

INVERSE DISTANCE WEIGHTED INTERPOLATION BASED ON  
IMAGE TOPOLOGY FOR SUPER-RESOLUTION

Barışcan Dedeş

A thesis submitted for the degree of  
*Master of Science*  
in  
*Electrical and Electronics Engineering*

Supervisor: Assoc. Prof. Dr. Hakan Güray Şenel

Eskişehir  
Anadolu University  
Graduate School of Science  
January, 2021

This work is dedicated to the memory of my aunt Hasibe...

## ABSTRACT

### INVERSE DISTANCE WEIGHTED INTERPOLATION BASED ON IMAGE TOPOLOGY FOR SUPER-RESOLUTION

Barışcan Dedeş

Department of Electrical and Electronics Engineering  
Anadolu University, Graduate School of Science, January, 2021

Supervisor: Assoc. Prof. Dr. Hakan Güray Şenel

Super-resolution is the process of producing a high resolution image by using a series of slightly different low resolution images. Super-resolution, generally, is composed of three stages: registration of low resolution images for determining the distances between them, interpolation or merging algorithms, and image enhancement steps. In the second stage, which is the most important one, pixels of low resolution images are combined with weights to compute high resolution image pixels. Blurring of edges and corners, of which their preservations are important for image understanding, is one of the most important problems to deal with for these algorithms. This thesis is about the enhancement of algorithms used in the second stage by integrating connectivity and connected relationships of pixels into computations. In this work, an algorithm to find connectedness diagrams for non-uniform data and the effect of connectedness information to edges on the preregistered synthetic images is shown. Among different interpolation methods, Inverse Distance Weight (IDW) algorithm is preferred and high resolution image pixels are computed by integrating connectedness information into this algorithm. Results, obtained from the newly developed algorithm are compared to cubic interpolation and conventional IDW methods and it is shown that the new algorithm yields a superior performance in edge preservation and noise suppression compared to others.

**Keywords:** Super-Resolution Imaging, Scattered Data Interpolation, Nonuniform Interpolation, Connectivity Map, Connectivity Map for Scattered Data, Connected Component Labeling for Scattered Data

## ÖZET

### SÜPER-ÇÖZÜNÜRLÜK İÇİN İMGE TOPOLOJİSİNE DAYALI TERS MESAFE AĞIRLIKLI İNTERPOLASYON YÖNTEMİ

Barıçan Dedeş

Elektrik Elektronik Mühendisliği Bölümü  
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ocak, 2021

Danışman: Doç. Dr. Hakan Güray Şenel

Süper-çözünürlük aralarında çok az fark bulunan birden fazla düşük çözünürlüklü görüntünün kullanılmasıyla yüksek çözünürlüklü görüntüler üretilmesi işlemidir. Süper çözünürlük, genellikle, düşük çözünürlükteki resimlerin aralarındaki uzaklığın tespiti için kullanılan imge çakıştırma, yeniden kurma algoritmaları ve görüntü iyileştirme aşamalarından oluşmaktadır. En önemli adım olan ikinci aşamada, düşük çözünürlüklü imgelerdeki piksellerin belirli bir şekilde belirli katsayılarla bir araya getirilerek, yüksek çözünürlüklü görüntüdeki piksellerin değerleri oluşturulmaktadır. Görüntünün anlaşılabilmesi için gerekli kenarların ve köşelerin yeterince belirli hale getirilememesi ve bulanıklaşma bu algoritmaların en büyük sorunlarından biridir. Bu tez, ikinci aşamada uygulanan yeniden kurma algoritmalarının, pikseller arasındaki bağlantı ve komşuluk ilişkilerinin hesaplamalara eklenerek geliştirilmesi konusundadır. Çalışmada dağınık veride bağlantısallık haritası çıkarma tekniği sunulmuş ve haritanın imgedeki kenar profiline etkileri önceden çakıştırılmış sentetik test görüntülerini üzerinde gösterilmiştir. Farklı interpolasyon yöntemleri arasında, ters mesafe ağırlıklı (IDW) interpolasyon yöntemi tercih edilmiş ve bağlantısallık haritası üzerine uygulanan interpolasyon sonuçlarının geliştirilen algoritma içinde kullanılmasıyla yüksek çözünürlüklü imgenin piksel değerleri hesaplanmıştır. Sonuçlar, kübik interpolasyon, geleneksel IDW yöntemleriyle karşılaştırılmış ve geliştirilen algoritmanın, kenar iyileştirme ve gürültü bastırma performanslarının diğer yöntemlere göre daha iyi sonuçlar verdiği gösterilmiştir.

**Anahtar Sözcükler:** Süper-Çözünürlük Görüntüleme, Dağınık Veri Aradeğerleme, Dağınık Veri için Bağlantısallık Haritası, Bağlantısallık Haritası, Dağınık Veri için Bağlantılı Bileşen Etiketleme

## **ACKNOWLEDGMENTS**

I would like to express my sincerest gratitude to my supervisor, Assoc. Prof. Dr. Hakan Güray Şenel, who has guided me throughout my thesis. I thank him for his guidance and encouragement, his good advice and support on academic, career and personal matters have been invaluable.

This thesis could not have been completed without the support of my company HAVELSAN Inc..

Also, I would like to thank my dear friends, Şan Gültekin and Ozan Fırat Özgül for the support they provided, which was above and beyond the call of duty. I would like to thank my team leader Burak Başlı for his support especially during the last few months of this work. Finally, I express my gratitude to Hasibe Dedeş for her encouragement, without whom I wouldn't have the courage to embark on this journey.

Bařışcan Dedeş

January, 2021

## **STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES**

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with “scientific plagiarism detection program” used by Anadolu University, and that “it does not have any plagiarism” whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Başkan Dedeş

## TABLE OF CONTENTS

	<u>Page</u>
<b>TITLE PAGE</b> . . . . .	i
<b>ABSTRACT</b> . . . . .	iii
<b>ÖZET</b> . . . . .	iv
<b>ACKNOWLEDGMENTS</b> . . . . .	v
<b>STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES</b>	vi
<b>TABLE OF CONTENTS</b> . . . . .	vii
<b>LIST OF TABLES</b> . . . . .	x
<b>LIST OF FIGURES</b> . . . . .	xi
<b>GLOSSARY OF SYMBOLS AND ABBREVIATIONS</b> . . . . .	xiv
<b>1. INTRODUCTION</b> . . . . .	1
1.1. Introduction . . . . .	1
1.2. Motivation . . . . .	3
1.3. Research Contributions . . . . .	4
1.4. Thesis Outline . . . . .	4
<b>2. BACKGROUND IN SUPER-RESOLUTION</b> . . . . .	6
2.1. Observation Model . . . . .	6
2.2. Image Registration . . . . .	7
2.2.1. Correlation Based Methods . . . . .	8
2.2.1.1. Cross-correlation Method . . . . .	8

2.2.1.2	Phase Correlation Method . . . . .	9
<b>2.2.2.</b>	<b>Feature Based Methods . . . . .</b>	<b>11</b>
<b>2.3.</b>	<b>Super-Resolution in Frequency Domain . . . . .</b>	<b>15</b>
<b>2.4.</b>	<b>Super-Resolution by Nonuniform Interpolation . . . . .</b>	<b>16</b>
<b>2.4.1.</b>	<b>Triangulation Based Methods . . . . .</b>	<b>17</b>
2.4.1.1	Delaunay Triangulation . . . . .	17
2.4.1.2	Linear Triangular Interpolation . . . . .	19
2.4.1.3	Cubic Triangular Interpolation: The Clough-Tocher Method . . . . .	21
<b>2.4.2.</b>	<b>Natural Neighbour Interpolation . . . . .</b>	<b>24</b>
<b>2.4.3.</b>	<b>Inverse Distance Weighted Interpolation . . . . .</b>	<b>27</b>
<b>2.4.4.</b>	<b>Radial Basis Function Interpolation . . . . .</b>	<b>28</b>
<b>2.5.</b>	<b>Synthetic Dataset Generation . . . . .</b>	<b>32</b>
<b>3.</b>	<b>CONNECTIVITY MAP . . . . .</b>	<b>35</b>
<b>3.1.</b>	<b>Fuzzy Sets and Fuzzy Topology . . . . .</b>	<b>35</b>
3.1.1.	<b>Fuzzy Sets . . . . .</b>	<b>35</b>
3.1.2.	<b>Fuzzy Topology . . . . .</b>	<b>37</b>
<b>3.2.</b>	<b><math>\alpha</math>-connectedness . . . . .</b>	<b>39</b>
<b>3.3.</b>	<b>Connectivity Map Algorithms . . . . .</b>	<b>40</b>
<b>4.</b>	<b>INTERPOLATION BASED ON IMAGE TOPOLOGY . . . . .</b>	<b>47</b>
<b>4.1.</b>	<b>Degree of Connectedness Map (DOCM) in Scattered Data . . . . .</b>	<b>48</b>
4.1.1.	<b>Natural Neighbour Concept in Scattered Data . . . . .</b>	<b>49</b>
4.1.2.	<b>Connectivity Map Algorithms for Scattered Data . . . . .</b>	<b>51</b>
<b>4.2.</b>	<b>Nonuniform Interpolation Utilizing DOCMs . . . . .</b>	<b>55</b>
<b>4.2.1.</b>	<b>Combination of Interpolation Values from DOCMs . . . . .</b>	<b>57</b>
4.2.1.1	<b>Selection of Interpolation Values from DOCMs . . . . .</b>	<b>58</b>
4.2.1.2	<b>Weighted Average of Interpolation Values from DOCMs . . . . .</b>	<b>62</b>

<b>5. RESULTS AND CONCLUSION . . . . .</b>	<b>66</b>
<b>5.1. Super-Resolution Results on Various Test Images . . . . .</b>	<b>66</b>
<b>5.1.1. Window (112x112) Test Image . . . . .</b>	<b>66</b>
<b>5.1.2. Military Vehicle (112x112) Test Image . . . . .</b>	<b>69</b>
<b>5.1.3. Resolution Chart (220x220) Test Image . . . . .</b>	<b>71</b>
<b>5.1.4. Church (240x240) Test Image . . . . .</b>	<b>73</b>
<b>5.1.5. Clock (240x240) Test Image . . . . .</b>	<b>75</b>
<b>5.1.6. Plate (160x260) Test Image . . . . .</b>	<b>77</b>
<b>5.1.7. Workers (268x364) Test Image . . . . .</b>	<b>79</b>
<b>5.1.8. Bridge (364x270) Test Image . . . . .</b>	<b>81</b>
<b>5.1.9. Sailboats (372x264) Test Image . . . . .</b>	<b>83</b>
<b>5.1.10. Lena (496x496) Test Image . . . . .</b>	<b>85</b>
<b>5.2. Computational Cost . . . . .</b>	<b>87</b>
<b>5.3. Conclusion . . . . .</b>	<b>88</b>
<b>REFERENCES . . . . .</b>	<b>89</b>
<b>CURRICULUM VITAE . . . . .</b>	<b>95</b>

## LIST OF TABLES

	<u>Page</u>
<b>Table 5.1.</b> Computation Time . . . . .	87
<b>Table 5.2.</b> Computation Profile . . . . .	87

## LIST OF FIGURES

	<u>Page</u>
<b>Figure 1.1.</b> SR grid . . . . .	2
<b>Figure 2.1.</b> Obervation Model . . . . .	7
<b>Figure 2.2.</b> Displacement between two LR images . . . . .	9
<b>Figure 2.3.</b> Correlation surface of cross-correlation and phase correlation .	11
<b>Figure 2.4.</b> Empty circle property of Delaunay triangulation . . . . .	17
<b>Figure 2.5.</b> A Voronoi diagram and its associated Delaunay triangulation .	18
<b>Figure 2.6.</b> Convex hull . . . . .	19
<b>Figure 2.7.</b> Linear interpolation inside a triangle . . . . .	20
<b>Figure 2.8.</b> Clough-Tocher Triangle . . . . .	22
<b>Figure 2.9.</b> 3D view of the control points of a third degree (cubic) Bézier triangle . . . . .	23
<b>Figure 2.10.</b> Three adjacent Bézier triangular patches (top view) . . . . .	24
<b>Figure 2.11.</b> Voronoi diagram of scattered dataset and newly created Voronoi tile around interpolation point . . . . .	25
<b>Figure 2.12.</b> Behaviour of IDW for various power parameters . . . . .	27
<b>Figure 2.13.</b> 1D example of radial basis function interpolation . . . . .	30
<b>Figure 2.14.</b> Sum of weighted basis functions from Figure 2.13 . . . . .	30
<b>Figure 2.15.</b> Gaussian RBF interpolation results with varying model parameter . . . . .	32
<b>Figure 2.16.</b> Sensor point spread function model . . . . .	33
<b>Figure 2.17.</b> Synthetically generated LR images of a car license plate . . . .	34
<b>Figure 2.18.</b> Synthetically generated LR images of a car license plate with additive Gaussian noise . . . . .	34

<b>Figure 3.1.</b>	Two paths connecting pixels $p$ and $q$ . . . . .	39
<b>Figure 3.2.</b>	4-connectivity . . . . .	41
<b>Figure 3.3.</b>	Connectivity map with different center pixels in a 7x7 window	43
<b>Figure 3.4.</b>	Connectivity map with different center pixels in a 7x7 window (continued) . . . . .	44
<b>Figure 3.5.</b>	Degree of Connectedness Maps of ridge and valley images . . .	45
<b>Figure 4.1.</b>	Unit squares filling space . . . . .	49
<b>Figure 4.2.</b>	Voronoi diagram . . . . .	50
<b>Figure 4.3.</b>	Delaunay triangulation and Voronoi diagram . . . . .	51
<b>Figure 4.4.</b>	Natural neighbours of a sample in a scattered dataset . . . .	51
<b>Figure 4.5.</b>	Operation windows in nonuniform super-resolution grid . . . .	53
<b>Figure 4.6.</b>	Degree of Connectedness Maps of irregularly sampled valley and ridge images . . . . .	55
<b>Figure 4.7.</b>	Center pixel in an interpolation window . . . . .	57
<b>Figure 4.8.</b>	Registered LR pixels of a sharp rising edge image . . . . .	58
<b>Figure 4.9.</b>	Averaged sums of interpolation values computed on original, DOCMb and DOCMd windows along edge direction . . . . .	59
<b>Figure 4.10.</b>	Averaged sums along edge direction of mean of interpolation values computed on DOCMb and DOCMd windows added to Figure 4.9 . . . . .	60
<b>Figure 4.11.</b>	Averaged sums along the edge direction of interpolation re- sults obtained using selective processing based on Equation 4.4 added to Figure 4.9 . . . . .	61
<b>Figure 4.12.</b>	Reconstructed ‘plate’ image using cubic, IDW and IDW <sub>sel</sub> . .	61
<b>Figure 4.13.</b>	Sobel edge detector on reconstructed ‘plate’ image using cubic, IDW and IDW <sub>sel</sub> . . . . .	62

<b>Figure 4.14.</b>	Averaged sums of interpolation values computed as weighted average of DOCMb and DOCMd window results along edge direction using different $\alpha$ parameters . . . . .	65
<b>Figure 5.1.</b>	HR ‘window’ image and one of the LR images . . . . .	67
<b>Figure 5.2.</b>	Super-resolution results of ‘window’ test image . . . . .	68
<b>Figure 5.3.</b>	HR ‘military vehicle’ image and one of the LR images . . . . .	69
<b>Figure 5.4.</b>	Super-resolution results of ‘military vehicle’ test image . . . . .	70
<b>Figure 5.5.</b>	HR ‘resolution chart’ image and one of the LR images . . . . .	71
<b>Figure 5.6.</b>	Super-resolution results of ‘resolution chart’ test image . . . . .	72
<b>Figure 5.7.</b>	HR ‘church’ image and one of the LR images . . . . .	73
<b>Figure 5.8.</b>	Super-resolution results of ‘church’ test image . . . . .	74
<b>Figure 5.9.</b>	HR ‘clock’ image and one of the LR images . . . . .	75
<b>Figure 5.10.</b>	Super-resolution results of ‘clock’ test image . . . . .	76
<b>Figure 5.11.</b>	HR ‘plate’ image and one of the LR images . . . . .	77
<b>Figure 5.12.</b>	Super-resolution results of ‘plate’ test image . . . . .	78
<b>Figure 5.13.</b>	HR ‘workers’ image and one of the LR images . . . . .	79
<b>Figure 5.14.</b>	Super-resolution results of ‘workers’ test image . . . . .	80
<b>Figure 5.15.</b>	HR ‘bridge’ image and one of the LR images . . . . .	81
<b>Figure 5.16.</b>	Super-resolution results of ‘bridge’ test image . . . . .	82
<b>Figure 5.17.</b>	HR ‘sailboats’ image and one of the LR images . . . . .	83
<b>Figure 5.18.</b>	Super-resolution results of ‘sailboats’ test image . . . . .	84
<b>Figure 5.19.</b>	HR ‘lena’ image and one of the LR images . . . . .	85
<b>Figure 5.20.</b>	Super-resolution results of ‘lena’ test image . . . . .	86

## GLOSSARY OF SYMBOLS AND ABBREVIATIONS

$\Delta$	Delta
$\epsilon$	Epsilon
$\gamma$	Gamma
$\lambda$	Lambda
$\mu$	Mu
$\omega$	Omega
$\rho$	Rho
$\sigma$	Sigma
$\theta$	Theta
$\nabla^2$	Laplacian
$\mathcal{F}$	Fourier Transform
$\mathcal{F}^{-1}$	Inverse Fourier Transform
FT	Fourier Transform
SR	Super-Resolution
SRR	Super-Resolution Reconstruction
HR	High Resolution
LR	Low Resolution
$Tr(\cdot)$	Trace of a Matrix
DOM	Degree of Membership
DOC	Degree of Connectedness
DOCM	Degree of Connectedness Map
DOCMb	Degree of Connectedness Map for Bright Pixels
DOCMd	Degree of Connectedness Map for Dark Pixels

# 1. INTRODUCTION

## 1.1. Introduction

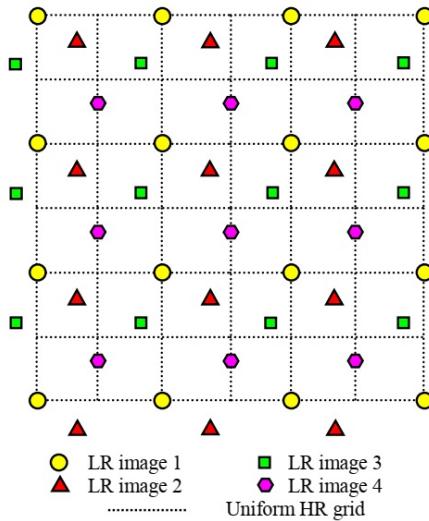
Images are typically composed of picture elements, i.e. pixels that are uniformly aligned to form a digital representation of real world scenes. One of the most important features associated with images is their resolution. Resolution directly affects the understanding of images. A high resolution (HR) image is desired for better human perception and performance of image understanding algorithms. For example, ship wake detection might be easier using HR satellite images. Similarly, a doctor can be more confident of the diagnosis with HR medical images.

Due to a number of factors, including the capabilities of image acquisition devices, optical equipment, coding and technology, resolution changes. In the last decade, image acquisition technology has advanced yielding very high resolutions. However, images and videos, acquired by using outdated technologies is a problem to tackle.

Traditionally, image resolution is limited by capturing device characteristics. Although it is nowadays possible to obtain an imaging system with large sensor and more pixels, in some applications it is preferred to use low resolution cameras due to cost, energy consumption and storage reasons. For example, surveillance cameras sacrifice resolution to some degree in lieu of long term operational benefits. In some other cases such as satellite imaging it may not be possible to replace image acquisition devices with imagers in desired resolutions. In medical imaging, one of the trade-offs is between extracting high resolution images and reducing the level of radiation patient receives since resolution is proportional to the amount of radiation used in the process [1]. To overcome resolution problems in these cases, one approach is to post-process captured low resolution images to obtain an HR image, in other words, hardware limitations are compensated by computational power.

*Super-resolution (SR) reconstruction* refers to the signal processing techniques

that generate high resolution (HR) images from multiple low resolution (LR) frames of the same scene that are acquired from slightly different perspectives. As imaging system takes subsequent shots, captured points on the scene changes as a result of the movement of either the camera or the scene relative to each other, for example, due to vibrations on a UAV or rotation of earth with respect to a satellite. These small movements can occur to yield sub-pixel shifts between scenes. Due to arbitrary movements between subsequent LR frames, gathered samples fall onto fractional valued locations, resulting in a super-resolution grid similar to the one shown in Figure 1.1. Here, each LR frame is assumed to be the decimated and aliased representations of the HR scene. The goal of super-resolution theory is to compute the HR scene, when only LR frames are available.



**Figure 1.1.** SR grid [2]

It is assumed that the information about the true scene, i.e. HR image, is present in subsequent LR frames. SR exploits this characteristic and reconstruction is possible only if there are shifts between frames.

Most SR methods can be divided into three stages. In first stage, relative positions of LR frames are estimated in a process referred to as image registration [3]. There are various methods to estimate shifts within LR frames [4, 5, 6, 7, 8, 9, 10]. Registered images form a nonuniform grid of points that are composed of LR pixels. Thus, it is necessary to estimate HR pixel values on the uniform grid.

There are various methods for the estimation of HR pixels. HR pixel estimation methods include frequency domain [11], projection onto convex sets [12, 13], regularized reconstruction [14, 15] and nonuniform interpolation [16] approaches. Finally, image restoration is generally applied to the resized image to remove blurring and noise. Some of these steps can also be implemented simultaneously, in [15], a maximum a posteriori (MAP) framework for jointly estimating image registration parameters and the high-resolution image is presented where estimation is performed on iterative optimization steps. This work will focus on the second, HR image estimation stage of SR reconstruction. Of the methods applied on this stage, nonuniform interpolation is one of the widely used techniques in the literature.

## 1.2. Motivation

Literature survey shows that Super-resolution theory generally focuses on the LR pixel locations with respect to uniformly distributed HR pixels and does not consider the connectivity information between sub pixels. Any LR pixel value, even if produced by a faulty registration process or a noise in the LR image, are taken into consideration without a filtering mechanism in conventional interpolation methods. Such problems generally do not produce unwanted results on flat intensity surfaces since the interpolation usually filters out the noisy element. However this factor adversely affects the edge preservation properties of nonuniform interpolation based SR techniques. Since edges constitute the most informative parts of images, any problem associated with edge performance has consequences on the image understanding algorithms and human perception. Motivation of this thesis is to solve the problem of unconnected LR pixels by detecting LR measurements belonging to the same object and performing interpolation using connectivity information between those samples. This way, it is aimed to minimize effects of unconnected measurements in interpolation grid and improve edge performance.

### 1.3. Research Contributions

Literature survey has shown that fuzzy image topology and connectedness concepts have not been utilized in super resolution reconstruction process. This thesis describes a novel algorithm that takes image topology into account during the HR image estimation stage of super-resolution reconstruction. The contributions of the research so far are as follows:

- **Connectivity Map for Scattered Data:** *Degree of connectedness map (DOCM)* computation algorithms in the literature are revised to work on nonuniform data. The resulting connectivity map disconnects scattered LR pixel peaks that are separated from the interpolation node by a valley in the brightness topology.
- **Interpolation Based on Image Topology Utilizing DOCMs:** Inverse distance weighted interpolation is applied on connectivity maps. DOCM properties in brightness and darkness topology are exploited in order to improve the results of interpolation along the edges in super-resolution imaging.

### 1.4. Thesis Outline

This thesis is organized as follows:

Chapter 2 discusses relevant background material in super-resolution reconstruction. The main stages and models in SR theory are presented. This chapter introduces Voronoi diagram and Delaunay triangulation concepts that are used in the computation of DOCM for scattered pixels and utilized by some of the nonuniform interpolation methods. Linear triangular, cubic triangular, natural neighbour, inverse distance weighting and radial basis function interpolation methods are detailed. Finally, a synthetic LR image dataset generation process is presented.

Chapter 3 summarizes the notions of fuzzy sets and fuzzy topology. The concept of degree of connectedness is given. Connected component labeling, degree of connectedness computation and degree of connectedness map computation algo-

rithms for uniform data in the literature are presented.

Chapter 4 provides degree of connectedness map computation algorithms that are revised to work on nonuniformly distributed data. New connectivity map computation algorithms are based on the concept of natural neighbourhood. Behaviour of degree of connectecness map in brightness and darkness topology is exploited in order to localize interpolation node along an edge profile. Based on estimated node locality, interpolation values computed using connectivity map windows in each topology are weighted in order to improve sharpness.

Chapter 5 demonstrates the performance of the proposed algorithm using synthetically generated test images. Results of the new algorithm are compared with the results of baseline cubic interpolation and conventional inverse distance weighted interpolation methods.

## 2. BACKGROUND IN SUPER-RESOLUTION

Earliest work on image super-resolution is a 1984 paper [11] by Tsai and Huang. They exploit shift and aliasing properties of the Fourier transform to relate the HR image with several shifted LR images. This approach, however, lacks the ability to cover complicated image degradation models. Since the frequency domain is known to have problems, focus of advanced SRR methods is on spatial domain in which it is generally possible to model all kinds of image degradations. Regardless of the domain, each method requires the values of shifts between LR images. Thus, it is necessary to accurately *register* LR images before any reconstruction process. This chapter summarizes basic methods used in these stages, initially observation model and image registration methods are given, then details for some reconstruction techniques are provided. Finally, a synthetic LR image generation process for the purpose of producing test data is given.

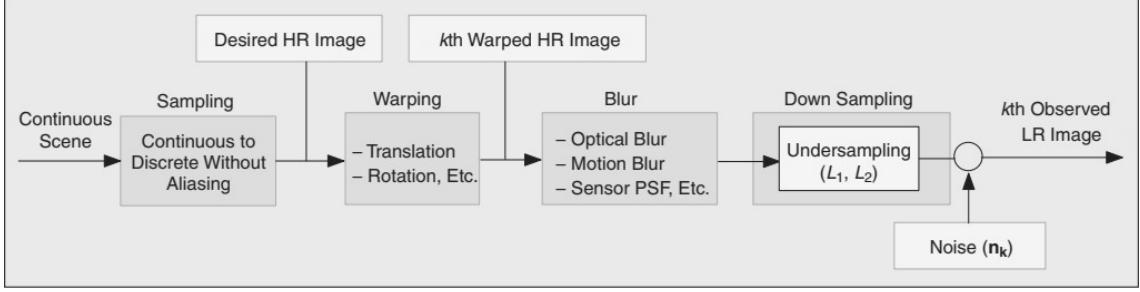
### 2.1. Observation Model

Imaging process can be thought as a camera capturing several LR frames which are downsampled versions HR scene with sub-pixels shifts between each other. Hence, an observation model relating the HR image to the LR frames is constructed as described in [17].

Let  $\mathbf{s}_k$  denote the k-th LR observation from the camera, ordered lexicographically as  $\mathbf{s}_k = [s_{1,k}, s_{2,k}, \dots, s_{M,k}]^T$  with  $M = N_1 \times N_2$  number of pixels in each LR frame. It is assumed that HR scene, denoted by  $\mathbf{z}$ , remains unchanged during the image capturing process. HR scene  $\mathbf{z}$  is also written as  $\mathbf{z} = [z_1, z_2, \dots, z_N]^T$ , where  $N = L_1 N_1 \times L_2 N_2$  with parameters  $L_1$  and  $L_2$  representing the down-sampling factors. With this notation, image observation model relating LR observations  $\mathbf{s}_k$  to HR scene  $\mathbf{z}$  is written as

$$\mathbf{s}_k = DB_k M_k \mathbf{z} + n_k, \quad k = 1, 2, \dots, p \quad (2.1)$$

where  $\mathbf{M}_k$  is the warp matrix representing motion information for  $k$ th frame,  $\mathbf{B}_k$  represents blur matrix (optical blur, motion blur etc.),  $\mathbf{D}$  is the subsampling matrix and  $\mathbf{n}_k$  is the lexicographically ordered noise vector. A block diagram depicting the observation model is given in Figure 2.1.



**Figure 2.1.** Obeservation Model [17]

The warp matrix represents translation and rotation motions. Since this motion is unknown,  $\mathbf{M}_k$  for each frame must be estimated from LR observations. In case of SR, blur matrix,  $B$  is approximated as a spatial averaging filter over HR image and assumed to be known [17]. Finally, the subsampling matrix,  $D$  creates aliased LR images scaled down by a factor of  $L_1$  and  $L_2$ . This model can be simulated with known warp and blur matrices and by adding random noise to generate synthetic data as described in Section 2.5.

## 2.2. Image Registration

Image registration is the process of matching image regions corresponding to the same scene in different images. In SRR, image registration is the stage where warp matrix  $\mathbf{M}_k$  is estimated. A comprehensive survey discussing various methods of 2D image registration is given in [3]. Focus of this work is on fusion phase of super-resolution reconstruction but a brief introduction to image registration is provided for coherence purposes. In literature, there are two main streams of image registration approaches [18]: intensity based methods and feature based methods. Intensity-based methods are based on maximization of a similarity measure to find true registration positions [5, 4]. These methods are also referred to as area-based

methods or template matching. The correlation coefficient is one of the most commonly used intensity patterns. Feature based methods [6], on the other hand, first extract salient objects in images. Then comes the matching step where registration parameters are estimated based on the locations of features in different frames.

### 2.2.1. Correlation Based Methods

In correlation based methods, a correlation coefficient for pairs is calculated as the frames are shifted and the points for which the maximum is achieved are set as the corresponding registration locations [3]. Fundamental limitation of correlation-based methods is its their inability to handle complicated geometric deformations (perspective transformations etc.) due to rectangular or circular search patterns used [18]. But it is applicable in SR case since sequential frames are from the same imaging modality and there are only small rotation and translation changes between LR images. Correlation based methods are studied in two categories: cross-correlation and phase correlation.

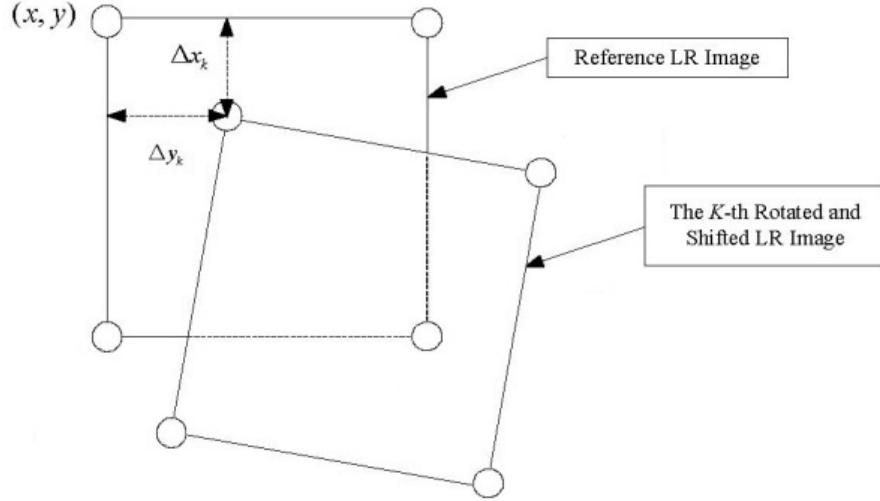
#### 2.2.1.1 Cross-correlation Method

Classical approaches use normalized cross-correlation (NCC) defined as [5]:

$$\gamma[\Delta x_k, \Delta y_k] = \frac{\sum_{x,y} (s_1[x, y] - \bar{s}_1)(s_k[x - \Delta x_k, y - \Delta y_k] - \bar{s}_k))}{\sqrt{\sum_{x,y} (s_1[x, y] - \bar{s}_1)^2 \sum_{x,y} (s_k[x - \Delta x_k, y - \Delta y_k] - \bar{s}_k)^2}} \quad (2.2)$$

where  $s_1$  is the reference image and  $s_k$  is the k-th shifted image.  $\bar{s}_k$  is the mean of the shifted image and  $\bar{s}_1$  is the mean of  $s_1[x, y]$ . In NCC, the pixels in each frame are normalized before comparing them by subtracting the frame mean and dividing by the standard deviation to avoid incorrect registration due to different illumination intensities between frames. Equation 2.2 includes only translation motion parameters of  $\Delta x_k$  and  $\Delta y_k$  but rotation parameters can also be easily included in this formulation as in [19, 20]. This measure of similarity is computed for each pair of the reference LR image and a rotated and shifted LR image and its maxi-

num is searched. To obtain subpixel accuracy required by SRR, the interpolation of the cross-correlation values must be used or LR images must be interpolated before computing the similarity measure (intensity interpolation) [21].



**Figure 2.2.** Displacement between two LR images

### 2.2.1.2 Phase Correlation Method

Another correlation-based technique is the phase correlation first proposed by Kuglin and Hines in 1975 [4]. It exploits the relationship between translation in the spatial domain and its corresponding effect in the frequency domain. Let  $f(x, y)$  and  $g(x, y)$  be shifted representations of the same scene, i.e.  $f(x, y) = g(x - \Delta x_k, y - \Delta y_k)$ . According to the Fourier shift property

$$F(u, v) = G(u, v) \cdot e^{-j(u\Delta x_k + v\Delta y_k)} \quad (2.3)$$

then, the normalized cross-power spectrum is given by

$$\frac{F(u, v)G^H(u, v)}{|F(u, v)G^H(u, v)|} = \frac{G(u, v)G^H(u, v) \cdot e^{-j(u\Delta x_k + v\Delta y_k)}}{|G(u, v)G^H(u, v) \cdot e^{-j(u\Delta x_k + v\Delta y_k)}|} = e^{-j(u\Delta x_k + v\Delta y_k)} \quad (2.4)$$

By taking the inverse FT of Equation 2.4, phase correlation function,  $p(x, y)$ , is found to be a delta-function located at the offset of the reference frame to the

translated frame

$$p(x, y) = \mathcal{F}^{-1}\{\mathrm{e}^{-j(u\Delta x_k + v\Delta y_k)}\} = \delta(x - \Delta x_k, y - \Delta y_k) \quad (2.5)$$

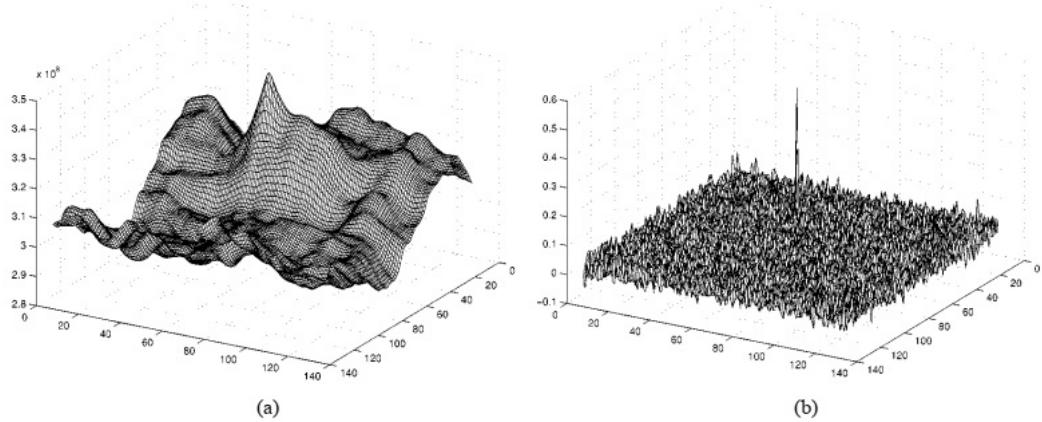
where  $\delta$  denotes the Dirac delta function and  $\mathcal{F}^{-1}$  is the inverse Fourier transform.

In theory, phase correlation surface should be 1 at the point of registration. However, in practice, phase correlation function is not an exact Dirac delta due to the presence of noise and existence of other complex types of displacements. Thus, the translation values are found by searching the maximum of  $p(x, y)$

$$(\Delta x_k, \Delta y_k) = \underset{(x,y)}{\operatorname{argmax}} p(x, y) \quad (2.6)$$

Phase correlation method is extended to handle rotation movements by using log-polar coordinate changes in which rotation is represented as a translational displacement along circumferential axis [22] or by using Fourier-Mellin Transform [23]. A simple way to obtain subpixel accuracy is through interpolation by zero-padding the cross-power spectrum before performing the inverse Fourier Transform. In this technique, accuracy is limited by the interpolation factor, i.e. how many times the spectrum is extended by padding. A different approach is suggested in [24], by including the phase correlation information of neighbouring pixels of main peak into displacement estimation. The reader is referred to [25] for a subpixel accuracy analysis of these phase correlation techniques.

Advantage of phase correlation over cross-correlation is the narrowness of phase correlation function peak and accompanied detection accuracy [24]. Figure 2.3 shows the correlation surfaces yielded by cross-correlation and phase correlation. The phase correlation produces a pointed peak at the point of registration. On the other hand, the cross-correlation provides a surface with widespread peaks. Phase correlation is also computationally less intensive than cross-correlation.



**Figure 2.3.** Correlation surface of (a) cross-correlation and (b) phase correlation [24]

### 2.2.2. Feature Based Methods

This approach is based on the extracted features from the images to achieve registration. They are recommended if the images contain distinctive object features. Scale-invariant feature transform (SIFT) algorithm proposed by Lowe [6] is the most widely used for feature extraction. SIFT features are invariant to translation and rotation which makes it suitable for super-resolution applications.

First and second order derivatives of a blurred image gives locations of edges and corners. This operation is referred to as Laplacian of Gaussian. Laplacian is normalized to achieve scale invariance:  $\sigma^2 \nabla^2 G$ , where  $\sigma$  is the scaling parameter of the Gaussian. But since Laplacian is computationally expensive, difference of Gaussian is used to approximate scale-normalized Laplacian of Gaussian. First stage of SIFT algorithm is scale-space extrema detection. For this purpose, several *octaves* of the original image are generated. Each octave is an image with half the size of the previous one. Then, a variable-scale Gaussian  $G(x, y, \sigma)$  is convoluted with these octaves to obtain the scale-space function:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \quad (2.7)$$

Two consecutive blurred images in an octave are subtracted to find difference of

Gaussian with various  $\sigma$  values for scale-space filtering purposes:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.8)$$

Resulting images are an approximation of scale invariant Laplacian of Guassian. After difference of Gaussian are found, local extrema are searched within images. Search is over scale and space, i.e. a sample is compared with all of its 8 neighbors in the current scale as well as 9 neighbours in the upper and lower scales. It is marked as a candidate keypoint if it is the smallest or largest value in this neighbourhood.

After potential keypoints are found, next stage is for achieving subpixel accuracy and refining results. To increase accuracy, Taylor series expansion of scale space function  $D(x, y, \sigma)$  is used to determine the interpolated location of the extremum:

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad (2.9)$$

If the intensity at the extrema is less than a threshold, it is discarded. Thus, keypoints with low contrast are eliminated. Difference of Gaussian gives higher response for edges since the gradient perpendicular to the edge are big and the other one (along the edge) is small. These are poor keypoint candidates because of their susceptibility to noise and they need to be removed. For this purpose, the ratio of principal curves at the keypoint location and scale is calculated:

$$\frac{Tr(\mathbf{H})^2}{|\mathbf{H}|} < \frac{(r+1)^2}{r} \quad (2.10)$$

where  $\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$ . Candidate keypoints with ratio of principal curves above this threshold (described in terms of  $r$ ) are discarded.

Third stage of SIFT algorithm is orientation assignment. In a neighbourhood around the keypoint location, gradient magnitudes and orientations are calculated:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.11a)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (2.11b)$$

As the neighbourhood function, a Gaussian weighted circular window is used where Gaussian variance is 1.5 times the scale of the keypoint. A histogram is created with 36 bins covering  $360^\circ$  of orientation. Histogram bin with the highest magnitude and any other bin with a magnitude above 80% that of the highest bin are converted into new keypoints with the same location and scale, but different orientations. Having multiple keypoints increases matching performance. In [6], keypoint detection is reported to be stable up to a  $40^\circ$  planar rotation.

Final stage of SIFT algorithm is creation of a vector to represent each keypoint referred to as the keypoint descriptor. A 16x16 pixel window, of which the keypoint is at the center, is considered. Gradient magnitudes and orientations in this window are taken. To achieve rotation independence, the keypoint orientation is subtracted from each gradient orientation in this window. Hence, each gradient orientation is defined relative to the keypoint orientation and gradient orientations will not change when the image is rotated. The gradient magnitudes around the keypoint are weighted with a Gaussian function with variance of half the width of the descriptor window. Then, this window is divided into 4x4 regions creating 16 sub-blocks. For each 4x4 sub-block, an 8 bin orientation histogram, with  $45^\circ$  resolution, is created. In the end, the total of 128 bin values are normalized and represented as a vector to form keypoint descriptor. The keypoint is uniquely identified by this feature vector. To achieve illumination independence, the vector values are thresholded and then renormalized. Details about thresholding parameter values used can be found in [6].

After features are extracted, the correspondence between the features detected in the reference LR frame and the rotated and shifted LR frame is established. For this purpose, closest match between images is identified using euclidean distance

from the keypoint descriptor vectors. These corresponding locations are then used to estimate the parameters of affine transformation model:

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} \quad (2.12)$$

where  $\Delta x_k$  and  $\Delta y_k$  are translation values and  $\theta_k$  is rotation angle between reference LR frame and k-th LR frame.  $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$  and  $\begin{bmatrix} x_k \\ y_k \end{bmatrix}$  are locations of the corresponding keypoints from each frame. A cost function based on this model is defined and minimized with respect to  $\Delta x_k$ ,  $\Delta y_k$  and  $\theta_k$ . A possible problem encountered during calculation of affine transformation parameters is that some of the obtained keypoint matches may not be correct and there may be outliers that do not fit the model. RANSAC algorithm can be used to eliminate these incorrect matches and improve estimation accuracy as described in [26, 27].

Other methods used for feature keypoint detection and feature description include Speeded Up Robust Features (SURF), ORB (Oriented FAST and Rotated BRIEF), KAZE and its accelerated version AKAZE. SURF is inspired by SIFT. SURF also approximates the Laplacian of Gaussian but instead with box filters and wavelet responses are used in orientation assignment steps [7]. SIFT and SURF algorithms are patented and require a license fee for usage. ORB is introduced as a free alternative to SIFT or SURF [8]. KAZE features are detected and described in nonlinear scale-space which makes blurring locally adaptive. This preserves the boundaries in an image while reducing noise, resulting in better distinctiveness [9]. Describing KAZE features is computationally expensive due to diffusion-filtering step used to generate nonlinear scales. To reduce complexity, a more efficient Fast Explicit Diffusion scheme is adopted. Moreover, descriptor is based on highly efficient Modified Local Difference Binary (MLDB). New algorithm is referred to as AKAZE [10]. A benchmark of mentioned techniques is available in [28]. These algorithms are all available in OpenCV (SIFT and SURF in non-free contrib module).

In this work, LR images are assumed to be shifted by only rotational and translational motions, which covers all possible shifts in 2D plane. It is a sufficient model when camera or scene movements are small enough to ignore scaling effects. Since the focus of this work is on the fusion phase, image registration algorithms are not implemented. Before executing the proposed algorithm for fusion phase, existing libraries can be used to estimate shifts between LR frames or artificial test data composed of sequences with known motion parameters can be chosen (i.e. LR frames can be generated from an HR image with given motion values).

Most registration algorithms require a series of operations, which sometimes yields wrong results due to complex shifts and rotations. Registration results is an important factor in the performance of SRR algorithms since the results generated by registration techniques are often used without any selection mechanism. Exclusion or inclusion of LR image pixel values depending on the relationship between other pixels can enhance SRR outputs.

### 2.3. Super-Resolution in Frequency Domain

Shift and aliasing properties of the Fourier Transform can be used to describe the relationship between the shots of the same scene if they are displaced by translational movements. In [11], authors exploit the Fourier Transform property to generate a single HR image from sequence of shifted LR images of the same scene. Let  $z(x, y)$  denote a continuous scene. Translation movements generate images of the same scene from slightly different perspectives:  $z_k(x, y) = z(x + \Delta x_k, y + \Delta y_k)$  with  $k = 1, 2, \dots, p$ .  $\Delta x_k$  and  $\Delta y_k$  are a priori fractional shifts. Continuous Fourier Transform of  $z$  is given by  $Z(u, v)$  and those of each shifted scene  $z_k$  is given by  $Z_k(u, v)$ . Then,

$$Z_k(u, v) = e^{j2\pi(\Delta x_k u, y + \Delta y_k v)} Z(u, v) \quad (2.13)$$

Shifted continuous scene  $z_k$  is sampled with sampling period  $T$  to acquire LR frame  $s_k[x, y] = z_k(xT + \Delta x_k, yT + \Delta y_k)$ .  $Z_k(u, v)$  is assumed to be bandlimited to  $L_1\pi/T$

and  $L_2\pi/T$  in respective axes. Aliasing property is used to relate the CFT of the HR image and the DFT of the k-th LR frame [17]:

$$S_k[u, v] = \frac{1}{T^2} \sum_{x=0}^{L_1-1} \sum_{y=0}^{L_2-1} Z_k \left( \frac{2\pi}{T} \left( \frac{u}{N_1} + x \right), \frac{2\pi}{T} \left( \frac{v}{N_2} + y \right) \right) \quad (2.14)$$

Combining Equations 2.13 and Equation 2.14 and by using lexicographic ordering for the indices x and y, a matrix vector form is obtained:

$$\mathbf{S} = \Phi \mathbf{Z} \quad (2.15)$$

where  $\mathbf{S}$  is a  $p \times 1$  column vector with the k-th element being the DFT coefficient  $S_k[u, v]$ ,  $\mathbf{Z}$  is a  $L_1 L_2 \times 1$  column vector containing the samples of the unknown CFT of  $z(x, y)$  and  $\Phi$  is a  $p \times L_1 L_2$  system matrix. Hence, in order to obtain the HR image,  $\Phi$  must be determined and an inverse problem defined by set of linear equations in Equation 2.15 has to be solved. The observation model used in this formulation is restricted to translational movements and it assumes a noise-free acquisition [29]. Beacuse of this restriction, focus of advanced SRR methods generally has been on spatial domain.

## 2.4. Super-Resolution by Nonuniform Interpolation

Image interpolation is the practice of increasing resolution before viewing or further processing tasks. After relative displacements between LR frames are calculated in image registration step, super-resolution grid shown Figure 1.1 is obtained where locations of each LR pixel are known. Up-scaling of LR frames onto HR grid is not possible via conventional methods due to nonuniform nature of the pixel positions. Techniques referred to as *scattered data interpolation* needs to be utilized. Scattered interpolation techniques can be broadly categorized as triangulation based interpolation, natural neighbour interpolation, inverse distance weighted interpolation and radial basis function interpolation [16]. Methods based on triangulation and inverse distance weights are given particular importance due to their relevance

to the proposed algorithm in this thesis.

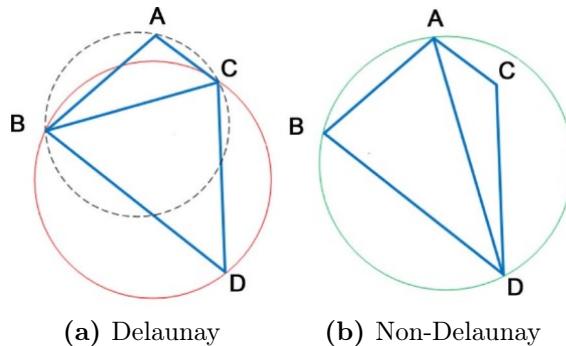
### 2.4.1. Triangulation Based Methods

In triangulation based methods, the nonuniform data set is triangulated and then an interpolation scheme is performed within each triangle using various weights on vertices. Hence, triangulation based interpolation methods are always local.

#### 2.4.1.1 Delaunay Triangulation

Let  $P = \{P_1, P_2, \dots, P_n\}$  denote a set of points in a plane, triangulation of  $P$  is a planar subdivision whose faces are triangles with vertices from the set  $P$  [30]. There may be many possible triangulations given a point set. In case of interpolation, however, it is better to avoid triangles with very small acute angles. One way to achieve this criteria is to use Delaunay triangulation, which maximizes the minimum interior angle of all triangles by utilizing MaxMin Angle Lemma [31].

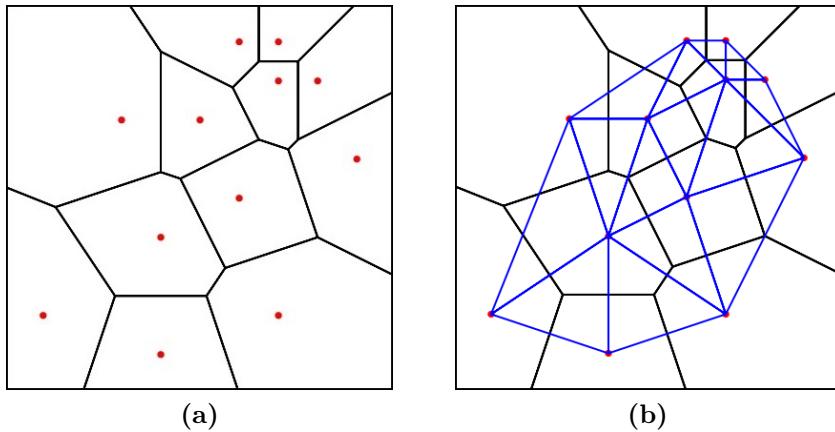
Delaunay triangulation requires to exclude any other point within the circumcircle of each triangle. BC edge that satisfies the empty circle property, as in Figure 2.4.a, is referred to as Delaunay edge. As shown in Figure 2.4.a, circumcircles of ABC and BCD triangles does not contain any point belonging to other triangle. However, Figure 2.4.b shows two triangles ABD and ADC, which are not Delaunay, because of the fact that the circumcircle of ABD includes point C, which belongs to ACD triangle.



**Figure 2.4.** Empty circle property of Delaunay triangulation [32]

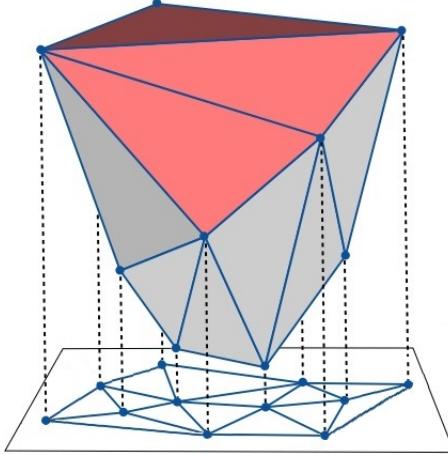
This definition of Delaunay triangulation is also true in reverse as indicated by Delaunay Lemma: if every edge in a triangulation is locally Delaunay, then the triangulation is a Delaunay triangulation [31]. Thus, a local property of each triangle leads to a global condition and this lemma enables the construction of simple algorithms that can generate Delaunay triangulation.

Delaunay triangulation is also defined as the straight-line dual of the Voronoi diagram [30]. If  $P = \{P_1, P_2, \dots, P_n\}$  is a set of points in a plane, Voronoi diagram decomposes the 2D space around each point  $P_i$  into convex polygons (referred to as Voronoi cells) such that an arbitrary point within the polygon is closer to  $P_i$  than any other point in  $P$ . Straight-line dual of a plane graph is then obtained by connecting two points sharing an edge in Voronoi diagram [33]. Figure 2.5 depicts an example Voronoi diagram and associated Delaunay triangulation.



**Figure 2.5.** (a) A Voronoi diagram and (b) its associated Delaunay triangulation [34]

There is also a relationship between convex hulls and Delaunay triangulations which can be exploited to construct the Delaunay triangulation of a set of points. In geometry, convex hull of a set of points is the smallest convex that includes those points. When the points in  $\mathbb{R}^2$  are lifted onto a paraboloid in  $\mathbb{R}^3$ , the Delaunay triangulation of original points is the downward projection of the ridges of the lower convex hull of lifted points [35]. A set of points and their lifting is shown in Figure 2.6. Downward-faces of generated convex hull is shaded in gray. Projection of the lower convex hull edges in Figure 2.6 on 2D surface is the Delaunay triangulation.



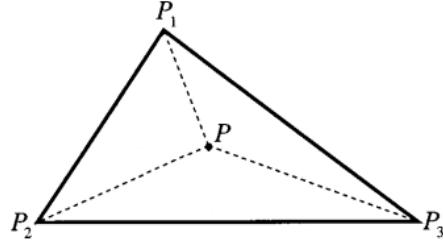
**Figure 2.6.** Convex hull

Availability of efficient algorithms for computing convex hull makes this approach a viable alternative to obtain Delaunay triangulation since only a simple projection step is required additionally. Another way to generate Delaunay triangulation, proposed by Charles Lawson [36], is through imposing empty circle property locally for each triangle. By Delaunay Lemma, resulting triangulation will be globally Delaunay. This method is referred to as edge flip algorithm and transforms any triangulation to conform the Delaunay properties in  $O(n^2)$  time.

Computer programs packages to generate and modify Delaunay triangulations and Voronoi diagrams are available as functions in MATLAB and Python (SciPy). Delaunay triangulation and these functions have a pivotal role in implementation of connectivity algorithms for nonuniform data.

#### 2.4.1.2 Linear Triangular Interpolation

Linear triangular interpolation is based on *barycentric* coordinates introduced by August Ferdinand Möbius [37]. Description of the method presented here is based on the work of Isaac Amidror [16]. Let points  $P_1$ ,  $P_2$  and  $P_3$  be vertices of a triangle located in 2D plane at locations  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$ , respectively. Let the values corresponding to these points be  $z_1$ ,  $z_2$  and  $z_3$ , respectively. Let  $P$  be an arbitrary interpolation point located at  $(x, y)$  whose value is to be estimated.



**Figure 2.7.** Linear interpolation inside a triangle

Location of any point  $P$  within the triangle  $P_1P_2P_3$  can be expressed as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = a_1 \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + a_2 \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + a_3 \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \quad (2.16)$$

Here, weights  $a_1, a_2, a_3$  are named barycentric coordinates and they add up to 1:

$$a_1 + a_2 + a_3 = 1 \quad (2.17)$$

Value of point  $P$  located at  $(x, y)$ , denoted by  $z$  is the average of values  $z_1, z_2$  and  $z_3$  weighted by barycentric coordinates:

$$z = a_1 z_1 + a_2 z_2 + a_3 z_3 \quad (2.18)$$

Using Equations 2.16 and 2.17 and with known location of the vertices, weights can be calculated as follows:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.19)$$

Weights are substituted in Equation 2.18 and the value of the point  $P$  is thus estimated.

### 2.4.1.3 Cubic Triangular Interpolation: The Clough-Tocher Method

Linear interpolation approaches are  $C^0$  continuous, meaning the interpolant itself is continuous across the triangle boundaries but its second derivative is not [16]. Hence, this method produces results that are not smooth across triangle boundaries.

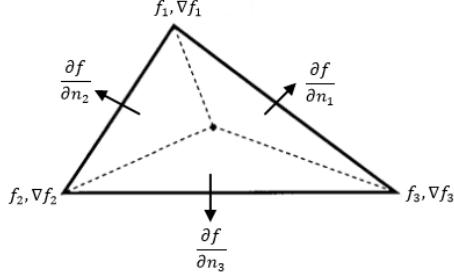
In order to achieve smoothness over the triangle boundaries, a function of order higher than 1 needs to be used in the interpolation. A popular method of choice is the Clough-Tocher method [38]. The Clough-Tocher interpolants were initially developed as a tool for finite element method of numerical analysis, hence referred to in the literature as such. But later they were demonstrated to be also useful in scattered data interpolation [39]. The Clough-Tocher method is based on cubic polynomials. Even though the interpolant is of the third order, the Clough-Tocher method is  $C^1$  continuous since it ensures smoothness only in the first derivative.

The Clough-Tocher method requires the computation of gradient and function values at each vertex and the normal directional derivative information along the edges. For 2D case, gradient is computed using of two partial derivatives along x and y axis. In order to achieve smoothness, these function values need to be equal across boundaries. Hence, they impose 12 constraints, 1 for each vertex value, 1 for each edge normal derivative and 2 (since gradient is computed along two edge directions) for each vertex gradient. There are 12 different constraints are imposed on the interpolation function. However, cubic interpolation polynomial:

$$p(x, y) = a_1x^3 + a_2x^2y + a_3xy^2 + a_4y^3 + a_5x^2 + a_6xy + a_7y^2 + a_8x + a_9xy + a_{10} \quad (2.20)$$

has only 10 degrees of freedom. Therefore, in a 12 axis search space, it is impossible to perform estimation using a polynomial function with 10 degrees of freedom. The only possible solution is to obtain additional the degrees of freedom. In order to increase degrees of freedom, triangle of interest (domain triangle or macro-triangle) is divided into three mini-triangles, as shown in Figure 2.8, by drawing a line from each vertex to the centroid [40]. Clough-Tocher belongs to a group referred to as a

split-triangle interpolation techniques.



**Figure 2.8.** Clough-Tocher Triangle

The aim of this split is to produce a cubic patch, known as Bernstein-Bézier patch, that satisfies  $C^1$  continuity condition over each mini-triangle shown in Figure 2.8. Here,  $(f, \nabla f)$  are sample values and gradient values at each vertex.  $\frac{\partial f}{\partial n}$  is the *the cross-boundary normal derivative*. It is more viable to use derivatives normal to the edges because the adjacent macro-triangles would then have the same data along the shared edge [41]. A Bernstein polynomial, of degree  $n$ , to be placed over each mini-triangle is defined as [40]:

$$B_{i,j,k}^n(t) = \frac{n!}{i!j!k!} u^i v^j w^k \quad (2.21)$$

where  $u, v, w$  are the barycentric coordinates. Then, Bernstein-Bézier polynomial of degree  $n$  is given by:

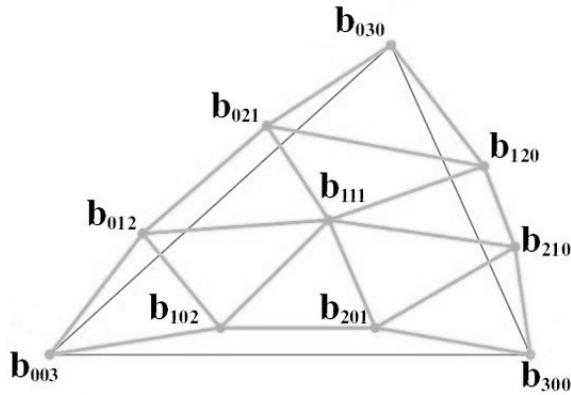
$$b^n(t) = \sum_{\substack{0 \leq i,j,k \\ i+j+k=n}} b_{i,j,k} B_{i,j,k}^n(t) \quad (2.22)$$

where  $b_{i,j,k}$  are called Bézier ordinates (or Bézier coefficients). The Bézier polynomials are formed as follows::

$$\begin{aligned} b^3(u, v, w) = & b_{3,0,0}u^3 + 3b_{2,1,0}u^2v + 3b_{1,2,0}uv^2 + b_{0,3,0}v^3 \\ & + 3b_{0,2,1}v^2w + 3b_{0,1,2}vw^2 + b_{0,0,3}w^3 \\ & + 3b_{1,0,2}uw^2 + 3b_{2,0,1}u^3w + 6b_{1,1,1}uvw \end{aligned} \quad (2.23)$$

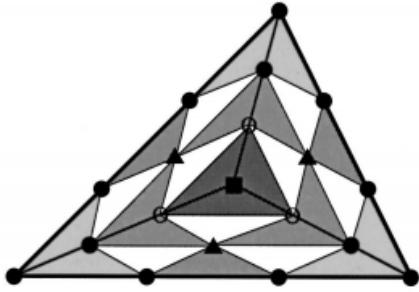
For each mini-triangle, i.e. one of the three triangles in Figure 2.8,  $b_{3,0,0}$ ,  $b_{0,3,0}$  and

$b_{0,0,3}$  correspond to the values at the vertices, and the seven other Bézier ordinates are the control points. All ordinates except  $b_{1,1,1}$  are referred to as boundary ordinates (since they are all on an edge or vertex). A lifted Bézier patch on top of one of the mini-triangles is shown in Figure 2.9. Each Bézier ordinate in 2D is analogous to control points of a 1D Bézier curve.



**Figure 2.9.** 3D view of the control points of a third degree (cubic) Bézier triangle

Bézier ordinate values at macro-triangle vertices are known but the boundary ordinate values are missing [16]. This missing data is approximated by the first order gradients at the vertices. For  $C^1$  continuity, micro-triangle edges along the border of each mini-triangle must be coplanar [38]. This gives a constraint derived from having common partial derivative values on boundary Bézier ordinates of each mini-triangle [42]. An original macro-triangle and three adjacent Bézier patches over mini-triangles are shown in Figure 2.10. In the figure, shaded micro-triangles must be coplanar with each other. Values of Bézier ordinates displayed by ●'s are determined from the sample values and the partial derivatives at the macro-triangle vertices. Bézier ordinates indicated by ▲'s are determined from *the cross-boundary normal* derivatives at the midpoint of each edge. Ordinates denoted by ○'s are determined as weighted average of neighboring ordinates (two ▲'s and a ●). Finally, weighted average of three surrounding ○ ordinates gives the value of the ordinate denoted by ■ [38].

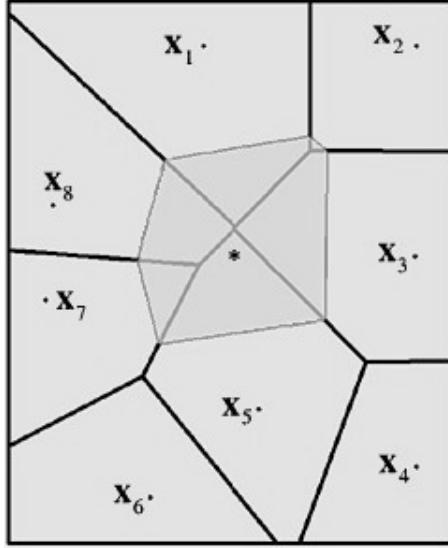


**Figure 2.10.** Three adjacent Bézier triangular patches (top view) [16]

Cubic triangulation interpolation is the baseline technique used in the fusion phase of SRR. It is a local method, and calculations are based on only the triangulated domain. This feature simplifies the calculations and a computationally efficient algorithm is achieved. Even large scattered data sets can be interpolated in a quick manner. It gives a smoother interpolating surface than linear interpolation based methods. However, third order polynomial used is not good at modeling the large intensity variations (i.e. edges), hence it produces poor results around edges [43]. It is more susceptible to noise since function is approximated using small number of pixels.

#### 2.4.2. Natural Neighbour Interpolation

Natural neighbour interpolation technique is first introduced by Robin Sibson [44, 45]. It is a local interpolation method and is based on Voronoi diagram of the underlying nonuniform data. Hence, this method requires a pre-processing step of Voronoi diagram construction. Initially a Voronoi diagram is created for the whole dataset. Then a new point is added to the diagram, the point at the center of Figure 2.11, modifying the existing polygon by creating a new tile around itself. The points whose Voronoi cells are chopped off upon insertion of the new point are referred to as the natural neighbours of the new point [46].



**Figure 2.11.** Voronoi diagram of scattered dataset and newly created Voronoi tile around interpolation point

Interpolation point value is determined from this modified Voronoi diagram. In natural neighbour interpolation, weights on neighboring data values are proportional to the fractional contribution to the area of newly created tile. These contributions in turn depend on the location of the interpolation point. Fundamental equation of natural neighbour interpolation for any point  $\mathbf{x}$  is:

$$z(\mathbf{x}) = \sum_{i=1}^n \lambda_i(\mathbf{x}) f(\mathbf{x}_i) \quad (2.24)$$

where  $f(\mathbf{x}_i)$  are known sample values at locations  $\mathbf{x}_i$  and weights  $\lambda_i$  are defined as:

$$\lambda_i(\mathbf{x}) = \frac{\text{area contributed by tile } i}{\text{area of newly created tile}} \quad (2.25)$$

Weights used in the natural neighbour interpolation are between 0 and 1 and add up to 1. The version of natural neighbour interpolation described by Equation 2.24 is only  $C^0$  continuous. To reach a  $C^1$  continuous solution, gradients at sample points must also be taken into account as explained in [16]. Sample values in Equation 2.24 are replaced by  $g_i(\mathbf{x}) = f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)$ . Weights are also

updated:

$$w_i(\mathbf{x}) = \sum_{i=1}^n \frac{\lambda_i(\mathbf{x})d(\mathbf{x}, \mathbf{x}_i)^{-1}}{\sum_i \lambda_i(\mathbf{x})d(\mathbf{x}, \mathbf{x}_i)^{-1}} \quad (2.26)$$

to obtain matching slopes at sample locations. Then, the final equation becomes:

$$z(\mathbf{x}) = w_i(\mathbf{x})g_i(\mathbf{x}) = \sum_{i=1}^n \frac{\lambda_i(\mathbf{x})d(\mathbf{x}, \mathbf{x}_i)^{-1}}{\sum_{i=1}^n \lambda_i(\mathbf{x})d(\mathbf{x}, \mathbf{x}_i)^{-1}} (f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i)) \quad (2.27)$$

where  $d(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{x}_i)^2}$  is the Euclidean distance between i-th sample location and the interpolation point. In natural neighbour interpolation, as the query point approaches to the sample point, the weight of that sample increases and reaches to 1.0. Weights of other samples decrease and this whole process is continuous. An advantage of natural neighbour interpolation is its robustness to biases introduced by sample clustering [45]. Sample density variations are tolerated by using weights based on fractional area contributions. Natural neighbour interpolation implemented in *griddata* function of MATLAB is  $C^1$  continuous whereas Python implementation in Scipy is  $C^0$  continuous as the stated implementation is that of Discrete Sibson Interpolation [47].

Natural neighbourhood interpolation is not used in this work but the concept of natural neighbourhood is utilized in modification of connectivity map computation algorithms for scattered data.

### 2.4.3. Inverse Distance Weighted Interpolation

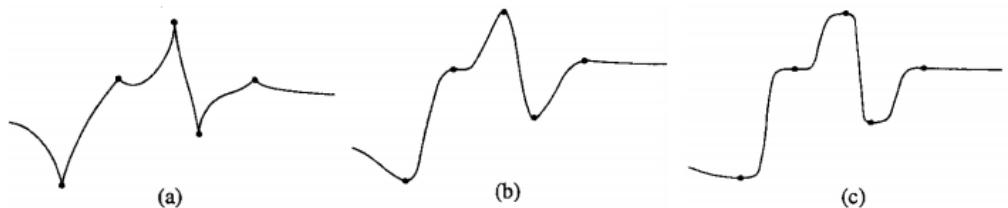
Inverse distance weighting (IDW) method is also referred to as Shepard's method [48]. IDW in this classical form is basically a zeroth-order kernel regression method where kernel function is the inverse distance function [49, 50]. Shepard's proposition to find an interpolated value at point  $[x, y]$  is:

$$z(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) f_i(\mathbf{x}) = \sum_{i=1}^n \frac{h_i(\mathbf{x})}{\sum_{i=1}^n h_i(\mathbf{x})} f_i(\mathbf{x}) \quad (2.28)$$

with

$$h_i(\mathbf{x}) = \frac{1}{d_i(\mathbf{x})^p}$$

where  $d_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2}$  is Euclidean distance between i-th sample location and the interpolation point and  $p \geq 1$  is a chosen exponent. Clearly,  $\sum_{i=1}^n w_i(\mathbf{x}) = 1$ . It is a global method, which means that every pixel in the interpolation grid has a weight on the interpolation result. When  $p \geq 2$ , data points close to the interpolation point has more weight on the result. As  $p$  gets smaller than 2, points farther away from the interpolation point tend to have more emphasis on the output. In 2D case, IDW has slope discontinuities at sample point coordinates if  $p \leq 2$ . When  $p \geq 2$ , interpolation results become less smooth [16]. 1D counterparts of these patterns are displayed in Figure 2.12. Therefore, 2 is known to be a good choice for the power parameter.



**Figure 2.12.** Behaviour for various power parameters: (a)  $p = 1$ , (b)  $p = 2$ , (c)  $p = 10$  [16]

An advantage of IDW is that it has no tunable parameter other than the power parameter, which is typically taken as 2. IDW tends to pick local variations better when sampling is sufficiently dense [51]. When  $r^2$  number of LR frames are available,

where  $r$  is the scaling factor, interpolation grid can be considered sufficiently dense for the IDW method.

The major disadvantage of IDW is that it is a global method and it is too sensitive to distant pixels with large values in the image [52]. IDW produces more blurred (but less noisy) images since the interpolation result is a weighted average. Although the conventional IDW method utilizes all points in the image, in practical implementations, the interpolation result is generally computed within a limited neighborhood. In practice neighbourhood size is determined by a radius parameter or by a parameter indicating the total number of points to be included in the computation [53]. However this requires a decision to be made about how to select these parameters.

In this work, IDW is selected to demonstrate how interpolation can be enhanced by taking connectivity between pixels into consideration. IWD generally uses more pixels than polynomial-based methods which makes it suitable to benefit more from connectivity information in a large window. When producing results, neighborhood size is selected to cover an equal to or larger than 7x7 pixel window. Negative effects of distant outliers are handled by using connectivity maps.

#### 2.4.4. Radial Basis Function Interpolation

Radial Basis Function (RBF) interpolation is another type of global interpolation method. RBF interpolation methods are known to supersede inverse distance weighting methods in some applications involving sparse data such as groundwater level [54]. RBF interpolation is applied to super-resolution problems [55]. However, they require special treatments such as adaptive parameter tuning [56]. In many cases, choice of basis functions used is not easy to make and even those functions that are known to work well are not always easy to justify mathematically [16]. Although this thesis does not focus on RBF based methods, information on RBF interpolation is also presented for the sake of thoroughness. RBF interpolation is first introduced by Hardy [57] and can be considered as a multivariate version of

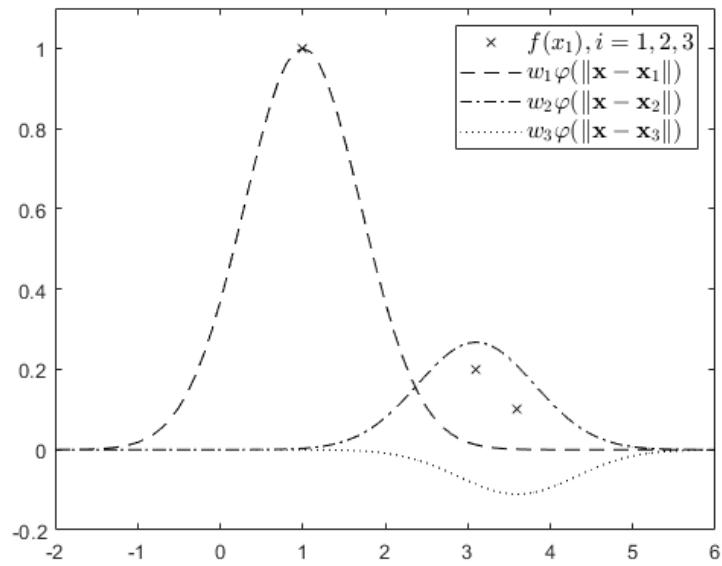
spline interpolation technique. In RBF, interpolant is of the form of sum of weighted functions called radial basis functions. Put in mathematical terms:

$$z(\mathbf{x}) = \sum_{i=1}^n w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (2.29)$$

where  $z(\mathbf{x})$  actually crosses over interpolation nodes, that is  $z(\mathbf{x}_i) = f(\mathbf{x}_i)$  for  $i = 1, 2, \dots, n$ . In matrix notation:

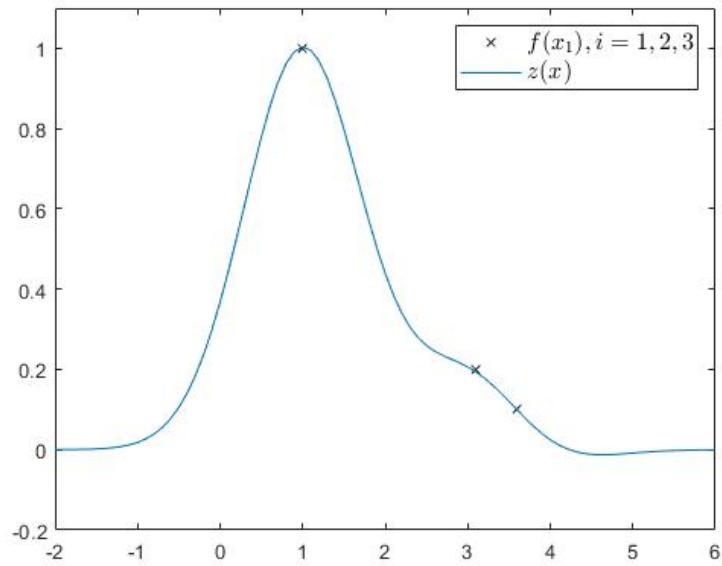
$$\begin{bmatrix} \varphi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \dots & \varphi(\|\mathbf{x}_n - \mathbf{x}_1\|) \\ \varphi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \varphi(\|\mathbf{x}_n - \mathbf{x}_2\|) \\ \vdots & \vdots & & \vdots \\ \varphi(\|\mathbf{x}_1 - \mathbf{x}_n\|) & \varphi(\|\mathbf{x}_2 - \mathbf{x}_n\|) & \dots & \varphi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_n) \end{bmatrix}$$

Solving the equation above gives the weights  $w_1, w_2, \dots, w_n$ . When multiplied with the basis function  $\varphi$ , these weights generate hills (when the weight is positive) or hollows (when the weight is negative) centered at sample point locations. Note that solution of this linear set of equations is unique and as a result obtained interpolant is also unique. For a 1D example, when set of linear equations of points at locations 1, 3.1, 3.6 with sample values 1.0, 0.2, 0.1 are solved using basis function  $\varphi(r) = e^{-r^2}$ , resulting weights scale the basis functions to be located on top of sample locations as shown in Figure 2.13 [58].



**Figure 2.13.** 1D example of radial basis function interpolation

When we sum all three basis functions, we obtain the interpolation function passing through all three sample points.



**Figure 2.14.** Sum of weighted basis functions from Figure 2.13

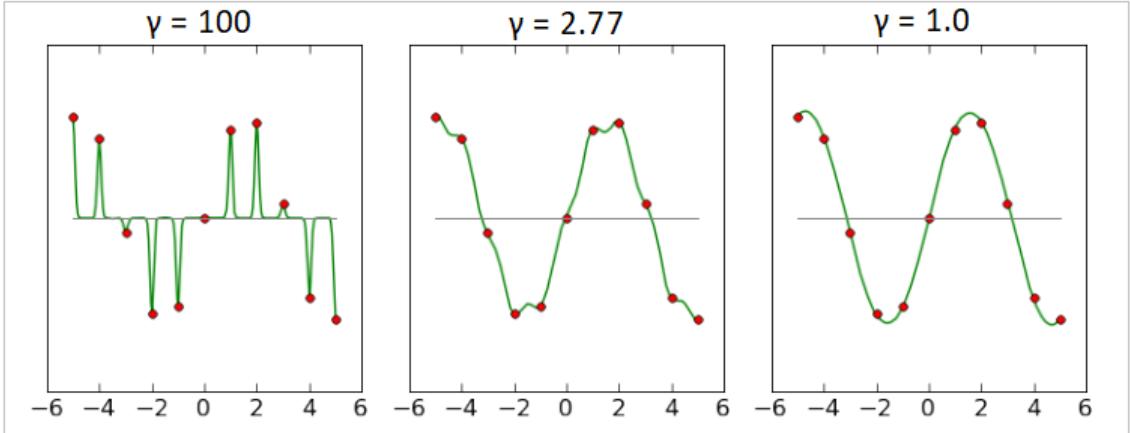
Basis functions used are typically a function of distance between sample points and interpolation point. For an interpolation point and single sample, distance can be denoted by  $r = \|\mathbf{x} - \mathbf{x}_i\|$ . Commonly used basis functions are [59]:

- **Gaussian:** It is a decaying exponential with formulation  $\varphi(r) = \exp(-\gamma r^2)$ . Major advantage of this function is its compactness. It becomes very small ( $\approx 1.2 \times 10^{-4}$ ) when  $r$  is greater than  $\frac{3}{\sqrt{\gamma}}$ . But this comes with the cost of having a parameter to tune. With a large  $\gamma$  value, interpolant will be sharp. It will get smoother as  $\gamma$  decreases. A small  $\gamma$  will make the linear system equation have a sparse matrix and an ill-conditioned one when  $\gamma$  is very small.
- **Multiquadric:** It is a non-compact function with the form  $\varphi(r) = \sqrt{r^2 + \gamma^2}$ . It is convex with values always above 0, hence requires a solution to a linear system with a dense matrix.
- **Thin-Plate Spline:** It is again a non-compact function but with a major advantage of having no tunable parameter. It has the form:

$$\varphi(r) = \begin{cases} r^k, & \text{if } k = 1, 3, 5, \dots \\ r^k \ln r, & \text{if } k = 2, 4, 6, \dots \end{cases} \quad (2.30)$$

Here,  $k$  is the dimension of the sample set. In case of surface interpolation,  $k = 2$  and basis function is simply  $\varphi(r) = r^k \ln r$ . This basis function, however, is also disadvantageous in terms of solving large scale matrices due to having non-zero elements at any point. Thin-Plate Spline is a special case of Polyharmonic Spline which includes an additional polynomial term.

In most software packages a Gaussian basis is used. Therefore it is crucial to choose a good radius. In Figure 2.15 effects of Gaussian basis function model parameter  $\gamma$  is displayed. When  $\gamma$  is large, interpolation is useless as it fails to capture a meaningful representation between samples. Second model parameter yields a better interpolation results however it lacks smoothness. Third model parameter captures the underlying sinusoidal in a smooth fashion. In practice,  $\gamma$  should be at least equal to inverse of average squared distance between samples,  $\frac{1}{r_{ave}} = \frac{1}{\|\mathbf{x}-\mathbf{x}_i\|_{ave}^2}$ , or smaller.



**Figure 2.15.** Guassian RBF interpolation results with varying model parameter [59]

RBFs are very good candidates to create smooth interpolation surfaces from large scattered samples sets. However, they are not adequate when the underlying data has large variations within very few sample intervals or the samples contain large measurement error and/or noise [58]. Selection of RBF parameters requires the knowledge of dominant image features and often they are chosen using heuristics.

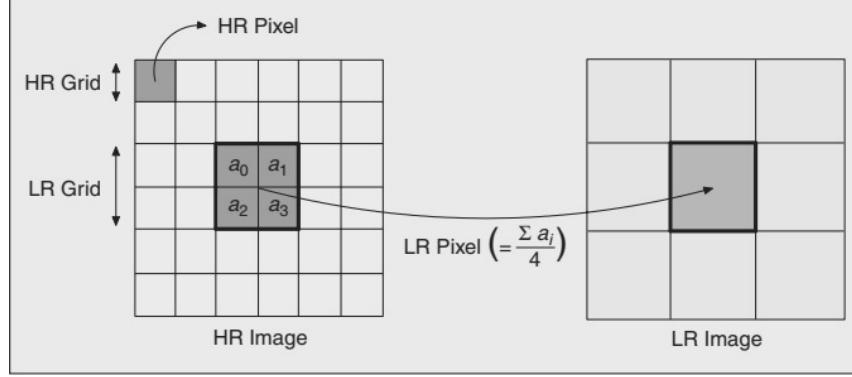
## 2.5. Synthetic Dataset Generation

The focus of this work is on reconstruction step, therefore it is useful to utilize datasets with known dislocation parameters between LR frames. Simplest way to achieve this is to simulate the observation model given in Equation 2.31 and to create LR frames based on the high resolution scene instead of acquiring contiguous individual LR frames by observing the scene. Image observation model relating LR observations  $\mathbf{s}_k$  to HR scene  $\mathbf{z}$  is:

$$\mathbf{s}_k = DB_k M_k \mathbf{z} + n_k, \quad k = 1, 2, \dots p \quad (2.31)$$

with  $\mathbf{M}_k$  as the warp matrix,  $\mathbf{B}_k$  the blur matrix (optical blur, motion blur and PSF blur),  $\mathbf{D}$  the subsampling matrix and  $\mathbf{n}_k$  additive noise. Initially, HR image is multiplied by different warp matrices to generate several other HR frames which are translated and rotated versions of the original image. In super resolution imaging,

point spread function of the capturing device sensors is the major factor of blur and is usually modeled with an average filter. For a scaling factor of 2, one method to generate synthetic LR images is by using a 3x3 averaging filter on HR image, and then down-sampling the result by 2 [60]. Another method is to calculate the mean of each 2x2 block in the HR image and to store the result as LR pixel value, thus achieving filtering and down-sampling in the same step as illustrated in Figure 2.16. The latter method is used in this work to simulate PSF blur.



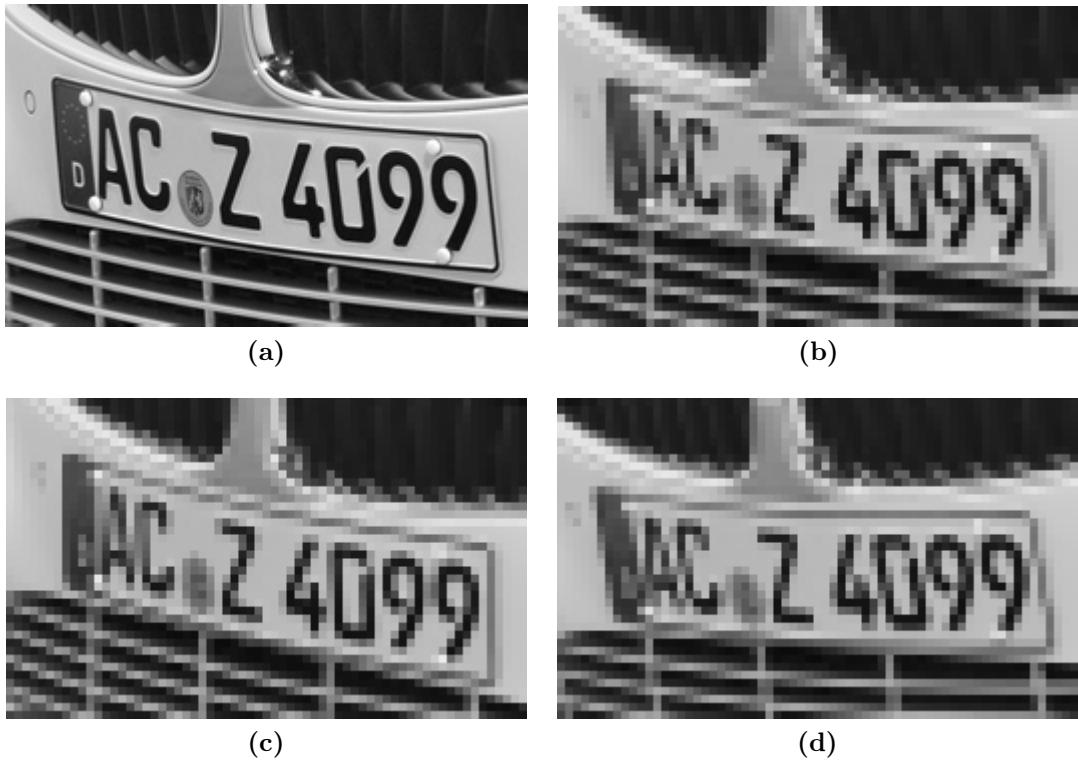
**Figure 2.16.** Sensor point spread function model [17]

In SR, some aliasing is always present even though the average filtering acts more or less as an anti-aliasing filter [17]. Under ideal conditions, where shifts are well distributed along all the directions, the minimum number of LR images necessary to reconstruct the SR image is equal to  $r^2$ , where  $r$  is the up-scaling factor [61]. For the experiments done in this work,  $r = 4$  was considered a trade-off between the resolution improvement and the number of required LR images.

Figure 2.17 displays a 260x160 section (converted to grayscale) portraying a license plate from a car image<sup>1</sup> and three synthetically produced LR images. LR images are scaled up for viewing purposes.

---

<sup>1</sup>Downloaded from <https://www.favcars.com/wallpapers-ac-schnitzer-99d-concept-e89-2011-184580.htm> (last accessed on 25 March 2020)



**Figure 2.17.** (a) HR image and (b), (c), (d) generated LR images of a license plate

Finally, the LR images are degraded by additive Gaussian noise to produce  $\mathbf{s}_k$  vectors consisting of lexicographically ordered LR image frames. Example LR images corrupted by a Gaussian noise with zero mean and variance of 50 are depicted in Figure 2.18.



**Figure 2.18.** Synthetically generated LR images of a car license plate with additive Gaussian noise

### 3. CONNECTIVITY MAP

Images are generally processed using sliding windows. Pixel values that are bounded within the window are used to calculate, for example, the filter output for the center of the window. Usually, window sizes are kept small to capture local variations and to discard the effects of objects distant to the center pixel. When operations on images are performed using large windows, results become inferior due to large intensity changes representing disconnected objects. To overcome the problem of large windows in median filtering, Senel proposed to use connectivity maps instead of actual pixels bounded by the window in order to reduce the effects of distant pixels on the output [62]. This chapter introduces of the concept of connectedness and summarizes the algorithms to produce connectivity maps. First of all, notions of fuzzy topology and concepts of degree of connectedness are given. Based on these notions, connectivity and its properties are examined. Then, a procedure to build connectivity map between the center pixel and other pixels is summarized. Definitions and algorithms given in this chapter are for uniformly distributed pixels.

#### 3.1. Fuzzy Sets and Fuzzy Topology

Notion of connectedness available for binary images is extended to intensity images by Rosenfeld in 1979 [63] by modeling multivalued images as *fuzzy sets*. If grayscale image values are in the  $[0, 1]$  range, a correspondence between the pixel value and its membership in a fuzzy set can be established. First, a definition of fuzzy set is given. Then, fuzzy topology notions are provided based on this definition.

##### 3.1.1. Fuzzy Sets

Fuzzy sets are first proposed by Lotfi Zadeh in 1965, extending the Set Theory [64]. Fuzzy sets are sets whose elements are paired with degree of membership to the set. Let  $\Sigma$  denote a set of points and  $p$  be a single element that belongs to  $\Sigma$ .

And let  $E$  be a fuzzy set such that  $E \subset \Sigma$  and  $p \in E$ . Membership values are between 0 and 1, 1 corresponds to full membership and 0 means the member is not included in the set.

$$\mu_E(p) = \begin{cases} 1, & \text{if } p \in E \\ 0, & \text{if } p \notin E \end{cases} \quad (3.1)$$

where  $\mu_E(p)$  is the membership function. If  $\mu_E(p)$  is close to 1, it is said that  $p$  has strong degree of membership (DOM) in  $E$ . If  $\mu_E(p)$  is close to 0,  $p$  has weak DOM in  $E$ . Zadeh extends the notions of inclusion, union, intersection, convexity etc. to fuzzy sets in his original work.

Fuzzy set  $E$  is *empty* if and only if its membership function is zero on  $\Sigma$ .  $K$  is a *subset* of  $E$ , i.e.  $K \subset E$ , if and only if  $\mu_K \leq \mu_E$ . And two fuzzy sets  $E$  and  $K$  are *equal* if and only if  $\mu_E(p) = \mu_K(p)$  for all  $p$  in  $\Sigma$ .

If  $\overline{E}$  denotes the *complement* of set  $E$ , then the membership function of  $\overline{E}$  is:

$$\mu_{\overline{E}}(p) = 1 - \mu_E(p), \quad \text{for all } p \in \Sigma \quad (3.2)$$

The *union* of two fuzzy sets  $E$  and  $K$  is a fuzzy set  $H$ .  $H = E \cup K$  and  $H$  has membership function:

$$\mu_H(p) = \max(\mu_E(p), \mu_K(p)), \quad \text{for all } p \in \Sigma \quad (3.3)$$

The union of  $E$  and  $K$  is the smallest set that contains both  $E$  and  $K$ .

The *intersection* two fuzzy sets  $E$  and  $K$  is  $A$ .  $A = E \cap K$  and  $A$  has a membership function:

$$\mu_A(p) = \min(\mu_E(p), \mu_K(p)), \quad \text{for all } p \in \Sigma \quad (3.4)$$

The intersection of  $E$  and  $K$  is the largest fuzzy set that is contained in both  $E$  and  $K$ .

With definitions given above, it is shown in [64] that De Morgan's law and distributive laws for conventional sets are also valid for fuzzy sets:

$$\text{De Morgan's laws } \begin{cases} \overline{K \cap E} = \overline{K} \cup \overline{E} \\ \overline{K \cup E} = \overline{K} \cap \overline{E} \end{cases} \quad (3.5)$$

$$\text{Distributive laws } \begin{cases} H \cap (K \cup E) = (H \cap K) \cup (H \cap E) \\ H \cup (K \cap E) = (H \cup K) \cap (H \cup E) \end{cases} \quad (3.6)$$

*Algebraic* operations on fuzzy sets can be defined as a way of combining fuzzy sets. Algebraic product of  $K$  and  $E$  is denoted by  $KE$  and it has the membership function:

$$\mu_{KE}(p) = \mu_K(p)\mu_E(p), \quad \text{for all } p \in \Sigma \quad (3.7)$$

Sum and absolute difference of fuzzy sets  $K$  and  $E$  are denoted by  $K+E$  and  $|K-E|$ , respectively:

$$\mu_{K+E}(p) = \mu_K(p) + \mu_E(p), \quad \text{for all } p \in \Sigma \quad (3.8a)$$

$$\mu_{|K-E|}(p) = |\mu_K(p) - \mu_E(p)|, \quad \text{for all } p \in \Sigma \quad (3.8b)$$

### 3.1.2. Fuzzy Topology

Fuzzy topology is the implementation of fuzzy set approach to the metric space. Concept of fuzzy topology is first introduced by Rosenfeld in [63] and [65]. Let  $I$  represent image intensity values at specific pixel locations. i.e.  $I(p)$  is the sample value at pixel location  $p \in \Sigma$ . Let  $\kappa = \max(I(p) \mid p \in \Sigma)$ . Then, degree of membership (DOM) of  $p$  is defined as its normalized intensity:

$$\mu_I(p) = \frac{I(p)}{\kappa} \quad (3.9)$$

where  $\mu_I(p)$  represents DOM of  $p$  in 'bright' pixel set. Since the membership function of the complement of set  $I$  is  $\mu_{\bar{I}}(p) = 1 - \mu_I(p)$ , DOM of  $p$  in the complementary

‘dark’ pixel set is  $1 - \mu_I(p)$ .

Other definitions needed to build connectivity maps are *path strength*, *connectedness* and *degree of connectedness (DOC)*. In the following discussions,  $E + p \subset \Sigma$  is used to denote a neighbourhood centered at  $p$ . Definition of path strength is the minimum DOM along a path  $\rho = \{p = p_0, p_1, \dots, p_n = q\} \subset E \subset \Sigma$ :

$$s_E(\rho) = \min_{1 \leq i \leq n} \mu_I(p_i) \quad (3.10)$$

Degree of connectedness of two pixels  $p$  and  $q$  is defined over all possible paths between  $p$  and  $q$ :

$$c_E(p, q) = \min_{\rho \subset E} s_E(\rho) \quad (3.11)$$

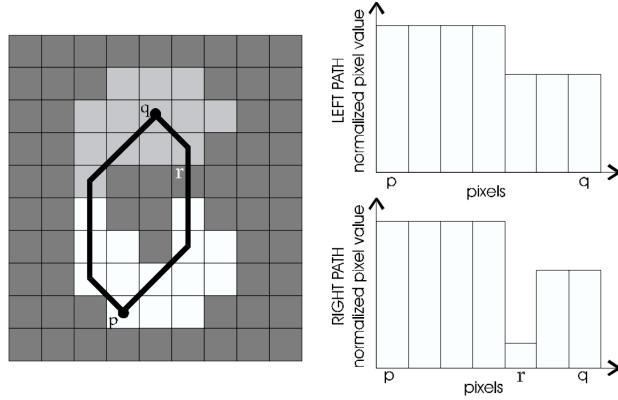
Connectedness of two pixels can be defined using normalized intensities. Points  $p$  and  $q$  in  $E$  are connected if and only if

$$\mu_I(p_i) \geq \min\{\mu_I(p), \mu_I(q)\} \quad (3.12)$$

is satisfied for every pixel  $p_i \in E$  on any path  $\rho^c = \{p = p_0, p_1, \dots, p_n = q\}$ . This definition of connectedness between two points is equivalent to

$$c_E(p, q) = \min\{\mu_I(p), \mu_I(q)\} \quad (3.13)$$

Two points and two different paths connecting them are given in Figure 3.1. Path strength of the path on the right has  $s_E(\rho_{right}) < \mu_I(q)$  and  $s_E(\rho_{right}) < \mu_I(p)$ . For the path on the left, however,  $s_E(\rho_{left}) \geq \mu_I(q)$ . Since there is a path that satisfies  $c_E(p, q) = \min\{\mu_I(p), \mu_I(q)\}$ ,  $p$  and  $q$  are said to be connected.



**Figure 3.1.** Two paths connecting pixels  $p$  and  $q$  [62]

DOM and DOC have the following properties [63], [65]:

- $\mu_I(p)$ ,  $s_E(\rho)$  and  $c_E(p, q)$  all have values in the closed interval of  $[0,1]$
- $c_E(p, p) = \mu_I(p)$
- $c_E(p, q) = c_E(q, p)$
- $c_E(p, q) \leq \min\{\mu_I(p), \mu_I(q)\}$

These properties are used in [62] by Senel to compute degree of connectedness in a computationally efficient way.

### 3.2. $\alpha$ -connectedness

Based on the definitions and properties given in the previous section, the concept of  $\alpha$ -connectedness is introduced in [62] to compute degree of connectedness (DOC) of two points in a set  $E$ ,  $c_E(p, q)$ , without having to compute the strength of all possible paths in the set  $E$ . Point  $p$  is said to be  $\alpha$ -connected to point  $q$  if

$$c_E(p, q) = \mu_I(p) \quad (3.14)$$

An equivalent definition can be made using binary threshold. First,  $B_t$  is defined as a binary image as follows:

$$B_t(x) = \begin{cases} 1, & \text{if } I(x) \geq t \\ 0, & \text{if } I(x) < t \end{cases} \quad (3.15)$$

where  $t$  is gray level threshold.  $B_{I(p)}$  is the binary image obtained by setting all sample values of  $I$  greater than or equal to  $I(p)$  to 1 and others to 0. Then,  $p$  is  $\alpha$ -connected to  $q$  if they are connected in binary image  $B_{I(p)}$ . Based on the DOM definitions,  $\alpha$ -connectivity is reflexive and transitive but it is not symmetric, i.e.  $p$  could be  $\alpha$ -connected to  $q$  but  $q$  will not be  $\alpha$ -connected to  $p$  if  $\mu_I(p) < \mu_I(q)$ .

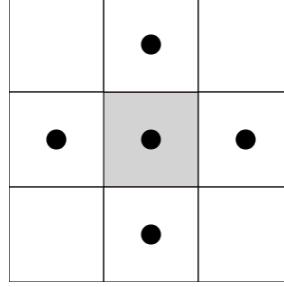
Equation 3.12 implies that  $p$  and  $q$  are connected if either  $p$  or  $q$  is  $\alpha$ -connected to the other or both  $p$  and  $q$  are  $\alpha$ -connected to each other. This property provides a less computationally costly way to construct connectivity map in a set of points than traversing all possible paths in the set  $E$ .

### 3.3. Connectivity Map Algorithms

Let  $t$  be the largest threshold that makes  $p$  and  $q$  stay in the same connected component in binary image  $B_t \cap E$  generated after thresholding  $I$  with  $t$ . In mathematical terms:

$$c_E(p, q) = \max\{t \mid (p \in B_t \cap E) \text{ AND } (q \in B_t \cap E)\} \quad (3.16)$$

and let  $\kappa$  be the maximum intensity value in an image  $I$ . Then, degree of connectedness between  $p$  and  $q$  is  $t/\kappa$ . Before giving the algorithm to compute the degree of connectedness between two pixels, binary connectedness needs to be defined. Let  $p \sim q$  denote the condition that  $p$  is binary connected to  $q$  in a thresholded binary image. Connectivity type considered here is 4-connectivity (or von Neumann neighborhood). The 4-neighbors of pixel  $p$  are denoted by  $N_4(p)$ . For a pixel  $p$  at  $(x,y)$ , set of 4-connectivity neighbors are given by  $N_4(p) = \{(x+1, y), (x-1, y), (x, y+1), (x, y-1)\}$ . When 8-connectivity is considered, pixels at diagonal corners must also be included.



**Figure 3.2.** 4-connectivity

Binary connectedness can also be thought of as connected component labeling problem where pixels  $p$  and  $q$  are required to have the same label. Using this analogy, connected component labeling algorithm in [66] that works on one component at a time can be modified as below to compute the binary connectedness. In the following algorithm, a value of 1 (true) in connectedness map  $J$  denotes a pixel that is connected to center pixel  $p$  in the thresholded binary image  $B_t$ .  $Q$  is defined as an unordered set of pixels initially empty representing *queue* of connected pixels, that is, a list of pixels whose 4-connectivity neighbours shall be checked for binary connectedness.  $Q_{|Q|}$  is used to pick out the last element of set  $Q$ . Then,  $N_4(Q_{|Q|})$  denotes the 4-connectivity pixels of last element in unordered set  $Q$ . Finally, let  $c$  denote the center pixel location of 4-connectivity neighborhood in the current iteration.

---

**Algorithm 3.1:** Binary connectedness between  $p$  and  $p_i$

---

```

initialize  $J(p_i) = 0$  for all  $p_i \in E$ 
set  $J(p) = 1$ 
add  $p$  to  $Q$ 
while  $Q \neq \emptyset$  do
    set  $c = Q_{|Q|}$ 
    remove  $Q_{|Q|}$  from  $Q$ 
    for all  $j \in \{1, 2, 3, 4\}$  do
        if  $(J(N_4(c)_j) = 0 \text{ and } J(c) = 1 \text{ and } B_t(N_4(c)_j) = 1)$  then
            set  $J(N_4(c)_j) = 1$ 
            add  $N_4(c)_j$  to  $Q$ 
        end
    end
end

```

---

Now consider the neighbourhood set  $E = \{p_i\}_{i=0}^{n-1}$  such that  $p = p_0$  and  $q =$

$p_{n-1}$  as a set of pixels from  $I$ . Define  $U$  as another set of pixels initially empty.  $B_{\mu_I(p_i)}$  is the binary image obtained by thresholding  $E$  at  $\mu_I(p_i)$ . Then, degree of connectedness (DOC) between  $p$  and  $q$ ,  $c_E(p, q)$ , can be calculated using the algorithm below [62]:

---

**Algorithm 3.2:** Degree of connectedness between p and q

---

```

for all  $p_i \in E$  such that  $\mu_I(p_i) \leq \min\{\mu_I(p), \mu_I(q)\}$  do
    threshold  $E$  at  $\mu_I(p_i)$  to generate  $B_{\mu_I(p_i)}$ 
    if  $p_i \sim p$  and  $p_i \sim q$  then
        add  $p_i$  to  $U$ 
    end
end
 $c_E(p, q) = \max\{\mu_I(p_i) \mid p_i \in U\}$ 

```

---

Utilizing Algorithm 3.2, degree of connectedness between a single pixel and others in a neighbourhood can be computed as described in Algorithm 3.3 below, resulting in DOCM (degree of connectedness map)  $c_E(p, p_i)$ .

---

**Algorithm 3.3:** Degree of Connectedness Map

---

```

initialize  $c_E(p, p_i) = 0$  for all  $p_i \in E$ 
set  $c_E(p, p) = \mu_I(p)$ 
for all  $p_i \in E$  such that  $\mu_I(p_i) \leq \mu_I(p)$  do
    threshold  $E$  at  $\mu_I(p_i)$  to generate  $B_{\mu_I(p_i)}$ 
    for all  $p_j \in E$  such that  $p_j \sim p$  in  $B_{\mu_I(p_i)}$  and  $c_E(p, p_j) \leq \mu_I(p_i)$  do
         $c_E(p, p_j) = \mu_I(p_i)$ 
    end
end

```

---

This process creates a map of how a pixel  $p$  is connected to other pixels in the window and it is referred to as *connectivity map*. In the neighbourhood  $E + p$  where the connectivity map is constructed, the value of center pixel remains unchanged, i.e.  $DOCM(p) = I(p)$ . Values of other points in the map specify  $\alpha$ -connectedness information between the center pixel and others. There are three possible scenarios in which connectivity map values are determined:

- In a set  $A + p$  when pixels are on the same level with the center pixel or they

are below  $p$  either on a ramp or separated by a peak from  $p$ , i.e.  $I(p_i) \leq I(p)$ :

$$DOCM(p_i) = I(p_i) \text{ for all } p_i \in A + p \quad (3.17)$$

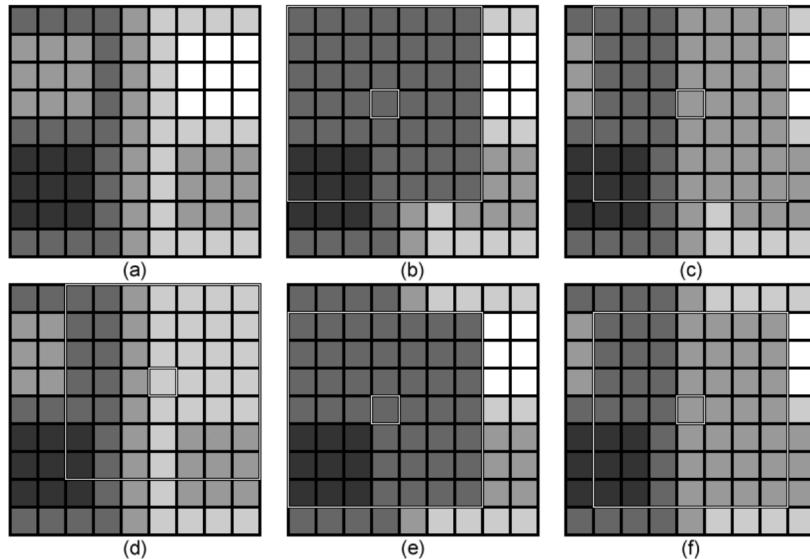
- In a set  $B + p$  when pixels  $p_j$  are above  $p$  either on a ramp or separated by a peak from  $p$ , i.e.  $I(p_j) > I(p)$ :

$$DOCM(p_j) = I(p) \text{ for all } p_j \in B + p \quad (3.18)$$

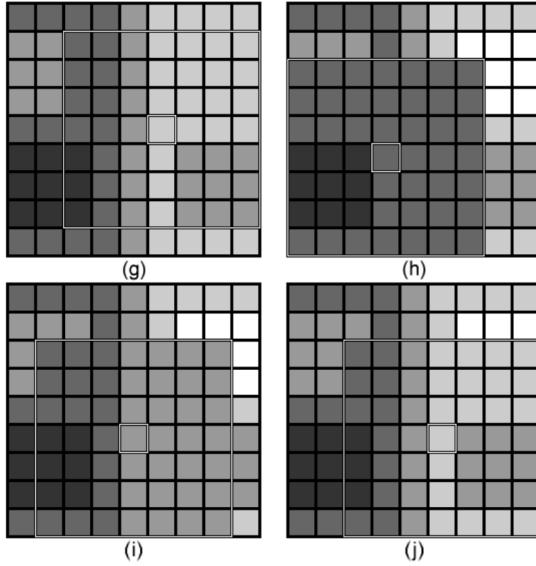
- In a set  $C + p$  where pixels  $p_k$  have arbitrary intensity values but form bright peaks in the window that are separated from  $p$  by a ridge or valley:

$$DOCM(p_k) = \kappa \times c_E(p_k, p) \text{ for all } p_k \in C + p \quad (3.19)$$

All three cases are displayed in Figures 3.3 and 3.4. In Figure 3.3.a original image  $I$  is displayed. Others display results of Algorithm 3.3 for different center pixels on a 7x7 window. In the figures, center pixel  $p$  is surrounded in white square. Contours of each 7x7 window ( $E + p$ ) are also shaded in white.



**Figure 3.3.** Connectivity map with different center pixels in a 7x7 window [62]



**Figure 3.4.** Connectivity map with different center pixels in a 7x7 window (continued)  
[62]

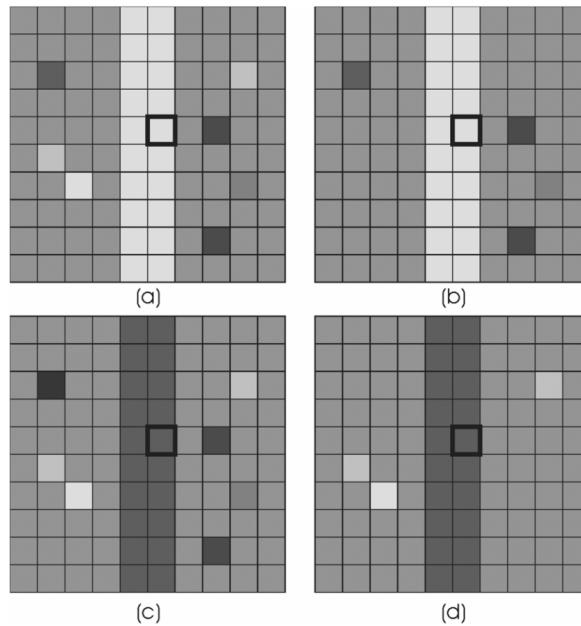
Degree of connectedness map (DOCM) emphasizes pixels values that are on the same level with the center pixel or at a level below the center pixel. This case is defined by Equation 3.17 and can be seen on the 3x3 bottom left dark patch in Figures 3.3.b-f and Figures 3.4.g-j. This case can also be observed on the bottom right patch in Figures 3.3.d and 3.4.g, 3.4.j.

DOCM decreases the values of pixels that are at a level above the center pixel depending on the pixel values separating the center pixel and the ones on the higher level. If the separating pixels are all on a higher level than the center pixel, then the values of pixels on a higher level become equal to the value of the center pixel. This is the case defined by Equation 3.18 and can be seen on the top left patch in Figures 3.3.b, 3.3.e, and 3.4.h.

If the pixels separating the center pixel and the ones that are on a higher level than the center pixel have values less than the center pixel, then the values of pixels on a higher level become equal to that of minimum valued pixel present among separating pixels. This is the case defined by Equation 3.19 and can be observed on the top left patch in Figures 3.3.c, 3.3.d, 3.3.f, 3.4.g, 3.4.i and 3.4.j.

One of the results of degree of connectedness map is that it removes uncon-

nected pixels to the center pixel within the operation window. And this behaviour is asymmetric for bright and dark pixels. When DOCM is computed for a window centered at one of the the bright pixels on the valley displayed in Figure 3.5.a, brighter pixels are removed as can be seen in Figure 3.5.b. On the other hand, darker pixels retain their value (Figure 3.5.b). When DOCM is applied on the image as it is, there is a bias of keeping dark pixels compared to bright pixels. Hence, DOCM in its original form is referred to as ‘DOCM for bright pixels (DOCM<sub>b</sub>)’ in [67]. A bias of keeping bright pixels exists when DOCM is computed on the negative image. In this case, DOCM for dark pixels (DOCM<sub>d</sub>) computed for a center pixel located on the ridge (Figure 3.5.c) results in an image patch where dark pixels are removed, keeping the bright pixels (Figure 3.5.d).



**Figure 3.5.** Degree of Connectedness Maps of ridge and valley images [67]

These properties of degree of connectedness maps can be exploited to obtain better results while processing images in large sliding windows. When the operation window is wide, large intensity changes are more likely to occur within the window. These intensity variations negatively effect the processing results. Using connectivity maps, effects of connected pixels can be emphasized compared to distant disconnected pixels in a wide processing window.

In [62], Senel proposed a *topological median filter* based on this property of connectivity maps. Topological median filter is the median filter applied on connectivity maps. There are four types of filters proposed: (1) TL is the median filter applied on DOCMb, (2) TD is the median filter applied on DOMCd. In addition to these, two composite filters are proposed: (3) TLD is the median filtering on connectivity map computed using TD filtered patch and (4) TDL is the median filtering on connectivity map computed using negative of TL filtered patch. TLD and TDL have improved edge preserving performance compared to conventional median filter when used appropriately. TDL is better at edge preservation when used on an image with dark objects on a bright background; TLD, on the other hand, is better at edge preservation when used on an image with bright objects on a dark background, compared to conventional median filter.

In a similar fashion, in [67], connectivity maps are exploited to prevent issues associated with large gradient kernels used for edge detection. Large kernels have better noise suppression but suffer from variations of distant pixels around edges, causing localisation problems. In order to reduce the effects of disconnected pixels, gradients are computed on DOCMb and DOCMd windows. Then a *topological gradient* is defined as a combination of these two gradients. Topological gradient is shown to have combined good noise suppression characteristics of a large kernel with good localization performance.

This work uses similar principles to improve the results of nonuniform interpolation along the edges in super-resolution imaging.

## 4. INTERPOLATION BASED ON IMAGE TOPOLOGY

Super-resolution (SR) can be divided into three steps: image registration, HR image estimation and restoration. First step, image registration, is the estimation of relative displacements between LR frames. Second stage is the estimation of HR pixel values. Final stage is the restoration for blur and noise removal [17].

Image registration stage is out of the scope of this work. Algorithms presented in this work are demonstrated on synthetic LR sequences with known relative dislocation parameters. Problem of blur and noise is not addressed either. Focus of this work is on the second, HR image estimation stage of super-resolution reconstruction. The aim of this study is to enhance the results of HR pixel value estimation by taking connectivity between pixels into consideration.

HR pixel estimation techniques widely used in the literature include frequency domain [11], regularized reconstruction [14, 15], projection onto convex sets [12, 13] and nonuniform interpolation [16]. Frequency domain methods, as the name implies, operate in frequency domain and are not suitable for incorporating connectivity information between LR pixels, which is available in spatial domain. Regularized SR reconstruction approaches typically solve for a constrained optimization problem minimizing the error between each LR frame and an a priori desired solution. Connectivity maps, which operate on sliding windows, can not be incorporated into linear systems of equations involving each entire LR image. For the same reason, projection onto convex sets techniques are not appropriate to benefit from connectivity maps. On the other hand, nonuniform interpolation approaches operate on nonuniform grid of registered LR pixels in spatial domain. It is possible, in windows on such a grid, to compute connectivity maps.

Conventionally, polynomial-based interpolation methods such as triangular cubic interpolation are used in SR due to their low computational cost [43]. These local methods utilize low-order polynomials to model an image signal (surface) using

a small number of pixels. However, polynomial functions are not good at modeling the large intensity variations (i.e. edges) [43]. Hence, the conventional methods produce poor results around edges.

In this chapter, concepts from fuzzy image topology, such as degree of connectedness, connectivity maps are applied to nonuniformly distributed samples and nonuniform interpolation. It is shown that an interpolation technique using greater number of pixels, when supplemented by connectivity information, produces superior results compared to conventional methods.

First, degree of connectedness map computation algorithms available for uniform pixels are revised for the case of nonuniformly distributed pixels utilizing the concept of *natural neighbourhood*. Second, inverse distance weighted interpolation scheme is applied on connectivity maps to enhance edges.

#### 4.1. Degree of Connectedness Map (DOCM) in Scattered Data

Let  $p$  and  $q$  denote two samples in the window of interest  $E$  within a scattered set of pixels  $S$ . i.e.  $p, q \in E$  and  $E \subset S$ . Definitions of *membership function*, *degree of membership*,  $\alpha$ -*connectedness* and *degree of connectedness* given in Chapter 3 do not assume a uniform grid. To generate DOCM for  $E$ , degree of connectedness  $c_E(p, q)$  must be computed between sample of interest and all other elements in the set  $E$ . DOC can be computed by finding the largest threshold that makes  $p$  and  $q$  stay in the same connected component in binary image  $B_t \cap E$  generated after thresholding a scattered set  $S$  with  $t$ :

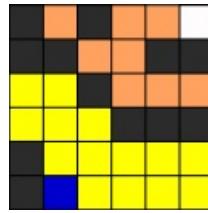
$$c_E(p, q) = \max\{t \mid (p \in B_t \cap E) \text{ AND } (q \in B_t \cap E)\} \quad (4.1)$$

When switched to a nonuniform grid, the only difference arises in the definition of *connected components* where a notion of neighbourhood plays a central role in its construction. Therefore, in order to compute connectivity between nonuniform samples, first a neighbourhood criteria needs to be defined. Then, Algorithm 3.1

from Chapter 3 can be revised with respect to this neighbourhood criteria to determine connectivity in binary image  $B_t$ . Algorithms 3.2 and 3.3 can be implemented in the same manner as before. A good candidate to replace 4-connectivity used in uniform case is *natural neighbourhood*.

#### 4.1.1. Natural Neighbour Concept in Scattered Data

In a uniform grid, pixels are assumed to be unit squares filling space as shown in Figure 4.1.



**Figure 4.1.** Unit squares filling space

In this partition of space, unit squares sharing an edge (4-connectivity) or sharing a corner (8-connectivity) are said to be neighbours. In case of nonuniformly distributed pixels, definition of neighbourhood is an issue [46]. In order to talk about a neighbourhood for scattered pixels, a similar partitioning of space based on pixel locations is necessary.

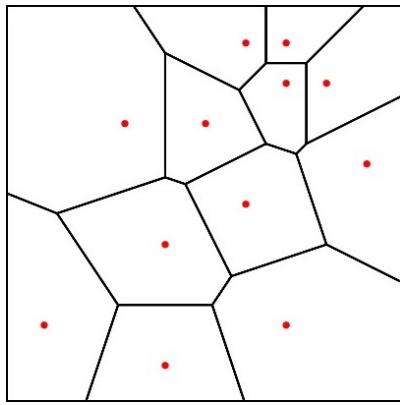
In geometry, a Voronoi diagram is a partition of a plane in such a way that for each element there is a corresponding region close to itself [30]. Consider a set of pixels  $P = \{p_1, \dots, p_n\}$  in 2D space. Voronoi diagram of the set  $P$  is the partition of the space around each pixel into convex polygons called *Voronoi tiles* (or *Voronoi cells*). If tiles corresponding to each pixel is denoted by  $\tau$ , we have  $n$  tiles  $\{\tau_1, \dots, \tau_n\}$  such that:

$$q \in \tau_i \iff d(q, p_i) \leq \min_{1 \leq j \leq n} d(q, p_j) \quad (4.2)$$

where  $d(p, q)$  is a function of Euclidean distance between two pixels. Voronoi diagram of a scattered set is displayed In Figure 4.2.

Voronoi diagram can be defined with distance metrics other than Euclidean

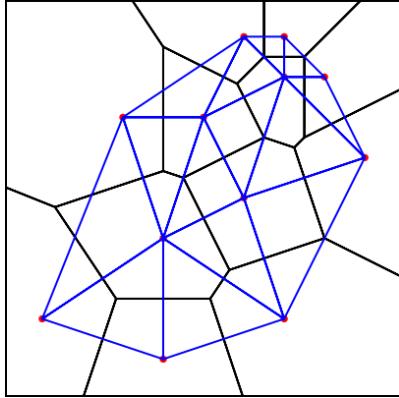
such as the Mahalanobis distance or Manhattan distance. In this work, Euclidean distance is preferred, firstly, because it is a more intuitive measure to interpret distance between features in an image. Secondly, maintenance of Voronoi diagrams with complex distance metrics is more complicated as they do not correspond to a triangulation [68]. As will be explained in Section 4.2, computation of connectivity maps requires insertion of a new point into nonuniform grid, which necessitate modification of the Voronoi diagram at each iteration.



**Figure 4.2.** Voronoi diagram

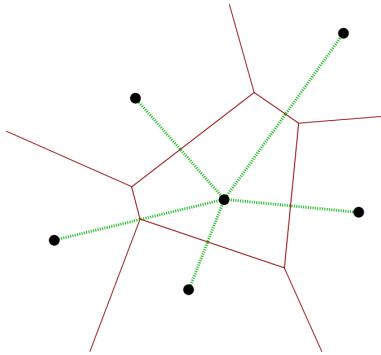
Within the set  $P$ , points  $p_i$  and  $p_j$  are defined as *natural neighbours* if there exists a point  $q$  such that  $q$  is closer to both  $p_i$  and  $p_j$  than it is to any other element of the set [68]. This notion of spatial proximity is embodied in the Voronoi diagram. In Figure 4.2, connection of the Voronoi diagram to the concept of natural neighbour can be seen. Natural neighbours are pixels that share one common edge in their Voronoi cell. Points on the intersection of tiles satisfy Equation 4.2 for each neighbouring Voronoi site [45].

The single most important feature of Voronoi diagrams is, for our application, their duality to Delaunay triangulations. Since Delaunay triangulation is the straight-line dual of Voronoi diagram [30], each vertex of a triangle corresponds to a Voronoi site and two vertices connected by a Delaunay triangle edge are natural neighbours as exemplified in Figure 4.3. Hence, *natural neighbours of a sample point  $p$*  are *union of vertices of all Delaunay triangles attached to vertex  $p$*  [69].



**Figure 4.3.** Delaunay triangulation and Voronoi diagram

Let  $N_{NN}(p)$  denote the natural neighbours of a pixel  $p$  in scattered set  $S$ . Since the samples of a nonuniform set are unevenly distributed, number of neighbours of each element might differ. Number of elements in  $N_{NN}(p)$  varies and is denoted by  $|N(p)_{NN}|$ .



**Figure 4.4.** Natural neighbours of a sample in a scattered dataset

Delaunay triangulations are widely studied in the literature and numerous algorithms and libraries are available to generate and manipulate them [70]. The same tools that exist for Delaunay triangulations can also be used to access to the entities of Voronoi diagrams and by extension to natural neighbours in a scattered data set.

#### 4.1.2. Connectivity Map Algorithms for Scattered Data

Algorithm 3.1 from Chapter 3 can be revised for nonuniform case utilizing the concept of natural neighbours. As mentioned before in Chapter 3, connectedness

of two pixels on a binary image can be obtained using connected component labeling algorithms.

In the algorithm below, connectedness map  $J$  denotes binary connectedness map where a value of 1 (true) stands for a pixel that is connected to center pixel  $p$  in binary image  $B_t$ .  $Q$  is an initially empty unordered set in which connected pixels are added iteratively. Elements of this set are pixels whose natural neighbours are yet to be checked for binary connectedness.  $Q_{|Q|}$  is used to pick out the last element of set  $Q$ . Then,  $N_{NN}(Q_{|Q|})$  denotes the natural neighbourhood pixels of the last element in the unordered set  $Q$ . Finally,  $c$  denotes the center pixel location of natural neighborhood in the current iteration.

---

**Algorithm 4.1:** Binary connectedness between  $p$  and  $p_i$

---

```

initialize  $J(p_i) = 0$  for all  $p_i \in E$ 
set  $J(p) = 1$ 
add  $p$  to  $Q$ 
while  $Q \neq \emptyset$  do
    set  $c = Q_{|Q|}$ 
    remove  $Q_{|Q|}$  from  $Q$ 
    for all  $j \in \{1, \dots, |N(c)_{NN}|\}$  do
        if ( $J(N_{NN}(c)_j) = 0$  and  $J(c) = 1$  and  $B_t(N_{NN}(c)_j) = 1$ ) then
            set  $J(N_{NN}(c)_j) = 1$ 
            add  $N_{NN}(c)_j$  to  $Q$ 
        end
    end
end

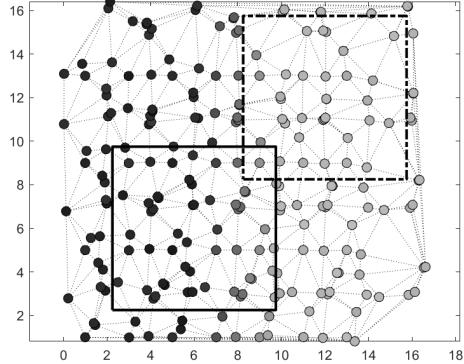
```

---

If  $J(p_i) = 1$  after the completion Algorithm 4.1, then  $p$  is said to be binary connected to  $p_i$ , denoted as  $p \sim p_i$ , in the thresholded binary image  $B_t$ . Then, we can continue to compute degree of connectedness,  $c_E(p, q)$ , between samples  $p = p_0$  and  $q = p_{n-1}$  in the window  $E = \{p_i\}_{i=0}^{n-1}$  which is a subset of  $S$ . In nonuniform case, size of  $E$  does not directly correspond to the dimensions of the chosen window.

Since sample are at subpixel locations, there may be more or less samples inside the window than the number pixels of the window. In uniform case, especially when 8-connectivity is used, a 5x5 window is not adequate to separate unconnected pixels [67]. In order for DOCM to have the effect of removing unconnected, therefore

unrelated pixels, windows larger than 5x5 should be utilized. In case of scattered data, this corresponds to choosing windows that enclose more than 36 sample points.



**Figure 4.5.** Operation windows in nonuniform super-resolution grid

Local pixel density varies between different regions of the nonuniform super-resolution grid. In Figure 4.5, number of pixels included in the 7.5x7.5 window centered at (6,6), displayed by solid line, is 56. On the other hand, number of pixels included in the 7.5x7.5 window centered at (12,12), displayed by dash-dot line, is 34. In order to accommodate the same number of pixels in each window, window size needs to be adjusted in each iteration. However, if window length is shrunk it cannot capture local variations in the image. When window size is increased it includes too much unconnected pixels. Therefore, a constant window size based on global pixel density is selected.

Due to dislocations between LR frames, some of the points in true scene are captured more than once in different LR frames. These points are referred to as *coincident points*. If the distance between two LR pixels is smaller than a small tolerance  $\epsilon$ , for example  $10^{-6}$ , they are also marked as coincident and average of their values is used as one sample [71]. As a result, there are less number of pixels on the interpolation grid than the total number of pixels included in LR frames. Hence, super-resolution grid pixel density varies for different dislocations between LR images. Average pixel density of the super-resolution grid is given by:

$$\lambda_s = \frac{\text{number of samples}}{\text{area of SR grid}} \quad (4.3)$$

Using  $\lambda_s$ , selected window length of a square window is equal to  $\sqrt{\frac{n_s}{\lambda_s}}$ , where  $n_s$  is desired number of samples in the window. Window size can be selected fractional valued.

Then,  $c_E(p, q)$  in a chosen window  $E$  can be calculated using the algorithm below:

---

**Algorithm 4.2:** Degree of connectedness between p and q

---

```

for all  $p_i \in E$  such that  $\mu_I(p_i) \leq \min\{\mu_I(p), \mu_I(q)\}$  do
    threshold  $E$  at  $\mu_I(p_i)$  to generate  $B_{\mu_I(p_i)}$ 
    if  $p_i \sim p$  and  $p_i \sim q$  then
        | add  $p_i$  to  $U$ 
    end
end
 $c_E(p, q) = \max\{\mu_I(p_i) \mid p_i \in U\}$ 
```

---

Utilizing Algorithm 4.2, degree of connectedness between a single pixel and others in a neighbourhood can be computed as described in Algorithm 4.3 below, resulting in DOCM (degree of connectedness map)  $c_E(p, p_i)$ .

---

**Algorithm 4.3:** Degree of connectedness map

---

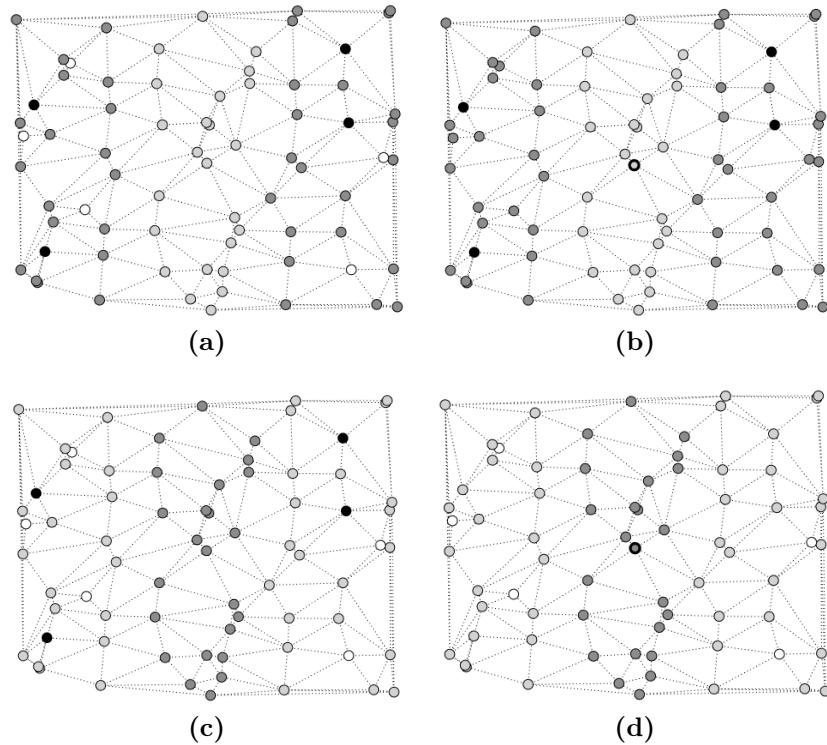
```

initialize  $c_E(p, p_i) = 0$  for all  $p_i \in E$ 
set  $c_E(p, p) = \mu_I(p)$ 
for all  $p_i \in E$  such that  $\mu_I(p_i) \leq \mu_I(p)$  do
    threshold  $E$  at  $\mu_I(p_i)$  to generate  $B_{\mu_I(p_i)}$ 
    for all  $p_j \in E$  such that  $p_j \sim p$  in  $B_{\mu_I(p_i)}$  and  $c_E(p, p_j) \leq \mu_I(p_i)$  do
        |  $c_E(p, p_j) = \mu_I(p_i)$ 
    end
end
```

---

The properties DOCMs exhibit in uniformly distributed pixels are also valid for nonuniformly distributed pixels. When DOCM is computed on the image as it is, it tends to remove brighter samples. When DOCM is implemented on the negative image, it can be observed that darker pixels are removed. These two complementary maps are dubbed DOCMb and DOCMd, respectively. Figure 4.6.a displays an irregularly sampled line on a constant image area corrupted by noise and the result of DOCM for bright pixels calculated for the entire set with respect to center pixels (circled with bold edge) is given in Figure 4.6.b. It can be observed

that noisy pixels with values higher than those of non-corrupted surrounding pixels are removed. On the other hand, noisy pixels with values smaller than those of non-corrupted surrounding pixels remain. Similarly, Figure 4.6.c displays a ridge on a constant image area and noisy pixels and the result of DOCM for dark pixels is given in Figure 4.6.d. In this case, it can be observed that noisy pixels with values higher than those of non-corrupted surrounding pixels remains. On the other hand, noisy pixels with values smaller than those of non-corrupted surrounding pixels are removed.



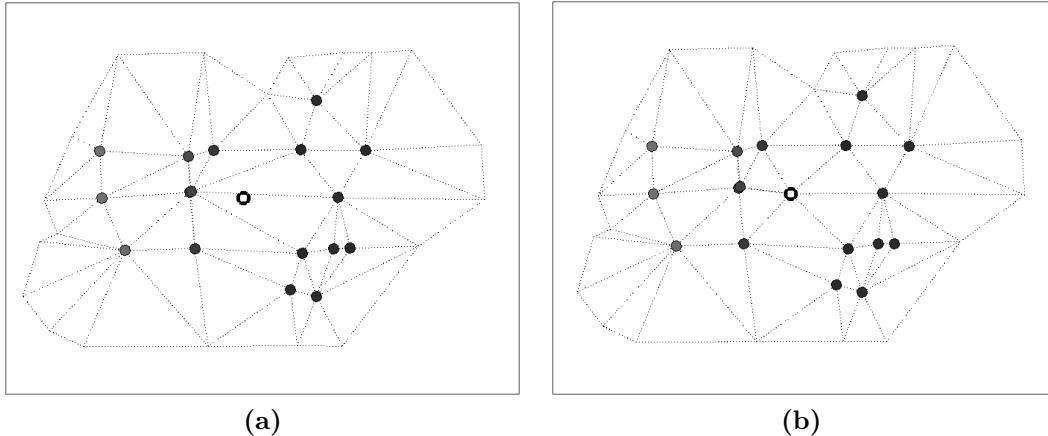
**Figure 4.6.** Degree of Connectedness Maps for (b) bright and (d) dark pixels of irregularly sampled (a) valley and (c) ridge images

## 4.2. Nonuniform Interpolation Utilizing DOCMs

Degree of Connectedness Maps for dark and bright pixels described in previous section can be utilized as a processing step in interpolation stage of super-resolution reconstruction. As mentioned in the previous section, operation window must include at least  $6 \times 6$  samples for DOCM algorithms to separate unconnected pixels. Therefore, interpolation technique chosen should also operate in large windows to be

able to benefit from these results. Local triangulation based methods such as cubic interpolation uses values only at the vertices of the triangle enclosing query point, thus they are not suitable for the task. Natural neighbour interpolation incorporates more than three samples yet all the pixels included in interpolation are neighbours. Therefore, it fails to fully exploit connectedness and DOCM concepts. Among the methods that make use of windows created by DOCMs is inverse distance weighted interpolation since it can operate on a larger window and assign weights to each sample in the set.

In order to compute connectedness maps, a center pixel is needed. However, the center pixel located at interpolation point is missing among LR pixels. In Figure 4.7.a location of interpolation point (center pixel for connectivity map) is shown with empty circle. The center pixel value needs to be estimated before applying connectedness algorithms since DOCM calculations require a center pixel value. The estimate can be generated using various methods. For instance, nearest LR pixel value can be taken or a simple local linear interpolation can be used. Then, this estimate is inserted into triangulation as shown in Figure 4.7.b and connectedness maps are computed on which IDW is applied. When computing DOCM, a new triangulation is not computed but rather the center pixel is inserted into triangulation computed for the entire interpolation grid. This approach has less computational cost since only a few edge flip operation are required rather than computing a new triangulation for the whole window. In addition, neighbourhood relations for pixels along the boundaries are preserved since the insertion operation is applied on the global grid. In Figure 4.7.a, border pixels inside the displayed window have the same neighbourhood relations in a different window since the neighbourhood relations depend on a single global triangulation.



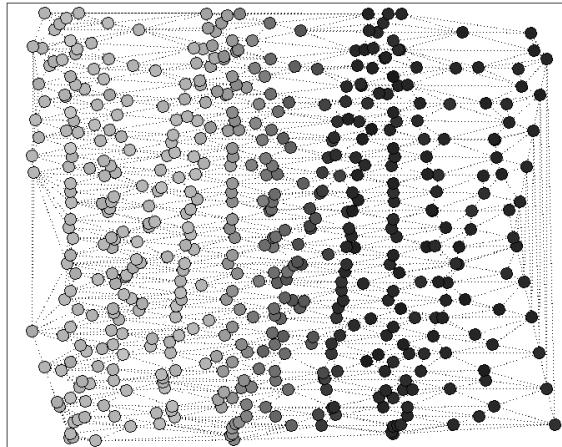
**Figure 4.7.** Center pixel (interpolation point) in an interpolation window: (a) not included in triangulation and (b) inserted into triangulation

Connectivity information is used to enhance interpolation results around edges by combining interpolation values computed on DOCMb and DOCMd.

#### 4.2.1. Combination of Interpolation Values from DOCMs

DOCMb and DOCMd values show different aspects of the encapsulated image part. DOCMb and DOCMd are close to each other on a smooth surface, however they become different along an edge. This property of DOCMd and DOCMb values can be utilized to enhance edges since transitions from dark to bright sections constitutes the most informative part of images. Along an edge, DOCMb usually provides values less than the original image intensity values within the window. On the other hand DOCMd tend to keep bright objects inside the image patch. Hence an interpolation applied on DOCMb values yields smaller values than the interpolation on the original image patch. Similarly, interpolation applied on DOCMd produces slightly higher values. This property is demonstrated using a synthetic image containing a sharp rising edge. The value of pixels on the low intensity part the edge is 40 and brighter part has a value of 180. 16 LR frames are produced using the procedure described in Section 2.5 Synthetic Data Generation. Noise is not added to the LR frames. Since the dislocation parameters between LR frames are known, resulting LR frames can be mapped into the interpolation grid. Figure 4.8 displays mapped

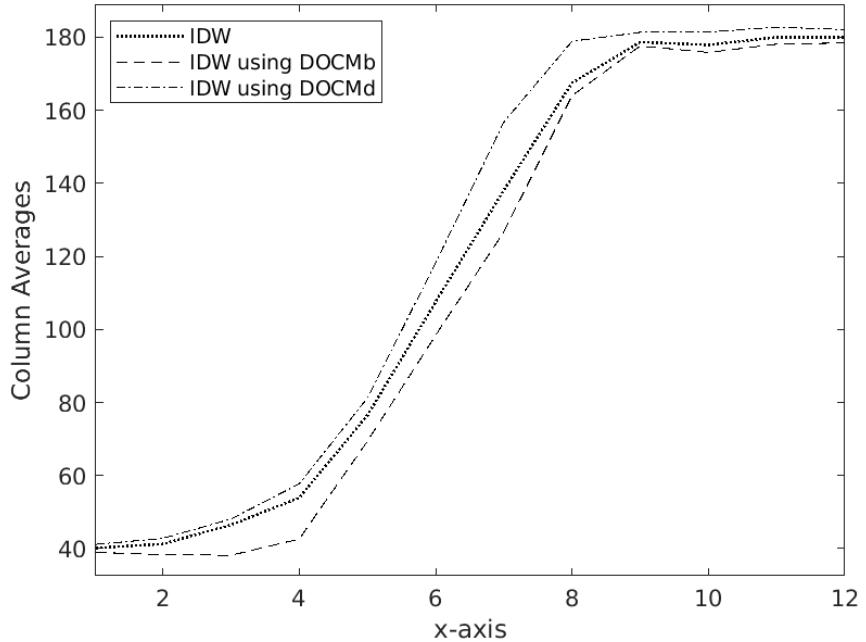
LR pixels. Note that LR pixels do not manifest a sharp rise due to the point spread function in the LR frame generation process. Obviously, interpolation on this grid does not reveal the sharp edge present in HR scene.



**Figure 4.8.** Registered LR pixels of a sharp rising edge image

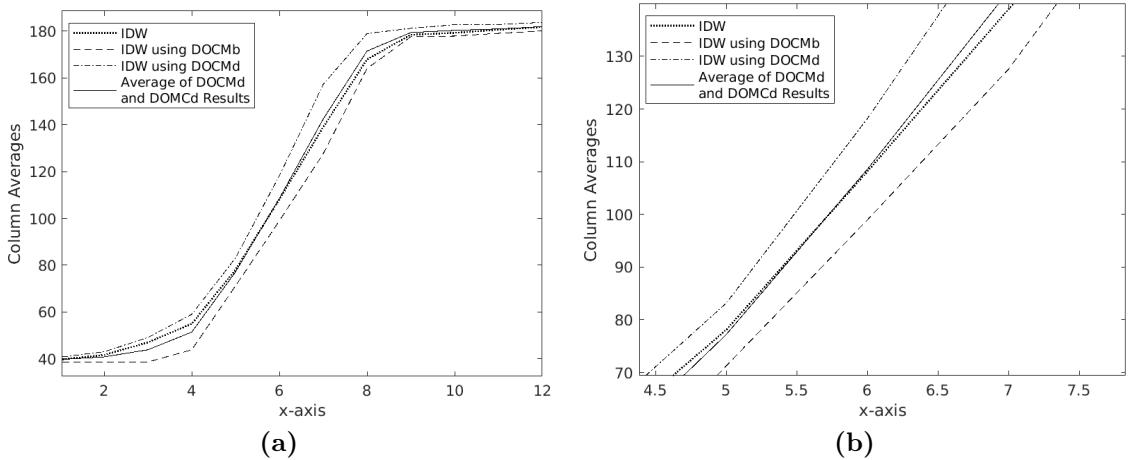
#### 4.2.1.1 Selection of Interpolation Values from DOCMs

Inverse distance weighted (IDW) interpolation method is used to reveal how DOCM<sub>b</sub> and DOCM<sub>d</sub> maps yield results within a large image window since other methods uses limited number of neighbouring samples. Power parameter of IDW is not modified, and its default value of 2 is used. Interpolation algorithm is applied on connectivity maps of containing an average of 49 samples each corresponding to a 7x7 window. Center pixel value is initially estimated using linear triangular interpolation. Then, IDW is applied on DOCM<sub>b</sub> and DOMC<sub>d</sub> computed based on this estimate. Reconstructed image pixel values are averaged vertically along the edge to visualize the edge profile as displayed in Figure 4.9.



**Figure 4.9.** Averaged sums of interpolation values computed on original, DOCMb and DOCMd windows along edge direction

In order to demonstrate how different aspects of DOCMb and DOCMd can be exploited to obtain sharper images, a fourth interpolation result is computed by averaging DOCMb and DOCMd results and displayed in Figure 4.10 below. Let  $z$  denote the interpolation value computed in the observation window using IDW interpolation method. Let  $z_b$  and  $z_d$  denote the interpolation values computed using IDW method in DOCMb and DOCMd windows, respectively. Average of DOCMb and DOCMd results,  $z_{avg} = \frac{z_b + z_d}{2}$ , is assigned as the interpolation value in each operation window. Fourth line added to Figure 4.10 is the interpolation result obtained using this averaging approach.

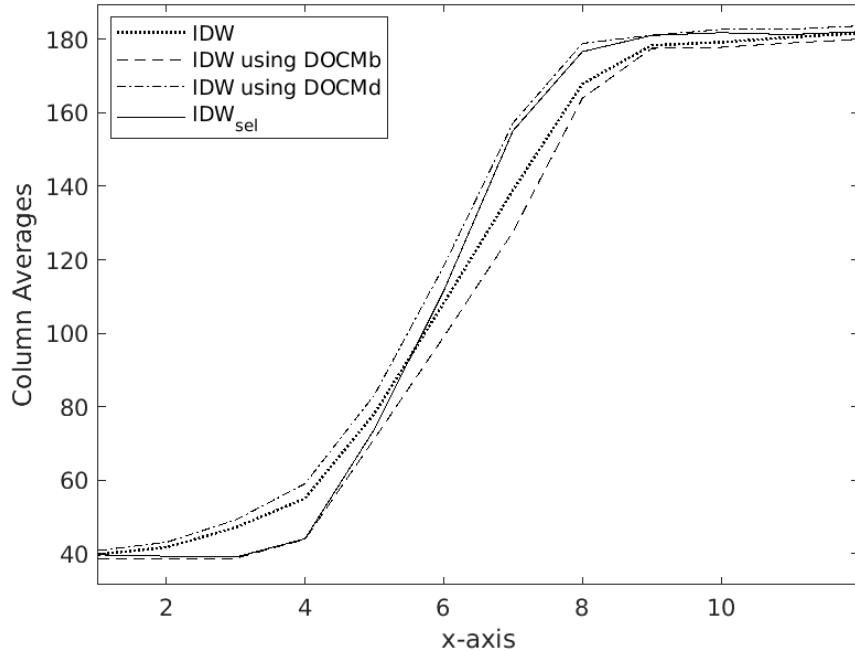


**Figure 4.10.** (a) Averaged sums along edge direction of the interpolation result obtained using average of DOCMb and DOCMd interpolation values is added to Figure 4.9 (b) closer look to inflection point in (a)

When DOCMb and DOCMd results are simply averaged, as can be seen in Figure 4.10 a sharper image is obtained. In the upper part of the edge,  $z_{avg}$  is slightly larger than  $z$  value, on the other hand  $z_{avg}$  becomes smaller in the lower part. Therefore, the comparison of  $z_{avg}$  and  $z$  values provide a way of detecting where the observation window is placed along the edge. When the  $z_{avg} = \frac{z_b+z_d}{2}$  is smaller than  $z$ ,  $z_b$  is selected to sharpen the edge. Reverse is true when  $\frac{z_b+z_d}{2}$  is higher than  $z$ . Based on this observation, the following simple logic could be established:

$$z_{sel} = \begin{cases} z_b, & \text{if } z > \frac{z_b+z_d}{2} \\ z_d, & \text{if } z \leq \frac{z_b+z_d}{2} \end{cases} \quad (4.4)$$

Interpolation result obtained using this selection based logic is referred to as IDW<sub>sel</sub> and its columns averages for the step edge is displayed in Figure 4.11 along with the results of conventional IDW, IDW applied on DOCMb and IDW applied on DOCMd.



**Figure 4.11.** Averaged sums along the edge direction of interpolation results obtained using selective processing based on Equation 4.4 added to Figure 4.9

As can be seen in Figure 4.11, this selection based process creates sharper images. In Figure 4.12 the reconstruction results are given for 'plate' test image using cubic and IDW interpolations on original grid and the result obtained by selection procedure. In this example, LR frames without additive noise are used.

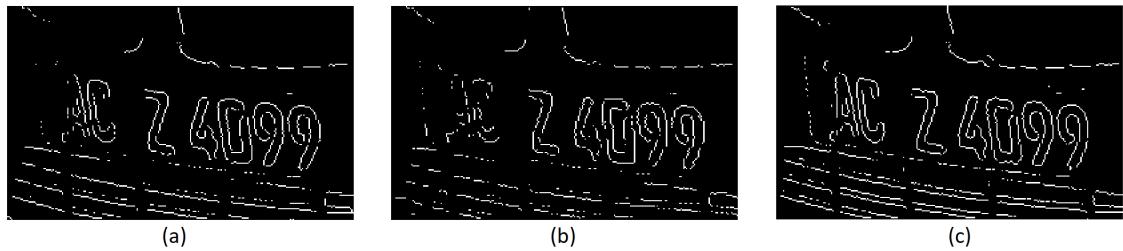


**Figure 4.12.** Reconstructed 'plate' image using (a) cubic, (b) IDW and (c) IDW<sub>sel</sub>

In Figure 4.12, it can be seen that selective processing of interpolation results of DOCMb and DOCMd windows yields visually better results. IDW<sub>sel</sub> result seems sharper compared to conventional IDW result. Therefore, selection of DOCMb and DOCMd interpolation considerably improved the interpolation result.

In order to reveal the effect of selection process on the edge performance, sobel edge operator of MATLAB is applied on each interpolation result and edge map is

given in Figure 4.13. In order to produce the results given in Figure 4.13, ‘edge’ function of MATLAB is applied to cubic interpolation result and the returned threshold value is recorded. The same threshold is provided to the ‘edge’ function to produce results in Figure 4.13.b and Figure 4.13.c. Utilization of selection based procedure reveals details that are not present in both cubic and regular IDW interpolations and letter contours are preserved better. It can be seen that the letter ‘A’ is more emphasized in  $IDW_{sel}$  than the others providing a better edge image. Cubic interpolation result seems to have superior results compared to conventional IDW but with the selection process the result of IDW is improved by integrating connected pixels to the center into the interpolation computation.



**Figure 4.13.** Sobel edge detector on reconstructed ‘plate’ image using (a) cubic, (b) IDW and (c)  $IDW_{sel}$

Selection based procedure has the potential to create jitters along the edge. There are irregular jumps between both locations and values of scattered data samples. Since the connectivity maps generated take any one sample as starting point in nonuniform data, reconstruction obtained by selection of different DOCMs results reflects the rapid changes in the underlying grid. In Figure 4.12.c, it can be observed that edge contours are not smooth compared to conventional IDW result. A balance between revealing edges and creating jitters can be reached by adding a weight parameter to the selection process.

#### 4.2.1.2 Weighted Average of Interpolation Values from DOCMs

In the selection based IDW technique, interpolation value can be one of the results based on DOCM<sub>b</sub> and DOCM<sub>d</sub>. Around the inflection point, the selection

procedure is shown to produce jitter. One solution to this problem can be using both DOCMb and DOCMd based interpolation results by giving weights to them. This approach is expected to yield results with less jitter along the edge. In this section, a weighted average of interpolation values from DOCMb and DOCMd is proposed to reduce the amount of jitter especially around the inflection point of a smoothly varying edge.

Let's define a new parameter  $\alpha$  as the distance between  $z_b$  and  $z_d$  divided by range of image intensity. This parameter is used to determine the weight assigned to DOCMb and DOCMd based interpolation values,  $z_b$  and  $z_d$ . It also requires a sign to indicate direction. It is demonstrated in previous section that if the  $\frac{z_b+z_d}{2}$  average is smaller than  $z$ , a value close to  $z_b$  must be selected to sharpen the edge. Then, the sign of  $\alpha$  is equal to the sign of  $z - \frac{z_b+z_d}{2}$ :

$$\alpha = \text{sign}(z - \frac{z_b + z_d}{2}) \left( \frac{z_d - z_b}{2^8} \right) \quad (4.5)$$

where  $2^8$  corresponds to an intensity range between 0 and 255. Using this parameter, weights of  $z_b$  and  $z_d$  can be determined. In Section 4.2.1.1, without any weight assignment, the average of DOCMb and DOCMd interpolation results is taken and is shown that the result has a sharper edge compared to conventional IDW results. Actually, this approach is a special case of weight assignment using alpha parameter when weights are simply 0.5. New weights can be calculated using:

$$w_b = 0.5(1 + \alpha) \quad (4.6a)$$

$$w_d = 0.5(1 - \alpha) \quad (4.6b)$$

The weight assignment can be further generalized by taking the integer powers of

$(1 + \alpha)$  and  $(1 - \alpha)$ :

$$w_b = 0.5(1 + \alpha)^n \quad (4.7a)$$

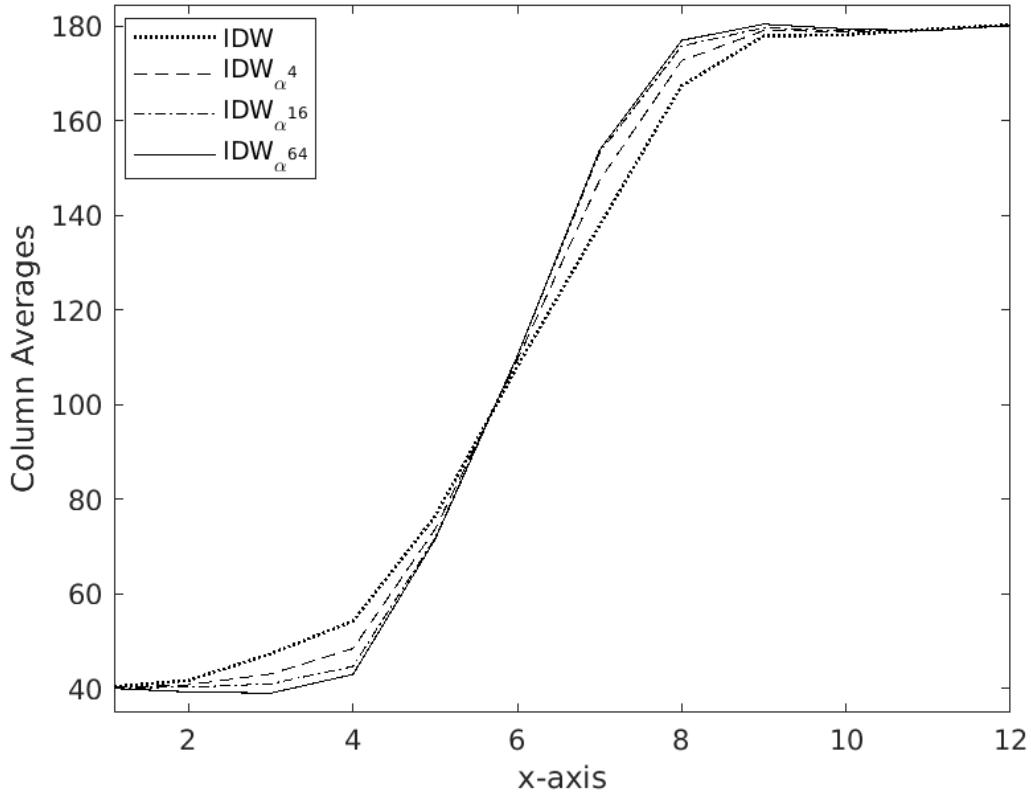
$$w_d = 0.5(1 - \alpha)^n \quad (4.7b)$$

Since  $\alpha$  has a value between 0 and 1, taking the power of  $(1 + \alpha)$  increases its value and a power of  $(1 - \alpha)$  is smaller than  $(1 - \alpha)$ . In order to improve sharpness, increasing powers of weight updates can be used.

After updating the weights,  $w_b + w_d$  may not add up to 1, hence new weights must be scaled before use, yielding the final equation to compute the interpolation value:

$$z_{\alpha^n} = \frac{w_b z_b + w_d z_d}{w_b + w_d} \quad (4.8)$$

Where  $n$  is the weight modification order. If  $z_b = z_d$ ,  $\alpha$  will be equal to zero and there is no update to weights. In fact, all  $z$ ,  $z_b$ ,  $z_d$  and  $z_{\alpha^n}$  values are equal in this case. An increased distance between DOCMb and DOCMd interpolation result indicates presence of an edge. When  $z_b$  and  $z_d$  are far apart, initial value of update term becomes higher. As the order increases, final results converges to that of selection based technique given in Section 4.2.1.1. These properties are demonstrated in Figure 4.14.



**Figure 4.14.** Averaged sums of interpolation values computed as weighted average of DOCMb and DOCMd window results along edge direction using different  $\alpha$  parameters

In Figure 4.14,  $IDW_{\alpha^4}$ ,  $IDW_{\alpha^{16}}$  and  $IDW_{\alpha^{64}}$  correspond to the result obtained using  $n$  values of 4, 16 and 64, respectively. It can be seen that improvement in the amount of obtained sharpness decreases at the larger values of  $n$ . This is in line with the exponential nature of power parameter.

1, 4 and 16 values of  $n$  represent recognizable changes in the sharpness of resulting images. In addition to these results, results obtained using the selection based procedure described in Section 4.2.1.1 and results of conventional cubic and IDW interpolations are presented on various test images in the next chapter.

## 5. RESULTS AND CONCLUSION

In order to test the proposed algorithm, synthetically generated LR sequences are employed since a widely accepted registered dataset for super-resolution applications is not available in the literature. Many works about super-resolution tend to keep their datasets disclosed and several datasets found are known to have registration issues. Eight bit HR images are first translated and rotated using randomly generated displacement parameters. Translational shifts of LR images are within a pixel boundary and maximum shift amount is limited by  $\pm 0.75$  pixel range. Also, rotational displacement is kept in the range  $[-3^\circ, 3^\circ]$ . Then, translated and rotated images are down-sampled by a factor of 4 using 4x4 mean filter. Finally, Gaussian noise with mean 0 and variance 25 is added to the down-sampled images.

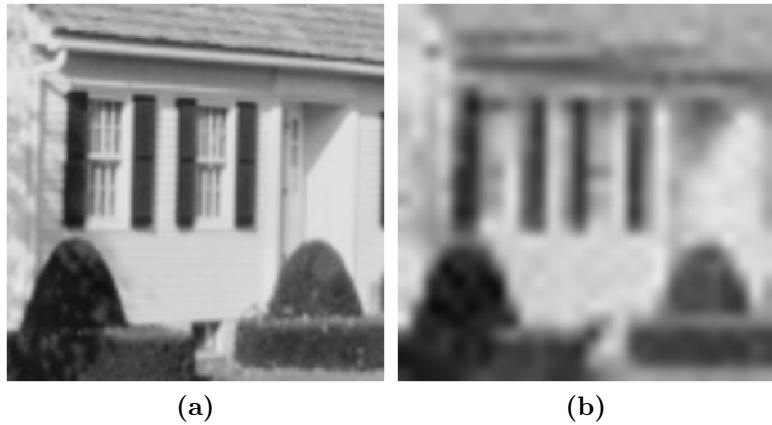
### 5.1. Super-Resolution Results on Various Test Images

The proposed topologically enhanced algorithm based on IDW is applied to all LR samples and HR images are generated. For comparison purposes, conventional IDW and cubic interpolation methods are also used and their results are compared to the proposed algorithm results. For cubic interpolation MATLAB's 'griddata' function is utilized. A MATLAB function to apply IDW is coded based on the description in [16, 45] and example implementation in [53]. Usually, interpolated images are deblurred by applying any deconvolution method that considers the presence of noise [17]. Since the focus of this work is on the interpolation stage, restoration is not applied to the interpolated images.

In the following sub-sections, the HR image and one of the LR frames is depicted. Then results of cubic, IDW,  $IDW_{\alpha^1}$ ,  $IDW_{\alpha^4}$ ,  $IDW_{\alpha^{16}}$  and  $IDW_{sel}$  are shown.

#### 5.1.1. Window (112x112) Test Image

In Figure 5.1.a HR 'window' image is depicted and in Figure 5.1.b one of the LR frames can be seen.



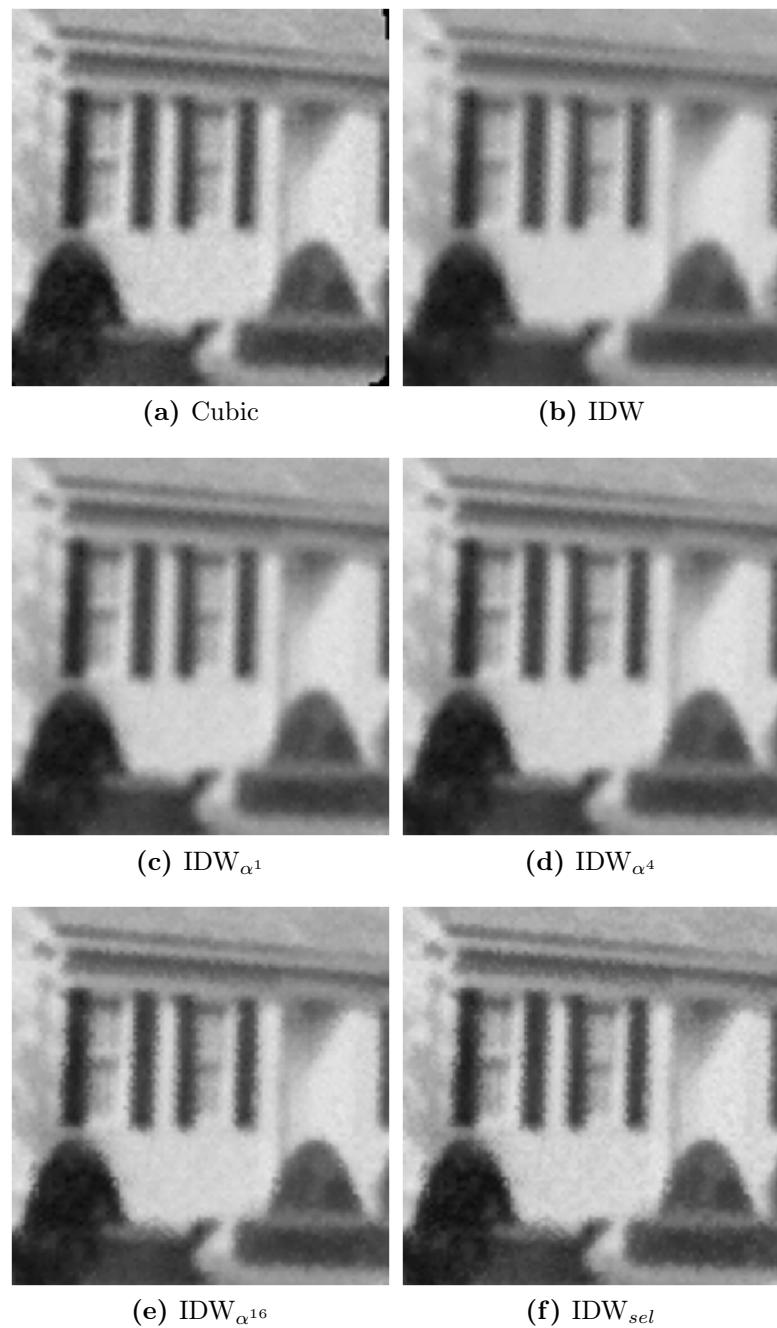
**Figure 5.1.** (a) HR ‘window’ image and (b) one of the LR images

Results of super-resolution algorithms for the ‘window’ test image is given in Figure 5.2. The conventional IDW result given in Figure 5.2.b is more blurry than the cubic super-resolution result given in Figure 5.2.a. IDW result has jitters in the region where two window lids are close. This result is obtained since the operation window includes samples from the other lid while computing the IDW interpolation values. The cubic interpolation result, on the other hand, does not have jitter between two window lids since operation window size of cubic interpolation is small and the interpolation value is only affected by nearby samples. However, the cubic interpolation result contains more noise than the IDW result because cubic interpolation values are computed using less number of pixels. IDW shows better noise suppression characteristics since a wider operation window is used.

In Figures 5.2.c and 5.2.d, results yielded by the proposed algorithm for  $n = 1$  and  $n = 4$  are depicted, respectively. Inclusion of connectedness information into the interpolation process filters out distant pixels and jitters as seen in Figure 5.2.b are removed. In Figures 5.2.c and 5.2.d, edges are noticeably sharper than the conventional IDW result. It can also be seen that noise suppression characteristic of IDW is preserved. It can be said that  $IDW_{\alpha^4}$  results are similar to a less noisy version of cubic interpolation results.

In Figures 5.2.e and 5.2.f, results of  $IDW_{\alpha^4}$  and  $IDW_{sel}$  are shown, respectively. Around the inflection point of the edge, the interpolation algorithm interpolation

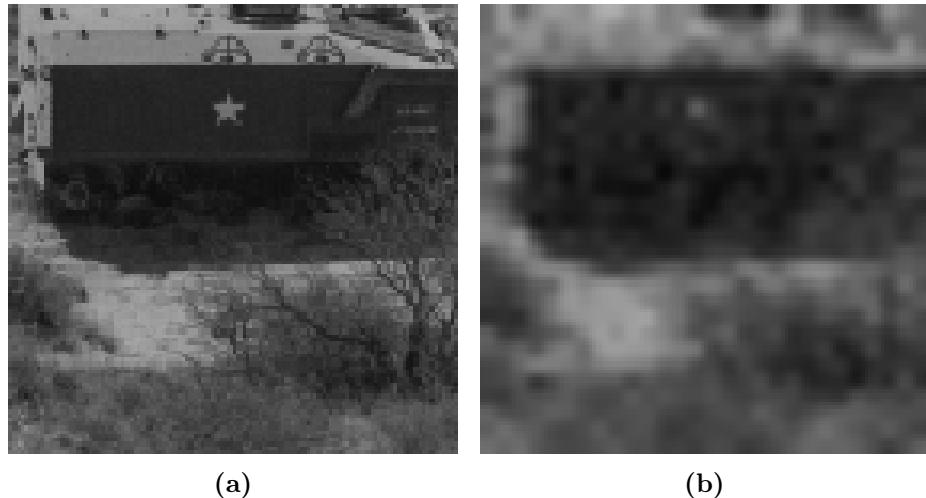
applied on DOCMb and DOCMd produces slightly different results. Differences of interpolation results based on DOCMb and DOCMd yields jitter in the middle of smoothly varying edges at the output. In Figures 5.2.e and 5.2.f, we observe this phenomenon. Image sharpness is increased compared to results in which smaller values of  $n$  were utilized, at the cost of jitter around edges. It is also observed that the amount of noise increases due to the same reason.



**Figure 5.2.** Super-resolution results of ‘window’ test image

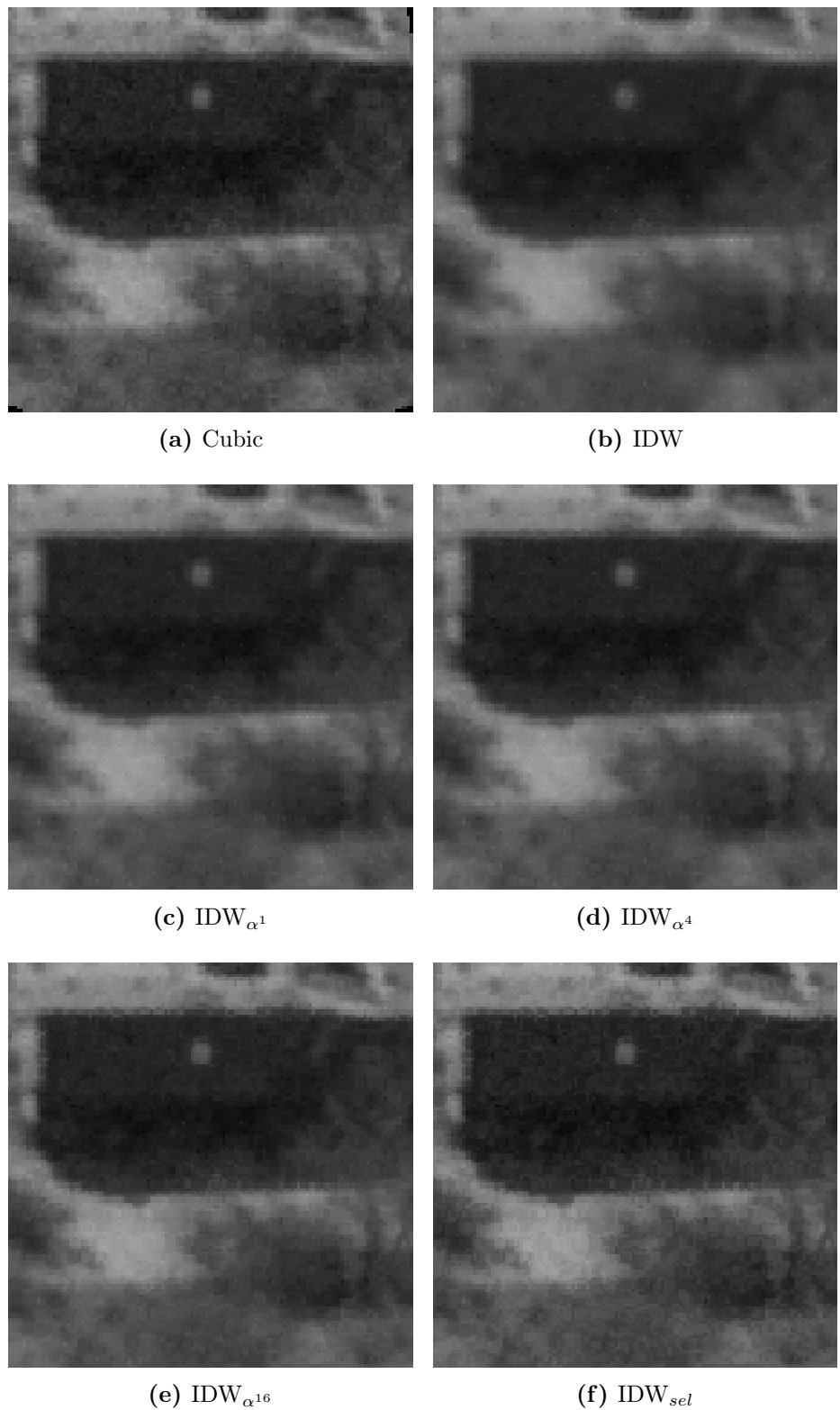
### 5.1.2. Military Vehicle (112x112) Test Image

In Figure 5.3.a HR ‘military vehicle’ image is shown and in Figure 5.3.b one of the LR frames can be seen.



**Figure 5.3.** (a) HR ‘military vehicle’ image and (b) one of the LR images

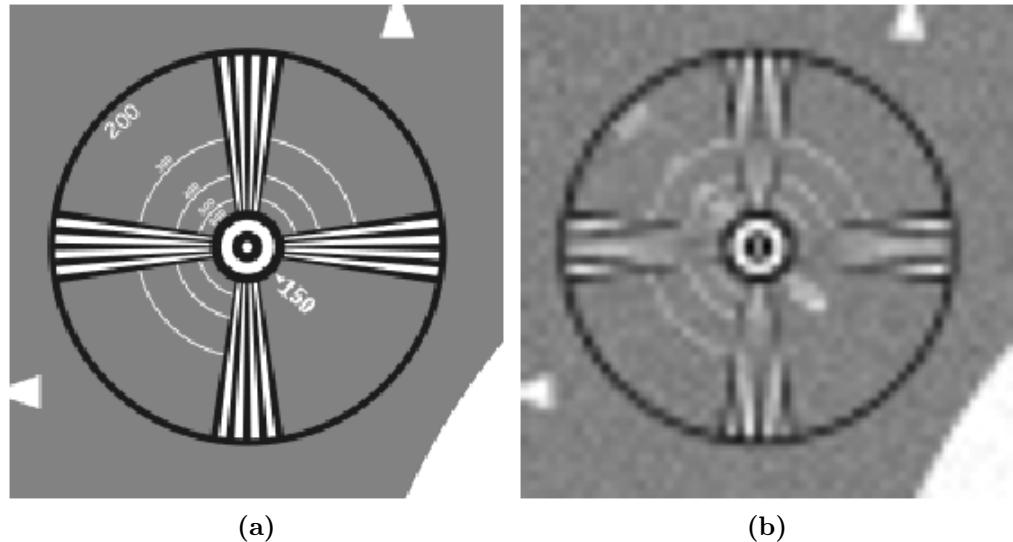
It seems that cubic interpolation preserves the edges better than the conventional IDW. However, cubic interpolation image is much noisier than IDW especially in flat surfaces, such as the side of the vehicle. IDW is smoother than the cubic interpolation result. When the cubic interpolation output is compared to  $IDW_{\alpha^n}$  results,  $IDW_{\alpha^n}$  seems to have less noise on flat surfaces. As the parameter n is increased, edges are enhanced with some jitter at the expense of more noise on flat areas.



**Figure 5.4.** Super-resolution results of ‘military vehicle’ test image

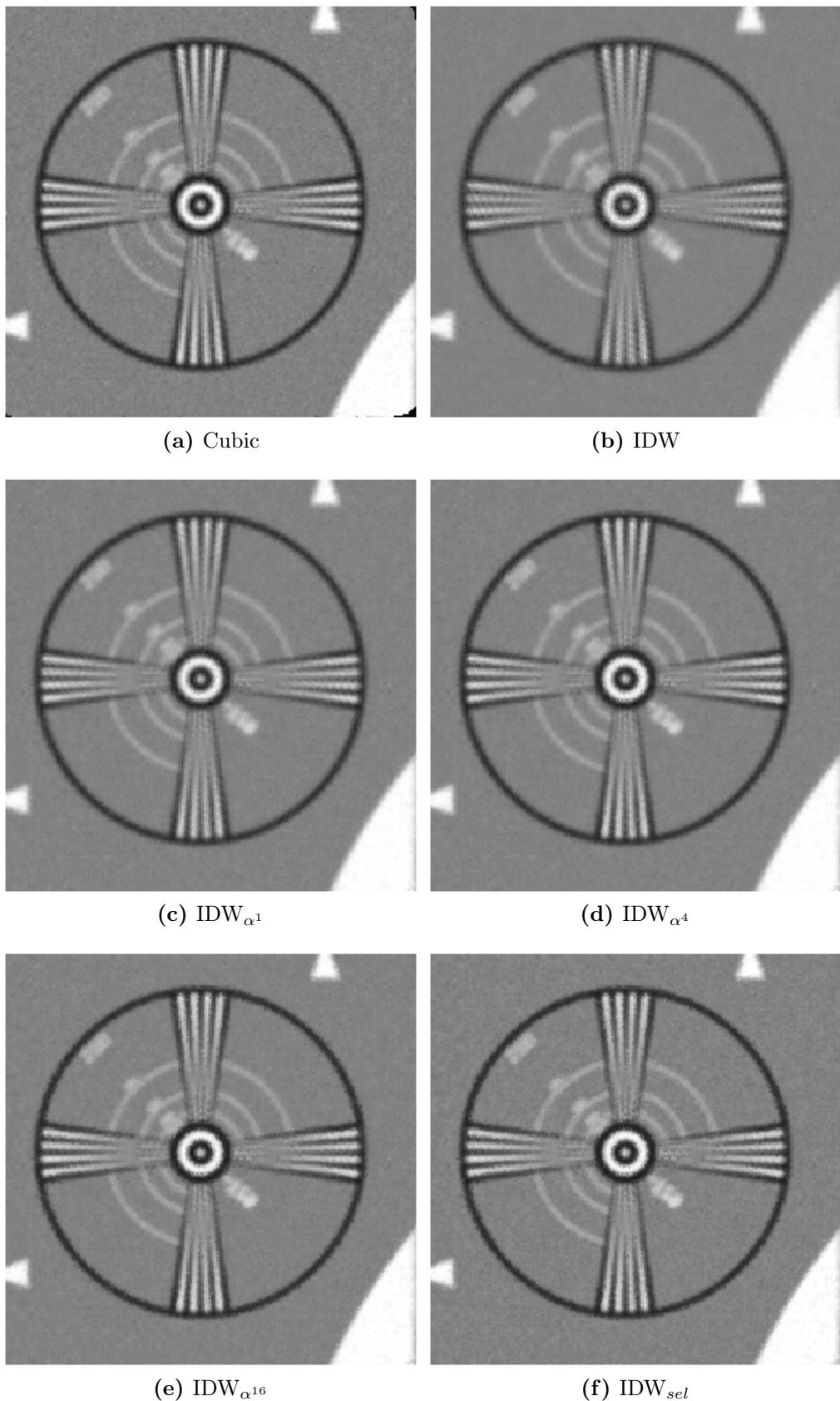
### 5.1.3. Resolution Chart (220x220) Test Image

In Figure 5.5.a HR ‘resolution chart’ image is given and in Figure 5.5.b one of the LR frames can be seen.



**Figure 5.5.** (a) HR ‘resolution chart’ image and (b) one of the LR images

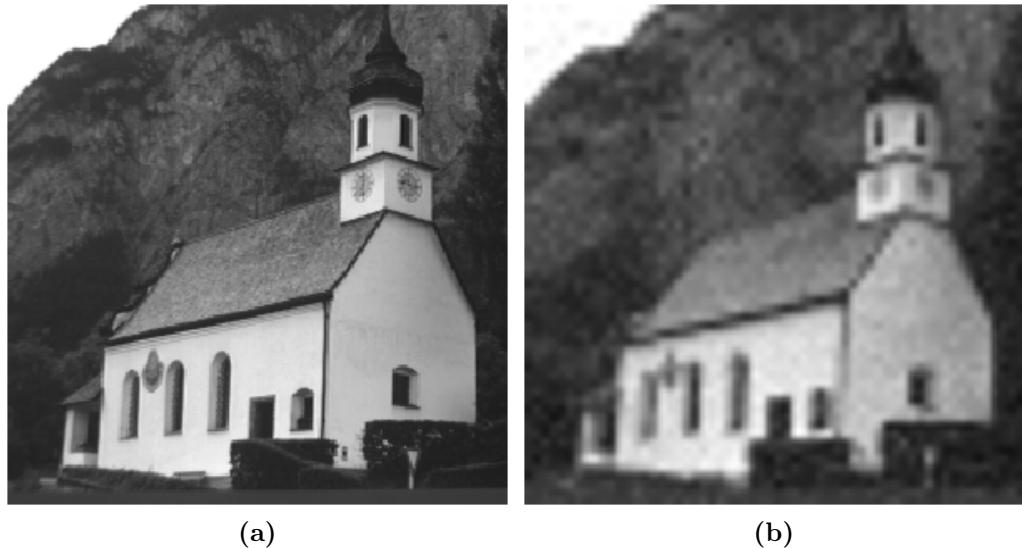
Results of super-resolution algorithms for the ‘resolution chart’ test image are shown in Figure 5.6. The same kind of behavior of cubic and IDW algorithms are visible as shown in above examples. IDW is blurred and cubic interpolation is sharper than IDW result but flat areas are contaminated with noise. Increasing the parameter  $n$  enhances edge performance, however, more noise is visible. As a compromise between noise and edge preservation, the parameter  $n$  with value 4 gives an acceptable output as compared to the cubic and the IDW results.



**Figure 5.6.** Super-resolution results of ‘resolution chart’ test image

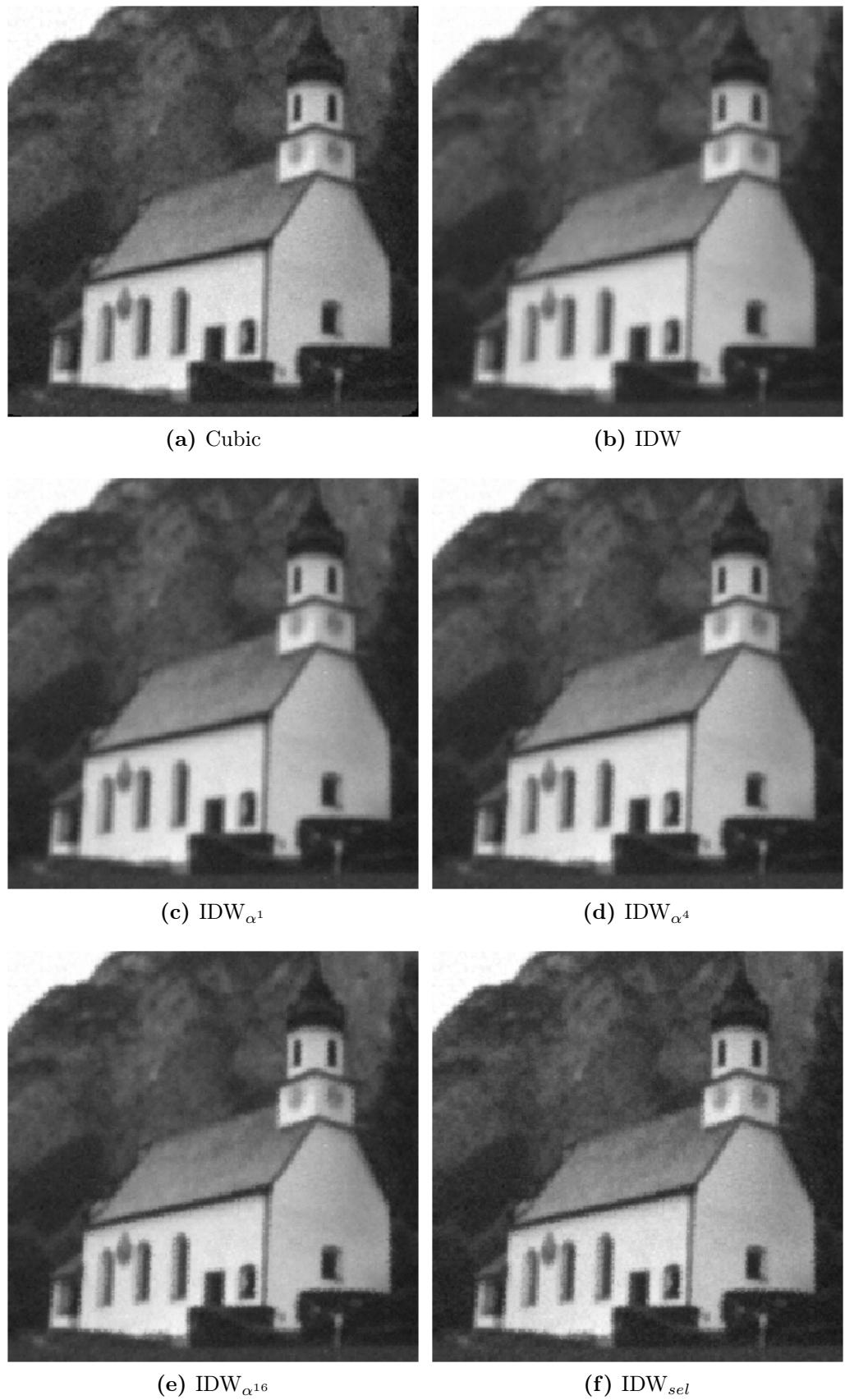
#### 5.1.4. Church (240x240) Test Image

In Figure 5.7.a HR ‘church’ image is depicted and in Figure 5.7.b one of the LR frames can be seen.



**Figure 5.7.** (a) HR ‘clock’ image and (b) one of the LR images

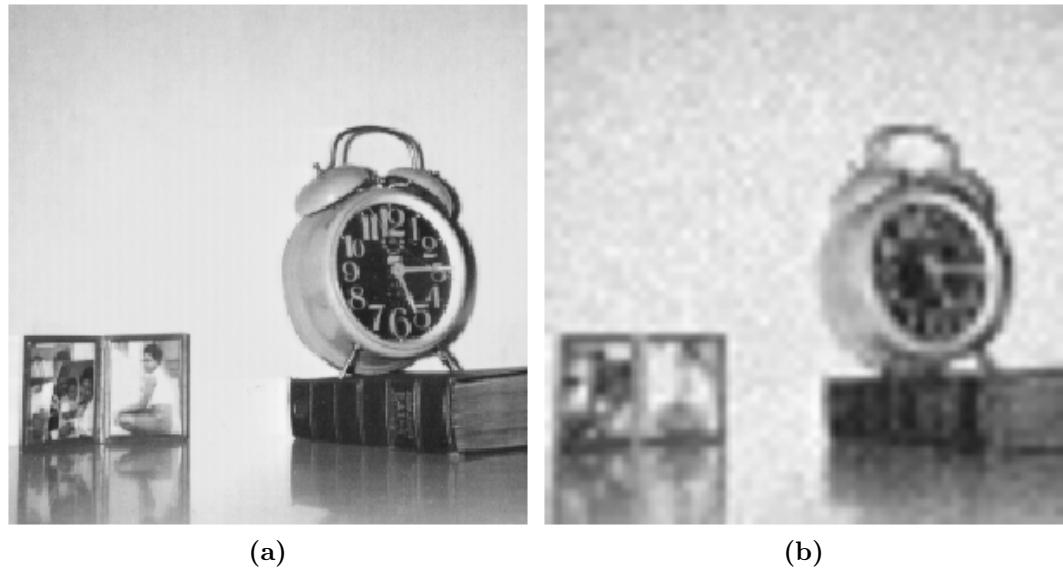
Results of super-resolution algorithms for ‘church’ test image is shown in Figure 5.8. It seems that  $IDW_{\alpha^n}$  results have thinner edge lines compared to IDW and cubic results on the front building corner and roof corners.  $IDW_{\alpha^n}$  results become more noisy when  $n \geq 16$  compared to  $IDW_{\alpha^n}$  results when  $n \leq 4$ .  $IDW_{\alpha^1}$  and  $IDW_{\alpha^4}$  results are smoother compared to cubic interpolation result.



**Figure 5.8.** Super-resolution results of ‘church’ test image

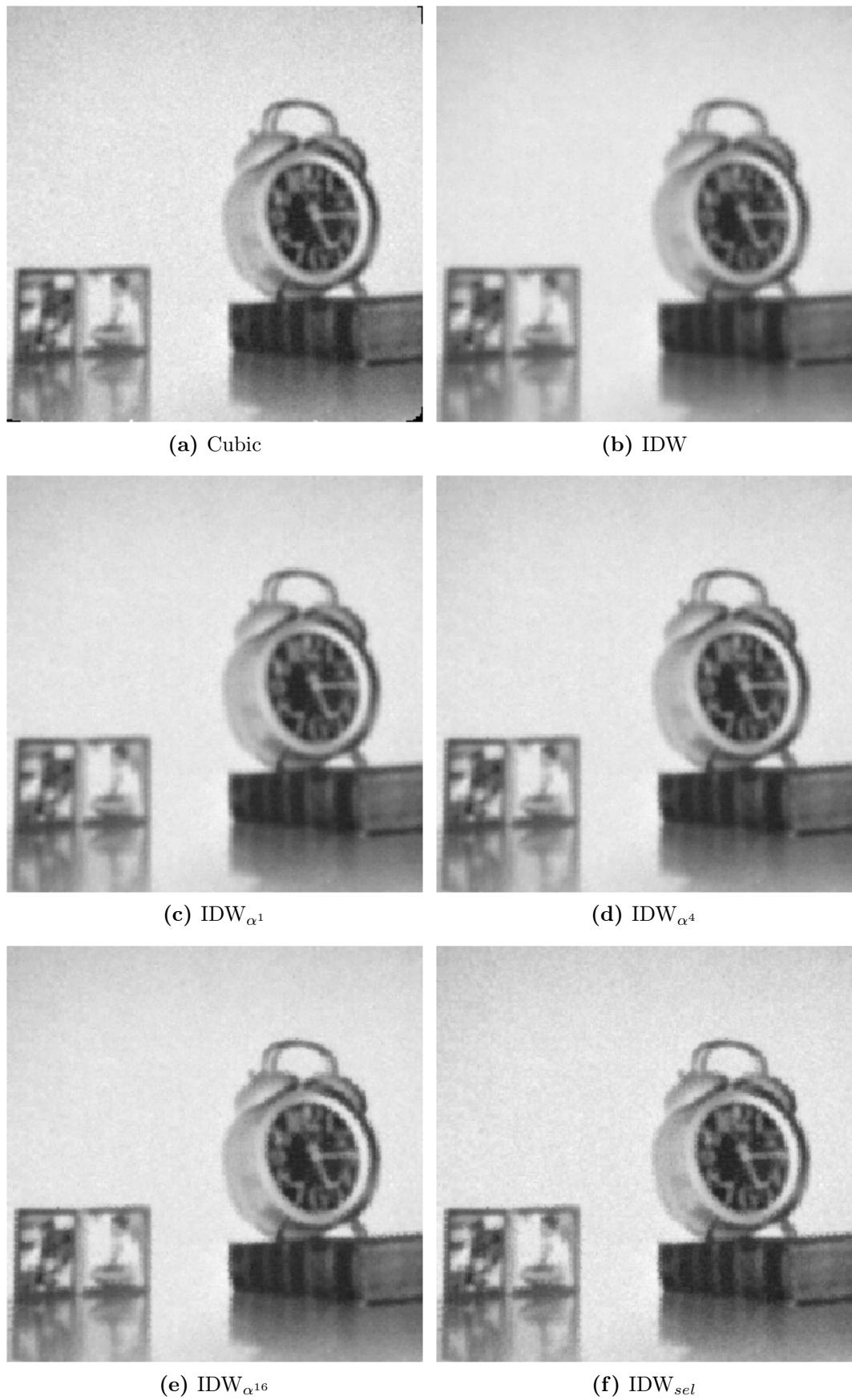
### 5.1.5. Clock (240x240) Test Image

In Figure 5.9.a HR ‘clock’ image is depicted and in Figure 5.9.b one of the LR frames can be seen.



**Figure 5.9.** (a) HR ‘clock’ image and (b) one of the LR images

Results of super-resolution algorithms for ‘clock’ test image is displayed in Figure 5.10. As can be seen from the wall IDW $_{\alpha^n}$  results contain less noise compared to cubic interpolation result. Numerals ‘6’ and ‘8’ seem to be more readable in IDW $_{\alpha^4}$ , IDW $_{\alpha^{16}}$  and IDW $_{sel}$  results compared to IDW interpolation result.



**Figure 5.10.** Super-resolution results of ‘clock’ test image

### 5.1.6. Plate (160x260) Test Image

In Figure 5.11.a HR ‘plate’ image is depicted and in Figure 5.11.b one of the LR frames can be seen.



**Figure 5.11.** (a) HR ‘plate’ image and (b) one of the LR images

Results of super-resolution algorithms for ‘plate’ test image is displayed in Figure 5.12. Letter contours are visibly sharper in  $IDW_{\alpha^4}$ ,  $IDW_{\alpha^{16}}$  and  $IDW_{\alpha^{sel}}$  results compared to cubic and IDW results. As can be seen from the white and light gray regions of the image  $IDW_{\alpha^n}$  results contain less noise compared to cubic interpolation result.



**Figure 5.12.** Super-resolution results of ‘plate’ test image

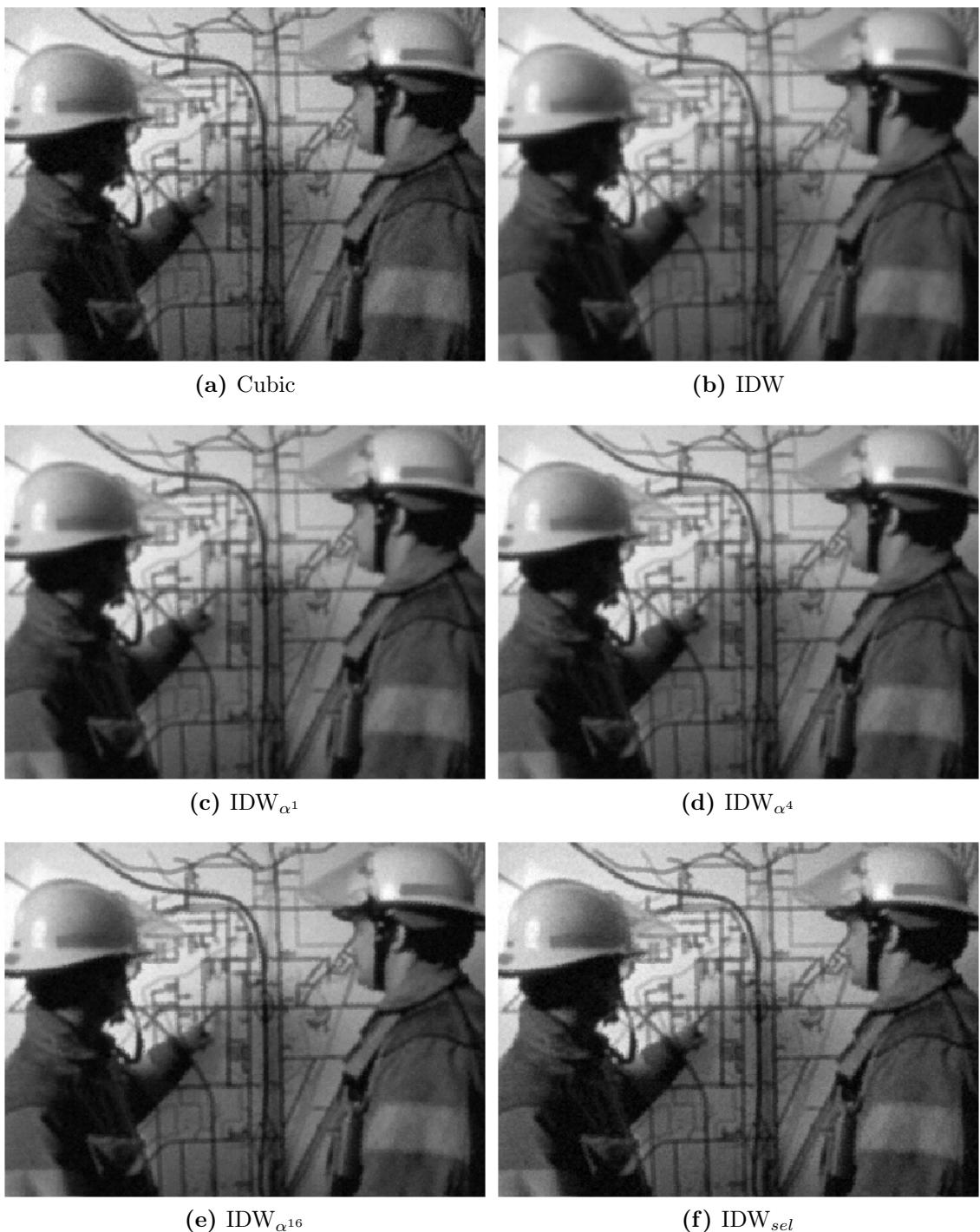
### 5.1.7. Workers (268x364) Test Image

In Figure 5.13.a HR ‘workers’ image is depicted and in Figure 5.13.b one of the LR frames can be seen.



**Figure 5.13.** (a) HR ‘workers’ image and (b) one of the LR images

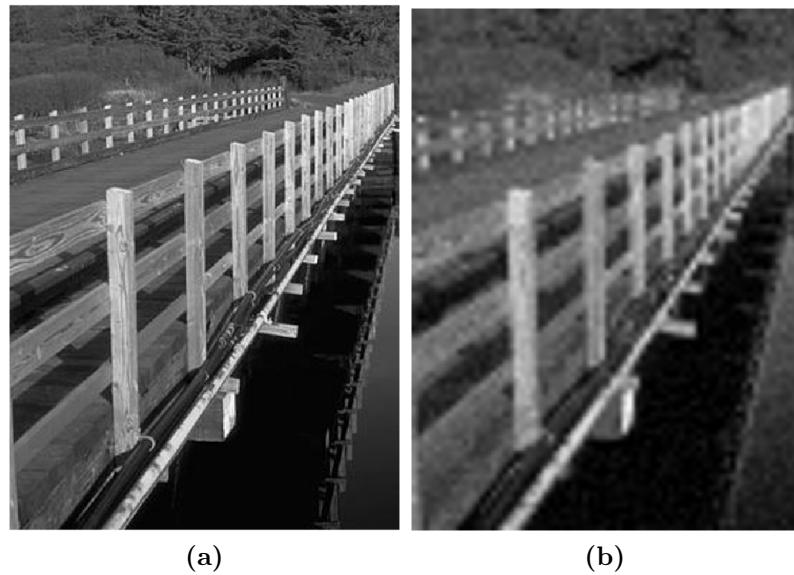
Results of super-resolution algorithms for ‘workers’ test image is displayed in Figure 5.14. Black lines (paths) on the maps seems thinner in  $IDW_{\alpha^4}$ ,  $IDW_{\alpha^{16}}$  and  $IDW_{sel}$  results compared to cubic interpolation result. As can be seen from the caskets and arms of the workers in the image  $IDW_{\alpha^n}$  results are contaminated by less noise compared to cubic interpolation result.



**Figure 5.14.** Super-resolution results of ‘workers’ test image

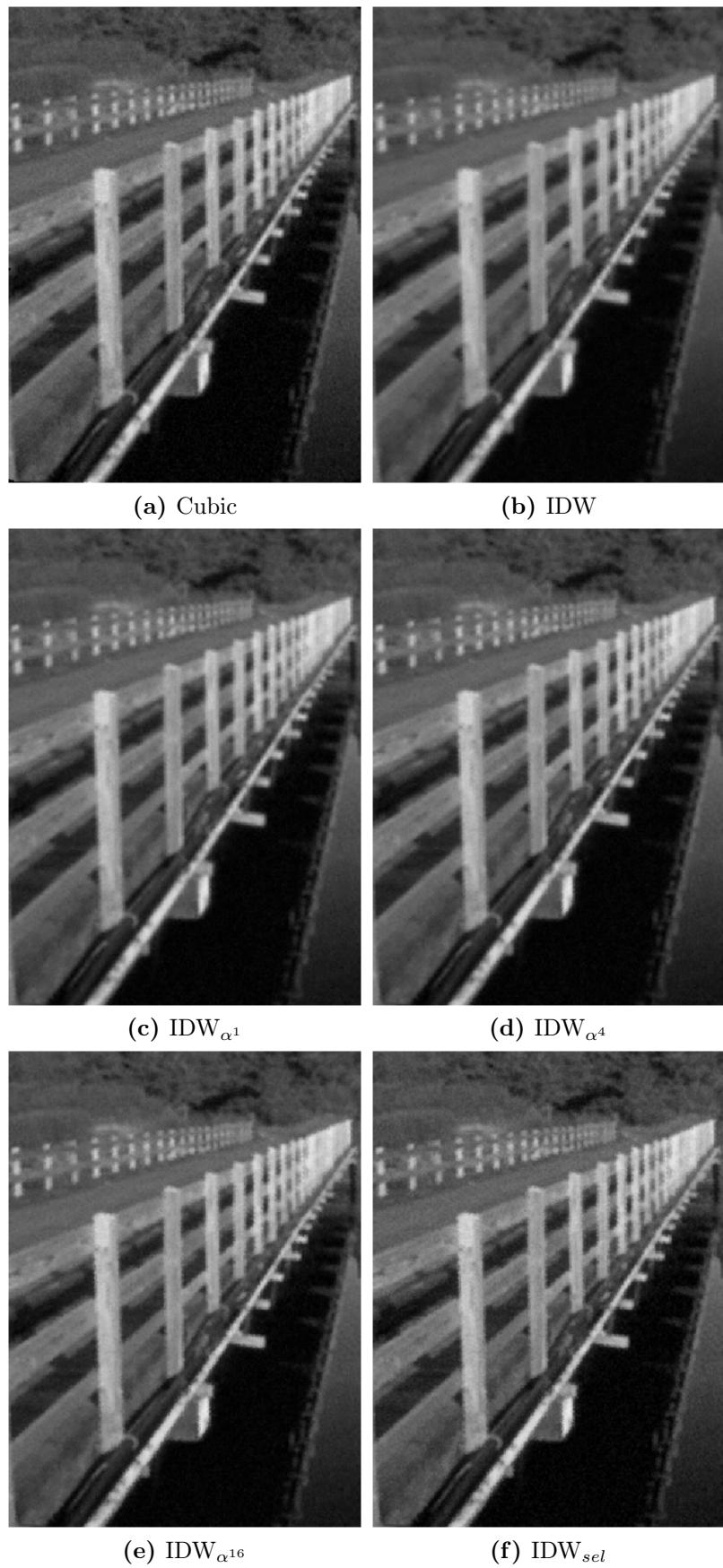
### 5.1.8. Bridge (364x270) Test Image

In Figure 5.15.a HR ‘bridge’ image is shown and in Figure 5.15.b one of the LR frames can be seen.



**Figure 5.15.** (a) HR ‘bridge’ image and (b) one of the LR images

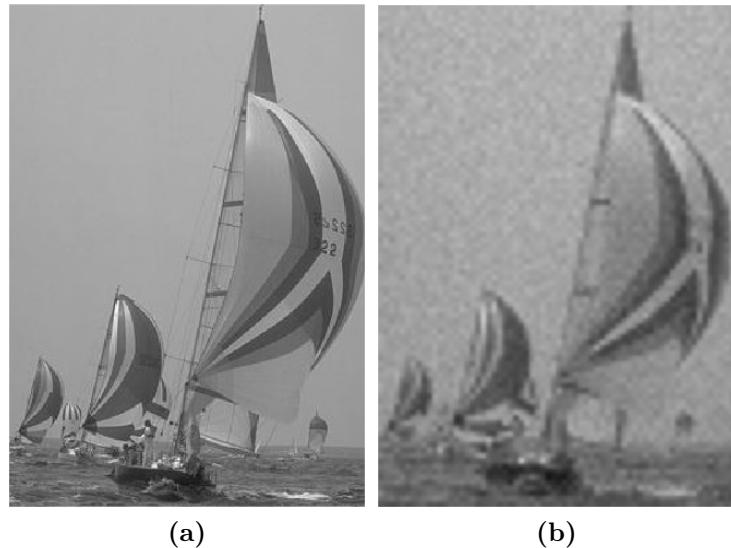
Results of super-resolution algorithms for ‘bridge’ test image is depicted in Figure 5.16. As can be seen from the still water detail int he image  $IDW_{\alpha^n}$  results contain less noise compared to cubic interpolation result. Edges of the wooden bridge arch seems to be sharper in  $IDW_{\alpha^{16}}$  and  $IDW_{sel}$  results compared to cubic interpolation result.



**Figure 5.16.** Super-resolution results of ‘bridge’ test image

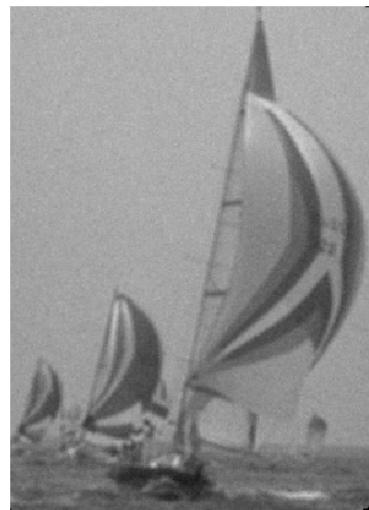
### 5.1.9. Sailboats (372x264) Test Image

In Figure 5.17.a HR ‘sailboats’ shown is given and in Figure 5.17.b one of the LR frames can be seen.



**Figure 5.17.** (a) HR ‘sailboats’ image and (b) one of the LR images

Results of super-resolution algorithms for ‘sailboats’ test image is depicted in Figure 5.18. IDW is blurred and cubic interpolation is sharper than IDW result but flat areas such as sail-cloth are contaminated with noise. When the cubic interpolation output is compared to  $IDW_{\alpha^n}$  results,  $IDW_{\alpha^n}$  seems to have less noise on flat surfaces. As the parameter n is increased, edges are enhanced with some jitter at the expense of more noise on flat areas.



(a) Cubic



(b) IDW



(c) IDW $_{\alpha^1}$



(d) IDW $_{\alpha^4}$



(e) IDW $_{\alpha^{16}}$



(f) IDW $_{sel}$

**Figure 5.18.** Super-resolution results of ‘sailboats’ test image

### 5.1.10. Lena (496x496) Test Image

In Figure 5.19.a HR ‘lena’ image is depicted and in Figure 5.19.b one of the LR frames can be seen.



**Figure 5.19.** (a) HR ‘lena’ image and (b) one of the LR images

Results of super-resolution algorithms for ‘lena’ test image is shown in Figure 5.20.  $IDW_{\alpha^1}$  and  $IDW_{\alpha^4}$  results are smoother compared to cubic interpolation result. As the parameter  $n$  is increased, edges are enhanced with some jitter as can be observed at the fur in the image. Increasing the parameter  $n$  enhances edge performance, however, more noise is visible.



**Figure 5.20.** Super-resolution results of ‘lena’ test image

## 5.2. Computational Cost

The major disadvantage of the proposed algorithm is its computational cost. Computation times for some of the test images are given in Table 5.1. Test images are processed on an Intel i7-7700 HQ CPU and 4 CPU cores are utilized.

**Table 5.1.** Computation Time

	Cubic	IDW	IDW using DOCM
Milatary Vehicle (112x112)	0.09s	5s	1329s (0.37 hrs)
Plate (160x260)	0.25s	25s	5286s (1.46 hrs)
Resolution Chart (220x220)	0.3s	30s	6732s (1.87 hrs)
Clock (240x240)	0.35s	40s	8763s (2.43 hrs)
Sailboat (372x264)	0.60s	95s	19258s (5.34 hrs)
Bridge (364x270)	0.62s	96s	19373s (5.35 hrs)

Computation profile of the proposed algorithm for ‘church’ test image is given in Table 5.2. First five functions that take the most time are listed. ‘Calls’ column displays how many times the function is called. ‘Total Time’ column represents the total duration of the function, including its child functions. ‘Self Time’ column displays the duration of time the function is run, excluding the time spent in its child functions. ‘binaryConnected’ is a known child function of DOCMdt.

**Table 5.2.** Computation Profile

	Calls	Total Time	Self Time
DOCMdt	28800	213.65min	42.72min
DOCMdt>binaryConnected	740908	170.92min	68.98min
setdiff	26710852	59.09min	10.01min
unique	53421704	50.97min	28.76min
ismember	53436104	33.05min	9.07min

As can be seen in Table 5.2, most of the CPU usage is dedicated to DOCM computation, i.e. Algorithm 4.3. Specifically, the function used to compute binary connectedness, i.e. Algorithm 4.1, takes the most time. ‘setdiff’ and ‘unique’ are MATLAB functions used to access natural neighbours inside the binaryConnected

function. ‘*isMember*’ is another MATLAB function used to check whether an interpolation point is already included in one of the LR frames. The last three functions mentioned may also be a part of other builtin MATLAB functions used (in addition to their explicit use to implement the proposed algorithm).

In order to reduce computation time, connected component labeling algorithm used on thresholded binary images (Algorithm 4.1) needs to be improved.

### 5.3. Conclusion

The main stages and models in SR theory are reviewed with a particular focus on nonuniform interpolation methods used in the HR image estimation stage. Conventionally, methods such as cubic triangular interpolation are preferred in this stage due to their low computational cost. However, conventional techniques are not good at modeling edges in images and they are susceptible to noise because of the low order of polynomials used and the small number of samples included in the interpolation.

Global interpolation methods, such as the inverse distance weighted interpolation investigated in this work, usually have better noise suppression characteristics compared to conventional methods. However, inverse distance weighted interpolation produces poorer results around edges compared to conventional techniques due to unconnected LR pixels present within the wide interpolation window.

In this thesis, an algorithm to solve the problem of unconnected LR pixels is proposed. First, LR measurements belonging to the same object are detected. For this purpose, degree of connectedness map computation algorithms are revised to work on nonuniformly distributed data. Then, behaviour of connectivity maps in brightness and darkness topology is exploited to improve the edge performance of inverse distance weighted interpolation. Knowledge about interpolation node location along an edge is obtained from relative intensity value of the interpolation result computed in the original window with respect to the interpolation result computed using degree of connectedness maps in brightness and darkness topology. In order to

improve sharpness, interpolation nodes closer to the high intensity part of an edge are assigned the interpolation values computed using degree of connectedness maps in darkness topology and vice versa. However, selective processing is shown to produce jitter along the edges. A tunable parameter is added to the proposed algorithm in order to control sharpness by giving weights to interpolation results computed using degree of connectedness maps. It is shown that this approach reduces the amount of jitter around the inflection point of a smoothly varying edge.

Synthetically generated LR sequences are employed in order to test the proposed algorithm. It is demonstrated that the proposed algorithm yields a superior performance in edge preservation and noise suppression compared to cubic interpolation and conventional IDW methods. The major disadvantage of the proposed algorithm is its computational cost.

## REFERENCES

- [1] Greenspan, H. (2008). Super-Resolution in Medical Imaging. *The Computer Journal*, 52(1):43–63.
- [2] Gilman, A., Bailey, D. G., and Marsland, S. R. (2008). Interpolation models for image super-resolution. In *Proceedings - 4th IEEE International Symposium on Electronic Design, Test and Applications, DELTA 2008*, pages 55–60.
- [3] Brown, L. G. and Brown, L. G. (1992). A Survey of Image Registration Techniques. *ACM COMPUTING SURVEYS*, 24:325—376.
- [4] Kuglin, C. D. and Hines, D. C. (1975). The Phase Correlation Image Alignment Method. In *IEEE International Conference on Cybernetics and Society*, pages 163–165, New York.
- [5] Lewis, J. P. (1995). Fast Template Matching. In *Vision Interface*, pages 120–123, Quebec City.
- [6] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- [7] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, pages 404–417. Springer, Berlin, Heidelberg.
- [8] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, pages 2564–2571. IEEE.
- [9] Alcantarilla, P. F., Bartoli, A., and Davison, A. J. (2012). KAZE Features. In *European Conference on Computer Vision*, pages 214–227. Springer, Berlin, Heidelberg.
- [10] Alcantarilla, P., Nuevo, J., and Bartoli, A. (2013). Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. In *Proceedings of the British Machine Vision Conference 2013*, pages 13.1–13.11. British Machine Vision Association.
- [11] Tsai, R. and Huang, T. (1984). Multiframe image restoration and registration. In *Image reconstruction from incomplete observations*, page 317. JAI Press Inc., Greenwich, CT.
- [12] Stark, H. and Oskoui, P. (2008). High-resolution image recovery from image-plane arrays, using convex projections. *Journal of the Optical Society of America A*, 6(11):1715.
- [13] Tekalp, A., Ozkan, M., and Sezan, M. (1992). High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration. In *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 169–172 vol.3. IEEE.

- [14] Kang, M. G. (1998). Generalized multichannel image deconvolution approach and its applications. *Optical Engineering*, 37(11):2953.
- [15] Hardie, R. C., Barnard, K. J., and Armstrong, E. E. (1997). Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Transactions on Image Processing*, 6(12):1621–1633.
- [16] Amidror, I. (2002). Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, 11(2):157.
- [17] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang (2003). Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, 20(3):21–36.
- [18] Zitová, B. and Flusser, J. (2003). Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000.
- [19] Zheng, L., Wang, Y., and Hao, C. (2011). Cross-Correlation Registration Algorithm Based on the Image Rotation and Projection. In *2011 4th International Congress on Image and Signal Processing*, pages 1095–1098. IEEE.
- [20] Berberidis, K. and Karybali, I. (2002). A New Efficient Cross-Correlation Based Image Registration Technique with Improved Performance. In *11th European Signal Processing Conference*, pages 1–4, Toulouse.
- [21] Tian, Q. and Huhns, M. N. (1986). Algorithms for Subpixel Registration. *Computer Vision, Graphics, and Image Processing*, 35(2):220–233.
- [22] Reddy, B. and Chatterji, B. (1996). An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271.
- [23] Qin-Sheng Chen, Defrise, M., and Deconinck, F. (1994). Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1156–1168.
- [24] Foroosh, H., Zerubia, J., and Berthod, M. (2002). Extension of Phase Correlation to Subpixel Registration. *IEEE Transactions on Image Processing*, 11(3):188–200.
- [25] Vera, E. and Torres, S. (2008). Subpixel Accuracy Analysis of Phase Correlation Registration Methods Applied to Aliased Imagery. In *16th European Signal Processing Conference*, Lausanne, Switzerland. IEEE.
- [26] Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.
- [27] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

- [28] Tareen, S. A. K. and Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–10. IEEE.
- [29] Milanfar, P. (2011). *Super-resolution imaging*. CRC Press, Boca Raton.
- [30] Rogers, C. A. (1964). Packing and Covering. *Cambridge Tracts in Mathematics and Mathematical Physics*, 54(6):450–450.
- [31] Edelsbrunner, H. (2000). Triangulations and meshes in computational geometry. In *Acta Numerica*, pages 133–213. Cambridge University Press.
- [32] Maddah, M. and Cao, C. (2017). Application of the Alpha Shape Method to Visualize and Analyze Surgical Motion. *Surgical Science*, 08:464–480.
- [33] Shamos, M. I. and Hoey, D. (1975). Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 151–162.
- [34] Gallier, J. (2011). Dirichlet-Voronoi Diagrams and Delaunay Triangulations. In *Geometric Methods and Applications*, pages 301–319. Springer, New York.
- [35] Brown, K. Q. (1979). Voronoi diagrams from convex hulls. *Information Processing Letters*, 9(5):223–228.
- [36] Lawson, C. L. (1972). Transforming triangulations. *Discrete Mathematics*, 3(4):365–372.
- [37] Möbius, A. F. (1827). *Der barycentrische Calcül ein neues Hülfsmittel zur analytischen Behandlung der Geometrie*. Verlag von Johann Ambrosius Barth, Leipzig, 1 edition.
- [38] Farin, G. (1985). A modified Clough-Tocher interpolant. *Computer Aided Geometric Design*, 2(1-3):19–27.
- [39] Lawson, C. (1977). Software for C1 Surface Interpolation. In *Mathematical Software*, pages 161–194. Elsevier.
- [40] De Berg, M., Cheong, O., Van Kreveld, M., and Overmars, M. (2008). *Computational geometry: Algorithms and applications*. Springer Berlin Heidelberg.
- [41] Farin, G. (1986). Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, 3(2):83–127.
- [42] Kong, V. P., Ong, S. H., and Saw, K. H. (2004). Range Restricted Interpolation Using Cubic Bezier Triangles. *IEEE Access*.
- [43] Siu, W. and Hung, K.-W. (2012). Review of image interpolation and super-resolution. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*.

- [44] Sibson, R. (1981). A brief description of natural neighbor interpolation. In Barnett, V., editor, *Interpreting Multivariate Data*, chapter 2, pages 21–36. John Wiley & Sons, New York.
- [45] Sibson, R. (1980). A vector identity for the dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87(1):151–155.
- [46] Boissonnat, J. D. and Gazais, F. (2002). Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry: Theory and Applications*, 22(1-3):185–203.
- [47] Park, S. W., Linsen, L., Kreylos, O., Owens, J. D., and Hamann, B. (2006). Discrete sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):243–252.
- [48] Shepard, D. (1968). A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM ’68, pages 517–524, New York, NY, USA. Association for Computing Machinery.
- [49] Chen, C., Zhao, N., Yue, T., and Guo, J. (2015). A generalization of inverse distance weighting method via kernel regression and its application to surface modeling. *Arabian Journal of Geosciences*, 8(9):6623–6633.
- [50] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., USA.
- [51] Watson, D. (1985). A refinement of inverse distance weighted interpolation. *Geo-Processing*, 2:315–327.
- [52] (2020). Inverse distance weighting interpolation/fitting - ALGLIB, C++ and C#.
- [53] Langella, G. (2021). Inverse Distance Weighted (IDW) or Simple Moving Average (SMA) Interpolation.
- [54] Kamińska, A. and Grzywna, A. (2014). Comparison of deterministic interpolation methods for the estimation of groundwater level. *Journal of Ecological Engineering*, 15(4):55–60.
- [55] Mareboyana, M. and Le Moigne, J. (2018). Super-resolution of remote sensing images using edge-directed radial basis functions. In *SPIE-Intl Soc Optical Eng*, page 35.
- [56] Hunt, T. (2004). *Image Super-Resolution Using Adaptive 2-D Gaussian Basis Function Interpolation*. PhD thesis, AIR FORCE INSTITUTE OF TECHNOLOGY.
- [57] Hardy, R. L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76(8):1905–1915.

- [58] Toit, W. D. (2008). *Radial basis function interpolation*. PhD thesis, University of Stellenbosch.
- [59] (2020). Introduction to RBF's - ALGLIB, C++ and C# library.
- [60] Mandanici, E., Tavasci, L., Corsini, F., and Gandolfi, S. (2019). A multi-image super-resolution algorithm applied to thermal imagery. *Applied Geomatics*, 11(3):215–228.
- [61] Scarmana, G. (2016). An Application of the Least Squares Plane Fitting Interpolation Process to Image Reconstruction and Enhancement. Technical report, University of Southern Queensland.
- [62] Senel, H. G., Peters, R. A., and Dawant, B. (2002). Topological median filters. *IEEE Transactions on Image Processing*, 11(2):89–104.
- [63] Rosenfeld, A. (1979). Fuzzy digital topology. *Information and Control*, 40(1):76–87.
- [64] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.
- [65] Rosenfeld, A. (1984). The fuzzy geometry of image subsets. *Pattern Recognition Letters*, 2(5):311–317.
- [66] Vincent, L., Vincent, L., and Soille, P. (1991). Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598.
- [67] Senel, H. G. (2009). Gradient estimation using wide support operators. *IEEE Transactions on Image Processing*, 18(4):867–878.
- [68] Bobach, T. and Umlauf, G. (2007). Natural Neighbor Concepts in Scattered Data Interpolation and Discrete Function Approximation. In *Visualization of Large and Unstructured Data Sets*, pages 23–35.
- [69] Cazals, F. and Giesen, J. (2004). Delaunay Triangulation Based Surface Reconstruction: Ideas and Algorithms. Technical report, R-5393, INRIA.
- [70] Bobach, T. (2009). *Natural Neighbor Interpolation - Critical Assessment and New Contributions*. PhD thesis, University of Kaiserslautern.
- [71] (2020). Interpolating Scattered Data - MATLAB & Simulink.

# CURRICULUM VITAE

Name-Surname : Barışcan Dedeş  
Foreign Language : English  
Birth Place and Year : Altındağ, Turkey / 1988  
E-mail : bariscand@eskisehir.edu.tr  
ORCID ID : 0000-0001-8089-5307

## **Education**

- January 2021, Anadolu University, Graduate School of Science  
Master of Science in Electrical and Electronics Engineering.
- December 2015, Polytechnic University of Milan, Department of Electronics,  
Information and Bioengineering  
Master of Science in Automation and Control Engineering.
- February 2012, Bilkent University, Faculty of Engineering  
Bachelor of Science in Electrical and Electronics Engineering.

## **Career Experience**

- 2019 - current, Signal Processing Software Engineer, HAVELSAN Inc., Ankara.
- 2016 - 2018, Control Systems Engineer, TUSAŞ Engine Industries Inc., Es-  
kişehir.