 3.9. IBP block diagram. Adapted from [45] ©2012 IEEE

$$y^{n+1} = (x^n W) D_S^2 \tag{3.12}$$

y^{n+1} is the simulated low-resolution image for the $(n+1)$th iteration, W is the degradation factor and D_S^2 is a down-sampling function. The simulated low-resolution image is compared to the original low-resolution input observation by subtraction to obtain an error. The error must be up-sampled in order to conform to size constraints of the high-resolution estimation. The error calculation can be described by equation (3.13).

$$x_e = (y - y^n) U_S^2 \tag{3.13}$$

y is the initial input low-resolution observation, y^n is the simulated low-resolution image of the nth iteration, U_S^2 is an up-sampling function and x_e is the error. The original high-resolution estimation is then filtered for edge projection and then added to the error, which is then projected onto the high-resolution estimation. Using the block diagram above, the estimation of the next high-resolution image can be described by equation (3.14).

$$x^{n+1} = x^n + x_e + HPF(x^0) \tag{3.14}$$

This iterative process is repeated for a predefined number of iterations or until a satisfactory error between the low-resolution input image and the low-resolution simulated image is met.

In order to simulate the generation of a low-resolution image, the high-resolution estimation needs to be down-sampled and degraded. During the down-sampling procedure the sampling frequency is decreased, which generates distortions in the high frequency components of the image, causing aliasing problems [26]. It may, therefore, be necessary to further filter the high pass filtered high-resolution estimation with a Gaussian filter in order to eliminate the distortions for the down-sampling procedure.

3.5 FEATURE DETECTION FOR SUPER-RESOLUTION

In order to implement feature detection, it is necessary to first identify which features of an observation of a scene need to be identified. Image edges are the most basic features of an image. Edges are a group of connected pixels, which together form a boundary line between two distinct regions of an image [46]. These are areas of an image where an abrupt change in pixel density occurs, for example, the boundary between a person's face and the background of an image. Blobs are regions of an image which are either darker or brighter than the neighbourhood of image pixels and are often regions of the same colour [37]. These are areas of an image which are typically separated by edges and often share the same pixel intensity values. These two types of features can be used to identify the smooth areas, edges and textured regions on an image.

3.5.1 Edge detection

Edge detection has many applications in digital image processing, including its use in pattern recognition. An edge can be identified by significant local changes of pixel intensity in an image and typically divides two distinct regions of an image. The Sobel edge detection algorithm is the simplest edge detection algorithm to implement. It involves convolving the image with an integer value filter [46]. It boasts the advantage of simplicity and is computationally inexpensive. However, the Sobel operator does not provide enough information about an edge to be useful for the purposes of applying it to super-resolution. The information obtained from applying the Sobel operator can be used to further identify edges by using Canny edge detection [36]. While slightly more complex, the Canny edge detection algorithm is still easy to understand and simple to implement.

3.5.1.1 Sobel edge detection

The Sobel Operator finds the approximate derivative in the horizontal and vertical directions [46]. Equations (3.15) and (3.16) are used to calculate the gradient at each data point.

$$\begin{aligned} G_x = &[f(x+1,y+1)+2f(x+1,y)+f(x+1,y+1)] \\ &-[f(x+1,y-1)+2f(x-1,y)+f(x-1,y+1)] \end{aligned} \tag{3.15}$$

$$\begin{aligned} G_y = &[f(x-1,y+1)+2f(x,y+1)+f(x+1,y+1)] \\ &-[f(x-1,y-1)+2f(x,y-1)+f(x+1,y-1)] \end{aligned} \tag{3.16}$$

G_x is the horizontal gradient and G_y is the vertical gradient. The net gradient G is then calculated according to equation (3.17).

$$G = \sqrt{G_x^2 + G_y^2} \tag{3.17}$$

The filter kernels for the horizontal and vertical gradients can be seen in equations (3.18) and (3.19).

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.18}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.19}$$

The gradient kernels G_x and G_y are convolved with the image to calculate the horizontal and vertical gradients of each pixel. Equation (3.17) is then used to find the gradient magnitude of each pixel. This will generate an image which shows only the high frequency edge information of the original image.

3.5.1.2 Canny edge detection

The Canny edge detection algorithm aims to reduce the number of signal feedbacks to a particular edge [36]. It focuses on preserving useful information and filtering out unessential information detected by the Sobel operator. It does this by performing two important processes, namely: identifying local maximums and hysteresis thresholding.

3.5.1.3 Local maximum identification

The process of identifying local maximums in a Sobel operator response is done to reduce the size of an edge to a single pixel so that the locations of the edges can be correctly identified within an image [36]. Identifying a local maximum requires that the orientation of the gradient is known. This information can be obtained from the output of the Sobel operator by using the horizontal and vertical gradients. The orientation is calculated, using equation (3.20).

$$\theta = \tan^{-1} \frac{G_y}{G_x} \tag{3.20}$$

Once the orientation is known, it can be used to identify the direction in which the edge is travelling. Depending on the accuracy that is required, the number of directions that the edge can travel, may be adjusted. For the purposes of simplicity, it is easiest to use eight compass points when defining the direction of the edge. Table 3.2 shows the four distinct directions of travel for an edge used in this design.

Table 3.2. Edge orientation

Edge orientation	**Rounded edge orientation**	**Direction of travel**	**Neighbour pixels to compare against**
22.5° - 337.5° and 157.5° - 202.5°	0°	West to east	North and south
22.5° - 67.5° and 202.5° - 247.5°	45°	South-west to north-east	North-west and south-east
67.5° - 112.5° and 247.5° - 292.5°	90°	North to south	East and west
112.5° - 157.5° and 292.5° - 337.5°	135°	North-west to south-east	South-west and north-east

Table 3.2 also describes which neighbour pixels the current pixel needs to be compared to in order to decide whether it is a local maximum. If the gradient magnitude of the current pixel is greater than that of its appropriate neighbours, then the current pixel's gradient magnitude is enlarged while the gradient magnitudes of the neighbouring pixels are suppressed. This allows the edge to only be defined by the one pixel instead of a larger collection of pixels. Figure 3.10 depicts the expected differences in the one dimensional responses of the Sobel and Canny operators when local maximums are found in an image.

Figure 3.10 shows that the initial convolution of the Sobel operator returns a gradient response, which spans over multiple pixels. This response is then limited to a single area by using local maximums.

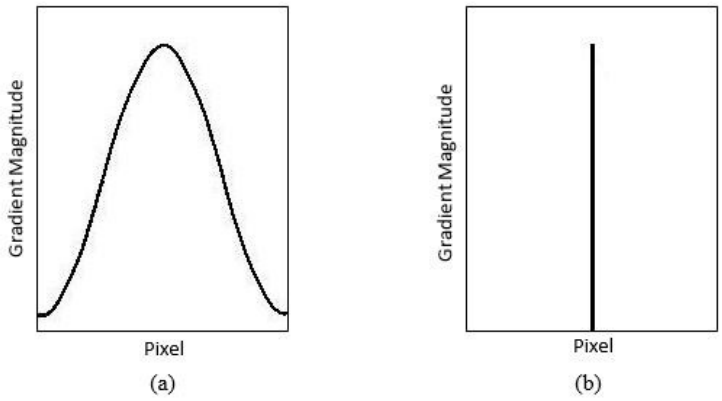

Figure 3.10. One dimensional response of (a) Sobel edge detection and (b) Canny edge detection

3.5.1.4 Hysteresis thresholding

Hysteresis thresholding is the process of identifying which edges found by the Sobel operator are useful and which contain unnecessary information. This is done by selecting two thresholds with which the image is filtered [47]. The first threshold (the lower bound) describes the gradient responses of pixels which are considered as unnecessary information. Any gradient values found below this threshold are suppressed or removed. The second threshold (the upper bound) describes the gradient responses of pixels, which are considered as useful information. Any gradient values found above this threshold are kept or enlarged. The areas between these two thresholds belong to the gradient values of pixels which are not low enough to be removed and not high enough to be kept. It is not yet known whether these values are useful or not. However, through pixel traversal, it can be determined whether these gradients are connected to useful information. If this is the case, these gradient values are preserved because they could contain useful information from a high gradient source. If this is not the case, these gradients can be removed. Figure 3.11 depicts an example of a one dimensional response when the two thresholds are applied.

Figure 3.11 shows how the red shaded area beneath the lower threshold will be removed because it is considered as unnecessary information. The green shaded area is above the upper threshold and is therefore considered as useful information, which will be preserved. The yellow shaded areas are found between the two thresholds and are also found to be connected to a gradient above the upper bound. Therefore they are also preserved.

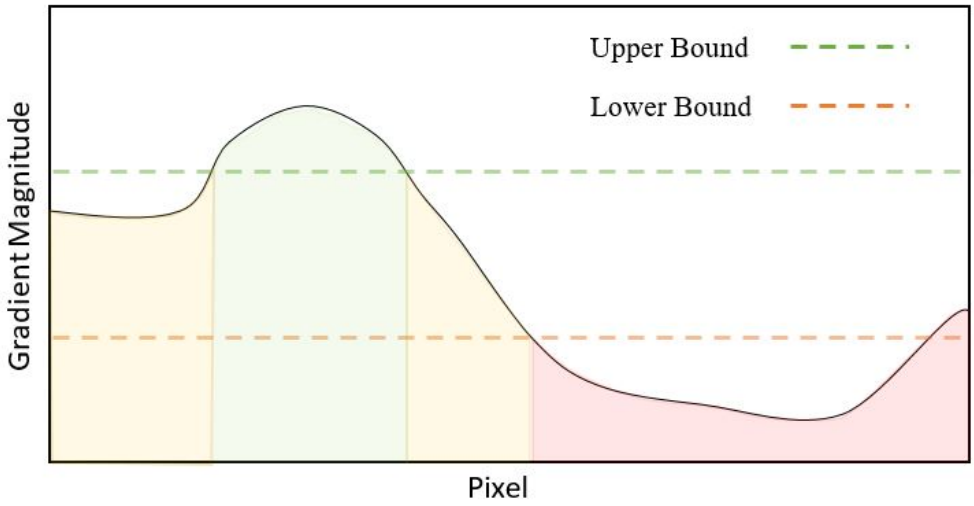

Figure 3.11. Hysteresis thresholding applied to a one dimensional response

3.5.2 Smooth region detection

Smooth regions can easily be identified by observing the pixel intensities and gradients of a neighbourhood of pixels [38]. The algorithm used to detect these types of blobs is a simple one to implement. First, a pixel is chosen to be examined. The pixel's intensity and gradient are compared to its eight neighbours. If a similarity is found, then the pixels with similarities are flagged as 'smooth'. Next, all pixels within that neighbourhood flagged as 'smooth', are compared to their eight neighbours. The process iterates in this way until no similarities are found within a neighbourhood of pixels, in which case the next pixel not yet flagged, is chosen to be examined. This process can be seen in practice in Figure 3.12.

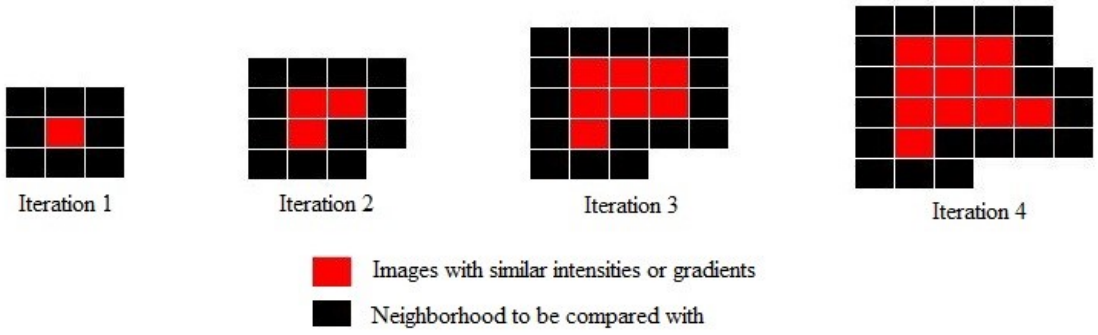

Figure 3.12. Smooth region detection

The red blocks represent pixels with similar pixel intensities. In the first iteration, it is found that the northern and north-easterly neighbours have similar intensities. The neighbourhood of these pixels is then examined in the second iteration to find further similarities. In this way, if sharp changes in pixel intensity are found, then the iteration stops and moves to the next region. This method can be combined with the edge detection algorithm to identify areas where sharp changes in intensity is known to occur, thus preventing the detector from examining pixels which are already known to be edges.

3.5.3 Texture detection

Given an input image, smooth regions are typically separated by the edges of the image and objects are surrounded by an edge boundary [34]. These characteristics of an edge allow us to classify pixels with high gradients as edges. However, a sharp change in pixel intensity could represent something other than an edge. Textures also share this trait. However, unlike an edge, the rapid change in pixel intensity does not separate smooth regions and is not the boundary of an object. The large gradient values in these regions are unpredictable and that can be used to classify them as textures.

Within a neighbourhood of pixel gradients, if a large change in pixel intensity occurs, then it can either be classified as a texture or an edge. Given that the Canny edge detection algorithm thins the edge values to a single pixel [36], the neighbourhood surrounding a pixel gradient can be examined to identify whether the edge is following a certain direction, or if the rapid changes are random or unpredictable. An example of this can be seen in Figure 3.13.

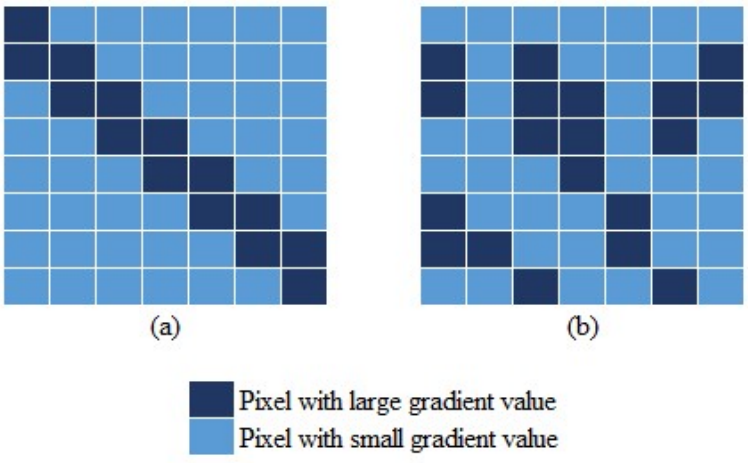

Figure 3.13. Pixel gradients. (a) Example of an edge and (b) example of a texture

The direction of an edge is predictable and typically separates two regions with smaller gradient values. If a neighbourhood of pixels with a number of large intensity values is found, then it is compared to different edge masks. If it is not similar enough to the mask, then it is considered a texture. An example of the north-west to south-east edge mask is shown in equation (3.21). In order to perform this operation, each gradient above a certain threshold must be enlarged and all other gradients must be suppressed.

$$E_{NW-SE} = \begin{bmatrix} 255 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 255 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 255 \end{bmatrix} \tag{3.21}$$

3.6 HYBRID SUPER-RESOLUTION SYSTEM

The hybrid system will use feature detection to combine the advantages of the three super-resolution algorithms that have been discussed. The flow diagram of the functionality of the final system is depicted in figure 3.14.

The system will begin by taking multiple low-resolution images as input, generated by down-sampling a high-resolution image, which will be used as a reference to measure the performance. It will then perform feature detection in order to determine the prominent features of the input images. Depending on the presence of different features, the hybrid system will choose which super-resolution algorithms to use in the construction of the final output image. In the case where all three features have similar proportions, the system will use all three algorithms. However, in the case where only two or less features are dominant, the system will decide to only use two or less super-resolution algorithms. The different cases can be seen in table 3.3. This is done to manage the complexity of the system and ensure that processing time is not wasted on insignificant information. The appropriate features are then extracted from each algorithm and combined to create the final output image.

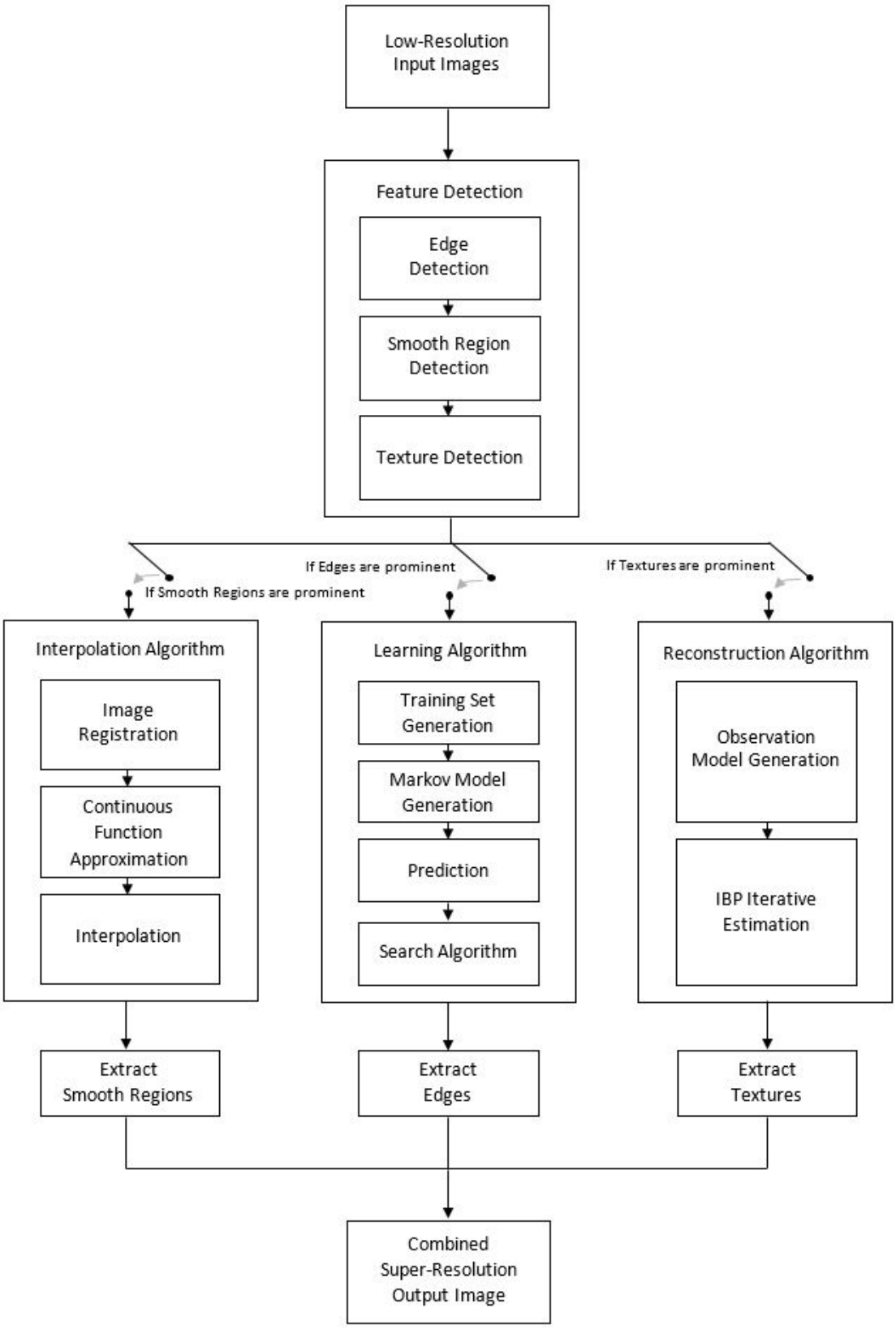

Figure 3.14. Hybrid system flow diagram

Table 3.3. Feature detection cases

Case No.	Edges	Smooth	Textures	Algorithms
1		✓		Interpolation
2	✓			Learning
3			✓	Reconstruction
4	✓	✓		Interpolation and Learning
5		✓	✓	Interpolation and Reconstruction
6	✓		✓	Learning and Reconstruction
7	✓	✓	✓	Interpolation, Learning and Reconstruction

3.6.1 Data set creation

In order for the hybrid system to be tested accurately, a data set of images with the appropriate features described in table 3.3 needs to be created. This is done by taking a large data set of 200 images pulled from standard test image databases including *Digital Image Processing, 3rd ed, by Gonzalez and Woods.* and *Digital Image Processing Using MATLAB, 2nd ed. by Gonzalez, Woods, and Eddins.* and categorising them, using the feature detection algorithm. Table 3.4 shows the different conditions that can be met for each classification, in which each feature's proportions are compared with one another, either being approximately similar or larger than a combination of features. Each image is then classified according to these criteria and added to the appropriate data set. For example, an image which has more smooth regions than edges and textures combined could be classified as primarily smooth (data set 1), but if the proportion of edges far out ways that of textures then it could also be classified in data set 4, therefore, a second criteria is created to subvert this.

Table 3.4. Data Set Classification

Data Set No.	Criteria 1	Criteria 2
1	Smooth > Edges + Textures	Edges ≈ Textures & Smooth > Edges
2	Edges > Smooth + Textures	Smooth ≈ Textures & Edges > Smooth
3	Textures > Smooth + Edges	Smooth ≈ Edges & Textures > Smooth
4	Smooth ≈ Edges & Smooth > Textures	N/A
5	Smooth ≈ Textures & Smooth > Edges	N/A
6	Edges ≈ Textures & Edges > Smooth	N/A
7	Smooth ≈ Edges ≈ Textures	N/A

3.7 CONCLUSION

In this chapter, the method for realising a hybrid multiple-image super-resolution system was explored. The interpolation, learning and reconstruction algorithms to be used in the system were explained in detail, in order to understand how the system would deal with different features detected from the input images. The final system was then created through the combination of these algorithms, together with feature detection.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

In this chapter, the results of the implementation of the methods described in the previous chapter are presented. The metrics for measuring the accuracy and efficiency of the system are discussed. Furthermore, the outcome of each step of the process is obtained to understand how the system flows. Finally, the final output super-resolution images are presented and assessed.

4.2 PERFORMANCE METRICS

The accuracy of the hybrid system cannot only be measured by it's visual appeal. Many metrics exist with which to measure and compare the performance of image processing techniques. The most widely used and accepted metrics are the Peak Signal to Noise Ratio and the Structural Similarity index. These metrics give a comprehensive and descriptive measure of how accurate an image processing algorithm is and will thus be used for the hybrid system.

4.2.1 Data sets

The results of the hybrid systems algorithms will be tested against a wide variety of images described in the previous chapter. Table 4.1 shows a description of each of the data sets used in this chapter.

Table 4.1. Data Sets

Data Set No.	Description
1	Primarily smooth regions
2	Primarily edges
3	Primarily textures
4	Smooth regions and edges
5	Smooth regions and textures
6	Edges and textures
7	Smooth regions, edges and textures

4.2.2 Peak signal to noise ratio

PSNR describes the ratio between the maximum possible power of a signal, in this case the highest possible pixel intensity of an image, and the power of corrupting noise which affects the fidelity of its representation [48]. In the context of super-resolution, the original high-resolution scene is regarded as the signal, while the output super-resolution image is regarded to have been affected by noise. When calculating PSNR, a higher value typically indicates that the reconstructed image is of a higher quality when comparing images of the same content. PSNR is based on the mean squared error between images, described in equation (4.1), when considering a w x h monochromatic image.

$$MSE = \frac{1}{wh} \sum_{k=0}^{w-1} \sum_{j=0}^{h-1} (g(k,j) - \tilde{g}(k,j))^2 \tag{4.1}$$

g and $\tilde{g}$ represent the original scene and the super-resolution output image respectively. In equation (4.2), MAX describes the highest possible pixel intensity in the image. In this context, the maximum intensity in an image represented by 8 bits per pixel is 255.

$$PSNR = 20\log_{10}(MAX) - 10\log_{10}(MSE) \tag{4.2}$$

4.2.3 Structural similarity

SSIM is a metric used to measure the similarity between images. It is a model which considers image degradation as a perceived change in structural information, as well as considering other perceptual phenomena such as luminance and contrast [48]. Structural information is defined by the strong inter-dependencies between spatially close pixels, which carry important information about the structure of objects within a scene.

The SSIM model is often calculated using a window, which is displaced pixel by pixel over the image. SSIM is calculated between two windows x and y, according to equation (4.3).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.3}$$

μ is the average pixel intensity of a window, σ^2 is the variance and c is a variable to stabilise division with weak denominators. This formula is obtained by combining three comparison measurements: luminance, contrast and structure. Equation (4.4), (4.5) and (4.6) describe how each of these measurements are obtained.

$$l(x,y) = \frac{(2\mu_x\mu_y + c_1)}{(\mu_x^2 + \mu_y^2 + c_1)} \tag{4.4}$$

$$c(x,y) = \frac{(2\sigma_x\sigma_y + c_2)}{(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.5}$$

$$s(x,y) = \frac{(\sigma_{xy} + c_3)}{(\sigma_x^2\sigma_y^2 + c_3)} \tag{4.6}$$

The resultant value obtained, is a decimal between -1 and 1, where a value of 1 indicates a completely identical image.

4.3 HYBRID SYSTEM OUTPUT STEPS

In order to understand the steps involved in the hybrid super-resolution system, it is necessary to begin with an original high-resolution image, which will be used as a reference for the performance and accuracy of the system. The entire system flow can simply be described by five steps:

1. Select a reference image as an original scene.

2. Down-sample the original scene to simulate discrete capture.

3. Perform feature detection to map features onto the image.

4. Perform interpolation, learning and reconstruction. super-resolution on the image.

5. Combine the results of each approach into a final high-resolution approximation.

4.3.1 Original continuous scene

The image 'lena.bmp' was selected as the original continuous scene. It has dimensions 512x512 pixels, as illustrated in figure 4.1. This image size is chosen in order to clearly show the visual effects that the super-resolution technique has on the final output image. When applied to large data sets, smaller images will be used in order to measure average performance on a much larger scale.

4.3.2 Discrete capture

The original scene is down-sampled in order to simulate discrete capture. This is done in order to simulate effects such as blur, degradation and sub-sampling, which typically effect the discrete capture of an image during the capture process. The result is a heavily degraded image, giving a very similar effect to that of capturing real continuous scenes in the world. The image shown in figure 4.2 will act as the low-resolution observation to be used by the three super-resolution algorithms.

Figure 4.1. Original continuous scene

Figure 4.2. Low-resolution observation

4.3.3 Feature mapping

Feature detection is performed to identify areas of the image which need to be sent to the different algorithms. The three distinct features are identified by the feature detection algorithm and are then grouped together to identify which features should be extracted from each algorithm. Ideally, large windows of pixels are identified by the feature map so that there is no frequent disconnect between pixels in the final result, which could have a negative affect on the final approximation.

Figure 4.3 shows the image blocks appropriate for each super-resolution algorithm.

Figure 4.3. Feature mapping

As seen in figure 4.3, a few features are sometimes classified differently from what may be expected. This is due to the algorithms reliance on the same parameters to classify a feature. Another consideration to make is whether it is feasible to identify a feature differently to its neighbor features in order to maintain neighborhood coherence. While this does not have a profoundly significant effect on the final result of the hybrid system, it may be worth while to investigate different feature detection algorithms which could provide more effective and robust solution

4.3.4 Super-resolution algorithms

The low-resolution observation is then sent to each super-resolution algorithm. However, the different algorithms require different information in order to successfully perform their task.

4.3.4.1 Interpolation

As discussed in chapter 3, the interpolation and reconstruction algorithms require multiple low-resolution observations in order to make accurate approximations. Therefore, the low-resolution observation is shifted in order to simulate a motion cue for non-uniform bi-cubic interpolation. The process follows a similar pattern to the one described in section 3.2.1, to ensure that sub-pixel shifts exist between the input images in order for the algorithm to have access to new information. An example of a group of shifted input images is shown in figure 4.4.

Figure 4.4. Shifting for interpolation. Shift (a), (b), (c) and (d)

Figure 4.5 shows the high-resolution approximation using interpolation, while table 4.2 shows the average PSNR and SSIM for interpolation super-resolution, when applied to different data sets. The average computation time for the execution of the system on each image is also given in order to draw a comparison between accuracy and speed.

Figure 4.5. High-resolution approximation using interpolation

Table 4.2. Average PSNR/SSIM for interpolation

Data Set	Computation Time (s)	PSNR	SSIM
1	10.35	24.39	0.87
2	8.50	20.35	0.69
3	7.54	22.44	0.75
4	7.98	23.01	0.77
5	8.78	23.86	0.80
6	8.23	21.97	0.71
7	9.50	22.95	0.76

Figure 4.5 shows the contrast of the interpolation result with the original high resolution image. As expected, the interpolation algorithm does not handle high frequency information well, but does return acceptable results for regions in which there are not drastic changes in pixel intensity. Table 4.2 shows how the interpolation algorithm has the highest structural similarity when applied to data sets with primarily smooth regions, but fails to perform adequately with other features. Since the algorithm is interpolating missing data using neighborhood information, high frequency information may suffer resulting in blurry artefacts, thus decreasing the signal to noise ratio.

4.3.4.2 Learning

The learning based method requires multiple images of similar content to create a training set. For a single approximation, these images can be hand-picked or the content of the image can be specified in order to make sure that the correct information is used. Ideally, for large data sets, the super-resolution system needs to first identify the type of content contained in the low-resolution algorithm and select appropriate images from the data set for the generation of the training set. Figure 4.6 shows an example of a group of images used to generate the training set for images with faces in them.

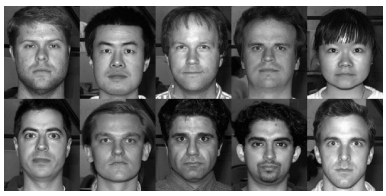

Figure 4.6. Training set images

Figure 4.7 shows the high-resolution approximation using learning, while table 4.3 shows the average PSNR and SSIM for learning super-resolution, when applied to different data sets. The average computation time for the execution of the system on each image is also given in order to draw a comparison between accuracy and speed.

Figure 4.7. High-resolution approximation using learning

Table 4.3. Average PSNR/SSIM for learning

Data Set	Computation Time (s)	PSNR	SSIM
1	18.54	27.21	0.82
2	17.23	27.56	0.86
3	16.59	26.52	0.79
4	20.56	27.53	0.83
5	19.26	26.78	0.80
6	21.78	26.91	0.81
7	17.99	27.01	0.81

Figure 4.7 shows the contrast of the learning result with the original high resolution image. As expected, the learning algorithm handles high frequency information much more effectively. Table 4.3 shows how the learning algorithm has the highest structural similarity when applied to data sets with primarily edges, but fails to perform adequately when there are drastic changes in pixel intensity within a finite region. The algorithm also provides a high structural similarity for smooth regions but at much greater computational cost when compared to the interpolation algorithm.

4.3.4.3 Reconstruction

Reconstruction follows the same shifting process as interpolation for each of its iterations. The process is iterated until an error rate of 0.85 is met between the simulated low-resolution image obtained from the high-resolution approximation and one of the original low-resolution input images or until a certain period of time elapses.

Figure 4.8 shows the high-resolution approximation using reconstruction, while table 4.4 shows the average PSNR and SSIM for reconstruction super-resolution, when applied to different data sets. The average computation time for the execution of the system on each image is also given in order to draw a comparison between accuracy and speed.

Figure 4.8. High-resolution approximation using reconstruction

Table 4.4. Average PSNR/SSIM for reconstruction

Data Set	Computation Time (s)	PSNR	SSIM
1	26.32	25.15	0.82
2	29.01	24.21	0.78
3	28.56	27.66	0.85
4	31.28	25.52	0.79
5	29.12	27.01	0.83
6	27.97	25.62	0.81
7	30.56	26.23	0.81

Figure 4.8 shows the contrast of the reconstruction result with the original high resolution image. As expected, the reconstruction algorithm is able to overcome the problem of rapidly changing intensities within a finite region, but sacrifices a significant amount signal to noise ratio. Table 4.4 shows how the reconstruction algorithm has the highest structural similarity when applied to data sets with primarily textures. It also manages to achieve acceptable results with other data sets but at the price of heavy computation times.

4.3.4.4 Feature extraction

Each of the algorithms returns a high-resolution approximation of the scene. The feature map is examined to check which pixels from which algorithm should be used for the final high-resolution approximation. Figure 4.9 shows the high-resolution result from each algorithm as well as the information extracted from each algorithm, which will form part of the final combined super-resolution result.

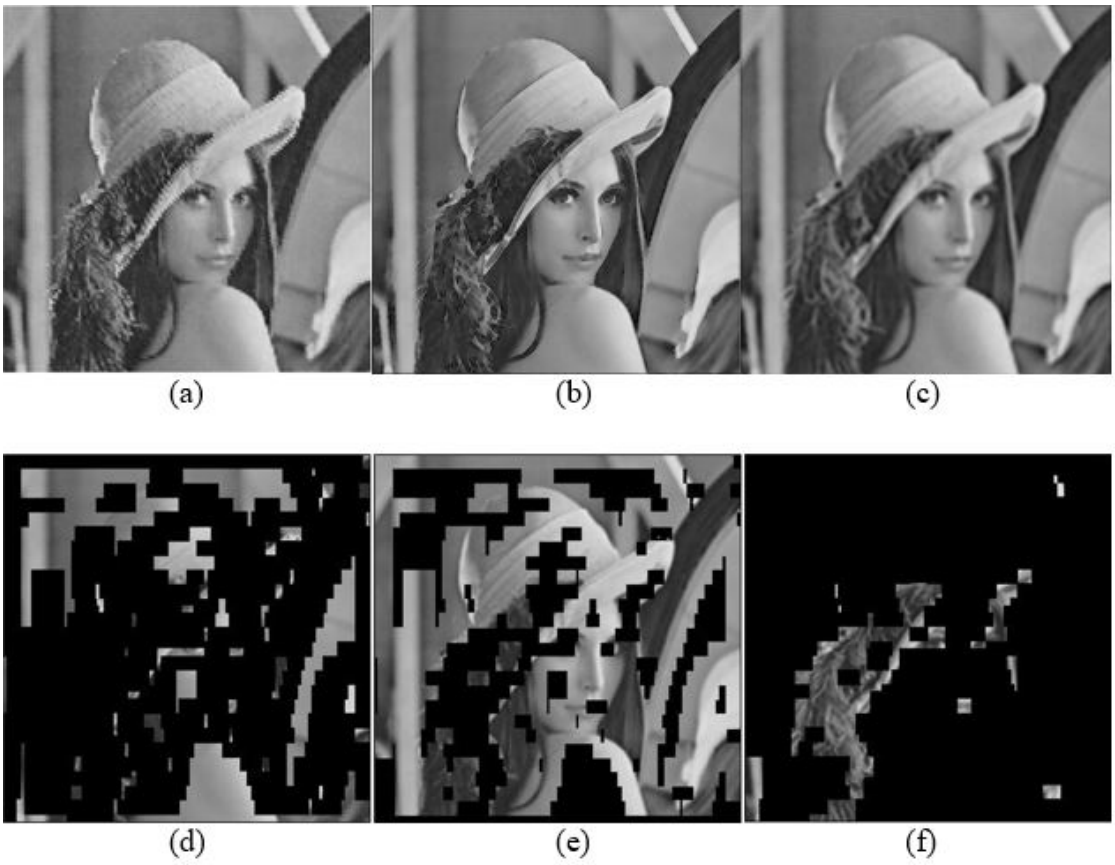

Figure 4.9. Features extracted from the different super-resolution algorithms. (a) Interpolation result, (b) learning result, (c) reconstruction result, (d) smooth regions extracted from interpolation, (e) edge information extracted from learning and (f) texture information extracted from reconstruction.

4.3.5 Combined super-resolution approximation

The features extracted from each of the algorithms are then combined to create the final super-resolution approximation. The result of this can be seen in figure 4.10.

Figure 4.10. Combined super-resolution approximation

In order to compare the visual aesthetic of the result, figure 4.11 shows the original continuous scene, the low-resolution observation and the final combined super-resolution approximation.

Figure 4.11. Final output result, (a) original continuous scene, (b) low-resolution observation and (c) combined super-resolution approximation

4.4 HYBRID SYSTEM RESULTS

In order to accurately measure the performance of the system, the algorithm must be applied to large data sets to obtain average computation times, PSNR and SSIM. The data sets chosen are all based on the different cases presented in table 4.1 in order to analyse how the system behaves when presented with different features.

Table 4.5, 4.6, 4.7 and 4.8 show the average PSNR and SSIM for different combinations of super-resolution algorithms when applied to different data sets. The average computation time for the execution of the system on each image is also given in order to draw a comparison between accuracy and speed.

In table 4.5, only the interpolation and learning algorithms are used; thus, textures are regarded as edges and are extracted from the learning result.

Table 4.5. Average PSNR/SSIM for interpolation and learning

Data Set	Computation Time (s)	PSNR	SSIM
1	29.11	25.80	0.85
2	25.24	23.97	0.77
3	24.89	24.50	0.78
4	28.20	27.54	0.84
5	28.65	25.33	0.81
6	30.12	24.43	0.78
7	27.58	25.00	0.78

In table 4.6, only the interpolation and reconstruction algorithms are used; thus, edges are regarded as textures and are extracted from the reconstruction result.

Table 4.6. Average PSNR/SSIM for interpolation and reconstruction

Data Set	Computation Time (s)	PSNR	SSIM
1	37.12	24.75	0.84
2	33.84	22.38	0.73
3	36.87	25.50	0.80
4	40.21	24.36	0.78
5	36.97	27.10	0.91
6	37.45	23.95	0.77
7	40.66	24.60	0.79

In table 4.7, only the learning and reconstruction algorithms are used; thus, smooth regions are regarded as textures and are extracted from the reconstruction result.

Table 4.7. Average PSNR/SSIM for learning and reconstruction

Data Set	Computation Time (s)	PSNR	SSIM
1	44.23	26.20	0.82
2	45.99	25.95	0.82
3	45.55	27.03	0.82
4	52.21	26.25	0.80
5	48.98	26.95	0.81
6	50.70	26.95	0.87
7	49.87	26.61	0.81

In table 4.8, all the algorithms are present; thus, all the corresponding features are extracted from their respective algorithms.

Table 4.8. Average PSNR/SSIM for interpolation, learning and reconstruction

Data Set	Computation Time (s)	PSNR	SSIM
1	54.89	24.56	0.89
2	55.71	26.45	0.83
3	52.78	26.12	0.82
4	59.64	27.65	0.84
5	57.25	27.35	0.91
6	58.02	27.89	0.87
7	60.10	28.01	0.90

4.4.1 Choosing algorithms for appropriate features

The more algorithms the system uses, the longer it takes to create the final super-resolution approximation. It can also be noted that there is no significant increase in accuracy when algorithms are applied to images with less than three prominent features. Thus, the approach of switching on necessary algorithms for significant features can be used to reduce average computation time with minimal effect on the accuracy of the system. Table 4.9 shows all the combinations of algorithms which can be chosen for appropriate features.

Table 4.9. Algorithms

Algorithm No.	Algorithms used
1	Interpolation
2	Learning
3	Reconstruction
4	Interpolation and learning
5	Interpolation and reconstruction
6	Learning and reconstruction
7	Interpolation, learning and reconstruction

Figure 4.12 to 4.18 show the performance of each combination of algorithm according to table 4.9 on the different data sets described in table 4.1. The combinations of algorithms are purposefully applied to all these data sets to contract their levels of performance. This information was used to decide which algorithms to use for sets of images which contained different features.

4.4.1.1 Smooth regions

According to figure 4.12, the interpolation algorithm has one of the highest SSIM indices of all the combinations of algorithms for smooth regions and also has the fastest computation time; thus, it was chosen as the only algorithm to use on smooth regions in the final hybrid system.

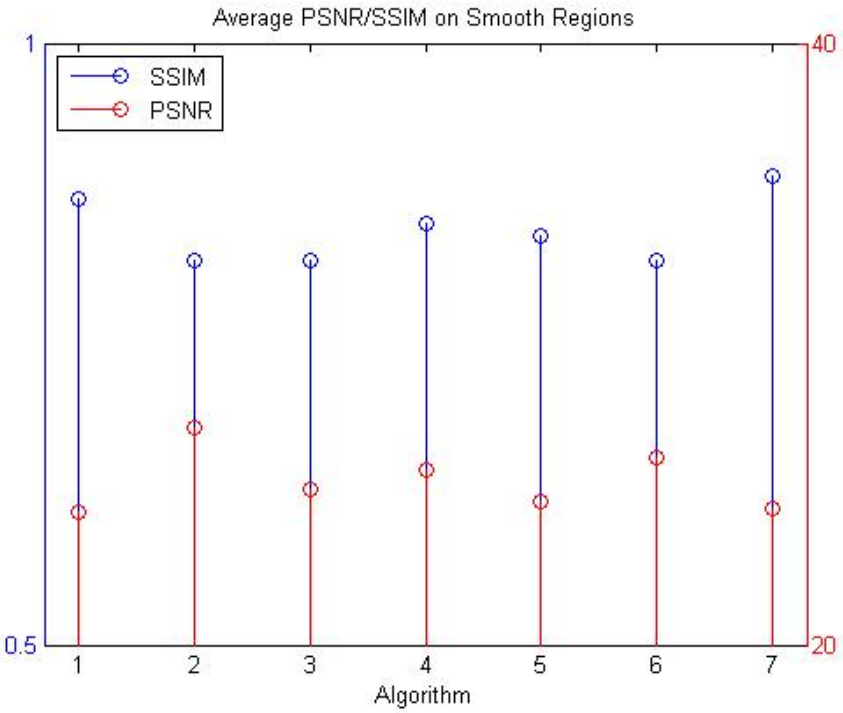

Figure 4.12. Average PSNR/SSIM for smooth regions

4.4.1.2 Edges

According to figure 4.13, the learning algorithm has the highest combination of PSNR and SSIM of all the combinations of algorithms for edges. For this reason it was chosen as the only algorithm to use on edges in the final hybrid system.

4.4.1.3 Textures

According to figure 4.14, the reconstruction algorithm has the highest combination of PSNR and SSIM of all the combinations of algorithms for textures. For this reason it was chosen as the only algorithm to use on textures in the final hybrid system.

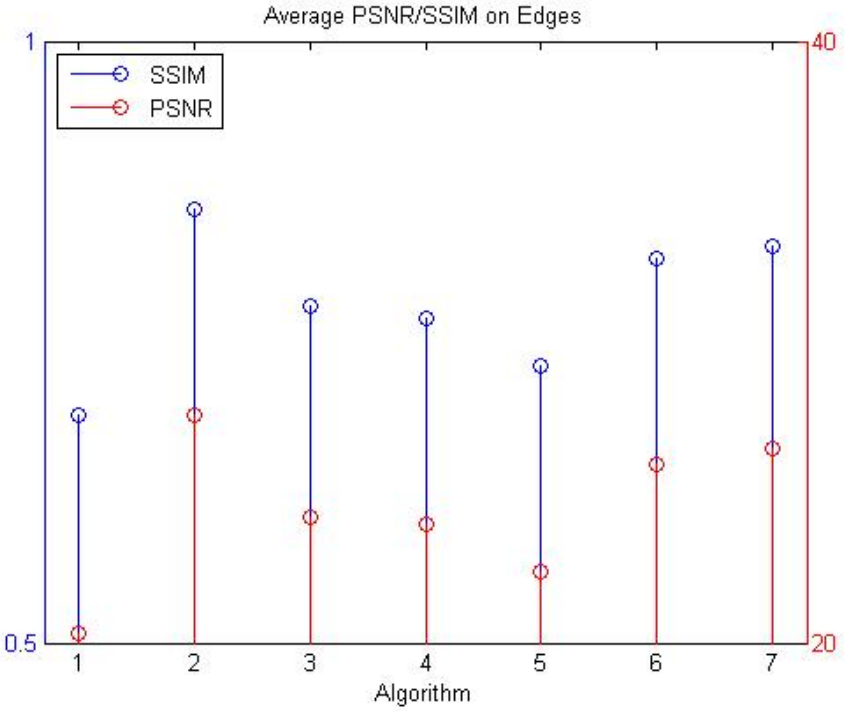

Figure 4.13. Average PSNR/SSIM for edges

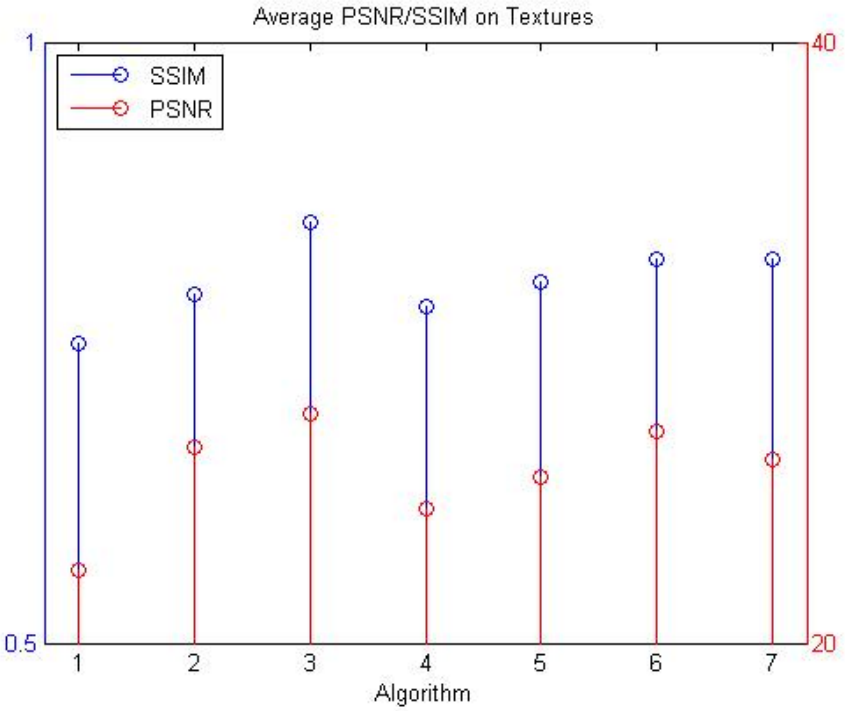

Figure 4.14. Average PSNR/SSIM for textures

4.4.1.4 Smooth regions and edges

According to figure 4.15, algorithm 2, 4 and 7 have the highest PSNR for smooth regions and textures. However, algorithm 4 has a slightly higher SSIM index and a much faster computation time than algorithm 7. Therefore, the interpolation and learning algorithms were used for this set of features.

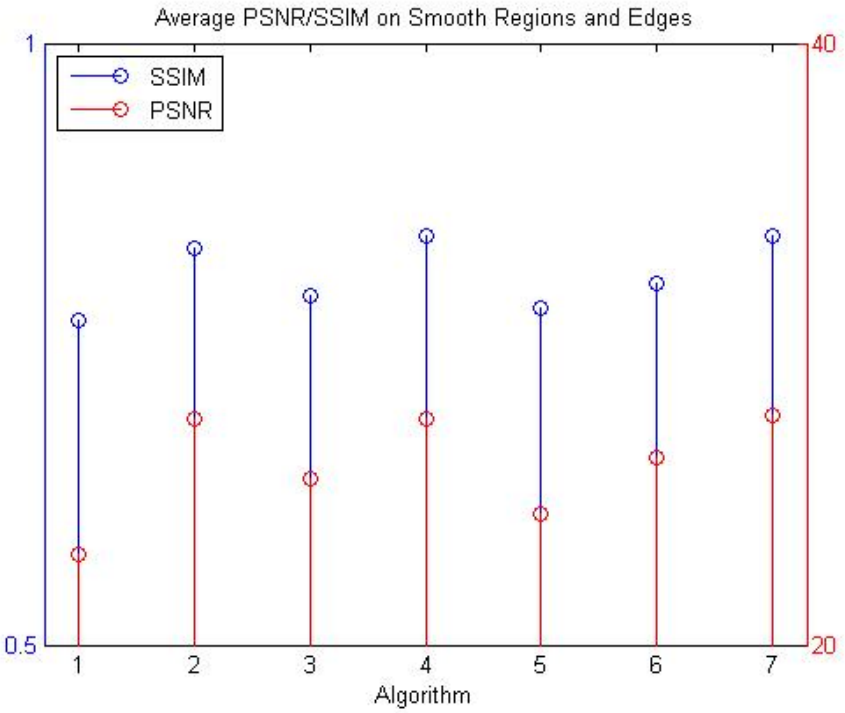

Figure 4.15. Average PSNR/SSIM for smooth regions and edges

4.4.1.5 Smooth regions and textures

According to figure 4.16, algorithm 5 and 7 have the highest combination of PSNR and SSIM for smooth regions and textures. However, algorithm 5 has a much faster computation time; therefore, the interpolation and reconstruction algorithms were used for this set of features.

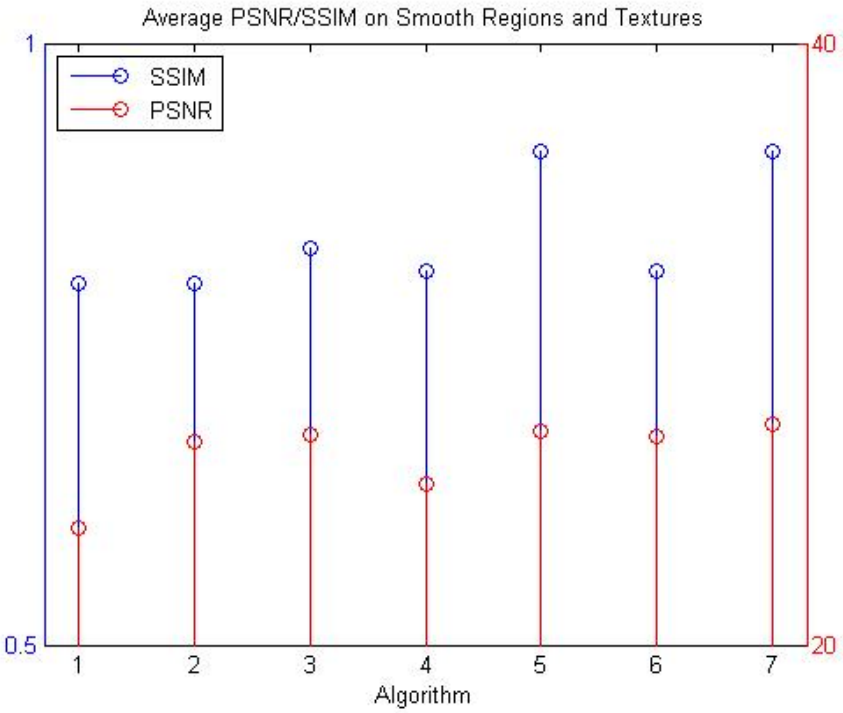

Figure 4.16. Average PSNR/SSIM for smooth Regions and textures

4.4.1.6 Edges and textures

According to figure 4.17, algorithm 6 and 7 have the highest combination of PSNR and SSIM. However, algorithm 6 had a slightly higher PSNR and a much faster computation time. Therefore, the learning and reconstruction algorithms were used for this set of features.

4.4.1.7 All features

According to figure 4.18, the combination of all three super-resolution algorithms results in the best PSNR and SSIM for data sets containing all the identified features. For this reason, all three algorithms were chosen to be used for this set of features in the hybrid system.

Using the selections of algorithms obtained from figure 4.12 to 4.18, table 4.10 shows the performance of the final hybrid super-resolution system, where the appropriate algorithms are applied to the correct data sets.

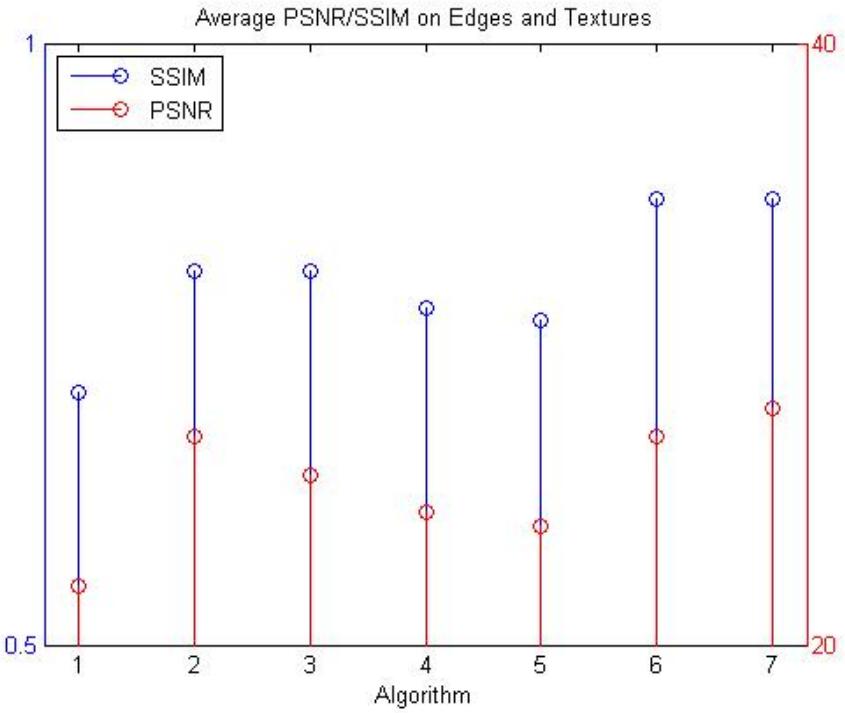

Figure 4.17. Average PSNR/SSIM for edges and textures

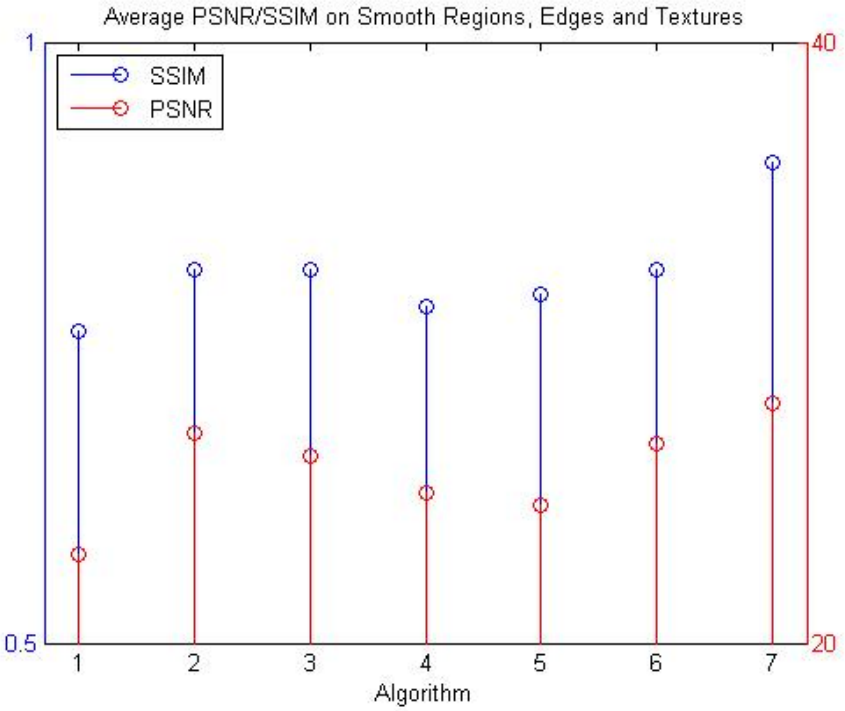

Figure 4.18. Average PSNR/SSIM for all features

Table 4.10. Average PSNR/SSIM for the hybrid system

Data Set	Computation Time (s)	PSNR	SSIM
1	10.35	24.39	0.87
2	17.23	27.56	0.86
3	28.56	27.66	0.85
4	28.20	27.54	0.84
5	36.97	27.32	0.91
6	50.70	27.91	0.87
7	60.10	28.01	0.90

Figure 4.19 shows a few more examples of the hybrid system applied to different types of images. Each of the output steps displayed here are described in detail in section 4.2. Larger versions for aesthetic comparison are available in Addendum A.

4.5 CONCLUSION

In this chapter, the results of the implementation of the hybrid multiple image super-resolution system were presented. The metrics for measuring the accuracy and performance of the system were discussed. Furthermore, the outcome of each step of the process was obtained, including the output of each of the super-resolution algorithms, in order to understand the steps involved in the flow of the system. Finally, the results of the hybrid system were presented, which took advantage of all the strengths of the super-resolution algorithms by switching necessary algorithms on and off, depending on prominent features.

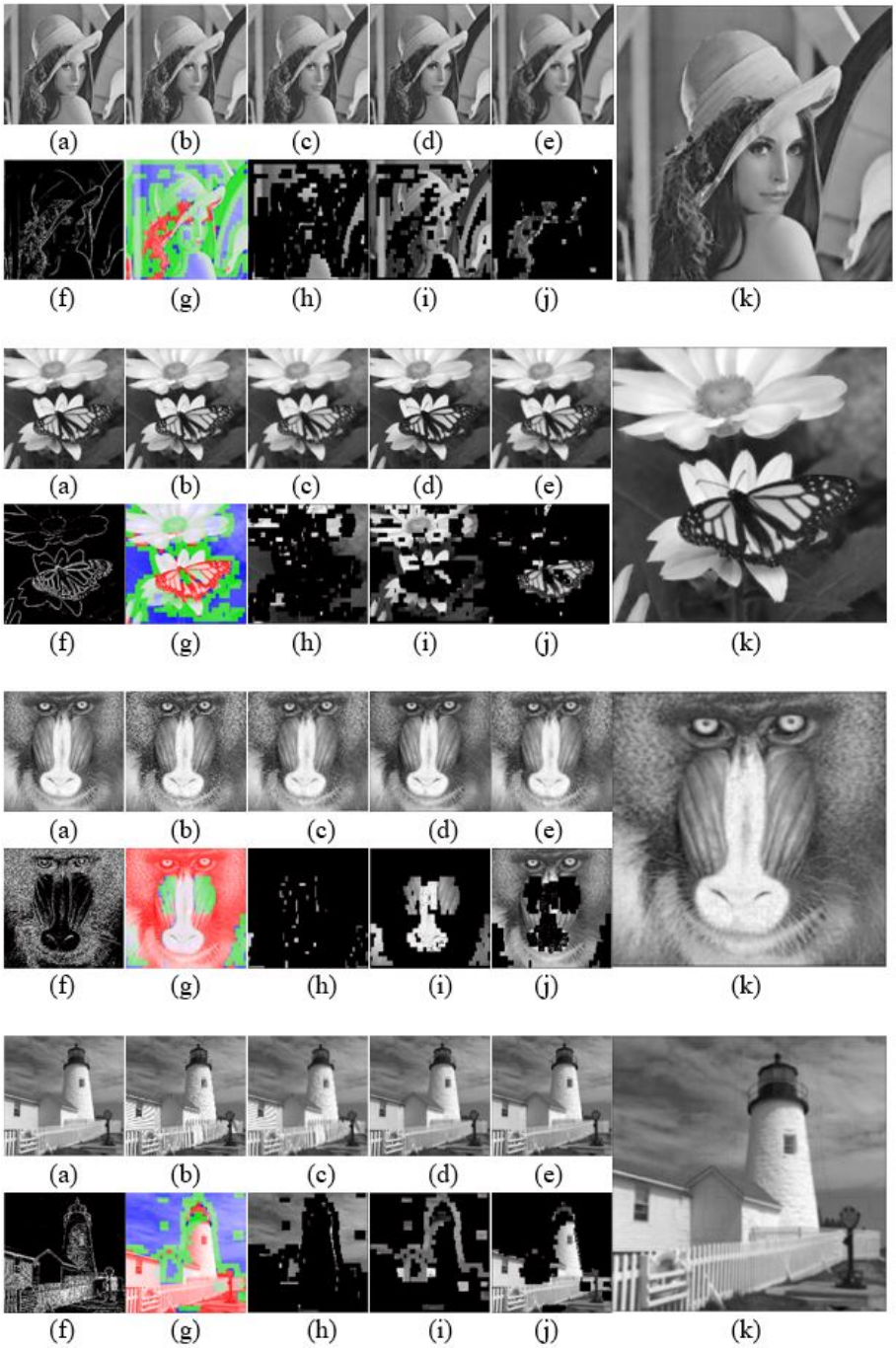

Figure 4.19. Hybrid system steps for different images. In the top row, (a) is the original scene, (b) is a low-resolution input image and (c), (d) and (e) are the interpolation, learning and reconstruction results respectively. In the bottom row, (f) is the feature detection result, with (g) the corresponding feature map. (h), (i) and (j) are the features extracted from (c), (d) and (e), according to (g). (k) is the combined super-resolution image.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

In this chapter, the results obtained for the hybrid system in the previous chapter are reviewed and discussed. Each algorithm involved in the system is discussed to understand its strengths and limitations in the application. Furthermore, the decisions made for the application of different combinations of algorithms on different feature sets are explored. Finally, the role of feature detection in the formation of the final product is discussed.

5.2 INTERPOLATION ALGORITHM

The interpolation algorithm is the first building block for realising the hybrid multiple image super-resolution system. It plays a significant role in the formation of the final super-resolution approximation. Its role is to enhance the resolving power of smooth regions in the low-resolution observation. The difference between the interpolated result and the low-resolution discretion is quite significant: undesired sharp edges were smoothed over and more detail and information is available. There is more definition in areas of high information density and textures are smoother in lower density regions. It is not expected for the approximation to match the original scene perfectly. Instead, the desired outcome is to ensure an increase in resolving power when compared to the lower resolution images.

The basic premise of increasing spatial resolution using interpolation, is the availability of multiple low-resolution images obtained from the same scene. The input images have varying global translation parameters defined by the sub-pixel shifts between their pixels. Missing pixel data can then be approximated to construct a high-resolution image, using neighborhood pixel data.

It can be seen from the interpolation results that when using four input images, the high-resolution approximations created from low-resolution images are 16 times larger than the down-sampled input images, according to equation (3.2). The increase in the size of the newly constructed approximation results in a large amount of missing data. When using four low-resolution input images, the amount of known data points is 25% of all the data points within the high-resolution grid. This means that the algorithm must estimate 75% of the output data points. The fact that the algorithm is able to estimate such a large amount of data with the limited data available and produce such satisfactory visual results, is quite impressive.

It is useful to consider the nature of results that are obtained when more than four images are used. When using eight images, the amount of known data decreases to 12.5% of the output data points. Theoretically, if sixteen images were used, then the amount of known data available decreases to 6.25% of the data points on the high-resolution grid. It can therefore be deduced that the interpolation algorithm experiences diminishing returns when increasing the number of input images. Theoretically, an infinite number of images could be provided to the system, but the resolving power of the new approximations would only experience small, if not negligible, improvements. This phenomenon suggests that the interpolation algorithm is not as efficient as it can be and if the approximation size is decreased, more accurate images can be reconstructed. Diminishing returns could still possibly be experienced but at a much slower rate.

The interpolation based approach was first introduced by Tsai and Huang in 1984 [9] and has been applied to many image processing problems where high-resolution images are in demand [10], including its application in satellite surveillance and medical technology. Since then it has undergone many iterations to become an accepted and frequently used image processing technique, which is the simplest algorithm of the three super-resolution algorithms used by the hybrid system. Its performance is rather low, relatively speaking. Its average PSNR when applied to varying data sets is often very low. This is because the interpolation algorithm struggles to deal with edge and texture information without the help of additional sharpening and restoration algorithms, making it look as if the output approximation

has been blurred. The improvements to this unwanted blur can be seen in the reconstruction algorithm, where the initial high-resolution estimation is the interpolation result. However, this improvement is negligible for smooth regions. Thus the interpolation algorithm's superior speed and efficiency for enhancing smooth regions is definitely worth its inclusion in the system.

5.3 LEARNING ALGORITHM

The learning based approach is a relatively new and innovative solution to the super-resolution problem [21] and has been applied to many image processing problems where high-resolution images are in demand, including its application in computer vision and robotics. Strictly speaking, it falls into the category of single image super-resolution [22]. However, the generation of a set of training data requires multiple low-resolution and high-resolution image pairs. The learning algorithm plays a significant role in the formation of the final super-resolution approximation. Its role is to enhance the resolving power of edges in the low-resolution observation. It is one of the most versatile and promising approaches, offering improved edge enhancement and finer detail. It uses a training set to learn the fine detail that corresponds to different image regions seen at a lower resolution [20]. Once it has learned this relationship, it is able to apply it to predict fine detail in other images. It takes advantage of the fact that a collection of image pixels has much less variability than a corresponding set of completely random variables.

The example based super-resolution algorithm works best when the training data resolution or noise degradation factor matches that of the images to which the algorithm is applied. It also thrives when training images are chosen from a similar domain. However, in the ideal case, a much larger training set would be used with a diverse range of training images to ensure that training data is generalised for cross-application use. This approach to super-resolution could be expanded to moving images with the use of multiple observations of the same pixel and the maintenance of coherence across subsequent frames.

The learning algorithm is slightly more complex than other algorithms. However, the concept is easy to understand and once grasped, the implementation of this algorithm is a simple task. It is one of the best performing algorithms out of the three super-resolution algorithms used by the hybrid system and the most computationally expensive algorithm, when considering the additional time needed to

generate training data for different types of image content. However, it consistently obtains high PSNR and SSIM values for multiple data sets, thereby solidifying its place in the final system.

5.4 RECONSTRUCTION ALGORITHM

The reconstruction algorithm plays a significant role in the formation of the final super-resolution approximation. Its role is to enhance the resolving power of textures in the low-resolution observation by attempting to remove the degradation effects of blur, sub-sampling and warping in order to acquire a desired high-resolution image. It is an iterative process, that estimates sub-pixel displacement and noise variance in low-resolution images. Thus, high-resolution images are approximated through expectation maximums. It was first introduced as an improvement to the interpolation algorithm, which suffered from heavy amounts of noise. Since then it has undergone many iterations to become an accepted and frequently used image processing technique, which is typically used in processing problems where high-resolution images are in demand, especially in images which suffer from heavy amounts of environmental noise.

The reconstruction algorithm shows satisfactory results with an error rate of 0.85. This could also be due to the relatively small pixel size of the images in the data set. However, the effectiveness of back-projection is undeniable when considering that the initial estimation for the high-resolution approximation is obtained by using interpolation. This approach is very effective at error correction, which can clearly be seen in the comparison of the initial estimation and the final approximation. High frequency information is very well-preserved and sharp edges are more defined. This approach is also very effective when heavy blur is present in an image and is also very good at preserving texture information. The IBP method itself is computationally very cheap at lower numbers of iterations, but can take much longer finding a solution if an unrealistic error maximum is defined.

The reconstruction algorithm is a relatively simple algorithm when compared to the other super-resolution algorithms used in the hybrid system. One iteration on its own is computationally very cheap. However, multiple iterations due to a high error rate specification can cause the algorithm to run for long periods of time, but this increase in computation time is worth the accuracy it provides for texture information and its ability to enhance information degraded by heavy blur.

5.5 FEATURE DETECTION

Edge detection is extremely useful in the identification of objects in an image. The Sobel operator is good at identifying high frequency information in an image. However, this information's usefulness in post-processing applications, such as super-resolution, is not highly valued. Too many unnecessary edges are detected, for example, the shaded areas on certain parts of a face or the small changes in intensity in a hat. When Canny edge detection is applied, most, if not all of these unnecessary edges, are removed. The edges are thinned out, making it much easier to discern where object boundaries are, while more prevalent edges are preserved. The edge information received from the Canny operator is ideal for the identification of edges for an example-based super-resolution algorithm. In addition, it can be seen that large collections of edges can often be identified as textured regions. This is because textured regions are areas with large collections of varying differences in pixel intensity, in other words, a large group of edges with different orientations. While this method of feature detection is not perfect, it does provide an effective and simple detection algorithm for the identification of texture regions for reconstruction based super-resolution, as long as the limit and expectation to which a cluster of edges is regarded a texture is defined. Similarly, smooth regions are identified as those regions with little to no edges at all. It should be noted that this detection algorithm could definitely be improved on and it may be useful to do so in future work.

5.6 HYBRID SUPER-RESOLUTION

The combined super-resolution approximation still contains some undesirable visual artifacts as a result of combining different pixels from different high-resolution approximations. Since the image blocks were approximated by different algorithms, the boundaries of the image blocks will have differing pixel values. This could potentially be rectified by adding Gaussian blur to those regions to blend the pixels together.

This version of the combination of pixels from different algorithms is not yet optimal. Since each algorithm receives the entire low-resolution observation in order to produce a high-resolution approximation, many pixels from these results are unused in the final product. This is almost unavoidable in the case of example-based super-resolution. Example-based super-resolution is heavily reliant on the input observation to be a recognisable and predictable image in order to apply the training set for

super-resolution. If an image block, containing only an edge of the image, is received, then it will be difficult to determine how best to enhance it because the training set will not know the context in which the edge was found. Luckily, this is not the case for interpolation and reconstruction. Therefore, the optimal solution for now would be be to send the entire image to the example based algorithm, send only the appropriate image blocks to the interpolation and reconstruction algorithms and then add the interpolation and reconstruction results to the example-based result.

Another problem that was encountered, was the difference in magnification size of each high-resolution approximation. By their nature, interpolation and reconstruction based algorithms enlarge the low-resolution observation onto a high-resolution grid in order to approximate missing pixel information. Therefore, each algorithm returns a different sized image, which needs to be scaled to the same size. When up-scaling or down-scaling the high-resolution approximations, a lot of information may be lost and degradation in image quality occurs. Therefore, a solution needs to be found to deal with the different magnification factors of each algorithm efficiently.

5.7 CONCLUSION

In this chapter, the results obtained for the hybrid system in the previous chapter were reviewed and discussed. The interpolation algorithm's fast computation speed and simple complexity made it the perfect candidate for the processing of the simplest identified feature - smooth regions. The learning algorithm's greater complexity and ability to enhance high frequency information efficiently allowed it to be used in cases where many edges were present in an image. The reconstruction algorithm's ability to deal with high levels of noise and blur in an image made it useful when dealing with features that contained windows of rapidly changing pixel intensity. Feature detection made the combination of these algorithms possible and by analysing different algorithms' behaviour when faced with different features, different combinations of algorithms were able to be applied appropriately to maximise performance and accuracy, while minimising computational complexity.

CHAPTER 6 CONCLUSION

6.1 CONCLUSION

This research has presented the development of a hybrid multiple-image, super-resolution image reconstruction system. Interpolation, learning and reconstruction are used to enhance the resolving quality of images. The results of each type of super-resolution approximation showed that each approach performed optimally on certain image features. This finding lead to the combination of these approaches to form a system, which can identify image features and apply the appropriate approach to optimally enhance resolving quality. The objective was to derive such a system, based on the three types of super-resolution approaches, which is able to improve the resolution of an image containing multiple image features; thus, producing a novel algorithm for creating high-resolution approximations of a scene by overcoming the hardware limitations of imaging systems in applications such as satellite technology, surveillance systems, computer vision and medical technology.

The super-resolution image reconstruction technique is capable of overcoming the inherent limitations of an imaging system and can improve the resolving quality of low-resolution images. Different methods and approaches to this technique have different practical and theoretical applications. The advancement of knowledge in this field will make it possible for image processing researchers to not only further investigate the use of multiple images in high-resolution imaging systems, but also make imaging technology affordable in a technologically advancing world.

6.2 BENEFITS OF THE STUDY

The super-resolution image reconstruction technique has become increasingly popular because it is capable of overcoming the inherent resolution limitations of an imaging system. By applying this post-processing technique to a captured image, it is able to produce clearer images with richer detail, more resolving power and more useful information.

The hybrid system which has been developed, is also capable of selecting appropriate algorithms for different features, ensuring that minimal computation time is wasted on approximating less accurate information. This selection process also ensures that the best algorithm is chosen to maximise the trade-off between computation time and accuracy.

6.3 RECOMMENDATIONS FOR FUTURE WORK

In order to further develop this technique to meet the needs of practical applications, further research could be done to explore the improvement of degradation models, motion estimation and algorithm efficiency. Solving the problem of varying magnification factors between different algorithms without resorting to up-scaling or similar algorithms, could assist in providing more accurate approximations. A more thorough investigation of feature detection algorithms for its application in super-resolution systems could assist in more accurately classifying features. Many modern applications, such as surveillance and computer vision, require a live sensor feed to perform optimally in the real world. Thus, the system could be expanded to enhance the quality of moving images in video with the use of multiple observations of the same pixel and the maintenance of coherence across subsequent frames. Of course, this would require the hybrid system to be further optimised to operate in real time.

REFERENCES

[1] S. Zhang and P. Huang, "High-resolution, real-time three-dimensional shape measurement," *Optical Engineering*, vol. 45, no. 12, pp. 123 601–123 601, 2006.

[2] L. Zhang, H. Zhang, H. Shen, and P. Li, "A super resolution reconstruction algorithm for surveillance images," *Signal Processing*, vol. 90, no. 3, pp. 848–859, 2010.

[3] D. Litwiller, "Ccd vs. cmos," *Photonics Spectra*, vol. 35, no. 1, pp. 154–158, 2001.

[4] L. Brown, "A survey of image registration techniques," *ACM computing surveys (CSUR)*, vol. 24, no. 4, pp. 325–376, 1992.

[5] T. Chen, P. Catrysse, A. Gamal, and B. Wandell, "How small should pixel size be," *Sensors and Camera Systems for Scientific, Industrial and Digital Photography Applications*, vol. 3965, pp. 451–459, 2000.

[6] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, "Signal-processing based method for aqcuiring very high resolution images with multiple cameras and its theoretical analysis," in *Proceedings of the IEEE Conference on Speech and Vision*, 1993, p. 19.

[7] Q. Qian and B. Gunturk, "Extending depth of field and dynamic range from differently focused and exposed images," *Multidimensional Systems and Signal Processing*, vol. 27, no. 2, pp. 493–509, 2015.

[8] R. Szeliski, “Image mosaicing for tele-reality applications,” in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, 1994, pp. 44–53.

[9] S. Park, M. Park, and M. Kang, “Super-resolution image reconstruction: a technical overview,” *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[10] L. Ziwei, W. Chengdong, C. Dongyue, Q. Yuanchen, and W. Chunping, “Overview on image super resolution reconstruction,” in *The 26th Chinese Control and Decision Conference*, 2014, pp. 2009–2014.

[11] P. Gajjar and M. Joshi, “New learning based super-resolution: Use of DWT and IGMRF prior,” *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1201–1213, 2010.

[12] N. Maheta, “A comparative study of image super resolution approaches,” in *3rd International Conference on Electronics Computer Technology*, vol. 1, 2011, pp. 126–133.

[13] P. Vandewalle, S. Susstrunk, , and M. Vetterli, “A frequency domain approach to registration of aliased images with application to super-resolution,” *EURASIP Journal on Advances in Signal Processing*, vol. 2006, p. 233, 2006.

[14] C. Bishop, A. Blake, and B. Marthi, “Super-resolution enhancement of video,” in *AISTATS*, 2003.

[15] W. Freeman and E. Pasztor, “Learning low-level vision,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1182–1189.

[16] J. Sun, N. Zheng, H. Tao, and H. Shum, “Image hallucination with primal sketch priors,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. II–729.

[17] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based super-resolution,” *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[18] H. Chang, D. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, p. 1.

[19] T. Chan, J. Zhang, J. Pu, and H. Huang, "Neighbor embedding based super-resolution algorithm through edge detection and feature selection," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 494–502, 2009.

[20] K. Zhang, X. Gao, X. Li, and D. Tao, "Partially supervised neighbor embedding for example-based image super-resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 230–239, 2011.

[21] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[22] S. Baker and T. Kanade, "Hallucinating faces," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 83–88.

[23] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 231–239, 1991.

[24] X. Zhang, Q. Liu, X. Li, Y. Zhou, and C. Zhang, "Non-local feature back-projection for image super-resolution," *IET Signal Processing*, vol. 10, no. 5, pp. 389–408, 2016.

[25] M. Irani and S. Peleg, "Super resolution from image sequences," in *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 2, 1990, pp. 115–120.

[26] ——, "Motion analysis for image enhancement: Resolution, occlusion, and transparency," *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.

[27] Z. Zhang, X. Wang, J. Ma, and G. Jia, "Super resolution reconstruction of three view remote sensing images based on global weighted POCS algorithm," in *International Conference on*

Remote Sensing, Environment and Transportation Engineering, 2011, pp. 3615–3618.

[28] Z. Tang, M. Deng, C. Xiao, and J. Yu, "Projection onto convex sets super-resolution image reconstruction based on wavelet bi-cubic interpolation," in *Proceedings of the IEEE International Conference on Electronic and Mechanical Engineering and Information Technology*, vol. 1, 2011, pp. 351–354.

[29] G. Chantas, N. Galatsanos, and N. Woods, "Super-resolution based on fast registration and maximum a posteriori reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1830–1830, 2007.

[30] P. Purkait and B. Chanda, "Super resolution image reconstruction through bregman iteration using morphologic regularization," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4029–4039, 2012.

[31] P. Rasti, H. Demirel, and G. Anbarjafari, "Iterative back projection based image resolution enhancement," in *8th Iranian Conference on Machine Vision and Image Processing*, 2013, pp. 237–240.

[32] Y. Tang and Y. Yuan, "Learning from errors in super-resolution," *IEEE Transactions on Cybernetics*, vol. 44, no. 11, pp. 2143–2154, 2014.

[33] K. Rumyantsev and D. Petrov, "Universal robust algorithm for detection of image features," in *12th World Congress on Intelligent Control and Automation*, 2016, pp. 3061–3065.

[34] C. Zhao, C.-L. Lin, and W. Chen, "Maximal margin feature mapping via basic image descriptors for image classification," in *IEEE International Conference on Industrial Technology*, 2016, pp. 775–780.

[35] A. Burlacu and C. Lazar, "Image features detection using phase congruency and its application in visual servoing," in *4th International Conference on Intelligent Computer Communication and Processing*, 2008, pp. 47–52.

[36] P. Kaur and B. Kaur, "2-d geometric shape recognition using canny edge detection technique," in *3rd International Conference on Computing for Sustainable Global Development*, 2016, pp. 3161–3164.

[37] K. T. M. Han and B. Uyyanonvara, "A survey of blob detection algorithms for biomedical images," in *7th International Conference of Information and Communication Technology for Embedded Systems*, 2016, pp. 57–60.

[38] M. Zhang, T. Wu, S. C. Beeman, L. Cullen-McEwen, J. F. Bertram, J. R. Charlton, E. Baldelomar, and K. M. Bennett, "Efficient small blob detection based on local convexity, intensity and shape information," *IEEE transactions on medical imaging*, vol. 35, no. 4, pp. 1127–1137, 2016.

[39] S. Chaudhuri, *Super-resolution imaging*. Springer Science & Business Media, 2001, vol. 632.

[40] K. Sharma and A. Goyal, "Classification based survey of image registration methods," in *4th International Conference on Computing, Communications and Networking Technologies*, 2013, pp. 1–7.

[41] E. Maeland, "On the comparison of interpolation methods," *IEEE transactions on medical imaging*, vol. 7, no. 3, pp. 213–217, 1988.

[42] W. T. Freeman and E. C. Pasztor, "Markov networks for super-resolution," in *Proceedings of the 34th Annual Conference on Information Sciences and Systems*, 2000.

[43] O. Schwartz and E. P. Simoncelli, "Natural signal statistics and sensory gain control," *Nature neuroscience*, vol. 4, no. 8, pp. 819–825, 2001.

[44] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[45] M. N. Bareja and C. K. Modi, "An effective iterative back projection based single image super resolution approach," in *International Conference on Communication Systems and Network*

Technologies, 2012, pp. 95–99.

[46] A. Jain, A. Namdev, and M. Chawla, "Parallel edge detection by sobel algorithm using cuda c," in *IEEE Students' Conference on Electrical, Electronics and Computer Science*, 2016, pp. 1–6.

[47] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, "Feature detection and tracking with the dynamic and active-pixel vision sensor (davis)," in *2nd International Conference on Event-based Control, Communication, and Signal Processing*, 2016, pp. 1–7.

[48] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *20th international conference on Pattern recognition*, 2010, pp. 2366–2369.

ADDENDUM A ELABORATED RESULTS

A.1 HYBRID SYSTEM RESULTS

Images obtained from the results of the hybrid system are available in an online album uploaded to Google Drive. These images have been made available for atheistic perusal and appear in an abbreviated form in Chapter 4. The link to the album is given below.

https://drive.google.com/open?id=0ByDatapHq2ClaGZFeG1kcGVuVEU

Following the link will guide you to an album in which all the relevant images can be found. The album has the following content:

- Figure 4.12. (Baboon)
- Figure 4.12. (Butterfly)
- Figure 4.12. (Lena)
- Figure 4.12. (Lighthouse)

Each of these folders contains all the relevant images described in Figure 4.12:

(a) Original Scene

(b) Low-resolution input image

(c) Interpolation approximation

(d) Learning approximation

(e) Reconstruction approximation

(f) Feature detection

(g) Feature map

(h) Smooth regions extracted from interpolation

(i) Edges extracted from learning

(j) Textures extracted from reconstruction

(k) Combined super-resolution result