Weakly Monotonic Averaging with Application to Image Processing

by

Timothy A. Wilkin
BSc (Hons) GCHE

Submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

Deakin University

May, 2014
DEAKIN UNIVERSITY
ACCESS TO THESIS - A

I am the author of the thesis entitled

Weakly Monotonic Averaging with Application to Image Processing

submitted for the degree of Doctor of Philosophy

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

“`I certify that I am the student named below and that the information provided in this form is correct`”

FULL NAME: Timothy Alan Wilkin

SIGNED: 

DATE: May 9, 2014

Signature Redacted by Library
DEAKIN UNIVERSITY CANDIDATE DECLARATION

I certify the following about the thesis entitled

Weakly Monotonic Averaging with Application to Image Processing

submitted for the degree of Doctor of Philosophy

(1) I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgement is given.

(2) The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

(3) That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

I also certify that any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

"I certify that I am the student named below and that the information provided in this form is correct"

FULL NAME: Timothy Alan Wilkin

SIGNED: ___________________________ DATE: May 9, 2014

Signature Redacted by Library
ACKNOWLEDGEMENTS

Completing a PhD thesis doesn’t happen without the generous support of many people. Having completed this journey in just over a year means that I owe some very big thanks to some very special people. To my beautiful wife Sarah and my daughters Emma and Chloe, you have my undying gratitude for your love and support throughout this journey: without you it would be a bleak and lonely life. To my extended family, thank you all for sticking by me, even when I wasn’t around to hear your encouragement.

I am lucky to work amongst many bright people at Deakin, who inspire and challenge me to achieve more every day. My sincere thanks go to my colleagues Dr. Lei Pan, Dr. Shaun Bangay, Dr. Michael Hobbs and Ms. Sophie McKenzie: thanks for the coffee, the chats and the many useful and rewarding discussions; to Lauren, for the chocolate, the chats and the open office door; and to my research students, Owen M., Daniel, Owen K., Ben and William - I’ve learned as much from you as you have from me.

A special thanks to Prof. Tomasa Calvo, with whom I have had the pleasure and good fortune to work beside and to learn from. To Prof. Humberto Bustince Sola and the GIARA team at Universidad Pública de Navarra in Spain: thank you for a fantastic, enriching and exceptionally productive two months in Pamplona - I had a great time and learned a lot (and even a little Spanish! Muchos gracias). My Associate Supervisors deserve special thanks: Assoc. Prof. Yong Xiang, Dr. Gang Li and in particular, Dr. Simon James (who perhaps loves maths even more than I do), thank you for keeping me on the right path!

Finally, there is always one person in the doctoral journey without whom it would never happen. I was blessed to have a supervisor so talented and knowledgeable, and who works even harder than I do. Very few supervisors would agree to a one year PhD, let alone suggest one and be able to cope with such a hectic work schedule. To Professor Gleb Beliakov, you have my eternal thanks for convincing me to take this on and for so expertly and thoughtfully guiding me to the end - there is no way that I would have achieved this without you and you may take pride in knowing that this happened because of you.
PUBLICATIONS

For the majority of publications listed herein, authors are listed alphabetically, as this was the consensus choice. In such cases the order does not specifically reflect the level of contribution of each author. The following publications resulted from the research presented in this thesis.


Other articles I have recently published:


Weakly Monotonic Averaging
with Application to Image Processing

Timothy A. Wilkin

Abstract

Structured information at pixel scales is essential in images produced by satellites, medical imaging systems and modern digital cameras, which generate images many times the resolution of display devices. Loss of this information degrades the visual quality of images and performance of computer vision algorithms. Traditional methods for image reduction and denoising act as low-pass filters, removing high frequency variations in images, without regard to whether these represent noise or visually relevant detail.

Both reduction and denoising may be formulated as aggregation problems over contiguous subsets of an image and recent application of monotonic averaging functions showed promising results in reducing images corrupted by additive Gaussian noise or impulse noise. However monotonic averages do not sufficiently address the issues of preserving fine detail, or removing non-Gaussian and non-additive noise. Given the prevalence of robust, non-monotonic means within the literature, this thesis proposes their application to image processing problems expressed as local averages, with the view to establishing the properties of functions suitable for image processing tasks.

Within this thesis a new class of averaging functions is established, based on the proposal of weak monotonicity - a restricted form of directional monotonicity. Sufficient conditions for the weak monotonicity of the Gini means, Lehmer means and generalised mixture operators are established. Penalty function methods are applied to demonstrate the weak monotonicity of robust estimators of location,
spatial-tonal filters, mode-like averages and density-based averages. Weak monotonicity is found to be a pervasive property of many means used where robustness to noise and outliers is essential.

A facial recognition task was used to assess reductions computed using weakly monotonic and monotonic averages. Comparative performance against clean, non-reduced images found that image reduction using mode-like averages improves recognition performance, even under significant corruption due to speckle or impulse noise. Other weakly monotonic averages are able to sustain recognition performance, while monotonic averages do not.

A modification of the mode-like average is proposed, using penalty weights constructed from a novel fuzzy measure of cluster compactness. Reduction of synthetic and real images shows that the mode-like averages are able to preserve structured information at pixel scales, while also reducing noise and corruption during image reduction. These results establish weak monotonicity as an important property of robust averages and as a formal framework for the unification of the previously disparate research areas of monotonic aggregation and non-monotonic means.
For Sarah, Emma and Chloe...
Contents

List of Figures 5

Chapter 1. Introduction 7
  1.1. Introduction 9
  1.2. Research Context 10
    1.2.1. Computer Vision 10
    1.2.2. Image Processing 13
    1.2.3. Texture space averaging methods 16
  1.3. Research Motivation 17
    1.3.1. The importance of “location” 18
    1.3.2. Image features as spatial-tonal clusters 19
  1.4. Research Problem 19
    1.4.1. Image processing problems as non-monotonic aggregations 19
    1.4.2. Research question(s) 20
  1.5. Research Outcomes 21
    1.5.1. Contributions to the theoretical study of aggregation functions 21
    1.5.2. Algorithmic contributions to image processing 22
    1.5.3. Applications/Proofs of Concept 22
    1.5.4. Significance of Research Outcomes 23
  1.6. Overview of Thesis 23
    1.6.1. Thesis logic 23
    1.6.2. Thesis layout 24

Chapter 2. Literature Review 27
  2.1. Introduction 29
  2.2. Aggregation Functions 30
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.1.</td>
<td>Notation</td>
<td>30</td>
</tr>
<tr>
<td>2.2.2.</td>
<td>Definitions</td>
<td>31</td>
</tr>
<tr>
<td>2.2.3.</td>
<td>Properties</td>
<td>33</td>
</tr>
<tr>
<td>2.3.</td>
<td>Examples and Main Families</td>
<td>37</td>
</tr>
<tr>
<td>2.3.1.</td>
<td>Means</td>
<td>37</td>
</tr>
<tr>
<td>2.3.2.</td>
<td>Medians</td>
<td>38</td>
</tr>
<tr>
<td>2.3.3.</td>
<td>Weighted Means</td>
<td>40</td>
</tr>
<tr>
<td>2.3.4.</td>
<td>Quasi-Averages</td>
<td>45</td>
</tr>
<tr>
<td>2.3.5.</td>
<td>Disjunctive and Conjunctive Functions</td>
<td>48</td>
</tr>
<tr>
<td>2.4.</td>
<td>Penalty-based Functions</td>
<td>50</td>
</tr>
<tr>
<td>2.4.1.</td>
<td>Penalty-based Averages</td>
<td>51</td>
</tr>
<tr>
<td>2.4.2.</td>
<td>Penalty Functions on Lattices</td>
<td>52</td>
</tr>
<tr>
<td>2.5.</td>
<td>Non-monotone Means</td>
<td>54</td>
</tr>
<tr>
<td>2.5.1.</td>
<td>Robust Estimators of Location</td>
<td>54</td>
</tr>
<tr>
<td>2.5.2.</td>
<td>Bajraktarevic Means</td>
<td>56</td>
</tr>
<tr>
<td>2.5.3.</td>
<td>Weighted Means and Outliers</td>
<td>60</td>
</tr>
<tr>
<td>2.6.</td>
<td>Images and Information</td>
<td>61</td>
</tr>
<tr>
<td>2.6.1.</td>
<td>Raster Images</td>
<td>62</td>
</tr>
<tr>
<td>2.6.2.</td>
<td>Digital Image Noise</td>
<td>64</td>
</tr>
<tr>
<td>2.6.3.</td>
<td>Noise Reduction Methods</td>
<td>66</td>
</tr>
<tr>
<td>2.7.</td>
<td>Averaging Problems in Image Processing</td>
<td>71</td>
</tr>
<tr>
<td>2.8.</td>
<td>Conclusions</td>
<td>73</td>
</tr>
<tr>
<td>3.1.</td>
<td>Introduction</td>
<td>77</td>
</tr>
<tr>
<td>3.2.</td>
<td>Weakly Monotonic Functions</td>
<td>79</td>
</tr>
<tr>
<td>3.2.1.</td>
<td>Definition</td>
<td>79</td>
</tr>
<tr>
<td>3.2.2.</td>
<td>Composition of Weakly Monotonic Functions</td>
<td>80</td>
</tr>
<tr>
<td>3.2.3.</td>
<td>Duality and (\phi)-transform</td>
<td>81</td>
</tr>
<tr>
<td>3.3.</td>
<td>New Results on Weakly Monotonic Means</td>
<td>87</td>
</tr>
<tr>
<td>3.3.1.</td>
<td>Penalty-based Averaging Functions</td>
<td>88</td>
</tr>
</tbody>
</table>
3.3.2. Estimators of Location
3.3.3. The Lehmer Mean
3.3.4. Gini means
3.3.5. Mixture functions
3.3.6. Spatial-Tonal Filters
3.3.7. Generalisations of Spatial-Tonal Filters
3.4. Mode-like Averages
3.5. Density-based Averages
  3.5.1. Density based medians and ML estimators
  3.5.2. Modified weighting functions
  3.5.3. Filtering Outliers
3.6. Conclusions

Chapter 4. Image Reduction as Weakly Monotonic Averaging
  4.1. Introduction
    4.1.1. Motivation for Image Reduction
  4.2. A Weakly Monotonic Image Reduction Operator
    4.2.1. Reduction Operator
  4.3. Block-based Image Reduction
    4.3.1. Image Representation
    4.3.2. Block-based Image Vectorisation
    4.3.3. Block Processing
  4.4. Experiments in Image Reduction: Facial Recognition
    4.4.1. Facial Recognition Problem
    4.4.2. Eigenfaces method for Facial Recognition
    4.4.3. The FERET Evaluation Protocol
    4.4.4. Experiment Design
    4.4.5. Results
    4.4.6. Discussion
  4.5. Conclusions

Chapter 5. Weighting Functions for Aggregation of Pixel Clusters
5.1. Introduction 153
5.2. Geometry-based measures of cluster compactness 154
   5.2.1. Distance-based measures 154
   5.2.2. Distance and Shape Measures 155
   5.2.3. Area-based measures 157
5.3. Structural Patterns in Image Blocks 158
5.4. A fuzzy measure of pixel cluster compactness 160
   5.4.1. A Sugeno-type fuzzy measure 162
   5.4.2. A fuzzy measure based on the Minimum Spanning Tree 166
   5.4.3. Decomposing fuzzy measure 168
5.5. Evaluation 174
   5.5.1. Reduction of Thin Curves 174
   5.5.2. Performance on Thin Curve Reduction Problem 176
   5.5.3. Reduction of Satellite Images 181
5.6. Conclusions 186

Chapter 6. Conclusions and Future Work 191
   6.1. Summary of Key Results 193
   6.2. Future Research 197

References 201
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Satellite imagery of “object 1”.</td>
<td>13</td>
</tr>
<tr>
<td>1.2</td>
<td>Satellite imagery of “object 2”.</td>
<td>13</td>
</tr>
<tr>
<td>1.3</td>
<td>Examples of image noise.</td>
<td>15</td>
</tr>
<tr>
<td>1.4</td>
<td>Local block-based image reduction.</td>
<td>17</td>
</tr>
<tr>
<td>2.1</td>
<td>Bayer pattern on imaging sensor.</td>
<td>63</td>
</tr>
<tr>
<td>2.2</td>
<td>Pictorial representation of a linear filter on a finite neighbourhood.</td>
<td>68</td>
</tr>
<tr>
<td>3.1</td>
<td>Mixture functions formed by various generator functions.</td>
<td>106</td>
</tr>
<tr>
<td>3.2</td>
<td>Contour plots for mixture functions formed by various generator functions.</td>
<td>107</td>
</tr>
<tr>
<td>4.1</td>
<td>Image block averaging operation.</td>
<td>131</td>
</tr>
<tr>
<td>4.2</td>
<td>Local block-based image reduction operation.</td>
<td>132</td>
</tr>
<tr>
<td>4.3</td>
<td>Normalised CMS curves showing noise affect on PCA-based facial recognition performance.</td>
<td>139</td>
</tr>
<tr>
<td>4.4</td>
<td>Normalised average performance vs noise level for reduced images.</td>
<td>140</td>
</tr>
<tr>
<td>4.5</td>
<td>Normalised performance of Mean reduction of images.</td>
<td>141</td>
</tr>
<tr>
<td>4.6</td>
<td>Normalised performance of Median reduction of images.</td>
<td>141</td>
</tr>
<tr>
<td>4.7</td>
<td>Normalised performance of Shorth reduction of images.</td>
<td>142</td>
</tr>
<tr>
<td>4.8</td>
<td>Normalised performance of LMS reduction of images.</td>
<td>142</td>
</tr>
<tr>
<td>4.9</td>
<td>Normalised performance of LTS reduction of images.</td>
<td>143</td>
</tr>
<tr>
<td>4.10</td>
<td>Normalised performance of PMODE reduction of images.</td>
<td>143</td>
</tr>
<tr>
<td>4.11</td>
<td>Normalised average performance vs noise level for reduced eigenfaces.</td>
<td>144</td>
</tr>
<tr>
<td>4.12</td>
<td>Normalised performance of the Mean reduction of Eigenfaces.</td>
<td>144</td>
</tr>
<tr>
<td>4.13</td>
<td>Normalised performance of the Median reduction of Eigenfaces.</td>
<td>145</td>
</tr>
<tr>
<td>4.14</td>
<td>Normalised performance of the LMS reduction of Eigenfaces.</td>
<td>145</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.15</td>
<td>Normalised performance of the LTS reduction of Eigenfaces.</td>
<td>146</td>
</tr>
<tr>
<td>4.16</td>
<td>Normalised performance of PMODE reduction of Eigenfaces.</td>
<td>146</td>
</tr>
<tr>
<td>5.1</td>
<td>Standard deviations in principal axes.</td>
<td>158</td>
</tr>
<tr>
<td>5.2</td>
<td>Several tonal clusters from a $3 \times 3$ block of pixels</td>
<td>159</td>
</tr>
<tr>
<td>5.3</td>
<td>Example clusters in $3 \times 3$ blocks.</td>
<td>160</td>
</tr>
<tr>
<td>5.4</td>
<td>Example of compact and scattered clusters.</td>
<td>161</td>
</tr>
<tr>
<td>5.5</td>
<td>Examples of adding a pixel to a cluster.</td>
<td>161</td>
</tr>
<tr>
<td>5.6</td>
<td>Estimated fuzzy measure using partially sub-additive Sugeno-based method.</td>
<td>164</td>
</tr>
<tr>
<td>5.7</td>
<td>Estimated fuzzy measure using partially super-additive Sugeno-based method.</td>
<td>166</td>
</tr>
<tr>
<td>5.8</td>
<td>Estimated fuzzy measure using Minimum Spanning Tree method.</td>
<td>169</td>
</tr>
<tr>
<td>5.9</td>
<td>Examples of basic components $C_1 - C_4$.</td>
<td>170</td>
</tr>
<tr>
<td>5.10</td>
<td>Decomposition of a subsets with 3 connected elements.</td>
<td>170</td>
</tr>
<tr>
<td>5.11</td>
<td>Fuzzy measure for cluster compactness computed from decomposition method.</td>
<td>172</td>
</tr>
<tr>
<td>5.12</td>
<td>Example of single pixel circle pattern used to test image reduction operators.</td>
<td>175</td>
</tr>
<tr>
<td>5.13</td>
<td>Reduction of circles using cluster-based mode-like averaging function.</td>
<td>177</td>
</tr>
<tr>
<td>5.14</td>
<td>Reduction of circles using comparison averaging functions.</td>
<td>178</td>
</tr>
<tr>
<td>5.15</td>
<td>Fuzzy measure values for MST-based cluster compactness.</td>
<td>179</td>
</tr>
<tr>
<td>5.16</td>
<td>Circle test image corrupted with 10% impulse noise.</td>
<td>179</td>
</tr>
<tr>
<td>5.17</td>
<td>Reductions of circle test image corrupted with 10% impulse noise.</td>
<td>180</td>
</tr>
<tr>
<td>5.18</td>
<td>Satellite image of the Port of Los Angeles.</td>
<td>182</td>
</tr>
<tr>
<td>5.19</td>
<td>Overlapped local block reduction operation.</td>
<td>183</td>
</tr>
<tr>
<td>5.20</td>
<td>Satellite image reduced using monotonic reduction operators.</td>
<td>185</td>
</tr>
<tr>
<td>5.21</td>
<td>Satellite image reduced using Shorth function.</td>
<td>186</td>
</tr>
<tr>
<td>5.22</td>
<td>Satellite image reduced using mode-like functions.</td>
<td>187</td>
</tr>
<tr>
<td>5.23</td>
<td>Difference images for reduced Port of Los Angeles images</td>
<td>188</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction
1.1. Introduction

The aggregation of several input values into a single representative output arises naturally as a problem in many practical applications and domains. There exists a wealth of well-studied problems across fields as diverse as economics, computer science, engineering and mathematics that admit a solution formulated as an aggregation. The research effort into aggregation problems has been disseminated throughout these various fields, with the subsequent mathematical formulation of aggregation having coalesced into a significant body of knowledge concerning aggregation functions. The study of this class of functions, their properties and their application to real problems has enabled a unifying perspective of aggregation problems across these relatively disparate fields, providing new insights and opening up new lines of research.

Recently, aggregation functions have been applied to problems in image processing, such as image scaling (reduction and zooming) and image reconstruction (replacement of missing or corrupt data). Solutions expressed as aggregations over subsets of the image are able to exploit parallelism, are relatively easy to implement and work directly in texture space, simplifying both the mathematical formalism of the problem and the analysis of results. The application of monotonic averaging functions to local block-based image reduction has particularly highlighted the strength of averaging approaches and has sparked considerable interest from the research community.

Unfortunately, most real data sources (particularly images) are corrupted by noise and outlier values, and monotonic averages are known to be sensitive to these corruptions, degrading the quality of the aggregation. However, there are many non-monotonic averages reported in the literature and these have proven to be robust to noise and outliers in other applications. Unfortunately, these functions are not considered aggregation functions by the current definition, which requires monotonicity, and thus have received little consideration by researchers in this field. Naturally, the question arises as to whether non-monotonic averages
are suitable in applications such as image reduction as it is not immediately clear that, in general, they will preserve visually relevant information.

This thesis addresses this over-arching question and investigates non-monotonic averaging functions within the context of the image reduction problem. It also considers the underlying question regarding the theoretical framework of averaging aggregations, which currently omits the non-monotonic functions. Significant novel results are presented showing the effectiveness of selected functions that fall within a newly proposed class of weakly monotonic averaging aggregations. Many of the important non-monotonic averages within the literature are shown to be weakly monotonic under certain conditions and it is shown that weak monotonicity is exactly the property required when averaging data corrupted by noise and outliers.

This chapter presents an introduction to this new research and contextualises the research problems investigated in the body of this thesis. The specific problem of image reduction is explained and its importance justified from two perspectives: that of a pre-processing operator in computer vision applications; and, as a necessity for the high quality display of images on ubiquitous smartphones and tablet devices.

1.2. Research Context

1.2.1. Computer Vision. The field of computer vision covers the capture, processing, analysis and understanding of visual imagery, typically as a component of more complex, intelligent and autonomous systems. The goal of computer vision is often stated as the modeling and understanding of the three dimensional world given one or more two dimensional images. In recent years the field has extended its coverage to include 2.5D (visual image plus depth image) and 3D point cloud data sources. However, continuous advancements in the development of inexpensive, robust and small solid state 2D imaging sensors, as well as small, powerful microprocessors, has led to the ubiquitous deployment of small, cheap cameras capable of high resolution imaging and “onboard” processing. From
smartphones and tablets, to vehicles, entertainment systems and toys, the application areas for machine vision are as diverse as they are challenging.

Notably, computer vision has had widespread success within the digital camera market itself, with most camera models now sold featuring facial detection and auto-focusing features. Some models have capabilities such as auto-shoot coupled to smile detection algorithms. All digital cameras employ digital image processing algorithms to produce “higher quality” images and this is particularly true of the “point-and-shoot” camera models.¹

There are many other applications of computer vision techniques arising from extensive development of imaging systems and technologies. For example, the automated detection and classification of skin cancers and brain lesions in medical imaging, or automatic detection and classification of animals on the African savannas from satellite imagery. Thus, while we now have the ability to formulate 3D images of the world, the more traditional problem of understanding the world using one or more 2D images remains an important research direction in computer science.

There are a variety of reasons as to why computer vision is an important research area, however the most significant is that through computer vision we may automate tasks which were traditionally only possible by humans, bringing quantifiable benefits of reliability, accuracy and speed. For example, some elements of quality assurance in production lines can now be handled by visual inspection and assessment of components and the speed and accuracy with which this can be done far exceeds that of human observers. Computers don’t require holidays, or work breaks and may be considered more reliable and predictable than humans.² As computers are ideally positioned to process massive volumes of digital data, they are also ideal for the automation of search tasks akin to “finding the needle in the haystack”. Within the realm of computer vision, such

¹This is also true of camera systems embedded into smartphones and tablets, which can be considered point-and-shoot models.
²Although obviously maintenance, upkeep and upgrades are now a new business cost.
problems occur when trying to recognise faces in massive crowds, or looking for small objects in satellite images.

An unfortunate but timely example demonstrating the need for automated analysis of visual satellite imagery is the recent tragic loss of Malaysian Airlines flight MH370 on March 8th, 2014, which claimed 239 lives. The aircraft is believed to have crashed into the southern Indian Ocean, 2000km off Australia’s west coast and during the early weeks of search for the crash site, visual observers in manned aircraft and visual analysis of satellite imagery was used in an attempt to locate debris. Several candidate objects were located in satellite images (depicted in Figures 1.1 and 1.2; *images courtesy of Australian Maritime Safety Authority and used with permission*), however none turned out to be debris from the aircraft. The size of debris objects, occurring at pixel scales in images containing billions of pixels, makes human analysis extremely difficult and time consuming. Efforts such as the crowd-sourcing of image analysis were also attempted, but were not evidently effective. As the search zone consists of many thousands of images on a daily basis, each covering areas of of the order of 100 square kilometers (with pixels scales corresponding to roughly one meter resolution), the automatic analysis of this imagery is essential for ensuring all candidate objects are detected and considered by human experts. Such automatic analysis would likely have significantly improved the efficiency and accuracy of the human search effort and may have lead investigators to the likely crash site before black box transponder signals fell silent. Without locating the wreckage, accident investigators are missing vital information necessary to determine the cause of the crash; a necessary step in trying to prevent further losses of life in the future.

The difficulty with conducting an automated analysis of the satellite images in this example is that debris occurs at pixel scale resolutions, meaning that algorithmic analysis is particularly sensitive to the two fundamental problems that underpin computer vision methods that operate in texture space. These problems are the *curse of dimensionality* and *corruption due to noise* and are central problems tackled within the sub-field of image processing.
1.2. RESEARCH CONTEXT

1.2.2. Image Processing. The algorithmic complexity of computer vision algorithms scales exponentially with the number of pixels in the image. While we
do not necessarily require real time processing of satellite or other images when *post hoc* analysis is being conducted, clearly this complexity will limit the volume of data that can be analysed in any given time period. Dimensionality reduction is usually performed in order to improve computational efficiency and one of the principle methods employed is image reduction. The source image is resampled at a lower resolution, with the objective that the same visual information is contained within the reduced image as was present in the source. In so doing, while the number of pixels used to represent the scene is reduced, the information relevant to the vision task should ideally be preserved. Under general conditions this will be true only if the sampling rate in the reduced image is at least twice the highest signal frequency within the source data. This is not the case in general, or under circumstances where variations in the source image occur at pixel scales.

For images with fine details at pixel scales relevant information will be lost by resampling at a lower rate. For computer vision tasks where there is sufficient information at lower frequencies to perform the analysis, this loss of information is usually acceptable. However, in contexts where the important information resides within these higher frequencies or where the loss of visual fidelity is undesirable, such losses must be prevented.

Another issue arising from resampling is that of spatial aliasing, whereby the sampling rate generates artificial frequencies within the data, causing visually apparent corruptions of the signal. Anti-aliasing during image reduction is commonly handled by first filtering the image using a low-pass filter, to remove frequencies higher than twice the new sampling rate, then resampling the image using an interpolation method (for example, bicubic interpolation), which computes a weighted average of pixel intensities in the neighbourhood of the new sample point.

The result of reduction using these approaches is an image missing high frequency information. When viewing an image reduced using these methods, at the same display resolution as the original, the loss of high frequency information diminishes the clarity of the reduced image, producing a blurring effect. The visual quality of the image is thus reduced. While this reduction in visual quality
1.2. RESEARCH CONTEXT

is not necessarily an issue in computer vision applications, where the information content is more relevant, the visual quality of reduced images is highly important when reduction is performed for the purposes of displaying images on screens with lower pixel resolution than the source image. This issue has become quite significant in recent years with the rapid growth in use of smartphones for capturing digital images.

Pixel counts in smartphone cameras are commonly around 12-16 megapixels (and the latest camera sensors are around 70 megapixels), however displays are typically constrained to approximately 2 megapixels. This represents a scenario where only a small fraction of the source pixels can be displayed at any one time. Clearly significant image reduction is required in order to display the entire image on the smartphone display, which, using current methods, significantly degrades the image quality and thus the perceived quality of the device for digital imaging.

The second problem commonly tackled within image processing is that of reducing the data corruption due to noise. Algorithmic approaches to noise reduction commonly assume that the captured image may be modeled by an additive random field comprised of discrete i.i.d random variables. Typically the noise variables are assumed to be normally distributed with zero mean, although other types of noise are also considered. Figure 1.3 depicts four different noise types commonly considered in image denoising problems. These are: impulse noise (also known as “salt & pepper” noise); Gaussian noise; speckle noise; and, Poisson noise.
The assumption of zero mean, additive noise admits a simple model of the relationship between the true image and captured image and permits a range of solutions based on filtering the image to remove high frequency intensity variations. Low pass filters may be implemented in a transform domain as a multiplication of a filter transfer function and the image transfer function, although for discrete signals, such as images, they may also be computed in texture space using a weighted average over a window centered on each pixel. From this perspective both image denoising problems and image reduction problems can be cast as problems of computing an average pixel intensity over a local subset of an image.

1.2.3. Texture space averaging methods. Visual information in images exists in the spatial and tonal distribution of values that represent the momentary light intensity field detected by the imaging sensor. At their heart, the aforementioned methods for image denoising and image reduction compute averages at specific points in this field, using the discrete sample points\(^3\) of the source image in this computation. The principle goal during these operations is, respectively, to decrease the signal variance and produce an image at a lower resolution, without degrading the image quality. Under assumptions of independent additive Gaussian noise, techniques based on the Weiner filter ensure an optimal maximum likelihood estimate of the true signal, in the sense of minimising the mean squared error between the true image and the estimated image. Unfortunately real images do not conform to ideal noise models and thus other solution methods for computing these averages should be considered.

Weiner deconvolution is equivalent to computing an arithmetic mean over local neighbourhoods within the image and thus suggests an alternative research direction for filtering and reduction, from the mathematical perspective of averaging functions. A recent series of articles formulated the image reduction problem as a local, block-based averaging problem [59] [58] [9], whereby the average value of each disjoint block of pixels in the source image is used as a representative value.

\(^3\)Discrete in both tonal and spatial values.
1.3. Research Motivation

This line of research is in its infancy, although it already shows promising results that invite deeper investigation. In particular, the work of Paternain and colleagues has focused on the application of monotonic averaging functions, such as the arithmetic mean and the median, to image reduction. These two well-known functions have been previously applied within denoising applications, and within the context of image reduction, were shown to provide good quality

of that block in the formulation of the reduced image (see Figure 1.4). The appeal of this approach is twofold: firstly, that the reduction problem can be solved easily within the texture space and in parallel, providing significant performance improvements over transform-based methods; and, secondly, by formulating the problem as a mathematical average, knowledge from the field of aggregation functions can be applied, both to obtain new averaging-based operators and to better understand the performance of reduction operators under realistic noise assumptions. By linking the image reduction task to the formal framework of aggregation functions and averaging aggregation in particular, it has been possible to define image reduction operators in an axiomatic manner [60], as well as define operators for colour image reduction via averaging operators on lattices [9] (where previously reduction had been done individually in each channel). This recent work has opened new avenues of research into the preservation of information and reduction of noise during reduction operations, which has not previously been considered in the filtering-interpolation approaches.
reduced images even in the presence of Gaussian noise and missing data. However, monotonic averages are known to be sensitive to outliers within the input data and thus are not robust under broader noise or data corruption assumptions.

1.3.1. The importance of "location". Denoising filters based on the arithmetic mean and the median, while having been applied in image denoising tasks and offering optimal performance under simple noise and signal models, have poor performance in general. Specifically, the arithmetic mean does not account for the 'location' of data values relative to other values and is thus sensitive to even a single significant outlier value. It also assumes that the input values are independent samples from the underlying distribution, which will not be the case in structured image data. The median on the other hand is robust to outliers, however in applications where there is significant signal variation across neighbouring pixels, the median may not effectively reduce this variation during reduction or denoising operations.

Within the signal processing field a variety of methods akin to weighted arithmetic means have been applied in denoising applications, with a consistent focus on the relevance of the location of a pixel within the local neighbourhood. In particular, the Gabor filter, which computes a weighted arithmetic mean in tonal space, uses a Gaussian kernel function as the basis for weighting individual pixel contributions within the tonal average. As with other filtering methods, this operator acts as a low pass filter on the data and removes high frequency information, causing blurring and aliasing along sharp intensity boundaries within the image.

More recently, the class of spatial-tonal filters have been applied to good effect in image denoising tasks, showing good capacity to preserve sharp edges in images and reduce aliasing affects. These filters apply weights based on both tonal separation and geometric separation, and therefore account for pairwise dissimilarity between pixels. While these filters compute a solution as a non-monotonic average of the input data, they are not typically formulated as averaging functions and not studied from that perspective. Indeed, because of their non-monotonicity,
they are not considered aggregation functions. It is clear though that averaging methods that account for pixel dissimilarity offer better performance than those that compute averages over intensity values alone and this warrants an investigation into non-monotonic averaging functions for image processing tasks such as image reduction and denoising.

1.3.2. Image features as spatial-tonal clusters. As described previously, a particular interest in this current research program is the preservation of small details and pixel scale features during image reduction operations, and the ability to distinguish structured detail from noise. This issue is particularly challenging, as there is no clear criterion as to what constitutes noise or fine detail, without prior knowledge of the image visual content. Intuitively, noise behaves randomly, whereas image features have tonal and geometric consistency. This vague notion, and the perspective of image reduction and denoising as an averaging operation, prompts a deeper investigation into suitable non-monotonic averaging functions that preserve structured detail, while omitting random noise from output images.

Further to this line of reasoning, it seems reasonable to posit that image features are represented locally by structured spatial-tonal clusters and that with regard to approaches to robust statistics, image noise and missing data can be viewed as outliers of these clusters. In particular, a local block of pixels within an image may be assumed to contain a principal cluster of values (in both spatial and tonal coordinates) representing an image feature and possibly a number of outlier values representing noise or corruption. In performing image reduction or denoising, the objective then is to identify this cluster and a representative value that may be used to either represent the local block in a reduction operation, or to replace a specific pixel in a denoising operation.

1.4. Research Problem

1.4.1. Image processing problems as non-monotonic aggregations. Methods for image reduction based on monotonic averages over pixel intensities cannot account for the structured nature of image data, and outliers from a local
pixel cluster are either rejected completely (in the case of the median function) or distort the average value (in the case of the arithmetic mean). In considering image reduction as an aggregation problem over local image blocks, an analysis of the requirements of reduction operator presents several key properties of the required aggregation operators:

(1) The aggregation must be averaging, so that the output pixel intensity lies within the bounds of the inputs pixel values within a given block (so as not to increase signal variance);

(2) The aggregation must be idempotent, such that constant intensity regions are represented by a single pixel of the same intensity; and,

(3) The aggregation should not be monotonic in all arguments, so as to provide the capacity to reject or downplay pixels that are outliers of the principal cluster in a local image block.

It is noted that this last requirement specifically excludes all existing aggregation functions as the formal definition, presented in Section 2.2, requires monotonicity in all arguments. However, there are many non-monotonic averaging functions described in the literature that are appropriate to single cluster aggregations, that have also been shown to be robust to outliers and noise. Subsequently, an investigation of these functions and the properties that lead to robustness is warranted, as is an evaluation of their suitability for image processing tasks expressed as averages over local image blocks. Additionally, extensions to current theoretical frameworks should be investigated so as to create a unified perspective of averaging aggregation functions that unites the previously disparate research fields of monotonic and non-monotonic averaging.

1.4.2. Research question(s). The issues discussed above lead to the following research objective:

- *Preserve structured information at pixel scales while reducing random noise in image processing tasks*

This will be considered from the perspective of algorithmic methods that are able to effectively discriminate between noise and structured visual information and
this thesis approaches this problem through the design of averaging functions that are robust in the presence of noise and outliers. Given this approach, the following specific questions are formulated to guide the research.

1. What general property or properties make non-monotonic averaging functions robust and effective for averaging problems in image processing?
2. Are mode-like averages suitable for representing the principle cluster within a local image block and do such averages preserve information content during image reduction operations?
3. Can fine image details, that would otherwise be filtered out by a monotonic averaging procedure, be preserved using non-monotonic averages?
4. How do we account for the geometric arrangement of pixels in the design of averaging functions suitable for image processing tasks?

1.5. Research Outcomes

Through answering the aforementioned research questions this thesis produces novel results in both the theoretical frameworks of averaging aggregation, as well as the application of averaging methods in practical image processing problems. Several significant contributions are summarised herein. For specific details, refer to the relevant chapters indicated in Section 1.6.

1.5.1. Contributions to the theoretical study of aggregation functions.

- This thesis proposes a definition for weakly monotonic functions and identifies a new class of averaging aggregation functions based on the relaxation of the monotonicity constraint.
1. INTRODUCTION

• Sufficient conditions and proofs for the weak monotonicity of several important classes of non-monotonic means are given. Notably, the Gini means, Lehmer means and mixture functions derived from the Means of Bajraktarevic, as well as the robust estimators of location, mode-like averages and density-based averages.

• Conditions for the preservation of weak monotonicity under transformation are studied and several key results are obtained regarding linear and nonlinear $\phi -$ transforms of weakly monotonic averaging functions, as well as for compositions of monotonic and weakly monotonic functions.

• Penalty-based methods and weak monotonicity are applied to propose generalisations of both the recently proposed density-based averages and the spatial tonal filters.

1.5.2. Algorithmic contributions to image processing.

• A weakly monotonic mode-like estimator is proposed and validated as a suitable averaging function for local block-based image reduction operations.

• A variation of the local reduction operation to use overlapping image blocks is proposed and validated for its ability to preserve fine scale image structure.

• An approach to formulating penalty weights based on novel fuzzy measures of cluster compactness is proposed and several methods for the construction of these measures are presented.

• A modification of the mode-like estimator to use weights based on cluster compactness is proposed and validated on real and synthetic images.

1.5.3. Applications/Proofs of Concept. This thesis presents proof of concept for:

• image reduction operators based on non-monotonic averages computed over local image blocks.
• a method for dealing with scale differences between images and models, for feature space methods of visual object recognition, proposing and validating a reduction of the feature space model.

1.5.4. Significance of Research Outcomes. This thesis proposes the concept of weak monotonicity in the context of averaging aggregation functions. Many averages used widely in practical applications are excluded from the field of aggregation functions because they lack monotonicity, yet it is precisely the lack of monotonicity which makes them robust when aggregating noisy values. The density-based means and robust estimators of location naturally fall into this new class of averages.

The analysis and practical illustrations performed in this thesis allow one to look at the class of weakly monotonic means as a unifying perspective of monotonic averages and many useful non-monotonic averages, joining these two previously disparate fields. Weak monotonicity provides a new view of averaging aggregation and invites further research into the extent that theorems related to monotonic averaging aggregations can be extended to cover weakly monotonic averaging aggregations.

The proofs of concept presented establishes the usefulness of weakly monotonic means in practical applications in image processing problems expressed as averages, extending current state of the art results relating to monotonic averages applied as local, block-based image reduction operators.

1.6. Overview of Thesis

1.6.1. Thesis logic. This thesis presents weakly monotonic averaging functions, a new classification of mathematical functions that encompasses all of the monotonic averaging aggregation functions, as well as many important non-monotonic means used extensively in applications where data are corrupted by noise and outliers. The properties of this class of functions are explored and results related to composition and transformation of weakly monotonic functions are presented. Proofs of the weak monotonicity of important classes of
non-monotonic means, such as the Gini means, mixture functions and robust estimators of location, are provided.

Given this theoretical framework, a novel weakly monotonic averaging function is proposed for application in image reduction tasks. This novel averaging function is evaluated in a context of image reduction as pre-processing in a computer vision task, where reduction significantly reduces computational complexity while also improving recognition accuracy in probe images corrupted by significant levels of speckle or impulse noise. The ability of various averaging functions to preserve information relevant to facial recognition is evaluated, showing the superior performance of weakly monotonic functions and specifically the proposed mode-like average.

A modification of this function is proposed in order to preserve fine visual details within images; a direct attempt to overcome the properties of the monotonic averages that remove this high frequency information during reduction. This modification applies fuzzy measure theory to formulate novel cluster compactness measures that take spatial organisation of pixels into account when selecting the significant cluster within a local image block. This variant of the mode-like average is tested against the previous version, as well as other monotonic and weakly monotonic functions. Both versions of the mode-like average are shown to preserve fine image details in real satellite images, where the other functions fail to do so.

These results show that weakly monotonic averages are useful as local block-based image reduction operators and confirm that weak monotonicity is an important property of averaging functions used in practical applications where data is corrupted by noise and outliers.

1.6.2. Thesis layout. Chapter 3 presents the theoretical contributions of this thesis, including the definition for weak monotonicity. Properties of weakly monotonic aggregations are investigated, including preservation of weak monotonicity under composition of aggregations, and linear and non-linear transformations. Sufficient conditions for the weak monotonicity of several important
classes of means are presented with proofs, and several robust estimators commonly applied in statistics are proven to be weakly monotonic functions. Classes of functions such as the spatial-tonal filters is investigated and density-based means are also proven to be weakly monotonic and further generalisations of these classes, based on the weak monotonicity property, are proposed.

In Chapter 4 a weakly monotonic mode-like averaging function is proposed, using penalty function methods. This function, using distance-based penalty weights, is evaluated for performance on image reduction as a pre-processing operation in a computer vision task, to determine its robustness to noise and outliers, as well as its capacity to preserve relevant information within the image necessary for the automated recognition task. It’s performance is compared to that of monotonic averages and several robust estimators of location, which, like the mode, are measures of the central tendency of the input data.

In Chapter 5 a modification to the mode-like average proposed in Chapter 4 is considered, that seeks to preserve fine image details that exist at pixel scales. This approach applies fuzzy measure theory to the design of a cluster compactness measure, which is used to determine the weights within the penalty formulation of the mode-like averaging function. This modified function is evaluated on synthetic and real test images against those functions considered in Chapter 4.

Finally, Chapter 6 presents the conclusions drawn from these research and considers the implications of the major results. Directions for future research building upon the outcomes of this thesis are proposed and briefly discussed.
CHAPTER 2

Literature Review
2.1. Introduction

The principal theorems of aggregation as a mathematical operation have been well-established within existing literature for many years. However, the outstanding question in many application areas is which aggregation function works best? A wide range of aggregation functions are presented in the literature, including the weighted quasi-arithmetic means, ordered weighted averages, triangular norms and co-norms, Choquet and Sugeno integrals and many more. Several recent books provide a comprehensive overview of this field of study ([10, 34, 86]).

This chapter presents the key mathematical foundations for aggregation, with a view to establishing the necessary properties of averaging aggregation functions. The subsequent exposition of penalty function methods and non-monotonic averaging functions builds upon this background to provide a foundation for the theoretical investigations and contributions of this thesis. These sections can be considered supporting material for those readers not familiar with aggregation and specifically with averaging functions.

In the latter portion of this chapter the focus turns to image processing and specifically the formulation of certain image processing tasks as averaging problems. A review of the current state of the art for representing and solving image denoising and image scaling problems is presented and leads to a unified view of robust averaging in the presence of noise.

Within this chapter, where appropriate, deference to existing literature has been shown with regards to notation; in particular favouring [10]. Where deviation from these existing standards has been exhibited and the notation is not obvious, a brief explanation will be provided. For those readers interested in the broader theoretical foundations of aggregation functions, the texts by Beliakov et al [10] and Grabisch et al [34] provide a consistent and thorough exposition.

The remainder of this chapter is structured as follows. Section 2.2 provides the necessary mathematical foundations that underpin aggregation functions, while section 2.3 presents the main families of aggregation functions and highlights some specific examples. Section 2.4 presents penalty based functions, which provide a
general framework for formulating both monotonic and non-monotonic averaging functions. Section 2.5 presents several important classes of means that are either non-monotonic, or not generally monotonic, for which new results on their monotonicity are provided in Chapter 3. Section 2.6 shifts the focus to the application domain of this thesis: digital images and image processing problems that admit solutions formulated as averages.

2.2. Aggregation Functions

2.2.1. Notation. This thesis is concerned with the problem of aggregating a finite set of input variables, \( X \doteq \{ X_{i,n} \mid n \in \mathbb{N}, i \in \{1, \ldots, n\} \} \), to produce a scalar aggregate, \( Y \), which summarises the inputs using an appropriate numeric value. The variables \( X_{i,n} \) are typically the attributes that characterise the input space, \( X \), or alternatively they may represent the same quantity from \( n \) unique sources. Within this thesis, where the application domain is image processing, the input variables denote the properties of pixels within an image. The specific semantic meaning of the relationship between the input vector \( X \) and output variable \( Y \) is domain and problem dependent. This will be explained further in Section 2.7.

Without loss of generality it is assumed that the domain of interest is any closed, non-empty interval of the extended real line, \( \mathbb{I} = [a, b] \subseteq \bar{\mathbb{R}} = [-\infty, \infty] \) and that each input variable may be transformed (translated and scaled) appropriately so that \( X' = \mathbb{I}^n \). Furthermore, for brevity, variables \( X_{i,n} \) will be written henceforth as \( X_i \), with \( i \in \{1, \ldots, n\} \), unless otherwise stated.

Tuples in \( \mathbb{I}^n \) are denoted by \( x = (x_i \mid i \in \{1, \ldots, n\}) \) and indicate the assignment of a specific value to the set of input variables \( X = x \), which is taken to mean \( (X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) \). \( \mathbb{I}^n \) is ordered such that for \( x, y \in \mathbb{I}^n \), \( x \leq y \) implies that each component of \( x \) is no greater than the corresponding component of \( y \), \( x_i \leq y_i \forall i \in \{1, \ldots, n\} \). Constant vectors are denoted by \( c = (c_1, \ldots, c_n) \) and particular use will be made of the constant vectors in the direction \( 1 = (1, 1, \ldots, 1) \), which are given by \( a = a \underbrace{(1, 1, \ldots, 1)}_{n \text{ times}} = a1 \), where \( a \in \mathbb{R} \) is a constant and \( n \) is implicit within the context of use.
2.2. AGGREGATION FUNCTIONS

The vector $\mathbf{x}^\uparrow$ denotes the result of permuting the vector $\mathbf{x}$ such that its components are in increasing order, that is, $\mathbf{x}^\uparrow = \mathbf{x}_\sigma$, where $\sigma$ is any permutation such that $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots \leq x_{\sigma(n)}$. Similarly, the vector $\mathbf{x}^\downarrow$ denotes the result of permuting $\mathbf{x}$ such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \ldots \geq x_{\sigma(n)}$. The common shorthand notation for a sorted vector, being $\mathbf{x}(\mathbf{i}) = (x(1), x(2), \ldots, x(n))$, will be used frequently. In such cases the ordering will be stated explicitly and then $x(k)$ represents the $k$-th largest or smallest element of $\mathbf{x}$ accordingly.

2.2.2. Definitions. The aggregate $Y$ is a scalar-valued quantity taking values $y \in \mathcal{Y}$ and aggregation represents the synthesis of $n > 1$ input values\(^1\) to produce a value that summarises, or represents these inputs. Given this notion of $y$ being a representative value, it is reasonable to expect that the output should fall within equivalent or appropriate bounds given the domain, $\mathcal{X} = \mathbb{I}^n$. Consequently the co-domain is assumed to be $\mathcal{Y} = \mathbb{I}$, expressing the most fundamental property of aggregation functions, that of bounds preservation.

**Definition 1.** A function $F : \mathbb{I}^n \to \mathbb{I}$ is **bounds preserving** if for $\mathbb{I} = [a, b]$ then $F(a) = a$ and $F(b) = b$.

The second fundamental property of aggregation functions is **monotonicity** with respect to all arguments, which says loosely that an increase in the value of one or more input values should be reflected by an increase in the output (or that the output should at least remain constant).

**Definition 2.** A function $F : \mathbb{I}^n \to \mathbb{R}$ is **monotonic** (increasing) if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{I}^n, \mathbf{x} \leq \mathbf{y}$ then $F(\mathbf{x}) \leq F(\mathbf{y})$.

A further restriction of monotonicity are those functions that are strictly increasing. I.e, they satisfy:

**Definition 3.** A **strictly monotonic** (increasing) function is one for which $\mathbf{x} \leq \mathbf{y}$ but $\mathbf{x} \neq \mathbf{y}$ implies $f(\mathbf{x}) < f(\mathbf{y})$ for every $\mathbf{x}, \mathbf{y} \in [a, b]^n$.

\(^1\)Herein we do not consider the special case of $n = 1$, where $f(x) = x$. 

The condition for monotonicity in Definition 2 is equivalent to the requirement that every univariate function $f_x(t) = F(x)$ with $x_i = t$ and the remaining components of $x$ being fixed, $f_x(t)$ is monotonic increasing in $t$, for all $t \in \mathbb{I}$.

**Remark 1.** Taking $e_i$ as a unit vector aligned to coordinate $x_i$, then if $F$ is directionally differentiable in its domain, monotonicity is equivalent to the condition that the directional derivative $D_{e_i}(F)(x) = \nabla F(x) \cdot e_i \geq 0$ at each point $x \in \mathbb{I}^n$, for all $i \in \{1, 2, ..., n\}$.

Given these definitions, the formal definition of an *aggregation function* is thus:

**Definition 4.** A function $F : \mathbb{I}^n \to \mathbb{I}$ is an *aggregation function* in $\mathbb{I}^n$ if and only if $F$ is monotonic increasing in $\mathbb{I}$ and $F(a) = a$, $F(b) = b$, with $\mathbb{I} = [a, b] \subseteq \mathbb{R}$.

This seemingly simple definition admits a very large family of functions, examples of which will be explored further in Section 2.3.

There are four principle classes of aggregation functions (see, for example, \[18\]): *averaging*, *conjunctive*, *disjunctive* and *mixed*, which are classified using the following definitions.

**Definition 5.** A function $F : \mathbb{I}^n \to \mathbb{R}$ has *averaging behaviour* (or is *averaging*) if for every $x \in \mathbb{I}^n$ it is bounded by

$$\min(x) \leq F(x) \leq \max(x).$$

In some texts (e.g. \[34\]) this behaviour is referred to as *internality*, however this term is reserved for labelling a specific restriction of aggregation behaviour (see Definition 11).

**Definition 6.** A function $F : \mathbb{I}^n \to \mathbb{R}$ has *conjunctive behaviour* (or is *conjunctive*) if for every $x \in \mathbb{I}^n$ it is bounded by

$$a \leq F(x) \leq \min(x), \text{ where } \mathbb{I} = [a, b].$$
Conjunctive functions include the classical example of logical AND.

**Definition 7.** A function $F : \mathbb{I}^n \rightarrow \mathbb{R}$ has **disjunctive behaviour** (or is disjunctive) if for every $x \in \mathbb{I}^n$ it is bounded by
\[
\max(x) \leq F(x) \leq b,
\]
where $\mathbb{I} = [a, b]$.

Disjunctive functions include the classical example of logical OR and may be formulated as the duals of the conjunctive functions (see Definition 21 and Subsection 2.3.5).

**Definition 8.** Mixed aggregation. An aggregation function $f$ is mixed if it does not belong to any of the above classes, i.e., it exhibits different types of behaviour on different parts of the domain.

The uninorms and nullnorms (Definitions 54 and 55 respectively) are examples of mixed aggregations.

General functions - and by extension, aggregation functions - can exhibit some or all of the following properties.

### 2.2.3. Properties

An idempotent element with respect to a binary operation $*$ is any element $t$ for which $t * t = t$. For example, with respect to multiplication, $\times$, the element 1 is idempotent, since $1 \times 1 = 1$. With respect to aggregation of a singleton all inputs are idempotent elements: i.e., $F(t) = t$. This property can be extended to $n$-ary functions using the diagonal section.

**Definition 9.** The **diagonal section** of any function $F : \mathbb{I}^n \rightarrow \mathbb{R}$ is the unary function $\delta_F : \mathbb{I} \rightarrow \mathbb{I}$ defined as $\delta_F(x) = F(x, \ldots, x)$ for all $x \in \mathbb{I}$.

Subsequently, an idempotent function is defined as:

**Definition 10.** A function $F : \mathbb{I}^n \rightarrow \mathbb{R}$ is called **idempotent** if for every input $x = (t, t, \ldots, t), t \in \mathbb{I}$ the output is $F(x) = \delta_F(t) = t$.

It follows from their definition that idempotent aggregation functions have averaging behaviour and that monotonic averaging functions are idempotent. See [33] for a proof.
Definition 11. A function is called **internal** if its value coincides with one of its arguments.

The above definition is due to Cauchy [19].

Definition 12. A function \( F : \mathbb{I}^n \to \bar{\mathbb{R}} \) has a **neutral element** \( e \in \mathbb{I} \) if for every \( t \in \mathbb{I} \) in any position it holds that \( F(e, \ldots, e, t, e, \ldots, e) = t \).

Definition 13. A function \( F : \mathbb{I}^n \to \bar{\mathbb{R}} \) has an **absorbing element** (annihilator) \( e \in \mathbb{I} \) if for any \( x_i = e \) in any position it holds that \( F(x_1, \ldots, x_{i-1}, e, x_{i+1}, \ldots, x_n) = e \), for all \( x \in \mathbb{I}^n \).

Of particular relevance to this thesis is the notion of shift-invariance: a constant change in every input should result in a corresponding change of the output.

Definition 14. A function \( F : \mathbb{I}^n \to \bar{\mathbb{R}} \) is **shift-invariant** (stable for translations) if \( F(x + a1) = F(x) + a \) for all \( a \in \mathbb{R} \), whenever \( x, x + a1 \in \mathbb{I}^n \).

Technically the definition of shift-invariance (which is also called difference scale invariance [34]) expresses stability of aggregation functions with respect to translations, rather than invariance. However, as the term shift-invariance is much in use (e.g., [18] [42]) it is adopted throughout this thesis.

Definition 15. A function \( F : \mathbb{I}^n \to \bar{\mathbb{R}} \) is **homogeneous** (with degree one) if \( F(ax) = aF(x) \) for all \( ax \in \mathbb{I}^n \) and \( aF(x) \in \bar{\mathbb{R}} \).

Aggregation functions that are shift-invariant and homogeneous are known as **linear aggregation functions**. Due to bounds preservation, idempotence follows from an aggregation function being shift-invariant, homogenous or both (i.e., linear). The canonical example of a linear aggregation function is the arithmetic mean.

Definition 16. A function \( F : \mathbb{I}^n \to \bar{\mathbb{R}} \) is **symmetric** if for all \( x \in \mathbb{I}^n \), its value does not depend on the order of its arguments. I.e.,

\[
F(x_1, x_2, \ldots, x_n) = F(x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)})
\]
for any permutation $\sigma = (\sigma(1), \sigma(2), ..., \sigma(n))$ of $(1, ..., n)$.

The following definition permits the construction of general $n$-ary aggregation functions from a single bivariate function.

**Definition 17.** A two-argument function $f$ is **associative** if $f(f(x_1, x_2), x_3) = f(x_1, f(x_2, x_3))$ holds for all $x_1, x_2, x_3$ in its domain.

Consequently $n$-ary functions are constructed by iteratively applying the bivariate case $f_2$ such that

$$f_n(x_1, x_2, ..., x_n) = f_2(f_2(...f_2(x_1, x_2), x_3), ..., x_n).$$

This thesis is principally concerned with continuous aggregation functions, in the sense that a small change in the inputs should result in (no more than) a small change in the outputs. Formally, this can be expressed by functions that have Lipschitz continuity.

**Definition 18.** An aggregation function $f$ is called **Lipschitz continuous** if there is a positive number $M$, such that for any two vectors $x, y$ in the domain of definition of $f$:

$$|f(x) - f(y)| \leq M d(x, y)$$

where $d(x, y)$ is a distance between $x$ and $y$. The smallest such number $M$ is called the Lipschitz constant of $f$ (in the distance $d$).

For a Lipschitz continuous function, a change in the input by some small amount $\delta = \|x - y\|$ means that the output will change by at most $M\delta$. Thus $M$ can be considered an upper bound on the rate of change of the function. Lipschitz continuity is important in this thesis wherein we consider small perturbations $\delta = (\delta_1, \delta_2, ..., \delta_n)$ on inputs, which denote noise and it is reasonably expected that the aggregated output, $f(x + \tilde{\delta})$, should not be substantially different from $f(x)$.

The concept of duality is important to the formulation of the disjunctive aggregation functions and is built on the definition of negation. The standard
negation on the interval $\mathbb{I} = [a, b]$ is given $N_1(t) = a + b - t$, which is a strong negation according to:

**Definition 19.** A univariate function $N$ defined on $\mathbb{I} = [a, b]$ is a **strong negation** if it is strictly decreasing and involutive. I.e., $N(N(t)) = t$ for all $t \in \mathbb{I}$.

The negation $N(t) = 1 - t^2$ is not a strong negation on $[0, 1]$, although it is strict.

**Definition 20.** A univariate function $N$ defined on $\mathbb{I} = [a, b]$ is a **strict negation** if its range is also $[a, b]$ and it is strictly monotonic decreasing.

The dual of a function is thus defined in terms of negation:

**Definition 21.** The **dual** $F^d_N$ (with respect to $N$) of an aggregation function $F : \mathbb{I}^n \to \overline{\mathbb{R}}$ is the function

$$F^d_N(x_1, x_2, ..., x_n) = N(F(N(x_1), N(x_2), ..., N(x_n)))$$

such that $N$ is a strong negation on $\mathbb{I}$.

When $N$ is the standard negation on $\mathbb{I} = [a, b]$ then $F^d = (a + b) - F((b + a - x_1), (b + a - x_2), ..., (b + a - x_n))$ and the missing subscript $N$ on $F^d$ implies the dual with respect to standard negation.

**Definition 22.** Given a strong negation, $N$, defined on $\mathbb{I}$, a **self-dual aggregation** function (with respect to $N$) is a function $F : \mathbb{I}^n \to \overline{\mathbb{R}}$ such that function

$$F(x) = N(F(N(x_1), N(x_2), ..., N(x_n)))$$

A final definition of specific interest is that of the $\phi$−transform.

**Definition 23.** Let $\phi : \mathbb{I} \to \mathbb{I}$ be a bijection. The **$\phi$−transform** of a function $F(x)$ is the function $F_\phi(x) = \phi^{-1}(F(\phi(x_1), \phi(x_2), ..., \phi(x_n)))$.

The negations are an important example of $\phi$−transforms.
2.3. Examples and Main Families

There are many examples of aggregation functions satisfying the definitions given above and herein several well known examples are presented, along with definitions of major families of aggregation functions.

2.3.1. Means. The arithmetic mean has historical significance as being, perhaps, the most widely known and frequently used aggregation function, particularly where the output is required to display averaging behaviour. Indeed, colloquially, when asked to compute the “mean” or “average” of a set of numbers, most people will compute the arithmetic mean. Within the mathematics literature the term mean is used synonymously with aggregation functions having averaging behaviour, however the relationship is only firmly established for monotonic functions. In Chapter 3 this relationship will be established for the broader class of weakly monotonic averaging functions.

Chisini’s definition of a mean as an average states that the mean of \( n \) independent variables \((x_1, ..., x_n)\), with respect to a function \( F \), is a value \( M \) for which replacement of each value \( x_i \) in the input by \( M \), results in the output being \( M \) (\cite{22}, stated in \cite{34}). I.e.,

\[
F(x_1, ..., x_n) = F(M, ..., M) = M.
\]

As was noted by de Finetti (\cite{26}, stated in \cite{34}), Chisini’s definition does not necessarily satisfy Cauchy’s requirement that a mean have averaging behaviour (internality) \cite{19}. However, by assuming that \( F \) is an increasing, idempotent function, then existence, uniqueness and internality (averaging behaviour) of \( M \) are restored to Chisini’s definition. The requirement that \( F \) be increasing is too strict given the aims of this thesis and as such, following many authors (e.g., Bullen \cite{15}), a mean herein is taken to be any function that is averaging (and hence idempotent).

**Definition 24.** A function \( M : \mathbb{I}^n \rightarrow \mathbb{I} \) is called a **mean** if it is averaging. I.e., \( \min(x) \leq M(x) \leq \max(x), \forall x \in \mathbb{I}^n \).
The arithmetic mean is thus defined as:

**Definition 25.** The *arithmetic mean* is the function \( A(x) = \frac{1}{n} \sum_{i=1}^{n} x_i \).

The arithmetic mean is homogeneous, shift-invariant, idempotent and averaging. As will be seen later, the arithmetic mean also corresponds to the output generated by minimising the sum of squared differences between the input values and the output, which is a common approach to solving regression and other statistical estimation problems. Two additional and simple means are the projection and the order statistic.

**Definition 26.** The *k*-th projection is the function

\[
P_k(x) = \sum_{i=1}^{n} w_i x_i, \quad w_i = \begin{cases} 
1 & \text{if } i = k \\
0 & \text{otherwise}
\end{cases}
\]

that selects the k-th element of \( x \) as the representative value.

Equivalently the projection is sometimes simply written as \( P_k(x) = x_k \). The projection is homogenous, shift invariant, idempotent, averaging and internal.

**Definition 27.** The *k*-th order statistic is the function \( S_k(x) = x_{(k)} \), being the value of the k-th smallest component of \( x \) (the k-th element of \( x_{\uparrow} \)).

The k-th order statistic is also homogeneous, shift-invariant, idempotent, averaging and internal. Order statistics are perhaps the simplest aggregations after the projection and are used commonly in aggregation of noisy data. Table 1 provides formulae for several common monotonic means. Note interestingly, that \( \min() \) and \( \max() \), being the limiting cases for averaging behaviour, are also means.

There are many other interesting means that appear within the literature (e.g.: the logarithmic means, Heronian means, Bonferroni means, etc.) and a thorough consideration of the class of means can be found in [15].

2.3.2. Medians. The median is also a widely known aggregation function having averaging behaviour. It is particularly used as a robust estimator of an
2.3. EXAMPLES AND MAIN FAMILIES

Table 1. Examples of monotonic means

<table>
<thead>
<tr>
<th>Mean</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic</td>
<td>( A(x) = \frac{1}{n}(x_1 + x_2 + \cdots + x_n) )</td>
</tr>
<tr>
<td>Geometric</td>
<td>( G(x) = \sqrt[n]{x_1x_2\cdots x_n} )</td>
</tr>
<tr>
<td>Harmonic</td>
<td>( H(x) = n\left(\sum_{i=1}^{n} \frac{1}{x_i}\right)^{-1} )</td>
</tr>
<tr>
<td>Minimum</td>
<td>( \min(x) = \min(x_1, x_2, \ldots, x_n) )</td>
</tr>
<tr>
<td>Maximum</td>
<td>( \max(x) = \max(x_1, x_2, \ldots, x_n) )</td>
</tr>
</tbody>
</table>

input vector, given its insensitivity to outliers. Indeed, like the other robust estimators of location presented in Chapter 3, it has a breakdown point of 50%, meaning that its value will not change when up to half of data values are outliers of the central tendency of the data.

**Definition 28.** The **median** is the function

\[
Med(x) = \begin{cases} 
\frac{1}{2}(S_k(x) + S_{k+1}(x)), & \text{if } n = 2k \text{ is even} \\
S_k(x), & \text{if } n = 2k - 1 \text{ is odd.}
\end{cases}
\]

**Definition 29.** The lower (upper) median is the function

\[
Med_l(x) = S_k(x)
\]

where, for the lower median, \( k = \lfloor \frac{n}{2} \rfloor \) is the nearest integer smaller or equal to \( \frac{n}{2} \), while for the upper median \( k = \lceil \frac{n}{2} \rceil \) (nearest larger integer).

**Proposition 1.** For any \( x \in [a,b]^n \) it holds that

\[
Med_l(x) \leq Med(x) \leq Med_u(x).
\]

In particular, if \( n = 2k - 1 \), it follows that \( Med_l(x) = Med(x) = Med_u(x) \), and, in general, \( Med(x) = \frac{Med_l(x) + Med_u(x)}{2} \). A related notion is the \( a \)-median:

**Definition 30.** Given a value \( a \in [0,1] \), the \( a \)-median is the function

\[
Med_a(x) = Med(x_1, \ldots, x_n, a, \ldots, a).
\]

We can immediately prove the following result.
Proposition 2. Given a value $a \in [0, 1]$, the $a$-median is given by

$$\text{Med}_a(x) = \begin{cases} 
\min(x) & \text{if } a \leq \min(x); \\
\max(x) & \text{if } a \geq \max(x); \\
a & \text{otherwise}.
\end{cases}$$

Proof. Observe that $\text{Med}_a$ is in fact the median of $n + (n - 1) = 2n - 1$ arguments, so we always recover the $n$-th greatest argument. $\square$

The concepts introduced up to this point are used under the assumption that all the inputs have the same importance. One way to account for the relative importance of the inputs is through the use of weights and the computation of weighted averages.

2.3.3. Weighted Means. In each of the examples where mention is made of constant weights $\Delta = (w_{i,n} | n \in \mathbb{N}, i \in \{1, ..., n\})$, unless otherwise stated, it should be assumed that the following definition holds.

Definition 31. A weight vector is a tuple $\Delta = (w_{i,n} | n \in \mathbb{N}, i \in \{1, ..., n\})$ such that:

1. $w_i \in [0, 1], \forall i \in \{1, ..., n\}$
2. $\sum_{i=1}^{n} w_i = 1$
3. $\exists j \in \{1, ..., n\} : w_j \neq 1/n$

The weighted arithmetic mean, known colloquially as the “weighted average”, is the inner product of a weight vector and input vector.

Definition 32. Given the weight vector $\Delta$, the weighted arithmetic mean is the function

$$M_{\Delta}(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n = \sum_{i=1}^{n} w_ix_i.$$ 

In the spirit of the weighted arithmetic mean it is also of interest to consider the case in which the inputs affect the output differently. The weighted median (treated in detail in [93]) is given by:
2.3. EXAMPLES AND MAIN FAMILIES

**Definition 33.** Let $\Delta = (w_1, \ldots, w_n) \in [0, 1]^n$ be a weighting vector with $w_i \geq 0$, $\sum w_i = 1$, and let $\mathbf{u}$ denote the vector obtained from $\Delta$ by arranging its components in the order induced by the components of the input vector $\mathbf{x}$, such that $u_k = w_i$ if $x_i = x^{(k)}$ is the $k$-th largest component of $\mathbf{x}$. The lower weighted median is the function

$$\text{Med}_{\Delta, l}(x) = x^{(k)},$$

where $k$ is the index obtained from the condition

$$\sum_{j=1}^{k-1} u_j < \frac{1}{2} \text{ and } \sum_{j=1}^{k} u_j \geq \frac{1}{2}.$$  

The upper weighted median is the function $\text{Med}_{\Delta, u}$ defined as in(1), where $k$ is the index obtained from the condition

$$\sum_{j=1}^{k-1} u_j \leq \frac{1}{2} \text{ and } \sum_{j=1}^{k} u_j > \frac{1}{2}.$$  

The arithmetic mean of $\text{Med}_{\Delta, l}$ and $\text{Med}_{\Delta, u}$ gives the weighted median $\text{Med}_{\Delta}$.

The upper and lower medians (and hence the median) are recovered by assigning the same weight to each of the inputs, $\Delta = (\frac{1}{n}, \ldots, \frac{1}{n})$.

**Proposition 3.** The lower and upper medians, the $a$-median and the lower and upper weighted medians are averaging homogeneous shift-invariant aggregation functions.

Introduced by Yager [92], Ordered Weighted Averages (OWA) associate weights not with a particular input, but rather with an input’s value relative to other inputs.

**Definition 34.** Given a weighting vector $\Delta$, the **Ordered Weighted Averaging** function is

$$\text{OWA}_{\Delta}(\mathbf{x}) = \sum_{i=1}^{n} w_i S_i(\mathbf{x}) = \sum_{i=1}^{n} w_i x^{(i)} = \langle \Delta, \mathbf{x}_{\downarrow} \rangle.$$
The median may be expressed as an OWA function with the weighting vector $\Delta = (0, \ldots, 0, 1, 0, \ldots, 0)$ for odd $n$ and $\Delta = (0, \ldots, 0, \frac{1}{2}, \frac{1}{2}, 0, \ldots, 0)$ for even $n$.

The concept of centered OWA operators was proposed by Yager in [95] and later also investigated in [103]. Here the weights are symmetric ($w_j = w_{n+1-j}$), strongly decaying ($w_i < w_j$ if either $i < j \leq (n+1)/2$ or $i > j \geq (n+1)/2$), and inclusive ($w_j > 0$). For example, consider the weighting vector with several non-zero values,

$$\Delta = (0, \ldots, 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{6}, 0, \ldots, 0),$$

and take into account as many central inputs. A special case of such weight vectors generates the $\alpha$-trimmed mean ($0 \leq \alpha \leq \frac{1}{2}$), which can be written as an OWA function with the weights

$$\Delta = (0, \ldots, 0, \frac{1}{n - 2[n\alpha]}, \ldots, \frac{1}{n - 2[n\alpha]}, 0, \ldots, 0),$$

where $[n\alpha]$ is the nearest integer no greater than $n\alpha$. Another interesting case is the Winsorized mean, in which extreme low and high values are replaced with other input values. An $\alpha\%$ Winsorized mean can be expressed as an OWA function with the weights

$$\Delta = (0, \ldots, 0, ([n\alpha] + 1)a, a, \ldots, a, ([n\alpha] + 1)a, 0, \ldots, 0),$$

with $a = \frac{1}{n}$. As special cases of OWA, $\alpha$-trimmed and Winsorized means are symmetric, homogeneous and shift-invariant aggregation functions. The weighted arithmetic means and OWAs are special (symmetric) cases of the Choquet Integral [23], which is defined with respect to a fuzzy measure (also called a capacity). Capacities were introduced by Choquet [23] as generalisations of the Lebesque integral to non-additive set functions, within the context of potential theory. Sugeno independently proposed fuzzy measures in [80], within the context of multi-criteria decision making, where they were used to model the interaction of criteria. Subsequently, through the following decade, they were developed as key tools for aggregation operations in decision theory.
2.3. EXAMPLES AND MAIN FAMILIES

Definition 35. Let $\mathcal{N} = \{1, 2, \ldots, n\}$. A discrete fuzzy measure is a set function $v : 2^\mathcal{N} \to [0, 1]$ which is monotonic (i.e., $v(A) \leq v(B)$ whenever $A \subseteq B$) and satisfies $v(\emptyset) = 0, v(\mathcal{N}) = 1$.

Furthermore, the following qualify fuzzy measures based on the measure of unions and intersections of subsets.

**Definition 36.** A fuzzy measure $v$ is called **submodular** if for any $A, B \subseteq \mathcal{N}$

\[
(3) \quad v(A \cup B) + v(A \cap B) \leq v(A) + v(B).
\]

It is called **supermodular** if

\[
(4) \quad v(A \cup B) + v(A \cap B) \geq v(A) + v(B).
\]

**Definition 37.** A fuzzy measure $v$ is called **additive** if for any two nonintersecting subsets $A, B \subset \mathcal{N}$, $A \cap B = \emptyset$:

\[
(5) \quad v(A \cup B) = v(A) + v(B)
\]

It is called **subadditive** if

\[
(6) \quad v(A \cup B) \leq v(A) + v(B).
\]

It is called **superadditive** if

\[
(7) \quad v(A \cup B) \geq v(A) + v(B).
\]

The final property of interest is symmetry.

**Definition 38.** A fuzzy measure $v$ is called **symmetric** if the value $v(A)$ depends only on the cardinality of the set $A$, i.e., for any $A, B \subseteq \mathcal{N}$, if $|A| = |B|$ then $v(A) = v(B)$. 
2. LITERATURE REVIEW

**Definition 39.** Given a parameter \( \lambda \in (-1, \infty) \), a **Sugeno \( \lambda \)-fuzzy measure** is a fuzzy measure \( v \) such that for all \( A, B \subseteq \mathcal{N}, A \cap B = \emptyset \) it satisfies

\[
v(A \cup B) = v(A) + v(B) + \lambda v(A) v(B).
\]

Given a fuzzy measure the **Choquet integral** is a piecewise linear, idempotent function defined as:

**Definition 40.** The discrete **Choquet integral** with respect to a fuzzy measure \( v \) is given by

\[
C_v(x) = \sum_{i=1}^{n} x(i) \left[ v(\{j \mid x_j \geq x(i)\}) - v(\{j \mid x_j \geq x(i+1)\})\right],
\]

where \( x \uparrow = (x(1), x(2), ..., x(n)) \) is an increasing permutation of the input \( x \) and \( x(n+1) = \infty \) by convention.

The Choquet integral can also be written as

\[
C_v(x) = \sum_{i=1}^{n} [x(i) - x(i-1)] v(H_i),
\]

where \( H_i = \{(i), (i+1), ..., (n)\} \) is the subset of indices of the \( n - i + 1 \) largest components of \( x \).

The Sugeno integral is similarly another aggregation function defined with respect to a fuzzy measure.

**Definition 41.** The **Sugeno integral** with respect to a fuzzy measure \( v \) is given by

\[
S_v(x) = \max_{i=1, ..., n} \min \{x(i), v(H_i)\},
\]

where \( x \uparrow = (x(1), x(2), ..., x(n)) \) is an increasing permutation of the input \( x \) and \( H_i = \{(i), (i+1), ..., (n)\} \).

The ordering of variables in the Sugeno and Choquet integrals is performed with respect to the value of the inputs. It would seem reasonable to permit the ordering of the inputs with respect to an auxiliary variable; for example, the significance of each input variable (which may vary over time), or some other
function of the input values themselves. Such ideas had their inception in the work of Mitchell and Estrakh [52] [53] and later a formal definition was provided by Yager and Filev [98]. Formally, an induced ordered weighted averaging (IOWA) function is defined by:

**Definition 42.** Given a weighting vector $\mathbf{w}$ and an inducing variable $\mathbf{z}$, the **Induced Ordered Weighted Averaging (IOWA)** function is

$$
IOWA_{\mathbf{w}} (\langle x_1, z_1 \rangle, \langle x_2, z_2 \rangle, \ldots, \langle x_n, z_n \rangle) = \sum_{i=1}^{n} w_i x_{\eta(i)}
$$

where the notation $\eta(\cdot)$ indicates that the inputs $\langle x_i, z_i \rangle$ are reordered such that $z_{\eta(1)} \geq z_{\eta(2)} \geq \ldots \geq z_{\eta(n)}$ and the convention that if $q$ of the $z_{\eta(i)}$ are tied, i.e. $z_{\eta(i)} = z_{\eta(i+1)} = \ldots = z_{\eta(i+q-1)}$, then

$$
x_{\eta(i)} = \frac{1}{q} \sum_{j=\eta(i)}^{\eta(i+q-1)} x_j.
$$

Finally, the fuzzy integrals can be extended to incorporate an auxiliary variable for inducing the ordering. The induced Choquet integral [94] is:

**Definition 43.** The **induced Choquet integral** with respect to a fuzzy measure $v$ and an order inducing variable $\mathbf{z}$ is given by

$$
IC_{v} (\langle \mathbf{x}, \mathbf{z} \rangle) = \sum_{i=1}^{n} x_{\eta(i)} \left[ v \left( \{ j \mid z_j \geq z_{\eta(i)} \} \right) - v \left( \{ j \mid z_j \geq z_{\eta(i+1)} \} \right) \right],
$$

where the notation $\eta(\cdot)$ indicates that the inputs $\langle x_i, z_i \rangle$ are reordered such that $z_{\eta(1)} \leq z_{\eta(2)} \leq \ldots \leq z_{\eta(n)}$ and $z_{\eta(n+1)} = \infty$ by convention.

2.3.4. **Quasi-Averages.** The power means (also known as root-power means) generalise the arithmetic means and are defined as

**Definition 44.** For $r \in \mathbb{R}$, the **power mean** is the function

$$
M_{[r]}(\mathbf{x}) = \left( \frac{1}{n} \sum_{i=1}^{n} x_i^r \right)^{1/r},
$$

if $r \neq 0$ and $M_{[0]}(\mathbf{x}) = G(\mathbf{x})$ (the geometric mean of $\mathbf{x}$).
This family of functions can be extended to the weighted power means.

**Definition 45.** Given a weighting vector $\Delta$ and $r \in \mathbb{R}$, the **weighted power mean** is the function

$$M_{\Delta,[r]}(x) = \left( \sum_{i=1}^{n} w_i x_i^r \right)^{1/r},$$

if $r \neq 0$ and $M_{[0]}(x) = G_{\Delta}(x)$ (the weighted geometric mean of $x$).

If we extend the range of $r$ to $-\infty$ and $+\infty$ then we obtain the augmented family of weighted power means, which are bounded by the limiting cases

$$M_{\Delta,[-\infty]}(x) = \lim_{r \to -\infty} M_{\Delta,[r]}(x) = \min(x)$$

and

$$M_{\Delta,[\infty]}(x) = \lim_{r \to \infty} M_{\Delta,[r]}(x) = \max(x).$$

Note that min and max are not themselves power means.

The quasi-arithmetic means further generalise the power means. Herein, consider:

**Definition 46.** A **generating function** (or scaling function) is any univariate, continuous, strictly monotonic function $g : \mathbb{I} \to \mathbb{R}$ for which $g^{-1}$ exists.

Thus, the following functions may be defined:

**Definition 47.** For a given generating function $g$, the **quasi-arithmetic mean** is the function

$$M_g(x) = g^{-1}\left( \frac{1}{n} \sum_{i=1}^{n} g(x_i) \right).$$

The weighted analogue of this function is given by:

**Definition 48.** For a given generating function $g$ and weighting vector $\Delta$, the **weighted quasi-arithmetic mean** is the function

$$M_{\Delta,g}(x) = g^{-1}\left( \sum_{i=1}^{n} w_i g(x_i) \right).$$
It is apparent then that the weighted power means are a subclass of the weighted quasi-arithmetic means, with generating function

\[
g(t) = \begin{cases} 
  t^r & \text{if } r \neq 0, \\
  \log(t) & \text{if } r = 0.
\end{cases}
\]

The OWA functions can also be generalised in the same manner \[103\].

**Definition 49.** Let \( g \) be a generating function and let \( \Delta \) be a weighting vector. Then the function

\[
GenOWA_{\Delta,g}(x) = g^{-1}\left(\sum_{i=1}^{n} w_i g(x_{i(i)})\right)
\]

is called a generalised OWA (also known as ordered weighted quasi-arithmetic mean).

Subsequently, given an auxiliary variable for determining an ordering over \( x \):

**Definition 50.** Given an inducing variable \( z \), weighting vector \( \Delta \) and generating function \( g \), the induced generalised ordered weighted average (I-GOWA) is given by

\[
I-GenOWA_{\Delta,g}(\langle x,z \rangle) = g^{-1}\left(\sum_{i=1}^{n} w_i g(x_{\eta(i)})\right),
\]

where \( \eta(\cdot) \) indicates that the inputs \( \langle x_i, z_i \rangle \) are reordered such that \( z_{\eta(1)} \geq z_{\eta(2)} \geq \ldots \geq z_{\eta(n)} \).

The same convention for ties is applied as that used for I-OWA (Eq. (10)). The following result allows for the concept of a quasi-median.

**Proposition 4.** Let \( F \) be an aggregation function on \( \mathbb{I}^n \) and let \( h : \mathbb{I} \rightarrow \mathbb{I} \) be a generating function. Then the \( h \)-transform

\[F_h(x) = h^{-1}(F(h(x_1), h(x_2), \ldots, h(x_n)))\]

is also an aggregation function on \( \mathbb{I}^n \).
The following definition follows from applying scaling functions $h : \mathbb{I} \to \mathbb{I}$ to the median.

**Definition 51.** Let $\text{Med}_w$ be a weighted median and $h$ be a generating function. The function

$$f(x) = h^{-1}(\text{Med}_w(h(x)))$$

is called a **weighted quasi-median** with respect to $h$.

If $h$ is a power function or a logarithm the resulting quasi-median is also homogeneous (of order 1).

It is not difficult to check that simple lower and upper medians satisfy $\text{Med}_h = \text{Med}$ with respect to any monotonic bijection $h : \mathbb{I} \to \mathbb{I}$, (i.e., they are invariant aggregation functions \cite{34,49}).

**2.3.5. Disjunctive and Conjunctive Functions.** The functions $\min(x)$ and $\max(x)$ are the limiting cases of the averaging aggregation functions. They are also the canonical examples of (respectively) conjunctive and disjunctive aggregation functions. Two important and widely known classes of conjunctive and disjunctive aggregation functions are the triangular norms and triangular conorms \cite{40}.

**Definition 52.** A **triangular norm** ($t$-norm) is a bivariate aggregation function $T : [0,1]^2 \to [0,1]$, which is associative, symmetric and has a neutral element of 1.

The dual of the $t$-norm is called a triangular conorm.

**Definition 53.** A **triangular conorm** ($t$-conorm) is a bivariate aggregation function $S : [0,1]^2 \to [0,1]$, which is associative, symmetric and has a neutral element of 0.

The triangular norms were introduced by Menger \cite{47} in the context of fusing distribution functions needed by the generalisation of triangular inequality of a metric on statistical metric spaces. They are now applied widely as a model of
fuzzy sets intersection and have also been applied in studies of many-valued logics and non-additive measures and integrals.

A generalisation of the t-norms and t-conorms are the uninorms, introduced by Yager [99].

**Definition 54.** A **uninorm** is a bivariate aggregation function $U : [0, 1]^2 \to [0, 1]$ which is associative, symmetric and has a neutral element $e$ belonging to the interval $]0, 1[$.

By modifying the interval over which $e$ may range to $[0, 1]$, the t-norm and t-conorm become limiting cases of the uninorms. Uninorms have mixed behaviour with respective to the magnitude of their inputs, such that they act as t-norms for low valued inputs and t-conorms for high valued inputs. The nullnorms on the other hand reverse this behaviour; that is, acting as t-conorms for low valued inputs and t-norms for high valued inputs. As with the uninorms, their definition is obtained by modifying the axiom concerning the neutral element [46].

**Definition 55.** A **nullnorm** is a bivariate aggregation function $V : [0, 1]^2 \to [0, 1]$ which is associative and symmetric, such that there exists an element $a$ belonging to the open interval $(0, 1)$ verifying

$$
\forall t \in [0, a], \quad V(t, 0) = t,
$$

$$
\forall t \in [a, 1], \quad V(t, 1) = t.
$$

Again, by permitting $a$ to range over the whole interval $[0, 1]$ the t-norms and t-conorms become special limiting cases.

While the uninorms and nullnorms are one way of building aggregation functions that mix conjunctive and disjunctive functions, the family of functions known as **T-S functions** is another approach that provides a uniform behaviour, in the sense that the behaviour of such a function does not depend on the part of the domain being considered. The first class of such operators was introduced by Zimmermann [104] for modelling human decision processes and subsequently these were generalised to broader classes of operators [29] [48] [69] [88].
Definition 56. Given $\gamma \in [0, 1]$, a t-norm $T$, a t-conorm $S$ and a continuous strictly monotonic function $g : [0, 1] \to [-\infty, \infty]$ such that $\{g(0), g(1)\} \neq \{-\infty, \infty\}$, the corresponding T-S function is defined as

$$Q_{\gamma,T,S,g}(x_1, x_2, \ldots, x_n) = g^{-1}((1 - \gamma) : g(T(x_1, \ldots, x_n)) + \gamma g(S(x_1, \ldots, x_n))).$$

The function $g$ is called the generating function of $Q_{\gamma,T,S,g}$.

2.4. Penalty-based Functions

Both the arithmetic mean and the median are known to be solutions to simple optimisation problems based on the minimisation of a measure of disagreement between the input values and the output value. The arithmetic mean of $x$ is given by

$$\mu = \arg \min_y \sum_{i=1}^n (x_i - y)^2 \quad (11)$$

and the median is

$$\mu = \arg \min_y \sum_{i=1}^n |x_i - y|. \quad (12)$$

In [17] it was demonstrated that averaging aggregation functions can be expressed equivalently as the solution to a minimisation problem of the form

$$F(x) = \arg \min_y P(x, y) \quad (13)$$

where $P(x, y)$ is a penalty function. One can think of $P$ as measuring the disagreement, or dissimilarity between the inputs $x$ and output $y$. It follows that $\mu = F(x)$ is a function having a value $\mu$ that minimises this dissimilarity given $x$. It is not necessary to explicitly state $F$, provided a suitable penalty function is given and the optimisation problem solvable. Subsequently it is sufficient to
solve $\mu = \arg \min_y \mathcal{P}(x, y)$ for all $x \in \mathcal{X}$. For $F$ to maintain the properties of monotonicity and bounds preservation, the following definition must hold for the chosen $\mathcal{P}$:

**DEFINITION 57. Penalty function.** The function $\mathcal{P} : \mathbb{I}^{n+1} \rightarrow \mathbb{R}$ is a penalty function if and only if it satisfies:

1. $\mathcal{P}(x, y) \geq c \quad \forall x \in \mathbb{I}^n, y \in \mathbb{I}$;
2. $\mathcal{P}(x, y) = c$ if all $x_i = y$; and,
3. $\mathcal{P}(x, y)$ is quasi-convex in $y$ for any $x$,

for some constant $c \in \mathbb{R}$ and any closed, non-empty interval $\mathbb{I}$.

The first two conditions ensure that $\mathcal{P}$ has a strict minimum and that a consensus of inputs ensures minimum penalty, providing idempotence of $F(x)$. The third condition implies a unique minimum (but possibly many minimisers that form an interval, in which case a convention for selecting the minimiser is required. The midpoint of the interval of minimisers is frequently chosen (for example, see [34], p.258). A function $P$ is quasi-convex if all of its sublevel sets are convex. That is, $S_\alpha(P) = \{ x | P(x) \leq \alpha \}$ are convex sets for all $\alpha$ (see, for example, [71]). Since multiplication by, or addition of a constant to $\mathcal{P}$ will not change the minimisation, $\mathcal{P}$ may be shifted (if desired) so that $c = 0$.

2.4.1. Penalty-based Averages. Penalty-based functions are not restricted to the class of monotonic functions. Non-monotonic averaging functions can be represented by a penalty function obtained by relaxing condition (3) above, which will be discussed in Section 2.5. In general, the following holds:

**THEOREM 1.** Any idempotent function $F : \mathbb{I}^n \rightarrow \mathbb{I}$ can be represented as a penalty based function $\mathcal{P} : \mathbb{I}^{n+1} \rightarrow \mathbb{I}$ such that

$$F(x) = \arg \min_y \mathcal{P}(x, y).$$

**PROOF.** The function $\mathcal{P}(x, y) = (F(x) - y)^2$ is one such penalty function and any strictly convex (or quasi-convex) univariate function of $t = F(x) - y$ can serve as a penalty function. \hfill \Box
It follows immediately that any mean can be represented by a penalty function.

**Corollary 1.** Any averaging aggregation function can be expressed as a penalty based function.

For a twice differentiable penalty function Calvo showed in [17] that a sufficient condition for monotonicity (and hence averaging behaviour) is

\[
\frac{\mathcal{P}_{x, y}}{\mathcal{P}_{y y}} \geq 0.
\]

### 2.4.2. Penalty Functions on Lattices.

The results for penalty-based functions can be extended to lattices, which will permit the formulation of aggregation solutions for coloured images. Consider first the following definitions.

**Definition 58.** Let \( L \) be a set. A **lattice** \( \mathcal{L} = (L, \leq, \wedge, \vee) \) is a poset with the partial order \( \leq \) on \( L \), and meet and join operations \( \wedge, \lor \), if every pair of elements from \( L \) has both meet and join.

**Definition 59.** Let \( S \) be a poset. A **chain** in \( S \) is a totally ordered subset of \( S \). The length of a chain is its cardinality.

**Definition 60.** If \( \mathcal{L}_1 = (L_1, \leq_1, \wedge_1, \lor_1) \) and \( \mathcal{L}_2 = (L_2, \leq_2, \wedge_2, \lor_2) \) are two lattices, their **Cartesian product** is the lattice \( \mathcal{L}_1 \times \mathcal{L}_1 = (L_1 \times L_2, \leq, \wedge, \vee) \) with \( \leq \) defined by

\[
(x_1, y_1) \leq (x_2, y_2) \iff x_1 \leq_1 x_2 \text{ and } y_1 \leq_2 y_2,
\]

and

\[
(x_1, y_1) \wedge (x_2, y_2) = (x_1 \wedge_1 x_2, y_1 \wedge_2 y_2),
\]

\[
(x_1, y_1) \lor (x_2, y_2) = (x_1 \lor_1 x_2, y_1 \lor_2 y_2).
\]

For representing colours in digital images, Cartesian products of finite chains will be used, with the length of each chain typically being 256. We note that all finite chains of the same length are isomorphic to each other, and hence we
can represent them as non-negative integers 0, 1, \ldots, K, and elements of product lattices as tuples \( x = (x_1, x_2, \ldots, x_m), x_i \in \mathbb{Z}_+ = \{0, 1, 2, \ldots\}. \)

**Definition 61.** Let \( f_1, f_2 \) be two aggregation functions defined on sets \( X_1 \) and \( X_2 \) respectively. The **Cartesian product of aggregation functions** is \( f = f_1 \times f_2 : X_1 \times X_2 \to Y_1 \times Y_2 \) defined by

\[
f(x_1, x_2) = (f_1(x_1), f_2(x_2)).
\]

Following on from representing the arithmetic mean and the median as penalty based aggregation functions (equations (11) and (12)), they may be defined on lattices as follows.

**Definition 62.** Let \( \mathcal{L} = (L, \leq, \wedge, \lor) \) be a product of finite chains. The geodesic distance between \( x, y \in \mathcal{L} \) is defined as the length of a chain \( C \) with the least element \( a = x \wedge y \) and the greatest element \( b = x \lor y \) minus 1,

\[
d(x, y) = \text{length}(C) - 1,
\]

since it corresponds to the smallest number of edges between vertices \( x \) to \( y \) in the covering graph of \( L \).

Note that all the chains with the least element \( a \) and the greatest element \( b \) on a product lattice in Definition 62 have the same length. This definition is equivalent to the following

\[
d(x, y) = \sum_{i=1}^{m} d_i(x_i, y_i) = \sum_{i=1}^{m} |x_i - y_i|,
\]

where \( d_i \) is the distance in the \( i \)-th chain in the product of \( m \) chains.

**Definition 63.** Let \( \mathcal{L} \) be a product of finite chains. Consider \( n \) elements \( x_1, \ldots, x_n \in \mathcal{L} \), that must be averaged. Let the penalty function be \( P: \mathbb{Z}_+^n \to \mathbb{R} \). The penalty based function on \( \mathcal{L} \) is \( f \) given by

\[
f(x_1, \ldots, x_n) = \mu = \arg \min_{y \in \mathcal{L}} P(d(x_1, y), d(x_2, y), \ldots, d(x_n, y)).
\]
Note that $P$ is quasi-convex in $y$, as in Definition 57. However, now $y \in \mathcal{L}$ rather than an element of a chain. To accommodate this within the definition of a quasi-convex function, the following results are required. A function $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ is quasi-convex on $X$ if all its level sets are convex. A function $f : \mathcal{L} \subseteq \mathbb{Z}_+^n \rightarrow \mathbb{R}$ is quasi-convex if its extension $\tilde{f} : X \rightarrow \mathbb{R}$ is quasi-convex, where $X \subseteq \mathbb{R}^n$ is the smallest set containing $\mathcal{L}$. Similarly $f : \mathcal{L} \rightarrow \mathbb{R}$ is convex if its extension $\tilde{f}$ is convex.

2.5. Non-monotone Means

2.5.1. Robust Estimators of Location. Robust estimators of location are averaging functions used to incorporate the position of data values relative to each other into the aggregation problem and thus they estimate the central tendency of the inputs. The mode is the most widely known of these estimators (possibly known to the ancient Greeks [74] [75]). The mode may be expressed in penalty form [17] as:

**Definition 64.** The Mode is the minimiser of the function

$$\text{Mode}(x) = \arg \min_{y \in \mathbb{E}} \sum_{i=1}^{n} p(x_i, y)$$

where $p(x_i, y) = \begin{cases} 0 & x_i = y \\ 1 & \text{otherwise} \end{cases}$.

Note that condition (3) of Definition 57 has been relaxed and the penalty function is not quasi-convex, in which case it may have several minimisers. In this case a convention is required as to which minimiser is selected (e.g., the smallest, or largest). Such a convention will be required for all non-monotonic means expressed as penalty-based functions.

The mode is idempotent and averaging, internal, homogeneous and shift-invariant. Functionally it is an estimator of the most significant cluster of input values, however it is not well defined when the input values are all unique, as in the tuple $(0.28, 0.21, 0.25, 0.36, 0.42, 0.83)$. In this example the first three values form a relatively compact cluster (at least when compared with the other potential clusters in this data), however as each value appears only once, any value would
satisfy Definition 64. A value within the cluster is a potential measure of central
tendency for this data, if the values outside the cluster are considered outliers.
While the median is robust to outliers, having a breakdown point of 50%, its
value of 0.32 is not a good estimate of a value within the principle cluster. The
arithmetic mean value of 0.39 is worse still as it is heavily biased by the last two
inputs. The robust estimators of location are thus averaging functions designed
to identify the principle cluster within the data and return a representative value
of this cluster (a cluster prototype).

The shorth [1], the Least Median of Squares (LMS) [72] and the Least Trimmed
Squares (LTS) [72] [73] are some of the many estimators found within the robust
statistics literature. Each is calculated using the shortest contiguous sub-sample
of \( x \) containing at least half of the values. The candidate sub-samples are the
sets \( X_k = \{x_j : j \in \{k, k+1, ..., k + \lfloor \frac{n}{2} \} \}, \quad k = 1, ..., \lfloor \frac{n+1}{2} \} \). The length of
each contiguous set is taken as \( \|X_k\| = |x_{k+\lfloor \frac{n}{2} \rfloor} - x_k| \) and thus the index of the
shortest sub-sample is

\[
k^* = \arg \min_i \|X_i\|, \quad i = 1, ..., \left\lfloor \frac{n+1}{2} \right\rfloor.
\]

**Definition 65.** The **Shorth** is the arithmetic mean of the shortest conti-
guous subsample containing half of the observations. It is the solution to the
minimisation problem given by

\[
\text{Shorth}(x) = \arg \min_{y \in I} \sum_{i=1}^{h} (x_i - y)^2, \quad x_i \in X_{k^*}, \quad h = \left\lfloor \frac{n}{2} \right\rfloor + 1.
\]

**Definition 66.** The **Least Median of Squares** estimator is the midpoint
of the shortest contiguous subsample containing half of the observations. It is the
solution to the minimisation problem given by

\[
\text{LMS}(x) = \arg \min_{y \in I} \text{Med} \{(x_i - y)^2 \mid x_i \in X_{k^*}\}.
\]

The objective in (15) is not quasi-convex in \( y \) and hence does not satisfy Definition
57, thus several optima may exist.

The Least Trimmed Squares estimator is defined as
**Definition 67.** The **Least Trimmed Squares** estimator is the arithmetic mean of the contiguous subsample containing half of the observations that has smallest variance. It is the solution to the minimisation problem given by

$$
LTS(x) = \arg \min_{y \in I} \sum_{i=1}^{h} r_{(i)}^2,
$$

where $r_{(i)}$ denotes the $i$'th smallest residual in $r = \{|x_j - y| | x_j \in X_k\}$ and $h = \left\lfloor \frac{n}{2} \right\rfloor + 1$.

Each of these functions is non-monotonic and has averaging behaviour. They are also symmetric, homogeneous and shift-invariant. Rousseeuw [73] states that ideally, all robust estimators of location of input data should have these properties. The properties of these robust estimators of location will be investigated further in Chapter 3. The mode is revisited in Section 3.4 in the context of introducing mode-like averages based on penalty functions.

**2.5.2. Bajraktarevic Means.** An important class of means that are not always monotonic are those expressed by the Mean of Bajraktarevic, which is a generalisation of the weighted quasi-arithmetic means. This requires a generalisation of the constant weighting vector $\Delta$, to that of a vector of continuous weighting functions.

**Definition 68.** A **weighting function** is any univariate, continuous function $w : \mathbb{I} \rightarrow [0, \infty)$.

**Definition 69.** Let $w(t) = (w_1(t), ..., w_n(t))$ be a vector of weighting functions $w_i : \mathbb{I} \rightarrow [0, \infty)$ and let $g : \mathbb{I} \rightarrow \mathbb{R}$ be a strictly monotonic function. The **mean of Bajraktarevic** is the function

$$
M_g^w(x) = g^{-1} \left( \frac{\sum_{i=1}^{n} w_i(x_i)g(x_i)}{\sum_{i=1}^{n} w_i(x_i)} \right).
$$
When \( g(x_i) = x_i \), the Bajraktarevic mean is called a *generalised mixture function* (or *generalised mixture operator*) and is given by

\[
M_w(x) = \frac{\sum_{i=1}^{n} w_i(x_i)x_i}{\sum_{i=1}^{n} w_i(x_i)}.
\]

For the further case where the weighting functions are the same, i.e., \( w_i(t) = w_j(t) \), the operator \( M_w(x) \) is simply a *mixture function* (or *mixture operator*)

\[
M_w(x) = \frac{\sum_{i=1}^{n} w(x_i)x_i}{\sum_{i=1}^{n} w(x_i)}.
\]

Alternatively, if we take the case of the mean of Bajraktarevic with \( w_i(t) = w_j(t) \), then we obtain the quasi-mixture operators

\[
M_{w^p}(x) = g^{-1} \left( \frac{\sum_{i=1}^{n} w(x_i)g(x_i)}{\sum_{i=1}^{n} w(x_i)} \right).
\]

Consequently, the mean of Bajraktarevic can be considered a generalised quasi-mixture operator.

A particularly interesting sub-class of Bajraktarevic means are weighted Gini means, obtained by setting \( w_i(t) = w_it^p \) and \( g(t) = t^p \) when \( p \neq 0 \), or \( g(t) = \log(t) \) if \( p = 0 \).
\[ G_{p,q}^\Delta(x) = \left( \frac{\sum_{i=1}^{n} w_i x_i^{p+q}}{\sum_{i=1}^{n} w_i x_i^q} \right)^{\frac{1}{p}} \]

Gini means generalise the weighted power means (for \( q = 0 \)) and hence include the minimum, maximum and the arithmetic mean as special cases. Another special case of the Gini mean is the Lehmer, or counter-harmonic mean, obtained when \( p = 1 \) and \( w_i = 1 \) for all \( i = 1, \ldots, n \). The Lehmer mean in \( n \) arguments is

\[ L_q(x) = \frac{\sum_{i=1}^{n} x_i^{q+1}}{\sum_{i=1}^{n} x_i^q}. \]

The contra-harmonic mean is the Lehmer mean with \( q = 1 \). The Lehmer mean is also a mixture function with weighting functions \( w_i(t) = t^q \). For \( q < 0 \) the value of the Lehmer mean at \( x \) with at least one component \( x_i = 0 \), is defined as the limit when \( x_i \to 0^+ \), so that \( L_q \) is continuous on \([0, \infty)^n\).

In general, averaging functions with weights that depend on the inputs are not monotonic and hence not classified as aggregation functions. Some results exist to show sufficient conditions for (increasing) monotonicity of the mixture functions. Consider functions given by Eqn. (19) and in the following three cases, consider weighting functions \( w : [0, 1] \to [0, \infty[. \) Then sufficient conditions for monotonicity are:

1. \( w(t) \geq w'(t) \) for any increasing, piecewise differentiable weighting function \([45] \);
2. \( w(t) \geq w'(t)(1 - t) \) for all \( t \in [0, 1] \): \([50], [51]\); or, if we fix the dimension \( n \) of the domain, \( n > 1 \): \([50]\).
3. \( \frac{w^2(t)}{(n-1)w(t)} + w(t) \geq w'(t)(1 - t), t \in [0, 1], n > 1 \): \([50]\).
Analogous results have been obtained for non-increasing weighting functions using duality [50] [70] [79]. Taking the dual weighting function \( w^d(t) = w(1-t) \), the resulting mixture function is the dual to \( M_w \); that is, \( M_{w^d} = 1 - M_w \).

As Mesiar showed [51], mixture functions can be written in penalty function form.

**Theorem 2.** The function \( M_w(x) = \arg\min_y \mathcal{P}(x, y) \) with penalty function

\[
\mathcal{P}(x, y) = \sum_{i=1}^{n} w(x_i)(x_i - y)^2
\]

is a mixture function.

**Proof.** (Due to Mesiar) This penalty attains a minima when

\[
\mathcal{P}_y(x, y) = -2 \left( \sum_{i=1}^{n} w(x_i)x_i - y \sum_{i=1}^{n} w(x_i) \right) = 0.
\]

It follows directly that the value of \( y \) at the minimum is

\[
y = \frac{\sum_{i=1}^{n} w(x_i)x_i}{\sum_{i=1}^{n} w(x_i)}
\]

and hence \( \mathcal{P}(x, y) \) defines a mixture function. \( \square \)

Evidently, for

\[
\mathcal{P}(x, y) = \sum_{i=1}^{n} w(x_i)(f(x_i) - f(y))^2;
\]

if \( f : \mathbb{I} \to \mathbb{R} \) is a continuous, strictly monotonic function, then the solution \( \mu = \arg\min_y \mathcal{P}(x, y) \) is the quasi-mixture function (Eqn. (20)) with generator function \( f(t) \). In a similar vein, taking the penalty function

\[
\mathcal{P}(x, y) = \sum_{i=1}^{n} w(x_i) |f(x_i) - f(y)|,
\]

the solution \( \mu \) is then a weighted quasi-median with generator \( f \), given by Definition 51 [51].
2.5.3. Weighted Means and Outliers. An important driver in the development of means with variable weights is the need to deal with outliers, which exist in most real data sets. Outliers are data that stand apart from the main trend of inputs and can be due either to errors in measurement or recording, or unaccounted for phenomena. In the first case outliers need to be eliminated so as not to contaminate the data, its interpretation or subsequent analysis. In the second case the outliers are themselves the main interest. In this thesis it is argued that most monotonic functions are not suitable for accounting for outliers (a notable exception is the median), because monotonicity in all arguments implies that the output is always affected by variation in any input, whether or not it is far from the main trend. Ideally the contribution of each input to the average should be weighted according to how far away this input is from the main trend (or majority) of the inputs, as has been proposed in the density-based averaging concept [2]. However, once input-dependent weights are introduced to eliminate the outliers, monotonicity is lost, since the product of a decreasing weight and increasing input (moving away from the remaining inputs) is not a monotonic function [50].

The Lehmer means, Gini means and generalised mixture operators will be investigated further in Chapter 3 where new results are presented regarding their non-monotonicity and explicitly, under what conditions each of these means exhibits a newly defined, weaker type of directional monotonicity. These new results, along with comparable results for the robust estimators of location and more broadly, other penalty-based non-monotonic averaging functions will help to bridge the gap between the theorems of aggregation functions - defined only for monotonic functions - and the wider class of non-monotonic means. In the remainder of this chapter the focus now turns to the application domain for this research, that of digital images and image processing problems.
2.6. Images and Information

Solid state imaging sensors in digital still and video cameras contain an array of discrete photodiodes that absorb electromagnetic radiation and produce an internal charge that is linearly proportional to the total energy delivered by incident photons in a given time interval. This charge can be read as an analogue signal and the manner in which the charge is transferred and amplified forms the basis of the difference between Charged Couple Device (CCD) and Complementary Metal-Oxide-Semiconductor (CMOS) based imaging sensors [57].

CCD-based devices have long dominated the marketplace, due to their high image quality and small pixel size. However, recent developments in CMOS-based active pixels (that incorporate charge amplification and logic circuits into the array), coupled with reductions in the size of microlens and filter arrays, has meant that CMOS imaging sensors now dominate the market for low-cost point and shoot cameras and camera-on-chip modules used in smartphones and other embedded imaging devices. Furthermore, several manufacturers now rely on large format CMOS sensors for their high end digital SLR cameras [56].

The result of these technology advancements is that low cost, high resolution, ‘smart’ digital imaging systems now pervade modern life. The consequence is that we now produce, transfer and store more digital imagery than ever before and at resolutions that grow exponentially each year [56]. Unfortunately, the reduction in pixel size that drives the growth in image resolution, particularly for the point-and-shoot market, comes at an inherent cost of increased image noise and unwanted artifacts, which must be dealt with by post-processing algorithms.

This rapid development of advanced digital imaging systems and the subsequent increases in both pixel counts within images and noise effects is the underlying driver for this research. It is within the context of high resolution images corrupted by noise that consideration is given to the problems of image scale reduction and image noise reduction (filtering), considering both as averaging problems. Before considering these problems and the current state of the art in
image processing algorithms, some consideration of images, their formation and properties is essential.

2.6.1. Raster Images. The raw output of the sensor array produces one image element per detector, with the intensity level of this element a function of the photon energy impacting that detector while the shutter was open. Both the image element and photodetector are now commonly referred to as a pixel. Thus, when using the word pixel, we interchangeably mean both the element of an image and its corresponding element of the imaging sensor array. Internal to the sensor, photon energy is converted to a voltage level, which is quantised to obtain a discrete intensity value within a finite range. Typically, the optical path of a digital camera transmits the full spectrum of incoming light and thus, without filtering, the sensor array would record the full visual spectrum intensity per pixel, producing what is commonly known as a greyscale (or luminance) image.

To capture coloured images the incoming light is filtered to permit transmission of photons within fixed bandwidth ranges. Earlier cameras based on CCD technology use dichroic prisms to split the light into multiple paths, with one sensing array per path. Naturally these cameras were more expensive and had more points of failure and sources of error and distortion than their modern CMOS-based counterparts. Most CMOS sensors use a single photodetector array and overlay a grid of micro-lenses and micro-filters. The filters are applied so as to further restrict the wavelength of incident photons onto each photodiode, to obtain discrete interlaced grids of red, blue and green sensitive detectors. A common layout of these filters is the Bayer pattern, depicted in Figure 2.1.

The raw Bayer pattern of colour intensities is reduced to a regular two dimensional grid of multi-colour pixels by a process known as demosaicing. The result of demosaicing (or the output of a colour-sensitive CCD sensor) is a multi-channel raster image (also called a bitmap) that encodes a discrete approximation of the original visual scene. A bitmap image can also be formed by rasterisation of a continuous image, such as a photograph, scalable vector graphics (SVG) image
or the output of analogue imaging system, by sub-sampling from the continuous image using an appropriate filter (such as, again, the Bayer filter) and then applying a demosaicing algorithm.

Most modern digital image and video cameras apply further post-processing to the raw bitmap image, performing operations such as white balancing, the removal of pattern noise, contrast and saturation adjustments, sharpening and image compression. This last operation reduces the colour depth of the image and results in smaller file sizes, but where ‘lossy’ compression algorithms are applied, further artifacts are introduced into the compressed image.

The colour depth of an image is determined by the quantisation, by the analogue-to-digital converter (ADC), of the voltage levels of the photodiodes and is directly related to the binary encoding of intensity for each pixel. The colour depth is expressed either as the number of bits per pixel (bpp) or the bits per channel (bpc). Using \( b \) bits, the number of quantised values that can be represented is \( 2^b \) and these are typically given in the range of \([0, K]\), where \( K = 2^b - 1 \).

There are two principle methods for encoding colour intensity: indexed colour and direct colour. In indexed colour, a pixel intensity value is used as an index into a colour lookup table, known as a pallete. The pallete is device dependent and thus indexed colours permit device independent image encoding. Unfortunately, differences between the palette of the encoding device and the display device result in unfaithful reproduction of the original scene. Indexed colour was used in older computer systems with graphical interfaces, with up to 12 bits per pixel,
beyond which the number of colour values and thus the size of the palette made it impractical to store the colour table in RAM.

Modern digital display devices rely on direct colour encoding, whereby each channel of the raster image corresponds to a dimension of the colour space used to represent colour values (or a mapping between the colour space of the image and colour space of the display is first applied to obtain transformed colour values). The most common space is the additive RGB colour model, which stores red, green and blue pixel intensities and traditionally uses 8 bits per channel, for a 24 bit encoding scheme. This equates to $16,777,216$ colour values, which is more than sufficient for human perception, which can distinguish between approximately $10,000,000$ distinct colours. Imaging displays directly use the RGB colour values per pixel to set the intensity value of the corresponding pixel within the display hardware. Where the display resolution differs from that of the encoded resolution, image scaling is required.

One of the problems arising from using a fixed quantisation (and thus finite colour depth) is that only a discrete, disjoint set of colours are represented directly. Where colours between those available in the palette are required, a process known as dithering can be applied to the image, which is equivalent to adding structured noise to the colour intensity data. This has the effect of reducing colour banding and sharp edges within the image so that the human eye naturally interpolates localised colour values. The effect generates the perception of smoother colour gradients and thus the inference that the missing colours are represented within the image. While introducing noise via dithering could be considered optional, unfortunately there are several other sources of the noise that unavoidably corrupt digital images.

### 2.6.2. Digital Image Noise

Given the nature of the technology used to capture, record and transmit digital images, it is not possible to have a clean, noiseless image. Data generated by solid state imaging sensors in response to incident light, suffers from the following types of noise (corruption) [56]:

(1) **Speckle noise.** So-called because of its grainy, speckled appearance over an image, this random variation in pixel intensity is caused by a variety of factors that perturb the measured output voltage of individual photodiodes. Internal to the imaging sensor thermal noise effects dominate, which is more pronounced at higher temperatures and in low illumination contexts. Random excitation of charge carriers and variabiity in amplifier circuits contributes electronic noise, as does electromagnetic interference from both internal and external sources. As a result of the central limit theorem, speckle noise is normally distributed. Common models assume it is an i.i.d. additive process, although speckle noise in certain imaging systems (such as synthetic aperture radar and ultrasound) are best described by multiplicative models.

(2) **Impulse noise.** Also described as salt-and-pepper noise due to the corruption of pixel values to either the minimum or maximum intensity. It is typically caused by errors in analogue-to-digital converters and bit errors in data transmission. Faulty photodetectors in the sensing array can also cause one-sided impulse noise.

(3) **Shot noise:** This random fluctuation in intensity is due to statistical quantum fluctuations; that is, fluctuations in the number of photons sensed at a given exposure level. For \( N \) sensed photons, the variation is of the order \( \sqrt{N} \). Shot noise dominates in lighter portions of the image (which have more incident photons per unit time) and follows a Poisson distribution. Shot noise at one pixel is independent of noise at another pixel.

(4) **Quantisation and Sampling noise:** These forms of noise derive from the discretisation of the domain and range of the original analogue visual signal. Discretising the domain (sampling) occurs because the CMOS or CCD sensor contains a regular grid of finite sized photodetectors that sample at discrete points of the incident light field. Discretisation of the output range of each detector (quantisation) causes an error between the
reported intensity and the sensed intensity. Both sources produce approximately uniformly distributed noise: for quantisation the root mean square noise level is proportional to the width of the quantisation interval and for sampling, the maximum spatial frequency that can be detected in the incoming light is inversely proportional to the pixel separation in the sensor.

Figure 1.3 on page 15 shows samples of each of these noise types on a uniform intensity backgrounds.

Noise in colour images can be decomposed into chroma noise (fluctuations in colour) and luminance noise (fluctuations in intensity). Chroma noise is considered more visually unappealing than luminance noise due to its unnatural look [56], whereas luminance noise appears similar to the graininess of images captured by film (generated by the random distribution of silver halide particles within the film substrate).

Noise reduction is perhaps the most fundamental problem within the field of image processing and a thorough review of the field is well beyond the scope of even specialist texts. The approaches can be broadly categorised though into local versus global methods and within these categories the major methodological themes can be presented herein. Global methods for noise reduction are those that are applied to the whole image, whereas local methods are applied to connected (typically overlapping) subsets of the image.

2.6.3. Noise Reduction Methods. The global methods for noise reduction operate in what is typically referred to as the transform domain. The image spatial-tonal data is transformed into an alternative space, operations are applied on the data within that space and the data is transformed back to the spatial-tonal space. For instance, transformation to a spatial-frequency domain by Fourier transform permits the application of low pass filters and frequency smoothing methods [36]. More popular though are the wavelet decomposition methods, which were popularised by the work of Donoho [24] [27] [28]. The central premise of wavelet methods is that white noise in the spatial domain is transformed into
white noise in the wavelet coefficient domain, but signal within the spatial domain is typically transformed into only a small band of coefficient values. The task then is to filter the coefficient values, or select only a subrange of values so as to omit the noise from the reconstructed image. Yaroslavsky provides a good review of transform domain methods [101].

Local methods for noise reduction typically operate in the spatial-tonal space of the image. The earliest methods of noise reduction in digital images were implementations of simple linear smoothing filters applied on bounded, overlapping subsets of the image. Such filters involve the convolution of the image signal with a radially symmetric kernel function and are typically expressed in analytic form by assuming continuous pixel addressing variables $x = (x, y)$ and that the sampled image (i.e., that detected by the imaging sensor) is piecewise constant in $I$. Taking $x, y \in \mathbb{R}^2$ and $u(x) \in I$ as the intensity at the given location, then the general form of a linear filter is

$$
\bar{u}(x) = \frac{1}{C(x)} \int_{y \in B_x} g(y - x)u(y)dy
$$

where $B_x \subset \mathbb{R}^2$ is a subset of the image centered on $x$, $C(x) = \int_{y \in B_x} g(y - x)dy$ is a normalisation constant and $g(y - x)$ is the chosen kernel. The ubiquitous choice for the kernel function is the Gaussian function

$$
g(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{z^Tz}{2\sigma^2} \right\},
$$

which produces the 2D isotropic Gaussian blur filter (attributed to Gabor [44]). For digital images the linear filters are generally implemented as a weighted average over a local region (window) centered on the pixel to be filtered. This operation can be expressed as a discrete summation centered on the pixel at location $(k, l)$ and having neighbourhood $B_x = \{(i, j) | i = 1, ..., m; j = 1, ..., n \}$

$$
\bar{u}_{kl} = \sum_{i=1}^{m} \sum_{j=1}^{n} h_{ij}u(k - \left\lfloor \frac{m}{2} \right\rfloor + i, l - \left\lfloor \frac{n}{2} \right\rfloor + j),
$$
where \( H = \{ h_{ij} \mid (i, j) \in B_x \} \) is the discrete impulse response function over the given neighbourhood. For example, given a \( 5 \times 5 \) neighbourhood, the impulse response function corresponding to a discrete Gaussian kernel with zero mean, unit variance is:

\[
H = \frac{1}{273} \begin{bmatrix}
1 & 4 & 7 & 4 & 1 \\
4 & 16 & 26 & 16 & 4 \\
7 & 26 & 41 & 26 & 7 \\
4 & 16 & 26 & 16 & 4 \\
1 & 4 & 7 & 4 & 1
\end{bmatrix}.
\]

The values in \( H \) represent weights applied to each pixel in the neighbourhood, such that (26) computes a weighted arithmetic mean. Figure 2.2 shows a pictorial representation of this operation, with two neighbourhood regions highlighted. Within the output image the value \( \bar{u}_{kl} \) is used as the intensity of pixel \((k, l)\). Images are typically padded at the boundaries so that \( k = 1, ..., M \) and \( l = 1, ..., N \) for an image of size \( M \times N \).

A simple box filter, corresponding to the arithmetic mean over the window, has constant weights of \( \frac{1}{|B|} \) (i.e., take \( C(x) = |B_x| \) and \( g(t) = g(0) = 1 \)).

The Gaussian filter is a low pass filter, meaning that it truncates all signal frequencies above its cutoff frequency. Most convolution-based filters (i.e., those represented by eqn. (24)) are low pass filters. Since fine image detail exists at high frequencies and sharp edges are also high frequency signals, then low pass filtering softens image gradients and removes important fine detail, making these
methods impractical for denoising. Additionally, while filters such as the Weiner filter may be optimal with respect to an assumption of white noise corruption of the image, most real images do not satisfy this assumption and thus optimal low pass filtering will introduce artifacts into the processed image.

To overcome such difficulties, nonlinear smoothing filters have been proposed by a variety of authors. The principal objective of these filters is to remove noise and preserve edges (and other high frequency, structured information within the image). A general class of such edge preserving filters are known as the *spatial-tonal filters* and also *neighbourhood filters* [100]. These filters take account of the dissimilarity between pixels, within both tonal and spatial dimensions, to determine the filtered value. The general form of these filters is given analytically by

\[
\tilde{u}(x) = \frac{1}{C(x)} \int_{y \in B_x} f \left( u(y) - u(x) \right) g \left( \| y - x \| \right) u(y),
\]

with \( C(x) = \int_{y \in B_x} f \left( u(y) - u(x) \right) g \left( \| y - x \| \right) dy \) the normalisation factor. The functions \( f \) and \( g \) represent the tonal and spatial weighting functions respectively. Versions of such filters include the bilateral filter [85], mean-shift [20] and local mode filtering [89], which have been shown to be equivalent to the partial differential equation methods (such as Perona and Malik’s anisotropic diffusion [66]). Furthermore, each of these implementations has been shown to be an equivalent form of the same filter [4] [90].

The criticism with both the linear and nonlinear filtering techniques described above is that they assume an additive white noise model of the spurious data within the image, such that the observed intensity \( u(x) \) at location \( x \) is given by

\[
u(x) = v(x) + \eta(x; \sigma).
\]

where \( v(x) \) is the true visual image transmitted by the optical system of the camera and \( \eta(x; \sigma) \) is a zero mean i.i.d. normal random variable with standard deviation \( \sigma \). Under this assumption the local filters attempt to reconstruct \( v(x) \) by computing an average over the local neighbourhood \( B_x \), which has the effect
of reducing the variance of the data in that region by a factor of \( \frac{1}{\sqrt{|B_\mathbf{x}|}} \). While this may seem a good approach, it has the effect that the filtered value at point \( \mathbf{x} \), namely \( \bar{u}(\mathbf{x}) \) approaches the average value of \( v \) taken over the region \( B_\mathbf{x} \), which will only be the same as the true value \( v(\mathbf{x}) \) when \( v(\mathbf{y}) = c \) for all \( \mathbf{y} \in B_\mathbf{x} \) (i.e., for flat images).

The noise sources described above do not generally produce a signal that adheres to the Gaussian model and in certain imaging systems (for instance, ultrasound) multiplicative noise models are more appropriate. Where the additive assumption does hold but the noise distribution is non-Gaussian, the linear or nonlinear filters will perform sub-optimally. As described in Section 2.3 though, there are many different averaging functions. A robust alternative to the local convolution-based filters are those based on order statistics, which incorporate Spatial and Rank (SR) information into filtering methods. These filters exploit spatial correlations in the data while retaining the robustness of strict rank order methods. This class of filters includes the Median filter \([38]\), the Weighted Median (WM) filter \([14]\), Center Weighted Median (CWM) filter \([41]\) and spatial-rank order selection filters \([5]\). They have been shown to provide good performance at removing impulse and shot noise from images, but perform poorly on speckle noise, particularly when the noise variance is high \([77]\).

There are also numerous fuzzy-rule-base extensions to these filters, including Arakawa’s Fuzzy Median (FM) filter \([3]\), which computes a weighted arithmetic mean of the inputs and the output of a median filter. The weights are determined by fuzzy rules which are learned from training over a reference image. The Fuzzy Weighted Median (FWM) filter was proposed by Taguchi \([81]\) and this filter varies in behaviour from the weighted median filter to the weighted average filter by varying the setting of a fuzzy boolean function that has a continuous output in \([0,1]\). Further extensions of this approach include the generalized fuzzy weighted median filter \([82]\) and Chiang’s adaptive centre weighted median filter \([21]\).

Again, these filters are most effective at dealing with impulse noise and in particular, the fuzzy extensions are able to deal with the situation where the
2.7. AVERAGING PROBLEMS IN IMAGE PROCESSING

The various methods of image noise reduction all compute, in some manner, an average of a set of pixels and use this value as the denoised pixel value in the output image. Another important problem in image processing is that of image scaling, also called image resizing. Magnification, or increasing the sampling rate of an image, has been investigated from several approaches, including super-sampling [36] and interpolation [43] [84]. Jurio et al recently proposed a novel method that generates an interval for each pixel from a weighted aggregation of pixels in a local neighbourhood [37]. Given the interval and a convex combination of its bounds they produced magnified images with significantly better quality than existing standard methods. The $K_\alpha$ operator used in this work, due to Atanassov, is related to the OWA operators of dimension two and thus represents an averaging aggregation function [16].

Image reduction, or the decrease of the sample rate, has also been investigated from a variety of approaches. The earliest methods involved sub-sampling from disjoint local neighbourhoods, taking for example, the top-left pixel in each block. Necessarily, because the sample rate of the image is reduced, by the Nyquist-Shannon sampling theorem, the maximum possible frequency that can be represented in the image is also reduced. If the image contained frequencies above the Nyquist limit, then the spectrum of the sub-sampled image will be distorted and information will be lost. This effect is known as aliasing and in
certain conditions can introduce spurious visual patterns (Moiré patterns) to the image.

As with noise reduction, global methods for image reduction have been proposed; for example using the discrete fuzzy transform \[63\] \[64\] \[65\]. Other global reduction methods include Least-Squares resizing \[55\] and resizing using sub-band discrete cosine transform \[54\].

As image reduction is also a type of low pass filtering operation it can be implemented as an averaging operation. Recent focus in this direction has investigated local block reduction operators based on averaging aggregation functions \[58\] \[59\] \[60\], penalty function representations of averaging aggregation \[61\] and reduction of coloured images as aggregations on discrete product lattices \[9\].

The central premise in these recent works has been image reduction as an averaging aggregation problem on local image blocks, which requires that the averaging function be monotonic. It is well known that monotonic averages are not robust to noise or outliers in the data and thus further research is required to establish the types of averaging functions that perform well on such problems in the presence of noise. It is within the context of this most recent literature on aggregation methods for image reduction that this thesis investigates non-monotonic averaging functions with applications to image processing. In particular, within this work, no assumptions will be made that the noise model is Gaussian, nor an additive process.

There are several assumptions though that seem reasonable in any image processing problem. Consider an operation applied to a local neighbourhood of an image to produce a representative value. It is reasonable to expect that this value lies within the intensity bounds of the subset, as this will ensure that processing an image will not arbitrarily brighten or darken it. Similarly, when processing an image of constant intensity (a 'flat' image), the output result of the operation applied over the whole image should be the same constant valued image. Finally, some kind of monotonicity seems reasonable; if the image is lightened uniformly then an operation over a subset should reasonably produce an increased value, or at least should not return a decreased value compared to the non-lightened
image. However, shift invariance seems too strong a condition, particularly if the additive noise assumption is surrendered, as does strict monotonicity.

In the next chapter the issue of monotonicity is addressed and a new class of weakly monotonic functions is presented. This class encompasses many means thought previously to be purely non-monotonic and conditions for weak monotonicity of these functions are presented.

2.8. Conclusions

The formal definition of aggregation functions requires monotonicity with respect to all arguments, which is an important constraint relevant to certain decision making contexts. Specifically, there are many situations in which an increase in one or more decision variables should not cause a decrease in the aggregate criterion, or output of the decision process. Monotonicity models human preferences in these contexts quite well. However, monotonic means are easily corrupted by outliers and thus in many practical applications, robust estimators such as the median, or the trimmed means (ordered weight averages) are used.

There are though many non-monotonic means that are particularly robust to outliers and which have been used widely in practical applications, receiving significant consideration in the research literature. In robust statistics for example, the robust estimators of location estimate the central tendency of the input data, either directly ignoring or discounting those inputs that lie far from this central measure. The density-based averages applied to stream processing of data are a recent extension of this idea. In image filtering, the spatial-tonal filters discount the contribution of pixel intensities as a nonlinear function of inter-pixel distance (in both spatial and tonal coordinates), effectively discounting pixels that are dissimilar to that being filtered. Unfortunately, none of these means are considered aggregation functions as they explicitly lack monotonicity and thus have not been studied by other researchers using the methods and knowledge stemming from aggregation function theory.
This thesis contributes to both the field of aggregation functions and non-monotonic means by defining a new class of functions having directional monotonicity in the direction \((1, 1, \ldots, 1)\), which is exactly the property required for robustness to outliers. The directional monotonicity property, proposed and studied in the next Chapter and therein called *weak monotonicity*, has an appealing interpretation in the image processing context: when the image is lightened or darkened uniformly, the relative ordering of pixel intensities should remain the same, as should the relative ordering of the average of any subset of these pixels. However, it is not essential that the average of all lightened (or darkened) pixel intensities increases (or decreases) by the same constant amount applied to the inputs, as would be implied by shift-invariance (stability to translation).

In the next chapter the properties of weakly monotonic functions are studied and applied to generate new results for existing non-monotonic means. Sufficient conditions for weak monotonicity of several important classes of means are also established and weak monotonicity is used to generalise several classes of non-monotonic averaging functions, including the density based means and the spatial-tonal filters. In the subsequent chapters weakly monotonic averages are evaluated in image reduction problems to assess their capability to preserve fine scale image detail while reducing unwanted noise and outliers and thus to support the claims that weak monotonicity is a desirable property of averaging functions applied to image processing problems expressed as a local average over image blocks.
CHAPTER 3

Weakly Monotonic Averaging Functions
3.1. Introduction

There are two fundamental motivations for the line of research described in this chapter. While the formal theories of aggregation functions provide a sound theoretical basis for their application in many varied domains, the constraint of monotonicity limits their utility. Within decision making problems, monotonicity with respect to all arguments has an important interpretation: an increase in the input of one or more criterion should not lead to the decrease of the overall score or utility of the decision option. However, in image processing (and many other domains), an increase in one input value above its neighbours may be due to noise or corruption and should not necessarily increase the intensity value that represents that region. Accordingly, the non-monotonic averaging functions used in such domains do not fit within the established theories regarding aggregation functions and they are typically dealt with from the signal processing or robust statistics perspectives.

As evidenced in Chapter 2 there exist many means that are not generally monotonic and thus not aggregation functions, with the mode, Gini Means, Lehmer means, Bajraktarevic means and mixture functions being particularly well known cases. It is highly desirable to have a formal mathematical framework for averaging functions that encompasses non-monotonic means and places them in context with the existing monotonic aggregation functions. Having a unifying framework will enable researchers and practitioners alike to further the understanding of aggregation problems and to develop, from a sound mathematical foundation, practical solutions to real world problems.

The second motivation is that there are many real world applications in which non-monotonic means have been shown to provide good aggregate values commensurate with the objectives of the aggregation. Perhaps the most significant example applications are those in the filtering and smoothing of images and signals, where averaging is the strategy of choice to reduce noise. However, the bulk of literature studying the nonlinear filters does not investigate their properties from the perspective of averaging functions and instead concentrates on
the proposal and evaluation of new functions that are often optimised for a given application or domain.

This chapter aims to extend the understanding of non-monotonic means and draw many existing means into a common framework with the averaging aggregation functions. This is achieved by relaxing the monotonicity condition for aggregation functions and presenting the class of weakly monotonic averaging functions in Section 3.2. It will be shown, through extensive examples and analysis, that many means appearing in the literature, that were previously thought to be purely non-monotonic, are in fact weakly monotonic.

Weak monotonicity is also exactly the property required for robust averages in the presence of noise and corruption of the input data. Understanding this new class of functions is vital to the creation of novel averaging functions suitable for robust averaging and thus, this chapter also contributes important results regarding composition and transformation of weakly monotonic averaging functions. In Section 3.3 the robust estimators of location are studied and proven to be weakly monotonic, even under certain classes of nonlinear transforms. Several sub-classes of the Bajraktarevic means are also analysed to provide sufficient conditions for weak monotonicity and important results regarding mixture functions illuminate why the spatial-tonal filters are robust, weakly monotonic averages.

In Section 3.4 the penalty-based mode-like averages proposed by Beliakov are examined and several new results are provided proving their averaging behaviour and weak monotonicity. Finally, in Section 3.5, the recently proposed density-based averages are investigated and their weak monotonicity is established. Using weak monotonicity as framework, generalisations of this class of functions are given that improve their robustness to outliers and permit many new density based averaging functions to be formulated. That weak monotonicity can be used to better understand and extend established as well as recently proposed classes of averaging functions lends considerable weight to significance of this concept within the broader mathematical field concerned with averaging functions and applications. The contents of this chapter have been published in the following peer-reviewed articles:
3.2. WEAKLY MONOTONIC FUNCTIONS

3.2.1. Definition. The definition of weak monotonicity proposed herein is prompted by applications and intuition, which suggest that it is reasonable to expect that a representative value of the inputs does not decrease if all inputs are increased by the same amount (or shifted uniformly). In this situation the relative positions of the inputs are not changed and hence it is reasonable to expect that their representative value changes if not by the same amount, then at least in the same direction. However, there are many applications in which a sufficient increase in one value should lead to a decrease in the representative value. This would occur when a value shifts from being considered a member of the principal cluster within the input data, to being considered an outlier of this cluster. For example, raising a pixel value slightly above neighbouring values should lead to an increase in the average value for that neighbourhood. However, further raising the value suggests that the pixel is a noisy outlier and
should be ignored or weighted down, thus decreasing the average value. Thus, it is accepted that an increase in one or more variables may lead to a decrease in the representative value (or vice versa). A formal definition of a function that conveys these properties is as follows.

**Definition 70.** A function $F$ is called **weakly monotonic** increasing (or directionally monotonic in the direction $1$) if $F(x+a1) \geq F(x)$ for any $a > 0$, $1 = (1, ..., 1)$, such that $x, x + a1 \in \mathbb{I}^n$.

**Remark 2.** If $F$ is directionally differentiable in its domain then weak monotonicity is equivalent to non-negativity of the directional derivative $D_1(F)(x) \geq 0$. Contrasting this against the directional derivatives of a monotonic function (Remark 1), it is clear that weakly monotonic functions are indeed a superset of monotonic functions, since they require monotonicity only in one direction.

**Remark 3.** Evidently monotonicity implies weak monotonicity, hence all aggregation functions are weakly monotonic. By Definition 14 all shift-invariant functions are also weakly monotonic. It is self evident that weakly monotonic increasing functions form a cone in the linear vector space of weakly monotonic (increasing or decreasing) functions.

Let us now establish some useful properties of weakly monotonic aggregations.

**3.2.2. Composition of Weakly Monotonic Functions.** Consider the function $F : \mathbb{I}^n \to \mathbb{I}$ formed by the composition $F(x) = A(B_1(x), B_2(x))$, where $A, B_1$ and $B_2$ are means.

**Proposition 5.** If $A$ is monotonic and $B_1, B_2$ are weakly monotonic, then $F$ is weakly monotonic.

**Proof.** By weak monotonicity $B_i(x+a1) \geq B_i(x)$ implies that $\exists \delta_i \geq 0$ such that $B_i(x+a1) = B_i(x) + \delta_i$, with $x, x + a1 \in \mathbb{I}^n$. Thus $F(x+a1) = A(b_1 + \delta_1, b_2 + \delta_2)$, where $b_i = B_i(x)$. The monotonicity of $A$ ensures that $A(b_1 + \delta_1, b_2 + \delta_2) \preceq A(b_1, b_2)$ and hence $F(x+a1) \geq F(x)$ and $F$ is weakly monotonic. \qed
By trivial extension, since all monotonic functions are also weakly monotonic, then if either of \( B_1 \) or \( B_2 \) is monotonic, then \( F \) is again weakly monotonic.

**Proposition 6.** If \( A \) is weakly monotonic and \( B_1, B_2 \) are shift-invariant, then \( F \) is weakly monotonic.

**Proof.** Shift-invariance implies that \( \exists a : B_i(x + a1) = B_i(x) + a \), with \( x, x + a1 \in \mathbb{I}^n \). Thus \( F(x + a1) = A(b_1 + a, b_2 + a) \), where \( b_i = B_i(x) \). The weak monotonicity of \( A \) ensures that \( A(b_1 + a, b_2 + a) \not\leq A(b_1, b_2) \) and hence \( F(x + a1) \geq F(x) \) and \( F \) is weakly monotonic. \( \square \)

3.2.3. **Duality and \( \phi \)-transform.** Standard monotonicity is preserved under \( \phi \)-transform, when \( \phi \) is a strictly monotonic function. Consider now functions of the form

\[
\phi(x) = (\phi(x_1), \phi(x_2), ..., \phi(x_n))
\]

with \( \phi \) any twice differentiable and invertible function.

**Proposition 7.** If \( A : \mathbb{I}^n \to \mathbb{I}^n \) is weakly monotonic and \( \phi : \mathbb{I}^n \to \mathbb{I} \) is a linear function then the \( \phi \)-transform \( A_\phi(x) = F(x) = \phi^{-1}(A(\phi(x))) \) is weakly monotonic.

**Proof.** \( \phi(x) = \alpha x + \beta \) and hence \( \phi(x + a) = \alpha(x + a) + \beta = \alpha x + \beta + \alpha a = \phi(x) + c \). Hence

\[
F(x + a1) = \phi^{-1}(A(\phi(x_1 + a), ..., \phi(x_n + a)))
\]

\[
= \phi^{-1}(A(\phi(x) + c1))
\]

\[
= \frac{A(\phi(x) + c1) - \beta}{\alpha}
\]

\[
\geq \frac{A(\phi(x)) - \beta}{\alpha}
\]

\[
= \phi^{-1}(A(\phi(x)))
\]

by weak monotonicity of \( A \). Hence \( F(x + a1) \geq F(x) \) and \( F \) is weakly monotonic. \( \square \)
If \( \phi(x) \) is a standard negation then the dual of \( A \) is a weakly monotonic function.

**Corollary 2.** The dual \( A^d \) of a weakly monotonic function \( A \) is weakly monotonic under standard negation.

For the \( \phi^- \) transform, if \( \phi \) is nonlinear then \( F \) may or may not be weakly monotonic for all \( x \), which can be observed by example.

**Example 1.** Take \( x = (1, 8, 16, 35, 47) \) and \( \phi(t) = \sqrt{t} \), then \( \phi(x) = (1, 2\sqrt{2}, 4, \sqrt{35}, \sqrt{47}) \) and \( \phi(x + 1) = (\sqrt{2}, 3, \sqrt{17}, 6, \sqrt{48}) \). If \( A \) is the Shorth (we prove the weak monotonicity of the Shorth in Section 3.3) then \( A(\phi(x)) = 5.61 \) and \( A(\phi(x+1)) = 2.84 \). As \( \phi^{-1}(t) = t^2 \) clearly \( 5.61^2 > 2.84^2 \) and \( F \) is not weakly monotonic.

Internal means are not necessarily weakly monotonic, as illustrated by the following example.

**Example 2.** Take \( x = (x_1, x_2) \in [0, 1]^2 \) and

\[
F(x) = \begin{cases} 
\min(x) & \text{if } x_1 + x_2 \geq 1 \\
\max(x) & \text{otherwise}
\end{cases}
\]

which is internal with values in the set \( \{\min(x), \max(x)\} \). Consider the points \( x = (\frac{1}{4}, 0) \) and \( y = (\frac{3}{4}, 0) \), then \( F(x) = \frac{1}{4} \) and \( F(y) = \frac{3}{4} \). It follows that \( F(x + \frac{1}{4}1) = \frac{1}{2} > F(x) \), however \( F(y + \frac{1}{4}1) = \frac{1}{4} < F(y) \). Hence this \( F \) is not weakly monotonic for all \( x \in \mathbb{R}^2 \).

Weak monotonicity is not preserved under all nonlinear transforms, although of course for specific functions this property will be preserved for some specific choices of \( \phi \).

**Proposition 8.** The only functions \( \phi \) which preserve weak monotonicity of all weakly monotonic functions are linear functions.

**Proof.** Consider the example of a shift-invariant and hence weakly monotonic function whose \( \phi^- \) transform is not weakly monotone in general. The function \( F(x, y) = D(y - x) + \frac{1+y}{2} \), where \( D \) is the Dirichlet function (which takes
values 0 at rational numbers and 1 at irrational numbers) is one such example. Taking generalised derivatives of $F$ gives

$$F_x = -D'(y - x) + \frac{1}{2} \quad \text{and} \quad F_y = D'(y - x) + \frac{1}{2},$$

from which it follows that $F_x + F_y = 1 \geq 0$. Now consider the $\phi$--transform of $F$ given by $F_\phi(x, y) = \phi^{-1}(F(\phi(x), \phi(y)))$ and the sum of its partial derivatives. It is sufficient to consider only the function $F(\phi(x), \phi(y))$ in which case it follows that

$$F(\phi(x), \phi(y))_x + F(\phi(x), \phi(y))_y = D'(\phi(y) - \phi(x))(\phi'(y) - \phi'(x)) + \frac{\phi'(x) + \phi'(y)}{2}.$$

The first term is zero only when $\phi'(x) = \phi'(y)$ for all $x, y$. That is, $\phi'(x)$ is constant and hence $\phi$ is a linear function. In all other cases the generalised derivative of $D$ takes values $+\infty$ and $-\infty$, and hence the sum of partial derivatives can be negative. \qed

Note that in Proposition 8 the product $F(x, y) = D(y - x) \cdot (x + y)$ (which is weakly monotonic but not shift invariant) could be used to obtain an analogous result.

It remains to be seen whether there are subclasses of weakly monotonic functions which preserve weak monotonicity under some non-linear $\phi$--transforms. To verify that there are such cases, consider the robust estimators of location (see Sections 2.5 and 3.3) that are calculated using the shortest contiguous sub-sample of $x$ containing at least half of the values. These estimators are shift-invariant and hence weakly monotonic. For clarity consider once again that the candidate sub-samples are the sets

$$X_k = \{x_j : j \in \{k, k + 1, ..., k + \left\lfloor \frac{n}{2} \right\rfloor \}, \ k = 1, ..., \left\lfloor \frac{n + 1}{2} \right\rfloor \}.$$
The length of each contiguous set is taken as \( \|X_k\| = \left| x_{k+\lfloor \frac{n}{2} \rfloor} - x_k \right| \) and thus the index of the shortest sub-sample is

\[
k^* = \arg \min_i \|X_i\|, \quad i = 1, \ldots, \left\lfloor \frac{n + 1}{2} \right\rfloor.
\]

Under the translation \( \bar{x} = x + a1 \) the length of each sub-sample is unaltered since

\[
\|\bar{X}_k\| = \left| \bar{x}_{k+\lfloor \frac{n}{2} \rfloor} - \bar{x}_k \right| = \left| (x_{k+\lfloor \frac{n}{2} \rfloor} + a) - (x_k + a) \right| = \left| x_{k+\lfloor \frac{n}{2} \rfloor} - x_k \right| = \|X_k\|
\]

and thus \( k^* \) remains the same. The following lemmas will be necessary to show that for some shift-invariant functions, such as the Shorth and LMS estimators, weak monotonicity is preserved under \( \phi \)-transforms with increasing invertible functions \( \phi \) such that \( \ln(\phi') \) is concave.

**Lemma 1.** Let \( \phi \) be any twice differentiable and invertible function. Then the function \( \phi(\phi^{-1}(x) + c) \) is concave if and only if: \( \ln(\phi') \) is concave when \( \phi' > 0 \); or, \( \ln(|\phi'|) \) is convex for \( \phi' < 0 \), for every \( c \in \mathbb{R}, c \geq 0 \) and \( x \in \mathbb{I}^n \).

**Proof.** Given the properties of \( \phi \) then

\[
(\phi^{-1})'(x) = \frac{1}{\phi'(\phi^{-1}(x))}, \quad \phi'(\phi^{-1}(x)) \neq 0
\]

\[
(\phi^{-1})''(x) = -\phi''(\phi^{-1}(x)) \cdot \frac{1}{[\phi'(\phi^{-1}(x))]^3}
\]

\[
g(x) = \phi(\phi^{-1}(x) + c)
\]

\[
g'(x) = \phi'(\phi^{-1}(x) + c) \cdot \frac{1}{\phi'(\phi^{-1}(x))}
\]

\[
g''(x) = \frac{1}{[\phi'(\phi^{-1}(x))]^3} \left[ \phi''(\phi^{-1}(x) + c) \phi'(\phi^{-1}(x)) \phi'(\phi^{-1}(x)) - \phi''(\phi^{-1}(x)) \phi'(\phi^{-1}(x) + c) \right].
\]

Thus \( g''(x) < 0 \) occurs under the following two circumstances:

\[
\phi''(\phi^{-1}(x) + c) \phi'(\phi^{-1}(x)) < \phi''(\phi^{-1}(x)) \phi'(\phi^{-1}(x) + c),
\]
3.2. WEAKLY MONOTONIC FUNCTIONS

\[ \phi' (\phi^{-1} (x)) > 0, \]

or

\[ \phi'' (\phi^{-1} (x)) \phi' (\phi^{-1} (x) + c) < \phi'' (\phi^{-1} (x) + c) \phi' (\phi^{-1} (x)), \]

\[ \phi' (\phi^{-1} (x)) < 0. \]

In the first case it follows that

\[ \phi'' (\phi^{-1} (x) + c) \phi' (\phi^{-1} (x) + c) < \phi'' (\phi^{-1} (x)) \phi' (\phi^{-1} (x)) > 0. \]

Hence \( \phi'' / \phi' = \left[ \ln \phi' \right]' \) is a decreasing function and thus \( \ln \phi' \) is concave. In the second case,

\[ \phi'' (\phi^{-1} (x)) < \phi'' (\phi^{-1} (x) + c) \phi' (\phi^{-1} (x) + c), \phi' (\phi^{-1} (x)) < 0. \]

and \( \phi'' / \phi' \) is increasing. Thus \( \phi'' / \phi' = \left[ \ln (-\phi') \right]' \) is an increasing function and thus \( \ln (-\phi') \) is convex. \( \square \)

**Lemma 2.** Let \( x \) be ordered such that \( x_i \leq x_j \) for \( i < j \) and let \( X_i \) denote the subset \( \{ x_i, \ldots, x_{i+h} \} \) for some fixed \( h \). Let \( \Delta X_i = \| X_i \| = | x_{i+h} - x_i | \) denote the length of the interval containing \( X_i \). If \( \phi \) is a concave increasing function then for \( i < j \), \( \Delta X_j \leq \Delta X_i \) implies that \( \Delta \phi (X_i) \geq \Delta \phi (X_j) \), where \( \Delta \phi (X_i) = | \phi (x_{i+h}) - \phi (x_i) | \).

**Proof.** For a concave function \( \phi \) the following holds: \( \phi (b) - \phi (a) \geq \phi (b + w) - \phi (a + w) \) for \( b > a \) and \( w > 0 \). Hence, let \( x_i = a, x_{i+k} = b, x_{j+k} = b + w \) and \( c = a + w = x_j - \delta \). Then

\[ \Delta \phi (X_i) = | \phi (b) - \phi (a) | \geq | \phi (b + w) - \phi (a + w) | \]

\[ = | \phi (x_{j+k}) - \phi (x_j - \delta) | . \]


3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

Table 1. Selected $\phi-$transforms that preserve weak monotonicity of robust estimators of location.

<table>
<thead>
<tr>
<th>$\phi(x)$</th>
<th>Increasing for</th>
<th>$y = \ln \phi'(x)$</th>
<th>$y''$</th>
<th>Concave</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^{ax}$</td>
<td>$a &gt; 0$</td>
<td>$ln a + ax$</td>
<td>0</td>
<td>$x \in \mathbb{R}$</td>
</tr>
<tr>
<td>$e^{ax^2}$</td>
<td>$a &gt; 0$</td>
<td>$ln 2a + ln x + ax^2, x &gt; 0$</td>
<td>$2</td>
<td>a</td>
</tr>
<tr>
<td>$e^{\frac{(x-a)^2}{2b^2}}$</td>
<td>all $a, b \in \mathbb{R}, x \leq a$</td>
<td>$ln \left(-\frac{2}{b^2}(x-a)\right) - \frac{(x-a)^2}{b^2}$</td>
<td>$\frac{-1}{x-a} - \frac{2}{b^2}$</td>
<td>$x \in (0, \frac{1}{\sqrt{2a}}]$</td>
</tr>
<tr>
<td>$x^p$</td>
<td>$x \geq 0, p &gt; 0$</td>
<td>$ln p + (p-1)ln x$</td>
<td>$\frac{x^p}{p^2}$</td>
<td>$p &gt; 1, x \geq 0$</td>
</tr>
<tr>
<td>$\frac{e^x + e^{-x}}{2}$</td>
<td>$x \geq 0$</td>
<td>$ln \frac{1}{2} + ln (e^x - e^{-x})$</td>
<td>$1 - \frac{(e^x + e^{-x})^2}{(e^x - e^{-x})^2}$</td>
<td>$x &gt; 0$</td>
</tr>
</tbody>
</table>

From $\Delta X_i \geq \Delta X_j$ it follows that $\delta \geq 0$ and given the monotonicity of $\phi$ then $\phi(x_j - \delta) \leq \phi(x_j)$ and hence $\Delta \phi(X_i) \geq \Delta \phi(X_j)$. □

Proposition 9. Consider $A$, a robust estimator of location based on the shortest contiguous half of the data and let $\phi$ be a twice differentiable strictly increasing invertible function such that $ln \phi'$ is concave. Then $A(x + a \mathbf{1}) \geq A(x)$ implies $\phi^{-1}(A(\phi(x + a \mathbf{1}))) \geq \phi^{-1}(A(\phi(x)))$.

Proof. Denote by $y = \phi(x)$ and $x = \phi^{-1}(y)$ (with functions applied component-wise) and we have $\phi' > 0$. We must show that

$$A(\psi_c(y)) = A(\phi(\phi^{-1}(y) + c)) \geq A(y).$$

By Lemma 1 function $\psi_c$ is concave. By Lemma 2 it holds that $\Delta X_i \leq \Delta X_j \Rightarrow \Delta \psi_c(X_i) \leq \Delta \psi_c(X_j)$ for $i \geq j$. Hence the starting index of the shortest half cannot decrease after the transformation $\psi_c$ and the result follows directly. □

Corollary 3. Let $A$ be a robust estimator of location based on the shortest contiguous half of the data. Let $\phi$ be twice differentiable strictly increasing function such that $ln \phi'$ is convex. Then the $\phi$-dual of $A$ is weakly monotonic.

Proposition 9 serves as a simple test to determine which $\phi$-transforms preserve weak monotonicity of averages such as the Shorth and LMS estimator. For example $\phi(x) = e^x$ does preserve weak monotonicity and $\phi(x) = \ln(x)$ does not. Table 1 shows some cases of functions $\phi$ that preserve weak monotonicity robust estimators under $\phi-$transform.
Table 2. Selected $\phi-$transforms that do not preserve weak monotonicity of robust estimators of location.

<table>
<thead>
<tr>
<th>$\phi(x)$</th>
<th>Increasing for</th>
<th>$y = \ln \phi'(x)$</th>
<th>$y''$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ln x$</td>
<td>$x &gt; 0$</td>
<td>$- \ln x$</td>
<td>(\frac{1}{x^2})</td>
</tr>
<tr>
<td>$\frac{e^x - e^{-x}}{2}$</td>
<td>$x \in \mathbb{R}$</td>
<td>$\ln \frac{1}{2} + \ln (e^x + e^{-x})$</td>
<td>$1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2}$</td>
</tr>
<tr>
<td>$\tan x$</td>
<td>$x \in \mathbb{R}/{(2n + 1)\frac{\pi}{2} : n \in \mathbb{N}}$</td>
<td>$\ln (1 + \tan^2 x)$</td>
<td>$2(1 + \tan^2 x)$</td>
</tr>
</tbody>
</table>

Table 2 shows several example functions that do not preserve weak monotonicity of the robust estimators of location under $\phi-$transform.

### 3.3. New Results on Weakly Monotonic Means

The means presented in Section 2.5 are, in general, not monotonic. Sufficient conditions for the monotonicity of some of these functions have been reported in the literature. In this section examples of weakly monotonic, but not necessarily monotonic averaging functions are presented and sufficient conditions for the weak monotonicity of several important classes of non-monotonic means are established.

While averaging behaviour of monotonic functions follows directly from their idempotence, this is not the case for non-monotonic functions. Thus, averaging behaviour must be established for each such function. For some of the functions below averaging behaviour has not been formally established previously and thus, where necessary, this will be performed. The following result will be useful for this purpose.

**Proposition 10.** Let $A_w$ be an averaging function with constant weights. Then the function $A_{u(x)}$ obtained from $A_w$ by allowing weights to be dependent on $x$ and enforcing normalisation of weights, such that $w_i = \frac{u_i(x)}{\sum u_i(x)}$ with $u_i \geq 0$ for all $x \in \mathbb{I}$, is bounded by $\min(x) \leq A_{u(x)}(x) \leq \max(x)$.

**Proof.** For every fixed $x$ the weights $w_i(x) = \frac{u_i(x)}{\sum u_i(x)}$ are fixed, non-negative and add to one. Then the value of $A_w(x)$ with such weights is bounded by the minimum and maximum, because $A_w$ is averaging. \(\square\)
3.3.1. **Penalty-based Averaging Functions.** Given that all averaging functions can be expressed as penalty-based functions, the following results permit the easy identification of weakly monotonic averaging functions from their penalty form. These results will also simplify several proofs later in this chapter. Note that herein Condition 3 of Definition (57) has been relaxed so that the averages are not necessarily monotonic functions.

**Theorem 3.** Let $F$ be a penalty-based averaging function defined by the minimisation of the penalty

$$
\mathcal{P}(x, y) = \sum_{i=1}^{n} w_i(x)D(x_i - y),
$$

where functions $w_i$ are constant under the translation $x \rightarrow x + c1$ and $D$ some function. Then $F$ is shift-invariant and hence weakly monotonic.

**Proof.** Since under the translation $x \rightarrow x + c1$ all $w_i$ remain constant then the minimum of $\mathcal{P}(x + a1, y)$ will be achieved at $y = \mu + a$, where $\mu$ is the minimiser of $\mathcal{P}(x, y)$. Properties of $D$ do not matter here, as its value is not affected.

**Remark 4.** This proof obviously holds for any shift-invariant univariate function $D(h(x_i) - h(y)) : \mathbb{I}^2 \rightarrow \mathbb{R}$ having a minimum $D(0)$ that measures the dissimilarity between the inputs $x_i$ and the value $y$.

**Theorem 4.** Let $f : \mathbb{I}^n \rightarrow \mathbb{I}$ be a shift-invariant function and $g$ be a function. Let $F$ be a penalty based averaging function with penalty $\mathcal{P}$ depending on the terms $g(x_i - f(x)) (x_i - y)^2$. Then $F$ is shift-invariant and hence weakly monotonic.

**Proof.** Let

$$
\mu = \arg \min_y P \left( g (x_1 - f(x)) (x_1 - y)^2, ..., g (x_n - f(x)) (x_n - y)^2 \right).
$$
Then
\[ \arg \min_y P(x + a1, y) = \arg \min_y P \left( g(x_1 + a - f(x + a1)) (x_1 + a - y)^2, ... \right. \]
\[ \left. ..., g(x_n + a - f(x + a1)) (x_n + a - y)^2 \right) \]

By the shift-invariance of \( f \)
\[ g(x_i + a - f(x + a1)) = g(x_i + a - f(x) + a) \]
\[ = g(x_i - f(x)) \]

and thus
\[ \arg \min_y P(x + a1, y) = \arg \min_y P \left( g(x_1 - f(x)) (x_1 + a - y)^2, ... \right. \]
\[ \left. ..., g(x_n - f(x)) (x_n + a - y)^2 \right) \]
\[ = \mu + a. \]

Again it is not necessary to restrict consideration to penalty functions with terms depending on \((x_i - y)^2\). Functions \( D \) that depend on the differences \( x_i - y \) with minimum \( D(0) \) will satisfy the above proof and satisfy the conditions on \( P \) with regards to the existence of solutions to (13). In particular, Huber type functions used in robust regression can replace the squares of the differences.

Further results regarding mixture functions are provided in Subsection 3.3.5 below. An extension of this result to weights that depend on all inputs will be useful when investigating density based means in Section 3.5.

3.3.2. Estimators of Location. Several robust estimators of location were presented in Section 2.5 and their definitions are restated below for clarity. The first of these and perhaps the most widely used estimator of location is the mode, being the most frequent input.
3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

Example 3. Mode: The mode (Defn. 64) is the minimiser of the penalty function

\[ P(x, y) = \sum_{i=1}^{n} p(x_i, y) \quad \text{where} \quad p(x_i, y) = \begin{cases} 0 & x_i = y \\ 1 & \text{otherwise} \end{cases}. \]

It follows that \( F(x + a1) = \arg\min_y P(x + a1, y) = \arg\min_y \sum_{i=1}^{n} p(x_i + a, y) \), which is minimised for the value \( y = F(x) + a \). Hence, \( F(x + a1) = F(x) + a \) and thus the mode is shift-invariant. By Definition 14 the mode is weakly monotonic.

The Least Trimmed Squares estimator (\cite{73}) rejects up to 50\% of the data values as outliers and minimises the squared residual using the remaining data.

Example 4. Least Trimmed Squares (LTS): The LTS (Defn. 67) uses the penalty function

\[ P(x, y) = \sum_{i=1}^{h} r_{(i)}^2, \]

where \( r_{(i)} = S_i(x) \), \( r_k = x_k - y \) and \( h = \lfloor \frac{n}{2} \rfloor + 1 \). If \( \sigma \) is the order permutation of \( \{1, \ldots, n\} \) such that \( r_{\sigma} = r_{\uparrow} \), then the minima of \( P \) occur when \( P_y = -2 \sum_{i=1}^{h} (x_{\sigma(i)} - y) = 0 \), which implies that the minimum value is \( \mu = \frac{1}{h} \sum_{i=1}^{h} x_{\sigma(i)} \).

Since \( S_k(x) \) is shift-invariant then \( S_i(x + a1) = r_{\sigma(i)} + a \) and thus

\[ P(x + a1, y) = \sum_{i=1}^{h} v_{\sigma(i)}^2, \]

where \( v_k = ((x_k + a) - y) \). It follows that the value \( y \) that minimises \( P(x + a1, y) \) is \( y = \mu + a \), hence the LTS is shift-invariant and thus weakly monotonic.

The remaining estimators of location presented compute their value using the shortest contiguous sub-sample of \( x \) containing at least half of the values. Recall, from Section 2.5, that the candidate sub-samples are the sets \( X_k = \{x_j : j \in \{k, k+1, \ldots, k + \lfloor \frac{n}{2} \rfloor \} \}, \ k = 1, \ldots, \lfloor \frac{n+1}{2} \rfloor \). The length of each set is taken as
\[ \|X_k\| = \left|x_{k + \lfloor \frac{n}{2} \rfloor} - x_k\right| \] and thus the index of the shortest sub-sample is

\[ k^* = \arg \min_i \|X_i\|, \ i = 1, ..., \left\lfloor \frac{n + 1}{2} \right\rfloor. \]

Under the translation \( \bar{x} = x + a1 \) the length of each sub-sample is unaltered since

\[ \|\bar{X}_k\| = \left|\bar{x}_{k + \lfloor \frac{n}{2} \rfloor} - \bar{x}_k\right| = \left|(x_{k + \lfloor \frac{n}{2} \rfloor} + a) - (x_k + a)\right| = \left|x_{k + \lfloor \frac{n}{2} \rfloor} - x_k\right| = \|X_k\| \] and thus \( k^* \) remains the same.

Consider now the Least Median of Squares estimator [72], which is the midpoint of \( X_{k^*} \).

**Example 5. Least Median of Squares (LMS):** The LMS (Defn. 66) can be computed by minimisation of the penalty function

\[ P(x, y) = \text{median}\{(x_i - y)^2 | y \in I, x_i \in X_{k^*}\}\]

The value \( y \) minimises the penalty \( P(x + a1, y) \), given by

\[ \min_y P(x + a1, y) = \min_y \text{median}\{(x_j + a - y)^2 | y \in I, x_j \in X_{k^*}\} = P(x, \mu), \]

is clearly \( y = \mu + a \). Hence, \( F(x + a1) = F(x) + a \) and the LMS is shift-invariant and weakly monotonic.

The Shorth ( [1]) is the arithmetic mean of \( X_{k^*} \).

**Example 6. Shorth:** The Shorth (Defn. 65) is given by

\[ F(x) = \frac{1}{h} \sum_{i=1}^{h} x_i, \ x_i \in X_{k^*}, \ h = \left\lfloor \frac{n}{2} \right\rfloor + 1. \]

Since the set \( X_{k^*} \) is unaltered under translation and the arithmetic mean is shift-invariant, then the shorth is shift-invariant and hence weakly monotonic.

**Example 7. OWA Penalty Functions:** Penalty functions having the form

\[ P(x, y) = \sum_{i=1}^{n} w_i S_i \left((x - y1)^2\right), \]
where \( S_i(x) \) is the order statistic (Defn. 27) define regression operators, \( F(x) \) \[97\]. Consider the following results dependent on the constant weight vector \( \Delta = (w_1, ..., w_n) \).

1. \( \Delta = (1) \) generates Least Squares regression and \( F \) is monotonic and hence weakly monotonic;
2. \( \Delta = (0, ..., 0, 1) \) generates Chebyshev regression and \( F \) is monotonic and hence weakly monotonic;
3. Since all the terms \( S_i((x - y)^2) \) are constant under transformation \((x, y) \rightarrow (x + a, y + a) \) (cf Theorem 4), the OWA regression operators are shift-invariant for any choice of the weight vector \( \Delta \).
4. For \( \Delta = \begin{cases} (0, ..., 0, 1, 0, 1/2, 1/2, 0, ..., 0) & n = 2k \text{ is even} \\ (0, ..., 0, 1, 0, ..., 0) & n = 2k - 1 \text{ is odd} \end{cases} \) then \( F \) is the Least Median of Squares operator and hence shift invariant and weakly monotonic; and
5. For \( \Delta = (1, ..., 1, 0, ..., 0), h = \left\lfloor \frac{n}{2} \right\rfloor + 1 \) then \( F \) is the Least Trimmed Squares operator and hence is shift invariant and weakly monotonic.

In the cases (3)-(5) above, the OWA regression operators are not monotonic.

### 3.3.3. The Lehmer Mean

As mentioned in Section 2, a special case of the Gini means (with \( p = 1 \)) are the Lehmer means, which are generally not monotonic. Lehmer means are mixture functions with weighting function \( w(t) = t^q \), which is neither increasing for all \( q \in \mathbb{R} \) nor shift-invariant. Mesiar has shown that under the constraint that \( w \) is increasing and differentiable, if \( w(x) \geq w'(x) \cdot (b - x), x \in [a, b] = \mathbb{I}, \) then \( M_\Delta \) is an aggregation function and hence monotonic (and by extension, also weakly monotonic) \[51\]. Additionally, \( M_\Delta \) is invariant to scaling of the weighting functions (I.e., \( M_{\alpha \Delta} = M_\Delta \forall \alpha \in \mathbb{R} \setminus \{0\} \)). In \[50\], it was shown that the dual, \( M_{\Delta}^d \), of \( M_\Delta \) is generated by \( w(1 - x) \).

Herein a sufficient condition for the weak monotonicity of the Lehmer means is presented. It will be useful to first establish the general properties of Lehmer means.
3.3. NEW RESULTS ON WEAKLY MONOTONIC MEANS

Theorem 5. The Lehmer mean \( L_q : [0, \infty)^n \to [0, \infty) \), given by

\[
L_q(x) = \frac{\sum_{i=1}^{n} x_i^{q+1}}{\sum_{i=1}^{n} x_i^q}, \quad q \in \mathbb{R}
\]

is

1. homogeneous;
2. monotonic (and linear) along the rays emanating from the origin;
3. averaging;
4. idempotent;
5. not generally monotonic in \( x \);
6. has neutral element 0 for \( q > 0 \); and,
7. has absorbing element 0 for \( q < 0 \).

Proof. Consider each of the following:

1. Homogeneous: Set \( x = \lambda u \) then

\[
L_q(x) = L_q(\lambda u) = \frac{\sum_{i=1}^{n} (\lambda u_i)^{q+1}}{\sum_{i=1}^{n} (\lambda u_i)^q} = \lambda \frac{\sum_{i=1}^{n} u_i^{q+1}}{\sum_{i=1}^{n} u_i^q} = \lambda L_q(u).
\]

Hence \( L_q \) is homogeneous with degree 1.

2. Monotonic (and linear) along the rays: Consider the generalised spherical coordinates (\[13\]) \( (r, \theta, \phi_1, ..., \phi_{n-2}) \), \( r \geq 0, 0 \leq \theta \leq 2\pi, 0 \leq \phi_i \leq \pi \) for the hypersphere \( S^{n-1} = \{ x \in \mathbb{R}^n : \| x \| = r \} \). We will restrict the angle variables so that \( x_i \in [0, \infty) \). The transformation to an orthonormal Euclidean basis \( E_n \) produces the vector \( x \) of length \( r \) having components
3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

\[ x_1 = r \cos(\phi_1) \]

\[ x_j = r \cos(\phi_j) \prod_{k=1}^{j-1} \sin(\phi_k), \quad j = 2, \ldots, n - 1 \]

\[ x_n = r \prod_{k=1}^{n-1} \sin(\phi_k) \]

where \( \phi_{n-1} = \theta \) and \( 0 \leq \phi_i \leq \pi/2 \), \( i = 1, \ldots, n - 1 \), \( r \geq 0 \). The Lehmer mean of the Euclidean vector \( x \) is therefore

\[
L_q(x) = \frac{[r \cos(\phi_1)]^{q+1} + \sum_{j=2}^{n-1} [r \cos(\phi_j) \prod_{k=1}^{j-1} \sin(\phi_k)]^{q+1} + [r \prod_{k=1}^{n-1} \sin(\phi_k)]^{q+1}}{[r \cos(\phi_1)]^q + \sum_{j=2}^{n-1} [r \cos(\phi_j) \prod_{k=1}^{j-1} \sin(\phi_k)]^q + [r \prod_{k=1}^{n-1} \sin(\phi_k)]^q}
\]

\[
= r \left( \frac{[\cos(\phi_1)]^{q+1} + \sum_{j=2}^{n-1} [\cos(\phi_j) \prod_{k=1}^{j-1} \sin(\phi_k)]^{q+1} + \prod_{k=1}^{n-1} \sin(\phi_k)]^{q+1}}{[\cos(\phi_1)]^q + \sum_{j=2}^{n-1} [\cos(\phi_j) \prod_{k=1}^{j-1} \sin(\phi_k)]^q + \prod_{k=1}^{n-1} \sin(\phi_k)]^q} \right)
\]

\[
= rf(\phi_1, \ldots, \phi_{n-1})
\]

Along rays emanating from the origin each \( f(\cdot) \) is constant and hence \( L_q(x) = \alpha_0 r \) is linear.

3. Averaging: Let \( x_{\sigma} = x_\downarrow \) and take \( a = x_1 \) and \( b = x_m \) denote the value of the largest and the smallest non-zero elements of \( x \) respectively. By homogeneity

\[
L_q(x) = a \frac{1 + \sum_{i=2}^{n} \left( \frac{x_i}{a} \right)^q}{1 + \sum_{i=2}^{n} \left( \frac{x_i}{a} \right)^q} = a \frac{1 + \alpha}{1 + \beta}
\]
Since \( x(i) \leq a \) for all \( i = 1, \ldots, n \) then \( \alpha \leq \beta \). Hence \( L_q(x) \leq a \). Similarly,

\[
L_q(x) = b \frac{1 + \sum_{i=1}^{m-1} \left( \frac{x(i)}{b} \right)^{q+1}}{1 + \sum_{i=1}^{m-1} \left( \frac{x(i)}{b} \right)^q}
\]

Since \( x(i) \geq b \) for all \( i = 1, \ldots, m-1 \) and \( x(i) = 0 \) for all \( i = m+1, \ldots, n \) then \( \delta \leq \gamma \).

Hence \( L_q(x) \geq b \). Thus, \( \min(x) \leq L_q(x) \leq \max(x) \) and \( L_q(x) \) is averaging.

4. **Idempotent**: For any vector \( x = (t, t, \ldots, t) \) we have that

\[
L_q(x) = \frac{\sum_{i=1}^{n} t^q}{\sum_{i=1}^{n} t^q} = t
\]

and hence \( L_q \) is idempotent.

5. **Not generally monotonic in \( x \)**: Take \( x = (1, 0) \) and \( y = (1, 1/2) \), then for \( q > 0 \), \( L_q(x) = 1 \) and \( L_q(y) = \frac{1+(1/2)^{q+1}}{1+(1/2)^q} = \frac{2^{q+1}+1}{2^{q+2}} < 1 \). Thus \( x < y \) and \( L_q(x) > L_q(y) \), hence \( L_q(x) \) is not generally monotonic in \( x \) for all \( q \in \mathbb{R} \).

6. **Has neutral element of 0 for \( q > 0 \)**: Consider \( x = (a, 0) \) then \( L_q(x) = \lim_{x_2 \to 0^+} \frac{a^{q+1} + x_2^{q+1}}{a^{q} + x_2^{q}} = a \) for \( q > 0 \).

7. **Has absorbing element of 0 for \( q < 0 \)**: Consider \( x = (a, 0) \) then

\[
L_q(x) = \lim_{x_2 \to 0^+} L_q(1, x_2) = \lim_{x_2 \to 0^+} \frac{a+x_2^{q+1}}{a+x_2^q} = \frac{a^2+x_2^2}{a^{q+1}} = 0.
\]

The sufficient condition for the weak monotonicity of Lehmer means depends on both the exponent \( q \) and the number of arguments \( n \). A relation between these two quantities is given by the following theorem.
Theorem 6. The Lehmer mean of \( n \) arguments,

\[
L_q(x) = \frac{\sum_{i=1}^{n} x_i^{q+1}}{\sum_{i=1}^{n} x_i^q}, \quad q \in \mathbb{R},
\]

is weakly monotonic on \([0, \infty)^n\) if \( n \leq 1 + \left(\frac{q+1}{q-1}\right)^{q-1} \).

Proof. The Lehmer mean for \( q \in [-1, 0] \) is known to be monotonic [31] and hence weakly monotonic in that parameter range. Herein a proof of the cases \( q > 0 \) and \( q < -1 \) is given. By Theorem 2, \( L_q(x) \) can be written as a penalty-based function (13) with penalty \( P(x, y) = \sum_{i=1}^{n} x_i^q (x_i - y)^2 \). Differentiation w.r.t \( y \) yields

\[
P_y(x, y) = -2 \sum_{i=1}^{n} (x_i^{q+1} - x_i^q y)
\]

At the minimum the equation \( P_y = F(x, y) = 0 \) holds, with the necessary condition that

\[
y = \frac{\sum_{i=1}^{n} x_i^{q+1}}{\sum_{i=1}^{n} x_i^q} = L_q(x)
\]

For any \( x_i = 0 \) the Lehmer mean is defined in the limit as \( x_i \to 0^+ \). The partial derivatives \( \frac{\partial L_q(x)}{\partial x_i} \) are given by the implicit derivative \( \frac{\partial y}{\partial x_i} = -\frac{F_{x_i}}{F_y} \), with

\[
F(x, y) = \sum_{i=1}^{n} x_i^{q+1} - y \sum_{i=1}^{n} x_i^q = 0.
\]

By differentiation \( F_y(x, y) = -\sum_{i=1}^{n} x_i^q \leq 0, \ \forall x_i \in [0, \infty) \) and thus the sign of the partial derivatives depends on the sign of \( F_{x_i} \), which is given by

\[
F_{x_i}(x, y) = (q+1)x_i^q - qx_i^{q-1}y.
\]

These derivatives can be either positive or negative. To establish weak monotonicity it must be shown that the directional derivative of \( L_q(x) \) in the direction
(1, 1, ..., 1) be non-negative. Since
\[
(D_1 L_q)(x) = \frac{1}{\sqrt{n}} \nabla L_q(x) \cdot 1 = \frac{1}{\sqrt{n} F_y(x, y)} \sum_{i=1}^{n} F_{x_i}(x, y),
\]
then the sign of the directional derivative is determined only by the sign of
\[
\sum_{i=1}^{n} F_{x_i}(x, y).
\]
Henceforth, working with the sorted inputs, \(x_{(1)} = x_{\downarrow}\) such that \(x_{(1)}\) is thus the largest input and \(x_{(n)}\) the smallest, consider first the case for \(q > 0\).

Examine first the term \(F_{x_{(1)}}\) and note that \(y \leq x_{(1)}\) for any input \(x\) since \(L_q(x)\) is averaging (Condition 3 of Theorem 5). Then it follows that
\[
F_{x_{(1)}} = (q + 1)x_{(1)}^q - qx_{(1)}^{q-1}y
\]
\[
\geq (q + 1)x_{(1)}^q - qx_{(1)}^{q-1} x_{(1)}
\]
\[
= x_{(1)}^q \geq 0.
\]
For the remaining \(x_i\) compute the smallest possible value of \(F_{x_i}\) by selecting the point of minimum value, which is attained for
\[
\frac{\partial F_{x_i}}{\partial x_i} = q (q + 1) x_i^{q-1} - q (q - 1) x_i^{q-2} y = 0.
\]
At the optimum either \(x_i^* = 0\) or
\[
q(q + 1) (x_i^*)^{q-1} - q(q - 1) (x_i^*)^{q-2} y = 0
\]
\[
\Rightarrow x_i^* = \left( \frac{q - 1}{q + 1} \right) y.
\]
At $x^*_i = 0$ it follows that $F_{x_i} = 0$ and at $x^*_i = \left(\frac{q-1}{q+1}\right) y$ the following holds:

$$F_{x_i}(x^*_i) = (q + 1) \left( \left( \frac{q-1}{q+1} \right) y \right)^q - q \left( \left( \frac{q-1}{q+1} \right) y \right)^{q-1} y$$

$$= (q - 1) \left( \frac{q-1}{q+1} \right)^{q-1} y^q - q \left( \frac{q-1}{q+1} \right)^{q-1} y^q$$

$$= y^q \left( \frac{q-1}{q+1} \right)^{q-1} (q - 1 - q)$$

$$= -y^q \left( \frac{q-1}{q+1} \right)^{q-1}$$

$$\geq -x^q_{(1)} \left( \frac{q-1}{q+1} \right)^{q-1}.$$ 

Since $(D_1 L_q)(x) \propto \sum_{i=1}^n F_{x_i}$ then

$$(D_1 L_q)(x) = c \left( F_{x_{(1)}} + \sum_{i=2}^n F_{x_{(i)}} \right)$$

and since each $F_{x_{(i)}} \geq -x^q_{(1)} \left( \frac{q-1}{q+1} \right)^{q-1}$ then

$$(D_1 L_q)(x) \geq c \left( F_{x_{(1)}} + (n - 1) \left( -x^q_{(1)} \left( \frac{q-1}{q+1} \right)^{q-1} \right) \right)$$

$$= c \left( x^q_{(1)} - (n - 1) \left( \frac{q-1}{q+1} \right)^{q-1} x^q_{(1)} \right)$$

$$= c x^q_{(1)} \left( 1 - (n - 1) \left( \frac{q-1}{q+1} \right)^{q-1} \right).$$

This expression is non-negative and hence $L_q(x)$ is weakly monotonic provided that

$$(n - 1) \left( \frac{q-1}{q+1} \right)^{q-1} \leq 1 \quad \text{or} \quad n \leq 1 + \left( \frac{q-1}{q+1} \right)^{1-q}, q > 1.$$ 

This condition implies directly that for $q = 1$, weak monotonicity only holds for $n = 2$. 
Now consider the case for $q < -1$. Clearly

$$F_{x_i} = \frac{(1-p)x_i + py}{x_i^{p+1}}, \quad p = |q| > 1$$

and note that these derivatives are defined in the limit for the case where $x_i = 0$. I.e., $F_{x_i}^{\pm}|_{x_i=0} = \lim_{x_i \to 0^{\pm}} F_{x_i}$. Examine the term $F_{x(n)}$ and note that $y \geq x_{(n)}$ since $L_q(x)$ is averaging. Thus

$$F_{x(n)} = \frac{(1-p)x_{(n)} + py}{x_i^{p+1}}$$

$$\geq \frac{(1-p)x_{(n)} + px_{(n)}}{x_i^{p+1}}$$

$$= \frac{1}{x_i^{p}}.$$

Again, the remaining $x_i$ are considered by seeking the minimum of $F_{x_i}$, given by

$$\frac{\partial F_{x_i}}{\partial x_i} = \frac{p(1-p)}{x_i^{p+1}} + \frac{p(p+1)}{x_i^{p+2}} y = 0.$$

This attains a minimum at $x_i = \left(\frac{p+1}{p-1}\right) y$ and substitution into $F_{x_i}$ gives

$$F_{x_i} \left(\frac{p+1}{p-1} y\right) = \frac{(1-p)\left(\frac{p+1}{p-1} y\right) + py}{\left(\frac{p+1}{p-1} y\right)^{p+1}}$$

$$= \frac{-1}{y^p} \left(\frac{p+1}{p-1}\right)^{-(p+1)}$$

$$\geq \frac{-1}{x_i^{p}} \left(\frac{p+1}{p-1}\right)^{-(p+1)}.$$

The directional derivative of $L_q(x)$ can be written as

$$(D_1 L_q)(x) = c \left( F_{x(n)} + \sum_{i=1}^{n-1} F_{x(i)} \right)$$

$$\geq c \left( \frac{1}{x_i^{p}} - \frac{n-1}{x_i^{p}} \left(\frac{p+1}{p-1}\right)^{-(p+1)} \right)$$

$$= \frac{c}{x_i^{p}} \left( 1 - (n-1) \left(\frac{p+1}{p-1}\right)^{-(p+1)} \right).$$
Note that the sign of this derivative does not change in the limit as $x(n) \to 0^+$ and is non-negative for

$$n \leq 1 + \left(\frac{p + 1}{p - 1}\right)^{p+1} = 1 + \left(\frac{q - 1}{q + 1}\right)^{1-q}, \quad q = -p.$$ 

Hence, in both cases ($q < -1, q > 0$) the requirement for a non-negative directional derivative and hence weak monotonicity of $L_q(x)$ is given by the same relation, $n \leq 1 + \left(\frac{q-1}{q+1}\right)^{1-q}$. For the case $-1 \leq q \leq 0$ this remains a sufficient condition for weak monotonicity, although clearly overly restrictive. $\square$

**Remark 5.** As suggested by one of the referees of [91], as

$$\left(\frac{q + 1}{q - 1}\right)^{q-1} = \left(1 + \frac{2}{q - 1}\right)^{\frac{q-1}{p}},$$

and the right hand side is increasing (with $q$) and approaches $e^2$ as $q \to 0$, there is a restriction for all $q > 1$ weak monotonicity holds for at most $n < 9$ arguments. This restricts the use of Lehmer means for positive $q$ in applications requiring some sort of monotonicity.

**Corollary 4.** The contra-harmonic mean (Lehmer mean with $q = 1$) is weakly monotonic only for two arguments.

### 3.3.4. Gini means.

In this section the result for the weak monotonicity of the Lehmer means is extended to broader class of Gini means. The Gini mean is a quasi-mixture operator where $w(x_i) = x_i^q$ and $g(x_i) = x_i^m$. Let $m = p - q$.

**Theorem 7.** The Gini mean

$$G^{p,q}(x_1, \ldots, x_n) = \left(\frac{n}{\sum_{i=1}^{n} x_i^p}\right)^{\frac{q}{p-q}}.$$
for \( p, q \in \mathbb{R} \) is weakly monotonic on \([0, \infty)^n\) if

\[
(n - 1) \left( \left( \frac{q}{p} \right)^{p-1} \left( \frac{q-1}{p} \right)^{q-1} \right)^{\frac{1}{p-q}} \leq 1
\]

and \( p > q \) and \( q \notin (0, 1) \) or \( p < q \), \( p \notin (0, 1) \).

**Proof.** Since \( G^{p,q} = G^{q,p} \) it is sufficient to only consider the case \( p > q \) and thus \( m > 0 \). To ensure weak monotonicity of the Gini mean it must be shown that the sum of its directional partial derivatives is non-negative. In the case \( q \in (0, 1) \) the partial derivative at \( x = (a,b) \) tends to \(-\infty\) when \( a \to 0^+ \), so the Gini mean is not weakly monotonic for such \( q \).

The following analysis is simpler using implicit differentiation. Recall that Gini means can be obtained by minimising the penalty function

\[
P(x, z) = \sum_{i=1}^{n} x_i^q (x_i^m - z)^2,
\]

which yields \( z = \frac{\sum_{i=1}^{n} x_i^{q+m}}{\sum_{i=1}^{n} x_i^q} \). The derivative of the penalty, \( P_z = -2 \sum_{i=1}^{n} x_i^q (x_i^m - z) \), can be written in terms of the implicit function

\[
F(x, z) = \sum_{i=1}^{n} x_i^q x_i^m - \sum_{i=1}^{n} x_i^q z.
\]

and it holds that \( \frac{\partial z}{\partial x_i} = -\frac{F_{x_i}}{F_z} = -\frac{P_{x_i z}}{P_{zz}} \geq 0 \iff P_{x_i z} \leq 0 \) where \( F_z = -\sum_{i=1}^{n} x_i^q \leq 0 \) and \( F_{x_i} = (q + m)x_i^{q+m-1} - qx_i^{q-1}z \). To determine a sufficient condition for weak monotonicity the directional partial derivative of the Gini mean must remain non-negative: \( D_u(G^{p,q}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} F_{x_i} \geq 0 \), \( u = (1,1,...,1) \). In the following, take \( x(\cdot) = x_{\infty} \), so that \( x(1) \geq x(i) \geq x(n), i = 1, ..., n \).

Consider the case \( p > q \geq 1 \). Examining the term \( F_{x(1)} \), note that since \( y \leq x(1) \) and \( z = y^{m} \leq x_{m(1)}^{m} \) for any input vector \( x \), it follows that

\[
F_{x(1)} = (q + m)x_{m(1)}^{q+m-1} - qx_{m(1)}^{q-1}z \geq (q + m)x_{m(1)}^{q+m-1} - qx_{m(1)}^{q-1}x_{m(1)} = mx_{m(1)}^{q+m-1} \geq 0
\]
due to $m > 0$ and $x_{(1)} \in [0, 1]$. Now consider the smallest possible value of $F_{x_i}$, for which $F_{x_i x_i}$ is calculated to find the critical points. This results in

$$F_{x_i} x_i = x_i^{q-2} \left((q + m)(q + m - 1)x_i^m - q(q - 1)z \right) = 0$$

if and only if $x_i^* = 0$ (in which case $F_{x_i}(0) = 0$) or $x_i^* = \left(\frac{q(q-1)z}{(q+m-1)(q+m)}\right)^{\frac{1}{m}}$. At the critical point it follows that

$$F_{x_i}(x_i^*) = (q + m) \left(\frac{q(q-1)z}{(q+m-1)(q+m)}\right)^{\frac{q+m-1}{m}} - q \left(\frac{q(q-1)z}{(q+m-1)(q+m)}\right)^{\frac{q-1}{m}} z.$$

From now on denote $k = \frac{q-1}{m}$. Then

$$F_{x_i}(x_i^*) = z^{1+k} \left(\frac{q(q-1)z}{(q+m-1)(q+m)}\right)^{1+k} - q \left(\frac{q(q-1)z}{(q+m-1)(q+m)}\right)^{k}.$$

Now consider $z = y^m \leq x_{(1)}^m$ and $z^{k+1} = y^{m(k+1)} \leq x_{(1)}^{m(k+1)}$ then

$$F_{x_i}(x_i^*) \geq x_{(1)}^{(1+k)m} \left(\frac{q}{q+m-1}\right)^{k+1} (-m) \left(\frac{q-1}{q+m}\right)^{k}.$$

Therefore the directional derivative of $G^{p,q}$ is

$$D_u(G^{p,q}) \geq \frac{1}{\sqrt{n}} x_{(1)}^{q+m-1} m + (n - 1) x_{(1)}^{q+m-1} \left(\frac{q}{q+m-1}\right)^{\frac{q+m-1}{m}} (-m) \left(\frac{q-1}{q+m}\right)^{\frac{q-1}{m}}$$

$$= \frac{m}{\sqrt{n}} x_{(1)}^{q+m-1} \left(1 - (n-1) \left(\frac{q}{q+m-1}\right)^{\frac{q+m-1}{m}} \left(\frac{q-1}{q+m}\right)^{\frac{q-1}{m}}\right)$$

$$\geq 0.$$
3.3. NEW RESULTS ON WEAKLY MONOTONIC MEANS 103

if

\[(n-1) \left( \frac{q}{q + m - 1} \right)^{\frac{q + m - 1}{m}} \left( \frac{q - 1}{q + m} \right)^{\frac{q - 1}{m}} = (n-1) \left( \left( \frac{q}{p - 1} \right)^{p-1} \left( \frac{q - 1}{p} \right)^{q-1} \right)^{\frac{1}{p-q}} \leq 1. \]

In the second case: \( q \leq 0 \) and \( q < p \). To analyze the directional derivative of \( G^{p,q} \), consider \( z = y^m \geq x^m_{(n)}. \) Then

\[ F_{x_{(n)}} = (q + m)x_{(n)}^{q+m-1} - qx_{(n)}^q \geq (q + m)x_{(n)}^{q+m-1} - qx_{(n)}^{q-1}x_{(n)} \]

\[ = mx_{(n)}^{q+m-1} \geq 0 \]

Now, taking into account that

\[ F_{x^*_{i}} \geq x_{(n)}^{q+m-1} \left( \frac{q}{q + m - 1} \right)^{q+m-1} \left( \frac{q - 1}{q + m} \right)^{\frac{q - 1}{m}} \]

it follows that

\[ D_u(G^{p,q}) \geq \frac{1}{\sqrt{n}} x_{(n)}^{q+m-1} m + (n - 1)x_{(n)}^{q+m-1} \left( \frac{q}{q + m - 1} \right)^{\frac{q+m-1}{m}} \left( \frac{q - 1}{q + m} \right)^{\frac{q-1}{m}} \]

\[ \geq 0 \]

if

\[(n-1) \left( \frac{q}{q + m - 1} \right)^{\frac{q + m - 1}{m}} \left( \frac{q - 1}{q + m} \right)^{\frac{q - 1}{m}} = (n-1) \left( \left( \frac{q}{p - 1} \right)^{p-1} \left( \frac{q - 1}{p} \right)^{q-1} \right)^{\frac{1}{p-q}} \leq 1. \]

By combining both cases with the symmetry \( G^{p,q} = G^{q,p} \) the result is obtained. □

3.3.5. Mixture functions. Consider mixture functions described in Section 2.5 and the generalised mixture functions (Eqn. (19)). The three cases for ensuring a monotonic, increasing function (presented in Section 2.5) are restated herein. In each case consider weighting functions \( w : [0, 1] \rightarrow [0, \infty) \), then sufficient conditions for monotonicity are:

(1) \( w \geq w' \) for any increasing, piecewise differentiable weighting function [45];
3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

(2) \( w(x) \geq w'(x)(1 - x) \) for all \( x \in [0, 1] \): \[50\] \[51\];

or, if we fix the dimension \( n \) of the domain,

(3) \( \frac{w^2(x)}{(n-1)w(1)} + w(x) \geq w'(x)(1 - x), x \in [0, 1], n > 1 \): \[50\].

Analogous results have been obtained for non-increasing weighting functions using duality. Taking the dual weighting function \( w^d(x) = w(1 - x) \), the resulting mixture function is the dual to \( M_w \); that is, \( M_{wd} = 1 - M_w \) \[50\]. Duality preserves both weak and standard monotonicity. The following results are provided for several interesting special cases of weighting functions.

**Proposition 11.** Let \( M_w(x) \) be a mixture function defined by eqn. (19) and \( w : [0, 1] \rightarrow [0, \infty) \). For the following generators, the functions \( M_w \) are:

- \( w(x) = e^{ax+b}, a \in [-1, 1] \): monotonic;
- \( w(x) = e^{ax+b}, a \in \mathbb{R} \): shift-invariant and hence weakly monotonic;
- \( w(x) = \ln(1 + x) \): weakly monotonic for \( n = 2 \) and \( x \geq 0.1117 \), \( n = 3 \) and \( x \geq 0.2647 \), and \( x \geq 0.4547 \) as \( n \rightarrow \infty \); and,
- \( w(x) = 1 - \sqrt{x} \): weakly monotonic for \( n = 2 \) only for all \( x \in [0, 1]^n \) and \( r > 1 \).

**Proof.** The first generator trivially satisfies the first condition above for \( a \in [0, 1] \), \( w'(x) \leq w(x) \), and for negative \( a \) the result is obtained using duality.

In the second case this generator fails conditions (1) and (2) above for \( a > 1 \). However

\[
M_w(x + t1) = \frac{\sum e^{ax_i+b+at}(x_i + t)}{\sum e^{ax_i+b+at}} = \frac{\sum e^{ax_i+b}(x_i + t)}{\sum e^{ax_i+b}} = M_w(x) + t.
\]

For the third generator the three conditions of monotonicity, stated above, are checked. From the first condition \( w(x) = \ln(1 + x) \geq w'(x) = \frac{1}{1+x} \) when \( x \geq LW(1) \approx 0.7632 \) (\( LW \) is the Lambert \( W \) function). Thus the mixture function generated by \( w(x) = \ln(1 + x) \) is monotonic in \([0.7632, \infty)\). From the second sufficient condition it follows that the mixture operator is monotonic in \([0.4547, \infty)\) and from the third condition it follows that the mixture operator generated by \( w(x) = \ln(1 + x) \) is monotonic in \([0.3708, \infty)\) for \( n = 2 \).
3.3. NEW RESULTS ON WEAKLY MONOTONIC MEANS

Considering weak monotonicity, it is the case that the mixture function with such a weighting function is weakly monotonic for such \( x \) that satisfy

\[
\ln(2) + (n - 1) \left( \ln(x + 1) + \frac{x - 1}{x + 1} \right) \geq 0.
\]

This inequality follows from the fact that the directional derivative of \( M_w \) is proportional to the sum of partial derivatives of \( F(x, y) = \sum_{i=1}^{n} \ln(1 + x_i)(x_i - y) \), that \( F_{x_i}(x_i, y) \geq \ln(1 + x(1)) \) and that all other partial derivatives achieve their minimum with respect to \( y \) when \( y = x(1) \). The directional derivative is the smallest when \( x(1) \) is the largest (\( x(1) = 1 \)), from which the above inequality is derived. By solving this inequality for \( x \) numerically, for fixed \( n \), it is found that for \( n = 2 \), \( x \) must be larger or equal to 0.1117, for \( n = 3 \), \( x \in [0.2647, \infty) \), and so on. When \( n \to \infty \) the smallest \( x \) approaches 0.4547..., which is consistent with standard monotonicity.

In the last case, let \( F(x, y) = P_y(x, y) \). The directional derivative of \( M_w \) is proportional to the sum

\[-\sum_i F_{x_i}(x_i, y) = \sum_i w'(x_i)(x_i - y) + w(x_i). \]

Since \( -F_{x_i}(x_{(n)}), y) \geq 1 - x_{(n)}^{1/r} \geq 0 \), and \( -F_{x_i}(x_i, y) \leq 1 - x_i^{1/r}(1 + \frac{1}{r}) + \frac{1}{r}x_i^{1/r-1}x_{(n)} \), for weak monotonicity the following must hold:

\[
1 - x_{(n)}^{1/r} + (n - 1) \left( 1 - x_i^{1/r}(1 + \frac{1}{r}) + \frac{1}{r}x_i^{1/r-1}x_{(n)} \right) \geq 0.
\]

The left hand side is the smallest when all \( x_i = 1 \), hence the directional derivative of \( M_w \) is non-negative when \( 1 - x_{(n)}^{1/r} + \frac{n-1}{r}(x_{(n)} - 1) \geq 0 \). Resolving this inequality for \( n \) results in

\[
n \leq r \frac{1 - x_{(n)}^{1/r}}{1 - x_{(n)}} + 1.
\]

The expression on the right decreases when \( x_{(n)} \to 1 \), and the limit of the ratio is \( \frac{1}{r} \). Therefore \( n \leq r \frac{1}{r} + 1 = 2 \).

Figure 3.1 depicts mixture functions formed by the generator functions described in Proposition 11. Contour plots for these averages are given in Figure 3.2. For the generator function \( g(x_i) = \exp\{ax_i + b\} \), as \( a \to \infty \) (respectively \(-\infty\)), the mixture function approximates \( \min(x_1, x_2, ..., x_n) \) (respectively \( \max(x_1, x_2, ..., x_n) \)).
3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

The next result is particularly interesting and is relevant in Subsection 3.3.6. It shows that if the weighting function is Gaussian, \( w(x) = \exp\left(-\frac{(x-a)^2}{b^2}\right) \), weak monotonicity holds irrespective of the value of \( a \) and for all \( b^2 \geq 1 \).

**Theorem 8.** Let \( A(x) \) be a mixture function defined by eqn. (19) with generator \( w(x) = e^{-(x-a)^2/b^2} \), then \( A \) is weakly monotonic for all \( a \in \mathbb{R} \), \( b \in \mathbb{R}/(-1,1) \) and \( x \in [0,1]^n \).

**Proof.** As with the previous proofs it is sufficient to show that the sum of the directional derivatives in the direction \( u = \frac{1}{\sqrt{n}} \{1,1,...,1\} \) is non-negative for all the given domains of \( a \) and \( b \). Consider first that the partial derivatives of \( A \)
3.3. NEW RESULTS ON WEAKLY MONOTONIC MEANS

are given by

\[
A_{x_i} = \frac{(w(x_i) + w'(x_i)x_i) \sum_{j=1}^{n} w(x_j) - w'(x_i) \sum_{j=1}^{n} w(x_j)x_j}{\left( \sum_{i=1}^{n} w(x_i) \right)^2}
\]

and

\[
D_u(A) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} A_{x_i} = \frac{1}{\sqrt{n} \left( \sum_{i=1}^{n} w(x_i) \right)^{\frac{3}{2}}} \sum_{i=1}^{n} \left( w_i + x_i w'_i \right) \sum_{j=1}^{n} w_j - \sum_{j=1}^{n} x_j w'_j \sum_{j=1}^{n} w_j .
\]
For brevity denote \( w_i = w(x_i) \) and \( w'_i = w'(x_i) = -2(x_i - a)w_i \). Observe that since the first term is strictly positive over all \( x_i \) it is sufficient to consider only the sign of the numerator of \( A_{x_i} \) to determine the sign of the directional derivative.

Given that \( w'_i = -\frac{2}{b^2}(x - a)w_i \), the above summations can be reorganised such that, up to a positive constant of proportionality,

\[
D_u(A) \propto \left( \sum_{i=1}^{n} w_i \right)^2 - 2 \frac{a}{b^2} \left( \sum_{i=1}^{n} x_i w_i \right)^2 - \frac{2}{b^2} \left( \sum_{j=1}^{n} x_j w_j \right) \sum_{i=1}^{n} w_i.
\]

Cancelling the terms with the factor \( \frac{2a}{b^2} \) gives

\[
D_u(A) \propto \left( \sum_{i=1}^{n} w_i \right)^2 - 2 \frac{a}{b^2} \left( \sum_{i=1}^{n} x_i w_i \right)^2 - \frac{2}{b^2} \left( \sum_{j=1}^{n} x_j w_j \right) \sum_{i=1}^{n} w_i.
\]

Consider subsequently the following equalities:

\[
\left( \sum_{i=1}^{n} w_i \right)^2 = \sum_{i=1}^{n} w_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} w_i w_j,
\]

\[
-2 \frac{a}{b^2} \left( \sum_{i=1}^{n} x_i w_i \right)^2 = -2 \frac{a}{b^2} \sum_{i=1}^{n} x_i^2 w_i^2 - \frac{2}{b^2} \sum_{i=1}^{n} \sum_{j \neq i} x_i w_j w_j = -\frac{2}{b^2} \sum_{i=1}^{n} x_i^2 w_i^2 - \frac{2}{b^2} \sum_{i=1}^{n} \sum_{j \neq i} \left( \frac{1}{2} x_i^2 + \frac{1}{2} x_j^2 \right) w_i w_j,
\]

\[
2 \frac{1}{b^2} \left( \sum_{i=1}^{n} x_i w_i \right)^2 = 2 \frac{1}{b^2} \sum_{i=1}^{n} (x_i w_i)^2 + 2 \frac{1}{b^2} \sum_{i=1}^{n} x_i x_j w_i w_j.
\]
Then, substituting the last equalities in the sum of the partial derivatives and simplifying,

\[ D_u(A) \propto \sum_{i=1}^{n} w_i^2 + \frac{1}{b^2} \sum_{i=1}^{n} \sum_{j \neq i} (b^2 - (x_i - x_j)^2)w_iw_j, \]

and this expression is non-negative for all \( x_i, x_j \in [0, 1] \), \( b^2 \geq 1 \) and independently of \( a \).

Thus Gaussian weighting functions become particularly important in the construction of mixture functions. By varying the parameters \( a \) and \( b \) the mixture functions are variously monotonic increasing, monotonic decreasing (both convex and concave) and unimodal quasi-concave, which will all be weakly monotonic functions.

Consider now linear combinations of weighting functions.

**Proposition 12.** Let \( w : [0, 1] \to ]0, \infty[ \) be given by \( w(x) = u(x) + v(x) \) where \( u, v : [0, 1] \to ]0, \infty[ \). Then the mixture operator \( M_w(x) \) defined by eqn. (19) is weakly monotonic if the mixture operators with generators \( u(x) \) and \( v(x) \) are also weakly monotonic.

**Proof.** Consider the function \( F^{(w)} \) related to the penalty function defining the mixture operator \( M_w \). It is a matter of a simple calculation to note that any partial derivative of \( F^{(w)}(x, z) = \sum_{i=1}^{n} w(x_i)(x_i - y) \) is

\[
F_{x_i}^{(w)} = u'(x_i)(x - y) + w(x_i)
= (u'(x_i) + v'(x_i))(x - y) + u(x_i) + v(x_i)
= (u'(x_i)(x - y) + u(x_i)) + (v'(x_i)(x - y) + v(x_i))
= F_{x_i}^{(u)} + F_{x_i}^{(v)}.
\]

Hence, \( F_{x_i}^{(w)} \geq 0 \) if \( F_{x_i}^{(u)} \geq 0 \) and \( F_{x_i}^{(v)} \geq 0 \). Therefore, directional derivative of \( M_w \) will be non-negative if the directional derivatives of \( M_u \) and \( M_v \) are non-negative. \( \square \)
Observe that the last result can be extended for any number of weighting functions, so that, if the weighting function of a mixture function \( M_w \) is in the form
\[
w = \sum_{i=1}^{r} u_i
\]
then the mixture function \( M_w \) is weakly monotonic if the mixture functions \( M_{u_i} \) are also weakly monotonic.

**Proposition 13.** Let \( w : [0, 1] \rightarrow [0, \infty[ \) be a weighting function given by \( w(x) = x + b, \ b > 0 \), then the corresponding mixture operator \( M_w \) is weakly monotonic if \( b \geq \frac{n-2}{n} x_{(1)} \geq \frac{n-2}{n} \).

**Proof.** It is enough to see that the directional derivative is non-negative. For that, first calculate the partial derivative of the function
\[
F(x, y) = \sum_{i=1}^{n} w(x_i)(x_i - y)
\]
to obtain the condition
\[
F_{x_i}(x, y) = w(x_i) + w'(x_i)(x_i - y) = (x_i + b) + (x_i - y) = 2x_i + b - y.
\]
For \( x \downarrow, a s \ y \leq x_{(1)} \) it follows that
\[
D_u(M_w) \geq (2x_{(1)} + b - x_{(1)}) + (n - 1)(b - x_{(1)}) = nb + (n - 2)x_{(1)} \geq 0
\]
if and only if \( b \geq \frac{n-2}{n} x_{(1)} \geq \frac{n-2}{n} \). \( \square \)

Regarding this result, observe that for \( n = 2, 3 \) or \( 5 \) then \( b \geq 0, b \geq \frac{1}{3} \) or \( b \geq \frac{3}{5} \) respectively. Therefore, mixture functions of the mentioned dimensions are weakly monotonic for \( b \geq 0, b \geq \frac{1}{3} \) or \( b \geq \frac{3}{5} \).

On the other hand, considering the first condition for monotonicity from Section 3.3.5, \( w(x) = x + b \geq 1 = w'(x) \) holds only if \( b \geq 1 \). Considering the second, more relaxed condition, \( w(x) \geq w'(x)(1 - x) \), also results in \( b \geq 1 \). Finally, the further relaxed condition \( \frac{w^2(x)}{(n-1)w(1)} + w(x) \geq w'(x)(1 - x) \) results in \( b^2 \geq 1 - \frac{1}{n} \).

**3.3.6. Spatial-Tonal Filters.** The well known class of spatial-tonal filters includes the mode filter [89], bilateral filter [85] and anisotropic diffusion [66] among others. This is an important class of filters developed to preserve edges within images when performing tasks such as filtering or smoothing. While these
filters are commonly expressed in integral notation over a continuous space, they
are implemented in discrete form over a finite set of pixels that take on finite
values in a closed interval. It can be shown that the class of functions is given
(in discrete form) by the averaging function

\[ F_\Delta^g(x; x_1) = \frac{\sum_{i=1}^{n} w_i g(|x_i - x_1|) x_i}{\sum_{i=1}^{n} w_i g(|x_i - x_1|)} \]

where the weights \( w_i \) are nonlinear functions of the locations of the pixels, which
have intensity \( x_i \). In all practical problems the locations are constant and hence
can be pre-computed to produce the constant weight vector \( \Delta = (w_1, w_2, ..., w_n) \).
The pixel \( x_1 \) is the pixel to be filtered/smoothed such that its new value is \( \bar{x}_1 = F_\Delta^g(x_1) \).

The function \( F_\Delta^g \) is nonlinear and not monotonic. It is trivially shown to be
expressed as a penalty-based function with penalty

\[ P(x, y) = \sum_{i=1}^{n} w_i g(|x_i - x_1|)(x_i - y)^2. \]

In image filtering applications it is known that this penalty minimises the mean
squared error between the filtered image and the noisy source image \([30]\). By
Theorem 4 it follows directly that the filter \( F_\Delta^g \) is shift-invariant and hence weakly
monotonic.

3.3.7. Generalisations of Spatial-Tonal Filters. Theorem 4 permits us
to generalise this class of filters to be those penalty based averaging functions
having penalty function

\[ P(x, y) = \sum_{i=1}^{n} w_i g(|x_i - f(x)|)(x_i - y)^2 \]

provided the \( w_i \) are constant under translations of \( x \), or even further using any
shift-invariant univariate function \( D : \mathbb{R}^2 \to \mathbb{R} \) (as discussed in Section 3.3)
The implication of replacing $x_1$ with $f(x)$ in the scaling function $g$ is that we may use any shift-invariant aggregation of $x$, which allows us to account for the possibility that $x_1$ is itself an outlier within the local region of the image. This is exactly the condition accounted for in the design of the fuzzy weighted median filters (and generalisations) discussed in Subsection 2.6.3 of Chapter 2. This provides an interesting result that invites further research into the application of weakly monotonic means to spatial-tonal filtering and smoothing problems and a comparison between this approach and those fuzzy approaches based on order statistics and trimmed means.

### 3.4. Mode-like Averages

As described previously, the mode (Definition 64) is a robust estimator of location that measures the central tendency of a data set. However, it is not well defined in cases where each of the inputs is distinct. The recently proposed mode-like averages [12] seek to overcome this deficiency by computing a representative value of the most significant cluster within the input data. When this cluster contains a single value in the majority this value should be returned, ensuring that the behaviour of this function collapses to that of the mode when there exists a most frequent value. As the mode is an internal function, then in all other cases the value returned is the input value that lies closest to the central tendency of the data. In this way the mode-like average computes a weighted maximum likelihood value from the data and is thus akin to the generalised M-estimators (see, for example, [35]) with the added constraint of being an internal function.

The mode-like average of Beliakov is given by minimisation of the penalty
\( P(x, y) = \sum_{i=1}^{n} w_i \rho(r_{(i)}) \)

with

\[
\rho(r_{(i)}) = \begin{cases} 
  r_{(i)} & r_{(i)} \leq \tau \\
  \beta \tau & r_{(i)} > \tau
\end{cases}
\]

\[
\tau = \alpha \max(\epsilon, r_{(2)}) \\
r_{i} = |x_i - y|
\]

\( \alpha > 0, \ 0 \leq \beta \leq 1 \)

The value \( \epsilon \) is the precision of the input data (i.e., the smallest difference value representable) and thus \( \tau \) ensures that the cluster radius is non-zero and encompasses any repeated values of \( y \) within the inputs (since in this case, \( r_{(2)} = 0 \)). Input values within the threshold \( \tau \) are assumed to be cluster members and their penalty term is equal to their weighted residual value. Inputs lying beyond this cluster threshold are assumed to be outliers; they are not omitted from the computation but rather they contribute a constant value to the total penalty for the given value of \( y \). The constant weights \( w_i \) are associated with the individual inputs and weight their significance given the application context. It was suggested by Beliakov that if all inputs are of equal importance then the weights may be omitted.

The parameter \( \alpha \) determines the cluster scale and since \( r_{(2)} \) is dependent on the inputs values, the cluster threshold is adaptive. For a given set of input data and choice of \( y \), the value \( \beta \tau \) is a constant and thus outlier pixels contribute to the penalty only as a constant times their weight.

The penalty \( P \) in Eq. (33) is designed to: a) penalise non-compact clusters; b) penalise small clusters; and, c) penalise values of \( y \) away from the cluster centre. For two clusters of equivalent scale, \( P \) will be smaller for the cluster containing more inputs and for two clusters containing the same number of inputs, the
Several novel data fusion operators have recently been proposed where the weights applied to the inputs are not constant \[2\] \[51\] \[62\]. Some of these operators are derived from the Bajraktarevic means (Defn. 69), by applying weights that depend on the inputs. The mixture operators described above are an example

3. WEAKLY MONOTONIC AVERAGING FUNCTIONS

penalty is smaller for the cluster with lower \( \tau \). Since this threshold is dependent on the input data then this averaging function adaptively determines outliers.

**Theorem 9.** Let \( F \) be a penalty-based function defined by minimisation of the penalty in Eq. (33). Then \( F \) has averaging behaviour.

**Proof.** There exists a permutation \( \sigma \) such that \( r_\sigma = r_\sigma \) and \( r_\sigma (1) = |x_\sigma - y| \). Assume that \( y < x_\sigma (1) = \min(x) \) for \( x_\sigma = x_\sigma \). Then \( x_\sigma = x_\sigma \) and \( r_\sigma (1) = |x_\sigma (1) - y| > |x_\sigma (1) - x_\sigma (1)| \). Since all \( w_i \geq 0 \) then \( P(x, y) > P(x, x_\sigma (1)) \) and \( y \) cannot be a minimiser of \( P \). Subsequently \( F(x) \geq x_\sigma (1) = \min(x) \). Similarly, assume that \( x_\sigma = x_\sigma \) and \( y > x_\sigma (1) = \max(x) \). Then \( r_\sigma (1) = |x_\sigma (1) - y| > |x_\sigma (1) - x_\sigma (n)| \) and once again, given positive weights, \( P(x, y) > P(x, x_\sigma (1)) \) and \( y \) cannot be the minimiser of the penalty. Subsequently, \( F(x) \leq x_\sigma (1) = \max(x) \). Thus \( F(x) \) has averaging behaviour.

**Proposition 14.** The mode-like average \( F \) given by minimisation of the penalty in Eq. (33) is shift-invariant and hence weakly monotonic.

**Proof.** Take \( P(x, \mu) \) as the minimum penalty corresponding to \( \mu = F(x) \) and note that each \( w_i \) is constant under translations of \( x \). There exists a permutation \( \sigma \) such that \( r_\sigma = r_\sigma \) and \( |x_\sigma (1) - y| = r_\sigma (1) \). Then \( r_\sigma (t) \) is the first non-zero residual, \( r_\sigma (t) = 0 \) and \( P(x, \mu) = \sum_{i=1}^n w_i \rho(|x_\sigma (1) - \mu|) \). Under the translation \( x + a \mathbf{1} \), then \( r_\sigma (t) = |x_\sigma (1) + a - y| \leq |x_\sigma (1) + a - y| \) and the order or the residuals is unchanged. Taking \( y = \mu + a \) gives \( r_\sigma (t) = |x_\sigma (1) + a - y| = |x_\sigma (1) - \mu| \) and thus the threshold index \( t \) is unchanged, ensuring that \( P(x + a \mathbf{1}, \mu + a) = \sum_{i=t}^n w_i \rho(|x_\sigma (i) - \mu|) = P(x, \mu) \). Thus \( F(x + a \mathbf{1}) = F(x) + a \) and \( F \) is shift-invariant and hence weakly monotonic.

3.5. Density-based Averages

Several novel data fusion operators have recently been proposed where the weights applied to the inputs are not constant \[2\] \[51\] \[62\]. Some of these operators are derived from the Bajraktarevic means (Defn. 69), by applying weights that depend on the inputs. The mixture operators described above are an example
of each weight depending on one input. Statistically grounded logic operators [62] are based on the minimisation of a penalty (the sum of absolute differences between the inputs and the output), which is weighted by a function of the inputs. The most recent proposal is density-based averaging [2], which was derived from a weighted arithmetic mean, but involves variable weights which depend on the density of inputs. Here the weighting functions depend not only on their respective inputs, but on all the inputs and hence they are a more general construction than Bajraktarevic means.

Herein the class of density-based averaging operators, first introduced in [96], are explored and several generalisations of the formulas in [2] are presented. The main motivation behind density-based means in [2] was to define analogues of the weighted arithmetic mean to filter outliers, by using weights that depend on the density of the data, with inputs closer to the main group of values having higher weights than those far away. The authors of [2] develop their means in the context of stream processing, in which only recent data are available and older data are represented by a few statistical quantities. Notably, by using a Cauchy kernel they develop recursive update formulas, suitable for on-line, real-time stream processing applications. To simplify the current presentation the formulas from [2] are restated without referring to the current time instance, assuming it is fixed. Once the mathematical properties are established they are easily ported to the stream processing context as in [2].

Let $d_{ij}$ denote the distance between inputs $x_i$ and $x_j$. In [2] the authors used Euclidean and cosine distances (in case the inputs $x_i$ are vectors themselves), although here consideration is restricted to Euclidean or Minkowski distances.

**Definition 71.** The density-based average is given by

$$ y = \sum_{i=1}^{n} w_i(x) x_i, $$
where

\begin{equation}
  w_i(x) = \frac{\delta_i}{\sum_{j=1}^{n} \delta_j} = \frac{K_C\left(\frac{1}{n} \sum_{j=1}^{n} d_{ij}^2\right)}{\sum_{k=1}^{n} K_C\left(\frac{1}{n} \sum_{j=1}^{n} d_{kj}^2\right)},
\end{equation}

and where \( K_C \) is the Cauchy kernel given by

\[ K_C(t) = (1 + t)^{-1}. \]

Note that the formula (34) is more general than the mixture operators because \( w_i \) depends on all input data, rather than merely \( x_i \). Note also that the penalty associated with (34) is

\begin{equation}
  P(x, y) = \sum_{i=1}^{n} w_i(x) (x_i - y)^2,
\end{equation}

which differs from Eq. (23) in that now the weights \( w_i \) depend on all of the components of the input vector. Consequently, the sufficient conditions for monotonicity of mixture functions discussed in Section 2.5 are not applicable.

Density-based means are not monotonic in the usual sense, which can be seen from the following example.

**Example 8.** Take \( n = 20, x = (1, 0, 0, \ldots, 0) \) and \( y = (3, 0, 0, \ldots, 0) \), and hence \( x \leq y \). Then \( w(x) = \left(\frac{7}{254}, \frac{13}{254}, \ldots, \frac{13}{254}\right) \) and \( w(y) = \left(\frac{29}{3658}, \frac{191}{3658}, \ldots, \frac{191}{3658}\right) \), which can be computed easily by noticing that for the vector \( x \): \( \sum_j d(1, j)^2 = n - 1 = 19 \) and \( \sum_j d(i, j)^2 = 1 \) for \( i \neq 1 \), and for \( y \): \( \sum_j d(1, j)^2 = 9(n - 1) = 171 \) and \( \sum_j d(i, j)^2 = 9 \) for \( i \neq 1 \). Then, applying (34) results in \( F(x) = \frac{7}{254} > \frac{87}{3658} = F(y) \) and hence monotonicity does not hold.

The issues of bounds preservation and weak monotonicity of (34) will be addressed by making use of Proposition 10 and Theorem 3. The following theorem can be established.

**Theorem 10.** The density based mean given by eqn. (34) is bounds preserving (and hence a mean as per Definition 24) and weakly monotonic.
Proof. The first assertion comes from Proposition 10 and the fact that (34) is based on the associated weighted arithmetic mean. The second assertion is supported by the fact that the weights depend only on the pairwise distances between the inputs, which remain constant under translation and then follows from Theorem 3 with \( g_i = w_i \), since all shift-invariant functions are weakly monotonic.\(^1\)

The weighted arithmetic mean is an example of the maximum likelihood (ML) estimator, which is based on the assumption that the inputs follow a Gaussian distribution around some value, estimated by the mean. If the distribution of inputs is different then other estimators, such as the median (when the distribution is Laplacian) are more appropriate. Thus, the following generalisations of the density-based means seem reasonable.

3.5.1. Density based medians and ML estimators. By modifying the penalty (36), density-based analogues of the median and other ML estimators are possible. The first generalisation arises by changing the penalty from

\[
P(x, y) = \sum w_i(x)(x_i - y)^2
\]

to

\[
(37) \quad P(x, y) = \sum w_i(x)|x_i - y|,
\]

from which a density-based weighted median is obtained. This approach was taken in [62] and further explored in [51], in the case of \( w_i \) being a function of one argument \( x_i \).

By using other functions instead of the squared or absolute differences other maximum likelihood (ML) type estimators are obtained. Each estimator will be bounds preserving and weakly monotonic and if all \( w_i \) are the same, will also be symmetric. These functions will reduce to the corresponding standard

\(^1\)We refer here only to the standard distances in linear vector spaces. The cosine distance mentioned in [2] is not a proper metric and the distances are affected by translation.
averaging aggregation functions (with constant weights) in the limiting case of equal distances between all data.

One interesting example is based on Huber’s loss function, defined by the penalty

\[ \mathcal{P}(\mathbf{x}, \mathbf{y}) = \sum w_i(\mathbf{x}) \rho(|x_i - y|), \]

where

\[ \rho(t) = \begin{cases} \frac{1}{2}t^2, & \text{if } t \leq \delta, \\ \delta(t - \frac{\delta}{2}), & \text{otherwise.} \end{cases} \]

Another example is the loss function \( \rho(t) = \log(\cosh(t)) \), which behaves similarly to Huber’s function. These two functions arose in robust statistics, where one aims at limiting the contribution of the outliers. Huber’s estimator behaves like the least squares estimator for data close to the average (the parameter \( \delta \) quantifies the notion of closeness), yet it behaves like the (more robust) median for data further away, when \( \rho \) changes to a linear function and hence limits the contribution of potential outliers.

### 3.5.2. Modified weighting functions.

The second generalisation of the density-based means is produced by modifying the weights. Note that any set of (non-negative) functions \( g_i \) which depend only on the distances between the inputs will ensure the result is a shift-invariant and hence a weakly monotonic function. Instead of adding all pairwise distances as in [2], a more localised estimator of density may be used, such as the variable kernel density estimators [83] and the nearest neighbour estimator. These estimators are more accurate that those based on all pairwise distances and can hence capture the notion of density more effectively. Specifically, such estimators limit the influences of the inputs far away from the point where the density is measured.

The nearest neighbour estimator can be expressed using a simple modification to the formula (35), by fixing \( i \) and ordering the distances (squared or otherwise) \( d_{i,(1)}(\mathbf{x}) \leq d_{i,(2)}(\mathbf{x}) \leq \ldots \leq d_{i,(n)}(\mathbf{x}) \). Now the \( k \)-th nearest neighbour density estimator is

\[ \delta^k_i(\mathbf{x}) = \frac{k}{2 \cdot n \cdot d_{i,(k)}(\mathbf{x})}, \]
3.5. DENSITY-BASED AVERAGES

with typically \( k \approx \sqrt{n} \).

A generalisation of the above formula is obtained by using a kernel \( K \):

\[
\delta_{i}^{K,k}(x) = \frac{1}{n \cdot d_{i,(k)}} \sum_{j=1}^{n} K \left( \frac{x_i - x_j}{d_{i,(k)}} \right).
\]

Finally, a trimmed mean of the distances can be used instead of the mean in (35); i.e.,

\[
\delta_{i}^{T}(x) = K \left( \frac{1}{k} \sum_{j=1}^{k} (d_{i,(j)}(x))^2 \right),
\]

or the median distance

\[
\delta_{i}^{med}(x) = K \left( median((d_{i,j}(x))^2) \right),
\]

in order to obtain a robust estimator of the density [72] [73]. Here on the contribution of the \( k \) nearest data (or at most half of the data in the case of median distance) is accounted for. In this way outliers are discarded in density calculations and hence the weights are more robust.

The weighting functions \( w_{i} \) are still computed by

\[
(38) \quad w_{i}(x) = \frac{\delta_{i}(x)}{\sum_{j=1}^{n} \delta_{j}(x)}.
\]

In all of the cases mentioned above the weighting functions will remain constant under translations and therefore the resulting density based mean will be shift-invariant and bound-preserving.

3.5.3. Filtering Outliers. The claim in [2] that density based means (34) filter outliers is only partially true, for isolated outliers. It is known that the arithmetic mean is not robust to outliers [73] and a single outlier is sufficient to drastically affect the value of the mean. Using variable weights alleviates this problem by downplaying the contribution of individual outliers, however, several closely grouped outliers will produce large density-based weights and hence contribute more to the corruption of the average. For instance, consider \( x = (0, 1, 2, 3, 100, 101) \). The density based mean given by (34) will produce \( y \approx 21.32 \).
because the weights computed by (35) are

\[ w \approx (0.194, 0.198, 0.202, 0.206, 0.101, 0.099). \]

The output is biased by the outliers and is not representative of the main cluster near \( y = 1.5 \). In contrast, using the median \( \delta^\text{med} \) in (35), the output value \( y \approx 1.52 \) is obtained, since the weights would be

\[ w \approx (0.154, 0.345, 0.345, 0.154, 0.00118, 0.000116). \]

Clearly the contribution of the outliers to this value are significantly diminished and thus these data are effectively filtered from the input.

While it is true that when the outliers constitute a relatively small minority, equation (35) will produce very small weights for the outliers, though this is not a guarantee of robustness, as the limit of (34) as \( x_i \to \pm \infty \) is not necessarily finite. This can be seen by examining the ratio \( \frac{w_i}{w_j} \) as \( x_i \to \pm \infty \). It is not difficult to see that \( \frac{w_i}{w_j} \to \frac{1}{n-1} \) in this case. Furthermore, as the percentage of outliers in the data increases, the total weight of the outliers will further increase.

To properly deal with outliers the arithmetic mean in (34) should be replaced by one of the robust estimators of location [73], such as those based on the least trimmed sum of squares principle, or the least median of squares, as was done in the simple numerical example above. This will ensure that the relative weight of the outliers \( \frac{w_i}{w_j} \to 0 \) as \( x_i \to \pm \infty \).

3.6. Conclusions

In this chapter a new framework for averaging aggregation functions has been proposed, based on the concept of weak monotonicity; a relaxation of the monotonicity constraint in the definition of aggregation functions. This class of functions includes all monotonic functions and thus, with respect to the means, all averaging aggregation functions are weakly monotonic averages. Additionally, all shift-invariant functions (those stable to translations) are also weakly monotonic. Weak monotonicity though is still weaker than shift-invariance, permitting a broader class of functions to be considered within this framework.
3.6. CONCLUSIONS

Sufficient conditions for the weak monotonicity of several important classes of means derived from the Means of Bajraktarevic were established. Other classes of means were shown to be weakly monotonic, including several of the robust estimators of location and the mode-like average. The recently proposed density-based averages were shown to be weakly monotonic and generalisations of this class, based on weak monotonicity, were also proposed. In the same vein, the weak monotonicity of spatial tonal filters was established and generalisations of this important class of edge-preserving filters were established.

In the next chapter several weakly monotonic averages are applied to the problem of image reduction, in the context of pre-processing in a computer vision task. The performance on this task is used as a basis for comparative evaluation between Beliakov’s mode-like average and several robust estimators of location, attempt to account for the location of data in their computation of an average. These weakly monotonic functions are evaluated against the arithmetic mean and median functions, to establish the possible benefits of using measures of central tendency in image reduction applications.
CHAPTER 4

Image Reduction as Weakly Monotonic Averaging
4.1. Introduction

4.1.1. Motivation for Image Reduction. Commercial digital cameras produce images containing up to 20,000,000 pixels on large format sensor arrays. This is approximately ten times the number of display pixels on a standard High Definition (HD) display and still more than twice the number of pixels available on professional high resolution displays. Subsequently it is not possible to display full images at their native resolution. Similarly, larger displays don’t imply higher resolution and large format TVs and monitors simply use larger pixels (which is why they are optimised for watching the screen from a further distance than smaller screens). While the resolution of display devices is necessarily constrained by the perceptual limits of human vision, the resolution of imaging sensors is constrained only by humanity’s mastery of engineering and understanding of physics. Indeed, camera sensors coming onto the market (in 2014) contain over 70 million pixels [76].

Remote sensing is another source of high resolution images. The GeoEye-1 satellite (launched in 2008) has the highest spatial resolution of a civilian satellite at 0.41 meters (panchromatic; 1.65m multi-spectral), orbiting at a distance of 681 kilometers. A single image swath covers 15.2 kilometers and is sampled by 20000 lines simultaneously, processing 35000 pixels per line. The resulting images contain 7 billion pixels. Both of these examples highlight the current capacity of digital imaging systems to capture imagery with extremely high pixel resolutions that greatly exceed the capabilities of modern display devices. In such cases, visualisation of the entire image field necessarily requires a reduction in image scale (presenting the same spatial information using fewer pixels). Another more common application of image reduction has been driven by the rapid uptake of smartphones and tablet computing devices having small screens and inbuilt high resolution cameras. Viewing images on the inbuilt screen requires image reduction.
Another important motivation for advancing the state of the art in image reduction algorithms is within the field of computer vision. Algorithmic developments in this area now make it possible to operate complex autonomous robots in complicated environments, to detect, identify and track visual targets in video and to perform high level abstract reasoning such as activity detection, or inferring environmental structure from pixel motion in 2D images. Many of the approaches to solving computer vision problems are bound by algorithmic complexity that scales exponentially with the number of pixels; a form of the “curse of dimensionality” [32]. As a pre-processing step, image reduction is frequently used to reduce the number of pixels processed and to thus reduce the time required to process the imagery and produce analysis results. Clearly though, image reduction in this context must not eliminate information that is relevant and necessary to the analysis task being performed and simplistic approaches such as filtering and sub-sampling cannot guarantee the preservation of small scale structured image features that may be important in a given vision task.

Methods for sub-sampling of images involve a three stage process: (1) low pass filtering of the image to limit the highest frequencies (below twice the required sampling rate); (2) sub-sample the filtered image at new pixel locations (using an interpolation technique); and, (3) perform histogram equalisation to fit the intensity histogram of the scaled image as closely as possible to that of the source image. While this approach effectively handles resizing in the presence of noise, it has the same failing as the filter-based noise reduction algorithms. That is, it assumes that important image detail does not exist within the portion of the spectrum being filtered. While this may be true of a snapshot portrait taken on a smartphone, it is certainly not the case in satellite imagery or medical scan data, where much of the important information exists at the limits of spatial and tonal resolution. Subsequently there is a clear need for image resizing algorithms that take into account fine image detail and preserve this (where possible) in the output image.
In this chapter a novel weakly monotonic averaging function is proposed, for use in a local, block-based image reduction algorithm. The evaluation methodology selected to assess the quality of this function is that of image reduction as preprocessing in a facial recognition task, where task performance is used to measure the quality of the reduction as compared to a baseline of no reduction on the same image data. In this way, any effects of image reduction can be evaluated in terms of their impact. This investigation extends the current literature that has considered only monotonic averaging functions within the context of local block-based reduction algorithms and the use of quality measures based solely on functions of pixel differences between the input and output images. The material in this chapter has been published in the following peer-reviewed article:


4.2. A Weakly Monotonic Image Reduction Operator

The mode-like average described by Eq. (33) has several key properties that are ideal for image reduction. Specifically, it returns an aggregate value representative of the significant cluster within the inputs, discounting the influence of outliers (which may be noisy or corrupted pixels, or those not representative of the principle cluster), while not disregarding them altogether. As the requirement of a block-based local reduction operator is to find a representative value of the image feature within the block, provided this feature represents the significant cluster, then the mode-like average should be suitable. This will still be the case even when the image feature is represented by a minority of block pixels, provided the outliers are not themselves tightly clustered. Furthermore, as eqn. (33) is an internal function then the pixels comprising the reduced image will be selected from the local image features that are deemed to be significant with respect to those pixels considered outliers.
4. IMAGE REDUCTION AS WEAKLY MONOTONIC AVERAGING

4.2.1. Reduction Operator. The operator proposed for image reduction by non-monotonic averaging is formulated as a penalty-based function based on eqn. (33) and given by

\[
\text{PMODE}(x) = \arg \min_{y \in \{x_1, \ldots, x_p\}} \{\mathcal{P}(x, y)\},
\]

with

\[
\mathcal{P}(x, x_j) = \sum_{i=1}^{p} w_{ij} \rho(x_i, x_j)
\]

and

\[
w_{ij} = \frac{d(z_i, z_j)}{\sum_{i=1}^{p} d(z_i, z_j)}.
\]

In line with Beliakov [12], the penalty term \(\rho\) is based on the cluster partitioning of the tonal residuals, \(r_{ij} = |x_i - x_j|\),

\[
\rho(x_i, x_j) = \begin{cases} 
    r_{ij} & r_{ij} < \tau \\
    \beta \tau & r_{ij} \geq \tau 
\end{cases},
\]

\(\tau = \alpha r_{(t)}\) and \(\alpha > 0\), \(0 \leq \beta \leq 1\), \(2 \leq t \leq p\).

Taking \(r_{(t)} = r_{\rightarrow}\) and since the average is internal (i.e., \(y \in \{x_1, x_2, \ldots, x_p\}\)) then \(r_{(1)} = 0\), since \(|x_i - x_i| = 0\ \forall i \in \{1, 2, \ldots, p\}\). Since \(r_{(2)} = 0\) when there are at least two identical pixels intensities, \(t\) is chosen so that \(r_{(t)}\) is the smallest non-zero residual, which is a variation on the choice of cluster scale described in [12].

As non-adjacent pixels may have similar intensity values but are less likely to be members of the same image feature, the weights \(w_{ij}\) penalise pixels which are not spatially close neighbours within a block. The weight formula given
above measures the dissimilarity between pixels based on their normalised distance within the local block, where \( d(z_i, z_j) \) is any suitable distance metric over the locations of pixels \( x_i \) and \( x_j \) within the \( m \times n \) block. Since these weights are constant for a given block size, they can be pre-computed during initialisation of the algorithm and within (39) they can be taken as a constant weight vector. From Propositions 9 and 14 it follows directly that the function defined by minimisation of (40) is averaging and weakly monotonic. It is also internal and idempotent. As \( r_0 > 0 \) then \( \tau \) scales if the data is scaled and while the penalty values will scale, the index \( j \) of the pixel that minimises \( P(x, x_j) \) will not. Thus (39) is also homogeneous.

### 4.3. Block-based Image Reduction

#### 4.3.1. Image Representation.** **

Herein an image of \( M \times N \) pixels is a set of scalar valued elements arranged in \( M \) rows and \( N \) columns. Thus an image may be represented by the matrix \( A_{M \times N} \), with elements \( a_{ij} \in [0, 1], i \in \{1, \ldots, M\}, j \in \{1, \ldots, N\} \). The set of all such images is defined by the set \( A_{M \times N} \). Each matrix \( A \) may be represented as a set of column vectors (or, without loss of generality, row vectors), such that \( A = \begin{bmatrix} a_1 & a_2 & \cdots & a_N \end{bmatrix} \), where \( a_i = \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{Mi} \end{bmatrix}^T \). Consequently, the vector \( X = \begin{bmatrix} a_1^T & a_2^T & \cdots & a_N^T \end{bmatrix}^T \) is an equivalent representation of the image as a point in the space \([0, K]^P\), where \( P = M \times N \) and \( K = 2^b - 1 \) (the number of bits used to encode an intensity value). It is assumed, without loss of generality, that when required the set of pixel intensities can be normalised to \([0, 1]\) and thus the intensity space is assumed to be \( \mathbb{I} = [0, 1]^P \). \( X \) is the resultant of vectorisation of the matrix \( A \).

In addition to an intensity value, each pixel has a location within the image matrix, which induces a fixed ordering on the intensities given the vectorisation procedure. For pixel \( X_k \) the vector \( z_k \) will denote its location and its intensity will be given by the scalar \( x_k = I(z_k) \), where the function \( I : \mathbb{N}^+ \times \mathbb{N}^+ \to \mathbb{I} \) is the piecewise-constant intensity field representing the image as detected by the imaging sensor having discrete sensor elements at integer coordinates.
It is proposed that the original image represented by $A$ be replaced with an alternative image $A'$ chosen from the set $A'_{M' \times N'}$, where $M' = \lfloor \frac{M}{m} \rfloor$ and $N' = \lfloor \frac{N}{n} \rfloor$, for a given choice of $m$ and $n$. The procedure for determining the reduced image is that each element $a'_{ij}$ of $A'$ is the solution to a local reduction operation applied to the corresponding subset of $A$, as depicted in Figure 4.1. The pivotal requirement is that the image $A'$ contain a subset the same visual information but at a lower pixel resolution, hence satisfying the fundamental aim of image reduction. An additional, important constraint applied herein is that the pixel intensity values $a'_{ij}$ in the output image should be chosen from the set of input pixel intensities; that is, the reduction operation should be internal (as per Definition 11).

This constraint has an important implication in vision tasks, in that now new intensity values are introduced into the image, as may occur using traditional approaches to image reduction (or filtering), or with non-internal averaging functions. In computer vision tasks - and particularly those relying on the spectral intensity of pixels to determine object classification - introducing spurious intensity data is likely to affect the performance on the task. Additionally, it is expected that this constraint will assist in preserving fine, pixel scale details and prevent the loss of important visual information. It is plausible though that this constraint, while preserving intensity values, will introduce spurious frequencies into the image. It remains to be seen as to whether or not this degrades the performance of the reduction operator.

4.3.2. Block-based Image Vectorisation. A local subset of an image, herein simply called a block, is any contiguous submatrix $A_{ij}$ of $A$, of size $m \times n$, such that $1 \leq m \leq M, 1 \leq n \leq N$ and $(i, j)$ are the row and column index in $A$ of the first element of this block. This submatrix can be vectorised to produce the vector $x = (x_1, x_2, ..., x_p)$, for $p = m \times n$, such that
4.3. BLOCK-BASED IMAGE REDUCTION

Figure 4.1. Image block averaging operation.

\[
A_{ij} = \begin{bmatrix}
a_{ij} & a_{i,j+1} & \cdots & a_{i,j+n-1} \\
a_{i+1,j} & a_{i+1,j+1} & \cdots & a_{i+1,j+n-1} \\
\vdots & \vdots & \ddots & \vdots \\
a_{i+m-1,j} & \cdots & \cdots & a_{i+m-1,j+n-1}
\end{bmatrix}
\]

(43)

Pixel \(a_{qr}\) in the block image has intensity \(x_k = a_{i+q-1,j+r-1}\), where \(k = (r - 1)m + q, \ q = 1...m, \ r = 1...n\) induces a column-major ordering on the block. The location of this pixel in the block is therefore \(z_k = (q,r)\).

The reduction operation on the block \(A_{ij}\) returns a scalar value

\[
\mu = F(x_1, x_2, \ldots, x_p)
\]

that is representative of this block. It is reasonable to require that \(\min(x) \leq \mu \leq \max(x)\) so that the reduction operator is averaging, and that the reduction be weakly monotonic, so that spurious intensities within this block are discounted in the computation of the average. The average \(\mu\) is constrained to be internal,
implying that $\mu \in \{x_1, x_2, ..., x_p\}$. That is, the reduction operation selects one of the pixel values from the block as being the pixel most representative of that block.

4.3.3. Block Processing. As with the work of Paternain [58] the source image will be vectorised using disjoint blocks. Figure 4.2 depicts this disjoint version of local block-based image reduction operation in terms of input blocks and output pixels. The set of $P' = M' \times N'$ image blocks are vectorised to produce the vectors $x_1, x_2, ..., x_{P'}$. These vectors can be processed in parallel to produce the scalar intensities $\mu_1, \mu_2, ..., \mu_{P'}$ corresponding to a vectorised version of the output image. The final image is reconstructed by placing value $\mu_k'$ at coordinate $(i', j')$ such that $i' = ((k' - 1) \% M') + 1, j' = (k' - i') / M' + 1$, and integer division is performed.

An algorithm to apply this aggregation function to a given image is given in Algorithm 1. The input vectors $w_k, k = 1, ..., p$ correspond to the weights for each candidate value $y = x_k$ and are thus $w_k = [w_{1k} \ w_{2k} \ \cdots \ \ w_{pk}]^T$. The operation sort($w_k, S$) reorders the elements of $w_k$ according to the order given by the index set $S$, corresponding to the permutation $\sigma$ such that $r_\sigma = r_\rho$. This index matrix is obtained as an output of the sort operation that produces the ordered residuals.
Algorithm 1 Disjoint block image reduction

```
Input: Matrix $A_{M \times N}$, parameters $m, n, \alpha, \beta, t, \epsilon$.
Output: Matrix $A'$ of dimension $\lfloor \frac{M}{m} \rfloor \times \lfloor \frac{N}{n} \rfloor$.

Init:
(1) $W_{p \times p} \leftarrow \left[ \begin{array}{c} w_1 \ w_2 \ \cdots \ w_p \end{array} \right]$
(2) $R_{p \times p} \leftarrow \left[ \begin{array}{c} r_1 \ r_2 \ \cdots \ r_p \end{array} \right] = [0]_{p \times p}$

Main:
(1) for $i'$ in $\{1, \ldots, \lfloor \frac{M}{m} \rfloor \}$ do
(2) for $j'$ in $\{1, \ldots, \lfloor \frac{N}{n} \rfloor \}$ do
(3) $i \leftarrow 1 + (i'-1)m$, $j \leftarrow 1 + (j'-1)n$
(4) $x \leftarrow \text{vectorise}(A_{ij})$
(5) for $k$ in $\{1, 2, \ldots, p\}$
(6) $r, S \leftarrow \text{sort}(\left| x - (x_k, x_k, \ldots, x_k) \right|, \uparrow)$
(7) $t \leftarrow \min\{l \mid r(l) > 0; \ l = 2, \ldots, p, \ r(l) \in r\}$
(8) $\tau \leftarrow \alpha \max(\epsilon, r(t))$
(9) for $l$ in $\{2, \ldots, p\}$ do
(10) if $r(l) \geq \tau$ then $r(l) \leftarrow \beta \tau$
(11) end
(12) $w \leftarrow \text{sort}(w_k, S)$
(13) $\mathcal{P}(x, x_k) \leftarrow w^T r$
(14) end
(15) $k^* \leftarrow \arg\min_{k=1,\ldots,p}\{\mathcal{P}(x, x_k)\}$
(16) $a'_{ij} \leftarrow x_{k^*}$
(17) end
(18) end
```

The two inner loops within this algorithm are efficiently implemented as matrix operations in MATLAB (or similar software) and the individual image blocks may be processed in parallel using suitable parallel matrix operations (again available in MATLAB), thus removing the outer loops. This algorithm is also trivially parallelised for implementation on SIMD architectures, such as modern GPUs.

4.4. Experiments in Image Reduction: Facial Recognition

4.4.1. Facial Recognition Problem. The task of facial recognition is an important problem within computer vision, with applications in biometric systems, security, surveillance, image tagging and even human computer interaction. As a practical application it represents a complex computational problem, with performance results heavily dependent on the images used to train and evaluate a
facial recognition model. As performance on this task is quantifiable it is appropriate to use the facial recognition task as a metric for assessing image processing algorithms.

An additional motivation for this choice was that image reduction is often a pre-processing step used in many computer vision applications as it reduces the computational complexity and subsequently the processing time. Additionally, the same PCA-based method used herein is an information-theoretic approach to reducing the dimensionality of the training and recognition problems. Hence, the goals of image reduction (to reduce the dimension of an image while retaining its significant information) are aligned well with the PCA-based methods for facial recognition. For these reasons the facial recognition task - and specifically the Eigenfaces method - was selected as a means by which performance of image reduction could be assessed in an objective manner, without the need to reconstruct images at the unreduced scale from their reduced versions.

4.4.2. Eigenfaces method for Facial Recognition. The Eigenfaces method for facial recognition [78] [39] [87] describes a set of faces by a feature space that spans the significant variations among that set. The algorithm uses Principal Component Analysis (PCA, also known as the Karhunen-Loève transformation) to identify the set of orthonormal basis vectors that maximise the variation along each dimension of the feature space (the eigenvectors of the space corresponding to the directions having largest eigenvalues).

A set of images of size $M \times N$ containing faces are expressed by the vectors $X_1, X_2, ..., X_Q$, each of length $P = M \times N$. Typically $Q \ll P$. This set has an average image defined by the vector $\bar{X} = \frac{1}{Q} \sum_{i=1}^{Q} X_i$, and each face differs from the average according to the vector $Z_i = X_i - \bar{X}$.

Using the technique of Turk & Pentland [87] the non-zero eigenvectors $u_i$ of the $P \times P$ matrix $C = \frac{1}{Q} \sum_{j=1}^{Q} Z_j Z_j^T$ can be computed efficiently by finding the eigenvectors $v_j, j = 1...Q$ of the $Q \times Q$ matrix $L$ that has elements $L_{jk} = Z_j^T Z_k$. Subsequently,
The vectors \( \mathbf{u}_i, i = 1...Q \) then satisfy the fundamental PCA problem

\[
\mathbf{u}_k = \arg \max_{\mathbf{u} \in [0,1]^P} \left[ \frac{1}{Q} \sum_{i=1}^{Q} (\mathbf{u}^T \mathbf{Z}_i)^2 \right]
\]

subject to the constraint that

\[
\mathbf{u}_j^T \mathbf{u}_k = \delta_{jk} = \begin{cases} 
1, & \text{if } j = k \\
0, & \text{otherwise}
\end{cases} \forall j, k = 1...P
\]

The eigenvalues of \( \mathbf{C} \) are given by

\[
\lambda_k = \max_{\mathbf{u} \in [0,1]^P} \left[ \frac{1}{Q} \sum_{i=1}^{Q} (\mathbf{u}^T \mathbf{Z}_i)^2 \right]
\]

however, given the sample size of training instances there will be at most \( Q \) non-zero values (the rationale for the approach used by Turk).

Any new image \( \mathbf{X} \) can be approximated by a linear combination of the basis vectors, such that

\[
\hat{\mathbf{X}} = \sum_{i=1}^{Q'} w_i \mathbf{u}_i
\]

where \( Q' \leq Q \) is a (possibly restricted) set of the eigenvectors corresponding to the largest eigenvalues and the weights \( w_i \) are determined by

\[
w_i = \mathbf{u}_i^T (\mathbf{X} - \bar{\mathbf{X}}).
\]

The set of weights for a given approximation form a tuple \( \mathbf{w} = (w_1, w_2, ..., w_{Q'}) \) which can be considered as a point in face space. The term Eigenface arises as
the eigenvectors $u_i$, when converted back into an $M \times N$ matrix $U$, form an image that depicts face-like structures and these images form a suitable basis for this face space.

Facial recognition can be carried out using a given set $U$ of Eigenfaces by generating, for a given query image (called a probe) its set of weights, according to Eq. (49). This point is then compared against a set of previously classified points in this space, known as the gallery set using an appropriate metric for measuring distance between points in this space. As the basis for defining points in face space is orthogonal, common metrics such as the $L_1$ and $L_2$ norms perform reasonably well. The distances to all gallery points are ordered from minimum to maximum and the nearest neighbour is returned as the best match.

4.4.3. The FERET Evaluation Protocol. The FERET evaluation protocol [67] was developed by the National Institute of Standards and Technology (NIST) as technical agents for the U.S. Department of Defence Counter Drug Technology Program. The greyscale version (original FERET database) used in this present study consists of 14,034 images of 1199 individuals. The FERET evaluation protocol determines the standard subsets of these images for training, gallery and probe images. The standard gallery consists of 1196 frontal images - one per selected subject. The Fb probe set used for experiments in this paper consists of 1195 images of the same subjects (again, 1 per subject, with 1 subject missing). The probe set images were collected at the same sitting as the gallery set images, with one randomly placed in the gallery and the other in the Fb probe set. The training set consists of 500 frontal images selected randomly from those images not in the gallery or probe sets. The training set does not guarantee that each person within the gallery and probe set is within the training set. The gallery and probe set design ensures that each subject in the probe set is within the gallery set, yet it guarantees that the images of these subjects are different. Unfortunately these images cannot be published due to copyright reasons, but they are accessible to interested readers through application to NIST.
Performance of a given algorithm on the facial recognition task is evaluated by computing the ordered distances between each image in the probe set and every image in the gallery set. The rank at which the correct classification is obtained is used to determine the success of the algorithm. A particularly common means of displaying this performance is the Cumulative Match Score (CMS) (also called the Cumulative Match Characteristic, or CMC), which expresses the probability that a correct match will be found at rank-n. The rank-1 CMS is the proportion of probe images that returned the correct classification as the nearest neighbour. The rank-n CMS is the proportion of probe images that returned the correct classification within the n-nearest neighbours. One can view the CMS as reporting the probability that a probe selected at random has a matching facial point from the gallery within a contour containing n gallery points. Clearly as this contour is expanded the CMS value should approach 1. In this work the CMS curves from the various experiments are normalised with respect to a baseline performance, since only one algorithm is being used under different scenarios. This will be explained further in the next section.

4.4.4. Experiment Design. All images in the FERET database were first transformed with a non-reflective similarity transform to ensure that the eyes and nose were aligned to base positions. The coordinates of these features in each image were supplied by NIST with the data set. The image was centered on the face and cropped to remove the background. Where necessary the resulting face region was scaled to maintain a constant interocular distance in the final image and an image size of $192 \times 192$ pixels. No histogram equalisation or illumination correction was applied to images, in order to minimise any potential bias of the results due to these pre-processing methods.

To conduct the experiments in this work each aligned image within the Fb probe set was duplicated and corrupted by noise. Images corrupted by speckle noise had each pixel’s intensity ($I_0$) modified according to $I = I_0(1+\eta)$, where $\eta$ is
a zero-mean normally distributed random number with variance $\sigma$ (not to be confused with the permutation $\sigma$ described in Section 4.2). Images corrupted by impulse (‘salt & pepper’) noise had a proportion $\sigma$ of their pixels set (uniformly) randomly to either 0 or 1. Data sets of corrupted probes for $\sigma = \{0.02, 0.05, 0.1, 0.2\}$ were constructed for both noise types.

Three experiments using the clean and corrupted data sets were conducted to obtain results from which the performance of the considered aggregation methods could be assessed. In all experiments, the training set and gallery set were comprised of clean images and both clean and corrupted versions of the probe sets were evaluated. The clean probe images correspond to $\sigma = 0$. The experiments were:

**Experiment 1: Baseline.** The Eigenfaces method was applied to non-reduced images and clean and corrupted versions of the Fb probe set were evaluated. CMS curves for corrupted probes were normalised against the curve for clean probes. This baseline (of recognition performance of non-reduced, clean probe images) would corresponds to the “best” performance expected of the Eigenfaces method on a given data set.

**Experiment 2: Reduction of Images.** Each reduction operator was applied to all clean and corrupted images within the database and the facial recognition model was trained using reduced, clean images in the training set. Clean and corrupted versions of the probe set were reduced.

**Experiment 3: Reduction of Eigenfaces.** Each reduction operator was applied to the set $U$ of Eigenfaces generated during experiment 1 and then to the gallery and probe images (both clean and corrupted image sets). The reduced Eigenfaces were used to transform the reduced gallery and probe set images into face space and CMS performance curves were computed. The baseline for normalisation was the performance of that operator in experiment 2 using clean, reduced images.

The operators evaluated in these experiments were: (arithmetic) Mean, Median, Shorth, LTS, LMS and PMODE (given by Eq. (39)) and each was applied
using a $3 \times 3$ block, generating reduced images of size $64 \times 64$ (i.e., $\frac{1}{9}$ their original size). For the PMODE operator the parameters $\alpha$ and $\beta$ were selected using subjective visual assessment of randomly selected corrupted images over a range of possible values for these parameters. For both noise types the values of $\alpha = 10$ and $\beta = 1.0$ gave good performance. The distance metric used in 41 was the infinity norm.

4.4.5. Results. Results for experiment 1 are presented in Figure 4.3, with Figure 4.3(a) showing the recognition performance of the Eigenfaces algorithm on non-reduced probe images corrupted by speckle noise. Figure 4.3(b) shows recognition performance of the algorithm on non-reduced probe images corrupted by impulse noise. Each CMS curve is normalised against performance on clean probe images. To reiterate, these scores are normalised against the performance on clean images.

Results for experiment 2 are presented in Figures 4.4-4.10. Figure 4.4 shows the normalised performance (averaged over 50 ranks) for each reduction operator
on each noise type and $\sigma$ value. The baseline for normalisation of these CMS curves was the performance on non-reduced clean images in experiment 1.

Figures 4.5-4.10 show the normalised CMS curves for the (arithmetic) Mean, Median, Shorth, LMS, LTS and PMODE reduction operators respectively, which were averaged over 50 ranks to produce the respective curves in Figure 4.4.

Results for experiment 3 are presented in Figures 4.11-4.16. Averaged normalised results for each reduction operator are presented in Figure 4.11. Figures 4.12-4.16 shows the normalised CMS curves for each operator, which as with experiment 2, were averaged over 50 ranks to produce the values in Figure 4.11.

4.4.6. Discussion. It is apparent from Figure 4.3 that, as expected, recognition performance degrades with increasing noise levels (increasing variance for speckle noise and increasing proportion of corrupted pixels for impulse noise). The shape of the curve for speckle noise corruption is interesting and worthy of brief consideration. The initial decline in performance with increasing rank (up to about rank 6) and the fact that this decline worsens with increasing noise
4.4. EXPERIMENTS IN IMAGE REDUCTION: FACIAL RECOGNITION

Figure 4.5. Normalised performance of Mean reduction operator applied to images. (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

Figure 4.6. Normalised performance of Median reduction operator applied to images: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
Figure 4.7. Normalised performance of Shorth reduction operator applied to images: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

Figure 4.8. Normalised performance of LMS reduction operator applied to images: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
4.4. EXPERIMENTS IN IMAGE REDUCTION: FACIAL RECOGNITION

**Figure 4.9.** Normalised performance of LTS reduction operator applied to images: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

**Figure 4.10.** Normalised performance of PMODE reduction operator applied to images: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
Figure 4.11. Normalised average performance versus noise level for reduced eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

Figure 4.12. Normalised performance of the Mean reduction operator applied to Eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
4.4. EXPERIMENTS IN IMAGE REDUCTION: FACIAL RECOGNITION

Figure 4.13. Normalised performance of the Median reduction operator applied to Eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

Figure 4.14. Normalised performance of the LMS reduction operator applied to Eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
Figure 4.15. Normalised performance of the LTS reduction operator applied to Eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.

Figure 4.16. Normalised performance of PMODE reduction operator applied to Eigenfaces: (a) Speckle-corrupted probes. (b) Impulse-corrupted probes.
variance suggests that the effect of noise is related to the relative scale of noise to
distance between the probe and gallery images in feature space. However, once
the distance increases sufficiently the effect of noise appears relatively constant
relative to baseline. This would suggest that noise causes a local reordering of
gallery points in the space and that the effect is of course diminished as the probe
image moves out of the local cluster around the gallery point. Interestingly, the
effect of impulse noise appears to be negligible at low levels of corruption yet
rises sharply above 10% of the pixels being set as outliers. This makes sense since
the probe image will be shifted toward the boundary planes of the image space,
which will transform to planar surfaces within the feature space, which is clearly
making discrimination more difficult.

Figure 4.4 shows the performance of each reduction operator for each noise
level, averaged over all 50 ranks of their respective normalised CMS curves, which
appear in Figures 4.5-4.10. The black dashed lines in Figure 4.4(a) and 4.4(b)
represent the average performance of the baseline, non-reduced clean and cor-
rrupted probes. That is, the black lines indicate the averages of the curves in
Figure 4.3. This graph provides for an easy comparison of the ability of each op-
erator to overcome noise and how image reduction clearly improves recognition
performance on noisy probe images. It is clear that each of the reduction oper-
ators, except the arithmetic mean, are able to reject speckle noise and improve
performance relative to baseline at the same noise level. This is to be expected
of robust averaging functions. However, performance of the Median operator on
speckle noise was not as good as the weakly monotonic averages. Again, this is
to be expected as the Median average over disjoint blocks may not sufficiently
decrease noise variance in the reduced image. The reductions performed with the
robust estimators of location show only slight decline in performance at increasing
noise levels (for both noise types), while the P-MODE operator actually improves
performance with increasing speckle noise and maintains performance under cor-
rruption due to impulse noise. This supports the claim that this operator selects
a value representative of the principle cluster in each block, since with increas-
ing noise, the noisy pixel intensities shift further from the tonal cluster, making
discrimination of the cluster easier. With impulse noise though, as more pixels
within a block are set to minimum or maximum values, the chance of having a
cluster of outliers increases. On average the P-MODE is able to reject outliers
effectively, but performance is slightly worse than in the speckle noise cases. The
P-MODE operator outperforms the other robust estimators of location in average
recognition performance and is clearly the better operator for this application.
This is evidenced by Figure 4.10, which shows that the P-MODE operator per-
forms nearly as well as recognition when analysis clean, non-reduced probe images
(with this baseline performance level represented by the black dashed line in this
figure).

Experiment 3 was conducted to evaluate the performance of reduction opera-
tors when applied to reduction of the feature space dimensionality. This is useful
in two scenarios: (1) where the probe images are captured at a different scale to
the recognition model that was previously trained (and the training images are
not available for resizing and retraining); and/or (2) the computation is to be
computed under constrained computational situations (such as a microprocessor
with limited memory resources) and a reduced model is essential. Figure 4.11
shows much the same performance as when reduction is performed pre-training
on the images, which is an interesting result that shows that image reduction
techniques can be effectively applied for dimensionality reduction in vector fea-
ture spaces and that retraining of the model to cope with different image scales
is not necessary (only reduction of the existing model). Figures 4.12-4.16 show
slight improvement in performance over the comparable curves in experiment 2
(recall that the baseline in these images is performance of that operator in exper-
iment 2), indicating that reduction of the trained model produces slightly better
discriminatory performance in feature space. This is likely due to their being too
many feature dimensions being retained in this model, such that the extra di-
mensions actually degrade performance. This suggests another unexpected novel
outcome of these experiments: that image reduction techniques may be applied
to determine the required number of feature dimensions necessary to sustain a
given recognition performance level.
4.5. CONCLUSIONS

Of final note is Figure 4.16, which again shows the P-MODE operator performs better than the other operators and in particular, on images corrupted by speckle noise, offers better performance when applied to reducing eigenfaces, rather than the original source images. Recalling that this operator actually improved performance over clean, non-reduced baseline images when reducing images directly, shows that that reduction of eigenfaces is clearly the best approach to use. This is reinforced by Figure 4.11 and realising that this means that by using the P-MODE operator and reducing eigenfaces, facial recognition performance is increased regardless of whether the probe images are noisy or not and performance is better than not applying image reduction techniques. Clearly image reduction should be performed in this domain, as it offers both efficiency improvements and accuracy improvements. Of course there will be a lower limit, beyond which reduction will remove too much information and accuracy will suffer. What this lower limit is remains to be seen and is a direction for future research in computer vision and image processing research.

4.5. Conclusions

This Chapter has presented a novel weakly-monotonic averaging aggregation function and applied it to the task of image reduction as a pre-processing step in a facial recognition problem. This operator was compared to several other weakly monotonic functions - the robust estimators of location - as well as the arithmetic mean and median (both monotonic averages). Facial recognition was used to assess the capacity of each operator to preserve visual information during reduction, while rejecting noise and several different levels of speckle and impulse noise were considered.

It was found that all of the weakly monotonic functions, as well as the median, were able to effectively reject impulse noise, even at significant levels of corruption. On speckle noise, the weakly monotonic functions outperformed the monotonic functions and the P-MODE averaging function proposed in this chapter outperformed the considered robust estimators of location. In all cases the
arithmetic mean performed poorly and recognition accuracy degraded with increasing noise levels. This result is to be expected as this operator is not robust to noise or outliers within the input data.

The experiments conducted showed that the proposed weakly monotonic, mode-like averaging function faithfully rejects outliers from the principle cluster in image blocks, even at higher noise levels. No significant performance difference was found between applying the reduction operators to the original images to produce the facial recognition model, as opposed to reducing the model eigenvectors trained on full size images. However, the latter operation permits far greater flexibility in computer vision applications, since it suggests that a feature vector based model can be trained offline, on sufficiently high resolution images and that the eigenvectors of this space can then be reduced using image reduction techniques to either determine the necessary number of feature dimensions for a given recognition rate, or to match the dimensionality of the trained model to that of the probe and gallery images used in a specific recognition application.

The over-arching result of these experiments is that weakly monotonic averaging functions do preserve image information relevant in a computer vision application and that reduction using these averages may actually improve task performance. However, the question as to whether the information that is preserved is visually relevant remains unanswered. The image content that permits accurate discrimination between faces using a distance metric in feature space may not be visually relevant or appealing to a human eye. In situations where image reduction is to be applied to enable visualisation of high resolution source images on low resolution devices, or where a lower resolution version of an image is necessary for initial assessment by a human, then it is essential that visual structure is preserved during the reduction operator. In the following Chapter this issue is addressed through the modification of the proposed P-MODE operator, to account for the visual structure of pixel groups in each local block.
CHAPTER 5

Weighting Functions for Aggregation of Pixel Clusters
In the previous chapter the penalty-based mode was proposed as an operator for image reduction in the presence of noise. This function produced a weakly monotonic average that coincided with the minimum of a weighted penalty function, such that the average intensity value was representative of the most compact spatial and tonal cluster. Spatial dissimilarity between pixels was accounted for in the penalty weights, which were computed using a distance function. While distance-based dissimilarity is used frequently in image processing algorithms, it assumes clusters are circular (hyper-spheres in the n-D case) and thus does not adequately account for the local spatial organisation of intensity values that represent fine detail in an image.

Given the statistical properties of noise within images it is highly unlikely that a spatially coherent set of pixels having similar tone are caused by noise and thus such sets are far more likely to signify important detail that should be preserved by the averaging operation. In weighting the penalty of tonally coherent sets, they should be favoured over any other set having the same tonal variation but a less coherent spatial organisation. Unfortunately distance based weights applied in the previous chapter do not permit explicit preferencing of one shaped cluster over another.

In this chapter, the aim is to replace the distance-based weights proposed in Section 4.2 with a weighting function that incorporates information regarding the spatial structure of a group of pixels, and that appropriately orders candidate clusters of pixels according to their spatial compactness. Consequently, the requirement is for a weighting function over the power set $2^P$, where $P$ is the index set for the input vector $\mathbf{x}$ and $w : 2^P \to [0, 1]$.

Some simple measures of compactness used in cluster validation problems, such as those based on ratios of the sum of inter-pixel differences to the number of pixels (i.e. average interset distance), or cluster radius, are not adequate, as they are not necessarily increasing in cluster cardinality and will not produce the desired ordering on compactness. On the other hand, this requirement is precisely
5. WEIGHTING FUNCTIONS FOR AGGREGATION OF PIXEL CLUSTERS

the monotonicity condition used in the definition of fuzzy measures and therefore it makes sense to look for a solution in the class of fuzzy measures. Before continuing with the development of suitable fuzzy measures of cluster compactness, some consideration of other cluster compactness measures is required so that the relevance and significance of this contribution can be established.

This material has been submitted for publication in the following articles:


5.2. Geometry-based measures of cluster compactness

5.2.1. Distance-based measures. The most prevalent approach within the literature for incorporating spatial information into a problem formalism, especially in applications such as noise reduction, is to weight the contribution of each pixel based on its distance from some fixed location. Pixels that are further apart are considered more dissimilar than pixels that are close, even if the intensity values are the same. Such approaches use a distance function on a normed vector space.

**Definition 72.** A function \( d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0, \infty) \) is a distance function if it satisfies

\[ d(x, y) = d(y, x) \quad \text{for all} \quad x, y \in \mathbb{R}^n \]

\[ d(x, z) \leq d(x, y) + d(y, z) \quad \text{for all} \quad x, y, z \in \mathbb{R}^n \]

In both instances the authors are listed alphabetically and not in order of contribution. H. Vu and G. Li assisted with the numerical computation of the measure values and G. Beliakov assisted in formulating suitable measure functions and drafting of the second paper. My contributions include (but are not limited to) the proposal to use fuzzy measures for cluster compactness and the design of the decomposing fuzzy measure, establishing suitable reference points and constraints for the measures, analysis of the outcome measures and evaluations of the measures in image reduction tasks. I also wrote the first paper and contributed significant portions of the second.
1. \( d(x, y) = 0 \iff x = y \) (strict minimum)
2. \( d(x, y) = d(y, x) \) (symmetry)
3. \( d(x, z) \leq d(x, y) + d(y, z) \) (triangle inequality)

The most commonly used distance functions belong to the Minkowski family, given by:

\[
d_k(x, y) = \left( \sum_{i=1}^{n} (x_i - y_i)^k \right)^{\frac{1}{k}}.
\]

These functions are stable to translation (shift-invariant), rotation invariant for \( k = 2 \), though not scale invariant. Commonly used examples of Minkowski distance functions are:

1. **Manhattan distance**: \( d_1(x, y) = \sum_{i=1}^{n} |x_i - y_i| \)
2. **Euclidean distance**: \( d_2(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = \sqrt{(x - y)^T (x - y)} \)
3. **Infinity distance**: \( d_\infty(x, y) = \max_{i \in \{1, \ldots, n\}} |x_i - y_i| \).

If a cluster prototype is taken as lying at the cluster centre, a general measure of membership can be defined on the data space based on the dissimilarity between pixels, which can be taken as a reciprocal function of the distance. The Euclidean distance is often used in such circumstances due to its rotational invariance. It leads to hyper-spherical clusters and assumes that pixels at equivalent distances from the centre, but at different orientations, are equally part of the same cluster and thus equally similar to the prototype.

5.2.2. **Distance and Shape Measures.** A more flexible approach uses a cluster specific distance function that incorporates both distance and shape (distribution in each axis) into the distance measure. For example, the Mahalanobis distance

\[
d(x, y; \Sigma) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)}
\]

incorporates the covariance matrix \( (\Sigma) \) of the data into the computation, which implicitly defines the dispersion of data in each dimension relative to the others.
WEIGHTING FUNCTIONS FOR AGGREGATION OF PIXEL CLUSTERS

(i.e., it provides a first order approximation of the shape of the cluster). The Mahalanobis distance is equivalent to the Euclidean distance computed in a properly scaled eigen-system of the covariance matrix. The determinant of this covariance matrix provides a measure of the volume of the ellipsoid containing the data (see Subsection 5.2.3 below).

While distance functions are often used as the basis for cluster membership functions, they are typically combined with a reciprocal scaling function to describe (often non-linearly) the degree of membership of the sample from the cluster. Typically, radial scaling functions (also known as kernel functions) are used for this purpose.

**Definition 73.** A **kernel function** is a function \( f : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \) satisfying

1. \( \lim_{r \to \infty} f(r) = 0 \); and,
2. \( \forall r_1, r_2 \in \mathbb{R}_0^+ : r_2 > r_1 \Rightarrow f(r_2) \leq f(r_1) \). (\( f \) is a monotonically decreasing function).

It is sometimes convenient to explicitly include \( f(0) = 1 \), so that all points at the cluster centre are considered equivalent to the prototype.

Examples of frequently used kernel functions are:

1. Cauchy function: \( f(r; a, b) = \frac{1}{r^a + b} \)
2. Gaussian function: \( f(r; a, b) = \exp \left( -\frac{1}{2} r^a \right) \)

The tonal and spatial tonal filters, which compute weighted averages as a method of denoising, rely almost exclusively on Euclidean distance and Gaussian kernel functions. For example, the anisotropic diffusion can be expressed in the continuous domain by the function

\[
F(x) = \frac{1}{C(x)} \int_{y \in B_x} \exp \left( -\frac{1}{h^2} \left| y - x \right|^2 \right) \exp \left( -\frac{(u(x) - u(y))^2}{\sigma^2} \right) u(y) dy,
\]

where \( u(x) \) is the pixel intensity at location \( x \). Filters such as this assume that the data is equally distributed in each axis, which is a reasonable assumption when the filter is applied over a ball \( B_x \) centered at \( x \). Even in discrete form \( B_x \) is
typically a square subset of an image and thus the data are distributed equally in each axis. More generally though the covariance matrix describes the distribution of data in each axis. For a two dimensional data set this is represented by the matrix

\[
\Sigma = \begin{bmatrix}
\sigma_{x}^2 & \sigma_{xy} \\
\sigma_{xy} & \sigma_{y}^2
\end{bmatrix}.
\]

The standard deviation \((\sigma)\) is, in the one dimensional case, a measure of the average distance of a data point from the maximum likelihood estimator of the cluster. In two dimensions an equivalent measure can be obtained by performing an eigenvector decomposition of \(\Sigma\). This yields a measure that more accurately represents the distribution of the data within the \textit{principal component} directions, given by the quantities:

\[
\sigma_1 = \sqrt{\sigma_{x}^2 \cos^2 \phi + \sigma_{y}^2 \sin^2 \phi + 2\sigma_{xy} \sin \phi \cos \phi},
\]

\[
\sigma_2 = \sqrt{\sigma_{x}^2 \sin^2 \phi + \sigma_{y}^2 \cos^2 \phi - 2\sigma_{xy} \sin \phi \cos \phi},
\]

\[
\phi = \arctan \frac{2\sigma_{xy}}{\sigma_{x}^2 - \sigma_{y}^2},
\]

which are depicted in Figure 5.1 in relation to the 1—standard deviation elliptical contour defined by \(\Sigma\) (in two dimensions). The values \(\sigma_1\) and \(\sigma_2\) can be used to scale distances in each principal direction, or they can be combined to form a single measure of the shape of the cluster.

5.2.3. Area-based measures. The eigenvector decomposition of \(\Sigma\) also shows that the determinant of the covariance matrix provides a measure of the area of the ellipsoid. Since

\[
\Sigma = R \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R^{-1},
\]

where \(R\) is a rotation matrix and \(\lambda_1, \lambda_2\) are the eigenvalues, the area of the ellipsoid is given by

\[
\text{Area} = \pi \sqrt{\det(\Sigma)} = \pi \sqrt{\lambda_1 \lambda_2}.
\]
with the columns of $\mathbf{R}$ being the normalised eigenvectors and $\lambda_1, \lambda_2$ being the eigenvalues of $\Sigma$. Since $|\mathbf{R}| = 1$ then $|\Sigma| = \lambda_1 \lambda_2$ and subsequently the area of the ellipsoid is

$$A = \pi \sqrt{|\Sigma|} = \pi \sqrt{\lambda_1 \lambda_2} = \pi \sigma_1 \sigma_2. \quad (50)$$

The area computed by (50) is invariant to rotations and translations of the data and is symmetric with respect to reflections. It would therefore be a possible measure of cluster compactness. Unfortunately it will not suffice in the current context where clusters are subsets of pixels in a regular grid, as in certain cases clusters produce an area of zero (specifically those representing a single row or column of the block). Additionally, clusters of non-adjacent pixels will produce larger areas than those having the same number of elements but more compactly arranged. Unfortunately though there is not simple function of these areas that produces a measure that is monotonic with respect to cluster cardinality.

### 5.3. Structural Patterns in Image Blocks

Consider a $3 \times 3$ block of pixels with grey level intensities such as that in Figure 5.2(a). Figures 5.2(b)-(d) depict several clusters of pixels within this block that contain pixels that are plausibly tonally similar. These clusters are possible structural patterns representing fine detail within the image. Membership in cluster within an $m \times n = p$ block of pixels will be denoted by an index set $\mathcal{S}$
corresponding to the elements of the binary vector $\mathbf{x} = (b_1, b_2, ..., b_p)$ that have the value 1, where $b_i = 1$ indicates that the $i$'th pixel is a member of the cluster and $b_i = 0$ indicates that this pixel is an outlier of the cluster. Thus, for the clusters in Figure 5.2.3(b)-(d), the index sets are, respectively, $\{1, 2, 5, 7\}$, $\{4, 7, 8, 9\}$ and $\{3, 6\}$ (assuming a column-major vectorisation of the block, as in eqn. (43)).

Considering these clusters, the requirement is to determine an appropriate precedence ordering over the $2^p$ clusters such that more compact clusters are preferred to less cohesive ones. This problem may appear relatively simple given the examples in Figure 5.2(b) and (c) and a reasonable compactness ordering for these samples may be $\{4, 7, 8, 9\} \prec \{1, 2, 5, 7\} \prec \{3, 6\}$, assuming that clusters with larger cardinality are better than those with smaller. However, there are many cases that may not be so obvious nor correctly differentiated using distance or shape based measures.

For example, consider the clusters $\{1, 2, 3\}$ and $\{1, 2, 5, 6\}$, depicted in Figure 5.3(a) and (b) respectively. Using average Euclidean inter-pixel distance (with smaller numbers describing more compact clusters), the $3 \times 1$ cluster would be slightly favoured ($\bar{d} = 1.333$) over the 4-element cluster ($\bar{d} = 1.344$). Using average Manhattan inter-pixel distance would definitely favour the $3 \times 1$ cluster ($\bar{d} = 1.333$ versus $\bar{d} = 1.667$). However, considering now the cluster $\{1, 2, 4, 5\}$ depicted in Figure 5.3(c): it would be favoured in the first case (having an average Euclidean inter-pixel distance of $\bar{d} = 1.138$) but would be equivalent to the $3 \times 1$ cluster when using Manhattan distance (having a value of $\bar{d} = 1.333$). While the result $2 \times 2(square) \prec 3 \times 1$ is enforceable given the Euclidean metric, in neither distance function case is the preference $2 \times 2(skewed) \prec 3 \times 1$ supported by this measure.
Given these considerations it does not seem reasonable to enforce a measure such as average inter-pixel distance, or the area of the cluster (as this will produce similar undesirable orderings). Thus an alternative approach to building a cluster compactness measure that ensures monotonicity with respect to set cardinality should be considered.

This measure has the following requirements. Let $S$ denote a subset of pixels in an $m \times n$ block and $SC$ a cluster size and compactness measure. The function $SC(S)$ must be:

(1) increasing in $|S|$;

(2) invariant with respect to translation, reflection and rotation (at least rotations by multiples of 90 degrees in the context of typical rectangular arrays of pixels); and,

(3) capable of discriminating between compact groups and sets of separated pixels of the same cardinality.

For convenience $SC$ will be normalised so that its range is $[0, 1]$. The proposal herein is that these requirements can be satisfied by a suitably constructed fuzzy measure of the power set $2^p$. In the remainder of this chapter several fuzzy measures of cluster compactness are presented and evaluated.

### 5.4. A fuzzy measure of pixel cluster compactness

For use as a penalty weight, a non-symmetric fuzzy measure (Defn. 35) is required so as to effectively differentiate between compact and scattered groups of pixels of the same cardinality, such as those depicted in Figure 5.4. However,
5.4. A FUZZY MEASURE OF PIXEL CLUSTER COMPACTNESS

Figure 5.4. Example of compact and scattered clusters.

Figure 5.5. Examples of adding a pixel to a cluster.

also required is symmetry (Defn. 38) among those subsets of $A$ representing translations, rotations or reflections of the corresponding groups of pixels.

Furthermore the fuzzy measure must be non-additive (Defn. 37). Consider a compact cluster of pixels - such as that shown in Figure 5.5 - and add another pixel to the cluster; the measure of this larger set should depend on how close this additional pixel is to the original cluster, rather than simply be the sum of the measure value of the two subsets of this cluster.

Herein three different approaches to constructing fuzzy measures that satisfy these requirements are presented. In the first case some additional information regarding the desired fuzzy measure values will be incorporated. A reference point is a specified value of $v$ for a particular subset $A$, which is believed to be a reasonable choice. For example, specifying $v(A) = 1$ for all compact $A$ of cardinality $|A| \geq k$, for some $k < n$ and where compactness means that each pixel in $A$ has a neighbour in $A$ (that is, for all $a \in A : d_H(a, A) = 1$ and $d_H$ is the Hausdorff distance) Furthermore, constraints such as $v(A) \geq v(B) + \delta$, for a
some $\delta > 0$, will be imposed when the measures of $A$ and $B$ should differ at least by $\delta$. It is assumed that there are $r$ reference points and $c$ inequality constraints.

5.4.1. A Sugeno-type fuzzy measure. The first approach considered is based on an analogue of the Sugeno $\lambda$–fuzzy measures given by eqn. 8, wherein all values $v(A)$ are immediately computed from $n$ independent values $v(\{i\}), i = 1, \ldots, n$, using an explicit formula

$$v\left(\bigcup_{i=1}^{m}\{i\}\right) = \frac{1}{\lambda} \left( \prod_{i=1}^{m} (1 + \lambda v(\{i\})) - 1 \right), \quad \lambda \neq 0.$$  

While Sugeno fuzzy measures are popular due to their simplicity and a small number of parameters, they are too restrictive for the current application as they do not differentiate between compact and non-compact groups (this information is not conveyed in the form of initial values of $v$ at the singletons). In the current context all $v(\{i\}), i = 1, \ldots, n$ would necessarily be the same (this value can be fixed at some small number, $\epsilon$). In contrast, consider an alternative measure satisfying

$$v(A \cup B) = v(A) + v(B) + \lambda_{AB} v(A)v(B), \quad (51)$$

with

$$\lambda_{AB} = \begin{cases} 0 & \text{if } d_H(A, B) \leq 1 \\ \lambda < 0 & \text{otherwise.} \end{cases}$$

The value of $v(A \cup B)$ will increase more if the subsets $A, B$ are geometrically close than when they are separated. Hence partial subadditivity is induced only for separated subsets. Since a subset $C$ can be obtained by unions of different subsets $A$ and $B$, for a consistent definition the following will be used:

$$v_\lambda(C) = \min_{A \cup B = C} v(A) + v(B) + \lambda_{AB} v(A)v(B). \quad (52)$$

Note that equation (52) implicitly accounts for translation, rotation and reflection symmetries because $d_H$ depends only on the intraset distances and not on geometrical orientation.
5.4. A FUZZY MEASURE OF PIXEL CLUSTER COMPACTNESS

The formula (52) has two parameters: the value of $\lambda \in (-1, 0)$ and a common value of $v(\{i\}) = \epsilon > 0$ at the singletons. For a fixed pair, $(\lambda, \epsilon)$, the values of $v(A)$ can be computed for all subsets $A$. Suitable values for these parameters can be obtained by fitting them at reference points and subject to the constraints specified.

This is achieved by first fixing some small $\epsilon$ and then fitting $\lambda$ by minimising the objective

$$F(\lambda) = \sum_{i=1}^{r} |v_\lambda(A_i) - v_i| + a \sum_{i=1}^{c} \max(0, -(v_\lambda(A_i) - v_\lambda(B_i) - \delta)),$$

where $r$ is the number of reference points $(v(A_i), v_i)$ with the desired values $v_i$, $c$ is the number of constrains of the type $v(A_i) \geq v(B_i) + \delta$ and $a$ is a tradeoff parameter controlling the relative importance of fitting the reference points or constraints. The objective $F$ is continuous as $v$ is a continuous function of $\lambda$, however it may have several local minima. The global optimum can be obtained numerically using the Pijavski global optimisation method [68] (implemented in GANSO software [11]).

In the following examples note that the indexes relate to a row-major ordering of the image blocks, to maintain consistency with the publications mentioned in Section 5.1. Given the rotational and reflective symmetry of the measures, it is irrelevant whether a row-major or column-major ordering is used.

**EXAMPLE 9.** The following illustrates the results of applying this approach to a $3 \times 3$ block. A reference point for $v(\{1,2,3,4,5,6\}) = 1$ is specified and the constraint $v(\{1,2\}) \geq v(\{1,3\}) + 0.05$ is imposed. The common value for $v(\{i\}), i = 1, \ldots, 9$ is fixed at $\epsilon = 0.2$.

The parameter $\lambda$ is estimated according to (53), where the tradeoff parameter $a$ is set to a default value of 1. The resulting optimal value of $\lambda = -0.4792$ was obtained in approximately three minutes on a typical desktop computer. Figure 5.6 shows the estimated values of the fuzzy measure for several representative clusters. It can be seen that, as required, clusters with higher cardinality receive larger measure values than clusters with lower cardinality and coherent clusters.
of a given cardinality have a higher measure than less coherent clusters of the same cardinality.

Remark 6. Since the reference value $v(\{1, 2, 3, 4, 5, 6\}) = 1$ was specified, it is possible that in the fuzzy measures computed by (52) with the provided $\varepsilon$ and estimated $\lambda$, some values are greater than 1. In such cases these values are reduced to unity by applying $v = \min(v, 1)$. This was the case in the previous example, with sets with cardinality $k \geq 6$ receiving a capped measure values of 1. This approach to capping $v$ was applied where necessary in subsequent examples to ensure that $v \in [\varepsilon, 1]$.

In addition to fitting $\lambda$ it is also possible to jointly $\varepsilon$. This can be achieved in one of two ways: 1) to set up a bi-level optimisation problem, where at the outer level a grid search is performed for $\varepsilon$, whereas at the inner level the global minimiser is obtained for $\lambda$ given the current value of $\varepsilon$; or, 2) solve a bi-variate global optimization problem using, for example, an Extended Cutting Angle method \cite{6} \cite{7} \cite{8} (also implemented in GANSO \cite{11}). In evaluating both options
it was found that the results are not particularly sensitive to the value of \( \varepsilon \) and thus the singleton measure value can be fixed at some reasonable value being the lower bound on the measure function.

An alternative to using a partially sub-additive measure (as in (51)) is to use a partially super-additive measure, given by (51) with

\[
\lambda_{AB}^+ = \begin{cases} 
0 & \text{if } d_H(A, B) \geq 1 \\
\lambda & \text{otherwise.}
\end{cases}
\]

The fuzzy measure is constructed by using

\[
(54) \quad v_\lambda(C) = \max_{{A \cup B = C}} v(A) + v(B) + \lambda_{AB}^+ v(A) v(B).
\]

In this case, rather than penalising clusters that have separated pixels, clusters which have more compact components are favoured. Fitting the parameter \( \lambda \) is done by solving problem (53) by restricting \( \lambda \) to \([0, \infty)\).

**Example 10.** A partially super-additive measure is computed using a reference point \( v(\{1, 2, 3, 4, 5, 6\}) = 1 \) and a constraint \( v(\{1, 2\}) \geq v(\{1, 3\}) + 0.05 \), as per the partially sub-additive example. The common value of the singletons is \( v(\{i\}) = \epsilon = 0.05 \). The optimal value for \( \lambda = 9.6766 \) (which took approximately three minutes of CPU time on a standard desktop PC). Figure 5.7 shows the estimated values of the fuzzy measure for several representative clusters.

It is worthwhile considering the computational complexity of the proposed method of solution. The CPU time for solving (53) is proportional to \( 2^n l(r + c) \), where \( l \) is the number of iterations of the Pijavski method, by which the quality of the minimizer \( \lambda \) is controlled. The typical CPU time (using Matlab and GANSO library) was approximately 3 minutes for \( p = 9 \) (3 \( \times \) 3 blocks). This is not excessive, as only one such fitting process is required to determine the measure, which is then fixed for use in the image reduction application. However, for larger \( p \) the CPU time grows considerably: for \( p = 25 \) it took approximately 24 hours to fit the fuzzy measure. Subsequently it is appropriate to consider alternative fuzzy measure construction techniques that are less CPU intensive.
5.4.2. A fuzzy measure based on the Minimum Spanning Tree. Minimum Spanning Trees (MST) have been used for clustering for several decades [102]. Given a connected weighted graph the MST is a subgraph (specifically a tree connecting all vertices) whose weight is the smallest. It is constructed from the adjacency matrix using Prim’s or Kruskal’s algorithms [25] and its complexity is quadratic in the number of vertices.

The weight of an MST may be taken as the sum of all edge weights in that tree, constructed from a complete graph connecting the elements of a cluster, with edge weights being pairwise distances between the pixels. In cluster analysis such MSTs are used to agglomerate data and partition it into several clusters by removing the edges of maximum weight. Here the requirement is a measure of compactness of a single cluster and hence only the weight of the MST will be used, rather than its structure.

The MST-based measure must satisfy the requirements given in Section 5.3. To do so, the following function is proposed

\[
SC(A) = W_0 - \frac{W(MST(A))}{\mu_D |A|^p} + 1,
\]
where \( p \) is the cardinality of the largest cluster, \( D \) is the largest possible distance between the elements of a cluster and \( W_0 = 1 + \frac{p(D + 1) - 1}{p^2D} \). For example, for a \( 3 \times 3 \) block and the Euclidean distance function, \( p = 9 \) and \( D = 2\sqrt{2} \) and hence \( W_0 = 1 + \frac{1}{9} + \frac{4}{81\sqrt{2}} \). For brevity denote \( W(\mathcal{A}) = W(\text{MST}(\mathcal{A})) \), which is the weight of the MST of set \( \mathcal{A} \).

The constant \( W_0 \) ensures that \( SC(\mathcal{A}) \in (0, 1] \) and it is assumed that \( SC(\emptyset) = 0 \). The values of \( T \) and \( M \) ensure appropriate scaling parameters so that the following holds.

**Proposition 15.** *The function SC in (55) is a fuzzy measure discriminating the compactness of clusters. It is invariant with respect to translation, reflection and rotation of these clusters.*

**Proof.** Given the MST, pairwise distances are invariant with respect to the mentioned transformations and hence \( SC \) is invariant as well. To show monotonicity and the range, note that in this context the minimum distance between the elements of \( \mathcal{A} \) is assumed to be one, and the largest is \( D \). It follows that \( \frac{W(\mathcal{A}) + p}{D} \leq \frac{W(\mathcal{A}) + p + 1}{|\mathcal{A}| + 1} \leq \frac{W(\mathcal{A}) + p}{|\mathcal{A}|} \).

The last inequality follows from the fact that \( \frac{a + 1}{b + 1} \leq \frac{a}{b} \) for positive \( a, b \) such that \( a \geq b \). The numerator above is clearly larger than the denominator as \( p \geq |\mathcal{A}| \). The above inequality implies that \( SC(\mathcal{A} \cup \{a\}) \geq SC(\mathcal{A}) \) for every \( \mathcal{A}, a \subseteq \mathcal{U} \), the largest possible cluster. Thus monotonicity with respect to cardinality is established.

When \( \mathcal{A} \) is itself a singleton, \( \{a\} \), it holds that \( SC(\{a\}) = W_0 - \frac{p + 1}{pD} = \frac{p(D + 1) - 1}{p^2D} > 0 \). When \( \mathcal{A} \) is the universal set then \( W(\mathcal{A}) = |\mathcal{A}| - 1 \) and thus

\[
SC(\mathcal{A}) = W_0 - \frac{p - 1}{pD} + 1 = 1,
\]

thus establishing the range. Hence \( SC \) is a fuzzy measure. Finally, consider two sets \( \mathcal{A}, \mathcal{B} \) of the same cardinality. The less compact the set the larger the weight.
168 5. WEIGHTING FUNCTIONS FOR AGGREGATION OF PIXEL CLUSTERS

\( W(MST(A)) \). Since \( SC \) is a decreasing function of \( W \), more compact sets will result in larger values of \( SC \). The proof is complete. □

EXAMPLE 11. The numerical results illustrating the formula (55) are presented in Figure 5.8. Unlike the Sugeno approach, the MST approach computes the fuzzy measure directly, rather than fitting it to the provided reference points or constraints. The measure values given in Figure 5.8 show monotonicity with respect to set cardinality and differentiation between compact and sparse clusters of equal cardinality, as required. One inconvenience of (55) is that the numerical values of \( SC(A) \) are clogged at the higher end of the range, so that the differences between more and less compact sets of cardinality more than three are in the second or third decimal place value. This may be rectified by raising \( SC(A) \) to some power \( q > 1 \), such that

\[
SC_q(A) = \left[ W_0 - \frac{W(MST(A))}{|A|^{pD} + 1} \right]^q.
\]

This distributes values throughout the range and more clearly differentiates numerically between the more compact and less compact sets, as in Figure 5.8(b). The CPU time using the both (55) and (56) was negligible.

5.4.3. Decomposing fuzzy measure. The final construction method for a fuzzy measure of cluster compactness is based on geometrical decomposition of the sets involved. The required geometrical symmetries of the clusters will be obtained by decomposing them into elementary components, obtaining invariant measures for these components and constructing the cluster measure as a function of the component measures. Consider Figure 5.9, which shows examples of four elementary building blocks of pixel clusters, herein called the basic components of a cluster. Each cluster set \( A \) can be represented in one or more ways through the union of (possibly overlapping) basic components, \( C_i \). These basic components account for all geometrical symmetries and are encoded using a hash function
5.4. A FUZZY MEASURE OF PIXEL CLUSTER COMPACTNESS

Figure 5.8. Estimated fuzzy measure using Minimum Spanning Tree method.

based on the average intrapixel distance

\[ h(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} d_H(a, \mathcal{A} \setminus a). \]
The number of basic components and their shape can be specified by the user according to their preference and applications.

To construct the fuzzy measure value for a given subset $A$, the possible decompositions into one or more basic components of the same type are determined, along with the number of ways to fit each such basic component into $A$. Each of the $C_i$ is assigned a value $u_i$. The value of $v(A)$ is computed as

$$SC(A) = \min(1, n_1u_1 + n_2u_2 + \ldots + n_lu_l),$$

where $l$ is the number of basic components, $u_i$ is the value of the $i$th basic component, and $n_i$ is the number of $i$th basic components in the decomposition of $A$. The following example illustrates the representation of a fuzzy measure value through a combination of the basic components.

**Example 12.** Take the subset $A$ as the cluster with 3 connected elements shown in Figure 5.10. The basic components of this set are also shown. The given
cluster shape results in three possible ways to fit the $C_1$ component ($n_1 = 3$), two ways to fit the $C_2$ component ($n_2 = 2$) and one way to fit the $C_3$ component ($n_3 = 1$). Note that overlapping is allowed. Computing the composition of a given cluster is a preprocessing step of the fuzzy measure computation and can be performed automatically through the use of the hash function $h$: the fact that $C_i \subseteq A$ is verified by computing $h$ for each subset of $A$ of the same cardinality as $C_i$. This is numerically efficient and it also guarantees that the decompositions are invariant with respect to geometrical transformations, as required.

For this example, take the weights of the basic components as $u = [0.1, 0.1, 0.05]$. Thus, the fuzzy measure value for the cluster in Figure 5.10 is given by:

$$v(A) = \min(1, 3u_1 + 2u_2 + 1u_3)$$

$$= \min(1, 3 \times 0.1 + 2 \times 0.1 + 1 \times 0.05)$$

$$= 0.55.$$  

The parameters of this model are the component measure, $u_1, \ldots, u_l$. These may be obtained by fitting to the available reference points and constraints, which represent the desired numerical values and relations between the values of $SC$ (as in the previous construction methods). Before proceeding with formulating a fitting problem, consider the following proof that the function $SC$ is a fuzzy measure. Let $U$ denote the universal set (the largest cluster in an $m \times n = p$ block of pixels).

**Proposition 16.** The function $SC$ in Eq. (57) is a fuzzy measure irrespective of the values $u_i \in [0, 1]| \sum_i n_i(U)u_i \geq 1$. $SC$ discriminates between more and less compact clusters.

**Proof.** Evidently $SC(\emptyset) = 0$ and $SC(U) = 1$. It remains to prove monotonicity with respect to cardinality and discrimination over compactness. For
monotonicity note that $A \subset B$ implies $n_i(A) \leq n_i(B)$ and consequently

$$\sum_i n_i(A)u_i \leq \sum_i n_i(B)u_i.$$  

For subsets of the same cardinality, more compact subsets $A$ allow fitting more basic components that are larger in size and they will have larger values of $n_i$ than less compact subsets. Hence more compact subsets produce larger values of $SC(A)$ and less.

\[\square\]

**Figure 5.11.** Fuzzy measure for cluster compactness computed from decomposition method.

It remains to fit the parameters $u_i$ to the available reference points and constraints. Because the parameters enter the measure value linearly then the following mathematical programming problem is appropriate:

\[
\begin{align*}
\text{minimize} & \quad F(u_1, \ldots, u_l) = \sum_{i=1}^{r} |SC(A_i) - v_i| \\
\text{subject to} & \quad SC(A_j) - SC(B_j) \geq \delta_j, \; j = 1, \ldots, c, \\
& \quad SC(A_k) = \min(1, \sum_{i=1}^{l} n_i(A_k) u_i), \\
& \quad \sum_{i=1}^{l} n_i(U)u_i \geq 1, \\
& \quad u_1, \ldots, u_l \geq 0.
\end{align*}
\]
which is subsequently converted, by introducing slack variables \( r_i^+, r_i^- \) (and denoting \( SC_i = SC(A_i) \)) into a linear programming problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{r} r_i^+ + r_i^- \\
\text{subject to} & \quad r_i^+ - r_i^- - SC_i = -v_i, \ i = 1, \ldots, r, \\
& \quad -SC_i + SC_k \leq -\delta_{ik}, \\
& \quad SC_k \leq \sum_{i=1}^{l} n_i(A_k)u_i, \\
& \quad SC_k \leq 1, \\
& \quad \sum_{i=1}^{l} n_i(U)u_i \geq 1, \\
& \quad r_i^+, r_i^- \geq 0, SC_i \geq 0, \\
& \quad u_1, \ldots, u_l \geq 0.
\end{align*}
\]

(59)

The number of variables is \( 2r + c + l + t \) and the number of constraints is \( r+c+2t+1 \), where \( t \) is the total number of subsets engaged in all of the constraints and reference points. Note that this number is smaller than the number of all possible subsets \( 2^p \), because only the values \( SC_i \) are required for those subsets. The values \( \delta_{ik} = \delta_j \) correspond to the pairs of sets \((A_j, B_j) = (A_i, A_k)\) in the \( j \)th inequality constraint. The problem is solved by using the standard simplex method.

**Example 13.** The output of this approach is again illustrated using an example \( 3 \times 3 \) block. The given reference point is \( v(\{1,2,3,4,5,6\}) = 1 \) and the constraint \( v(\{1,2\}) \geq v(\{1,3\}) + 0.05 \) is imposed. Four basic components, as depicted in Figure 5.9, are used to construct the fuzzy measure. Their weights are estimated by solving Problem (59).

The values obtained for the components are \( u = [0.0042, 0.0799, 0.0228, 0.1169] \). Figure 5.11 shows the estimated values of the fuzzy measure for several representative clusters.
5.5. Evaluation

5.5.1. Reduction of Thin Curves. The objective in designing and implementing penalty weights based on cluster compactness was to specifically address the preservation of fine detail and structure within images undergoing scale reduction. Thus, to test the effectiveness of these weights a synthetic test problem was created, for which visual assessment of reduction operators could be easily conducted. The image in Figure 5.12 depicts a series of concentric circles as thin curves having a width of one pixel. This pattern contains a large variety of cluster patterns within small $3 \times 3$ blocks and in performing reduction on this image, any operator must cope with the problem that the important image detail represents a minority of pixels at all scales greater than $2 \times 2$ blocks. Furthermore, versions of this test data for different lines widths are easily generated and may be assessed against the ground truth for the preservation of continuous lines in the reduced images.

A test image $I$ is constructed according to

$$I = \max(C \cdot F, B),$$

where $C$ is a binary image depicting the set of $m$ circles of radii $\{r_1, r_2, ..., r_m\}$, $F$ is a foreground noise field and $B$ is a background noise field. These fields were generated as uniformly distributed 8 bit pixel intensities in the ranges $[245, 255]$ and $[0, 50]$ respectively.

A local block-based reduction operator based on eqn. (39) was constructed by replacing the distance-based weights (given by eqn. (41)) with a cluster-based weight, such that

$$P(x, y) = w(A) \sum_{i=1}^{p} \rho(x_i, y),$$

where

$$w(A) = 2 - v(A)^q \geq 1$$
and \( \nu \) is a fuzzy measure computed using the aforementioned approaches. \( A \) denotes the subset of pixels in the current block having intensities that satisfy \( r_k < \tau \) as per eqn. (42). These pixels define, for a given value of \( y \), the candidate cluster within the tonal space for that block. This choice of penalty function means that for two candidate clusters of equal cardinality and equivalent pixel intensity differences within the cluster, the more spatially coherent cluster will have a lower penalty and thus be preferred as the significant cluster of the block. Conversely, if the spatial patterns are equivalent, the cluster with the more compact tonal range will be preferred. As with the method described in Chapter 4 the candidate representative values \( y \) are taken from the set of input pixel values for that block, so that the output image is a proper subset of the input image.

Several versions of this penalty-based averaging function were constructed using the proposed fuzzy measures given by equations (55) and (57). For the minimum spanning tree based measure, exponent values \( q = 1, 2, 4 \) were chosen and for the fuzzy decomposition method the single value \( q = 1 \) was used, since this function already provided a well-distributed set of measure values in \([0, 1]\). The measure values are depicted in Figure 5.15. Three test images having circle

\[ \text{Figure 5.12. Example of single pixel circle pattern used to test image reduction operators.} \]
widths of 1, 1.5 and 2 pixels were used to assess the capacity of the averaging functions to cope with clusters that ranged from the minority to the majority of a 3 × 3 block. The mode-like average with cluster penalty weights was compared against the mode-like average with penalty weights (presented in Chapter 4), as well as the other monotonic and non-monotonic averaging functions considered in that study. These were the arithmetic mean, median, Shorth and Least Median of Squares (LMS).

5.5.2. Performance on Thin Curve Reduction Problem. Image reduction was performed over disjoint blocks of size 3 × 3 (producing a 1/4 scale image) and the resulting images are shown in Figures 5.13 and 5.14. It is apparent from these results that the median, shorth and LMS functions do not preserve the relevant curves of one pixel thickness. This is to be expected as each of these functions has a breakdown point of 50% and they will reject up to half of the inputs if they are outliers, without degradation of the average. In this case the average returned for most blocks is from the background field. In the case of the arithmetic mean, while the circles are preserved, their peak intensity is diminished and they are spatially broadened, which is also as expected. While the resulting image contains the desired visual continuity of the curves, the structural detail of these curves (specifically radial intensity gradients) have been corrupted by the reduction.

The cluster-based mode-like averaging function performs consistently better than the other averaging functions on this test image, including the distance-based mode-like average, which is most similar to it. In particular, at line widths of one pixel thickness, each of the proposed fuzzy measure construction techniques shows similar performance. For the powered minimum spanning tree approach, the exponent value shows little effect on the results, indicating that the spread of measure values across cluster cardinalities plays only a minor role in the penalty minimisation problem. The obtained measure values for the MST and DFM construction methods are given in Figure 5.15.
The final test verifies the robustness of each of the averaging functions to impulse ("salt & pepper") noise while reducing curves having a thicknesses of one pixel. Figure 5.16 shows the sample image which was formed by adding random values of 0 or 1 (sampled uniformly and applied to 10% of pixels) to the

**Figure 5.13.** Reduction of circles using cluster-based mode-like averaging function.
5. WEIGHTING FUNCTIONS FOR AGGREGATION OF PIXEL CLUSTERS

**Figure 5.14.** Reduction of circles using comparison averaging functions.

A logical image used to depict the circles. This had the effect of randomly setting 5% of foreground pixels to background values and 5% of background pixels to foreground values. At this rate it is expected that each local block being reduced had, on average, one corrupt pixel switched from the foreground to background, or vice versa. All other parameters were equivalent to the previous tests.
5.5. EVALUATION

Figure 5.15. Fuzzy measure values for cluster compactness obtained using: (a) MST, \( q=1 \); (b) MST, \( q=2 \); (c) MST, \( q=4 \); and (d) DFM, \( q=1 \).

Figure 5.16. Circle test image corrupted with 10% impulse noise.

Each of the averaging functions described above were tested using blocks of size \( 3 \times 3 \) and results are shown in Figure 5.17, where the lower left quadrant has been zoomed to highlight the capacity of each operator to cope with this type of noise.

Again it is apparent that the cluster-based mode-like averaging function is able to identify and ignore many of the corrupt foreground pixels (i.e., those that should be background but were changed to foreground). However, comparing the
results to the first column of Figure 5.13 shows that corrupted background pixels (i.e., foreground pixels switched to the background) do cause some additional loss of continuity of the circular curves. This is to be expected because even one foreground pixel switched to background in a $3 \times 3$ block would reduce the cardinality of the foreground pixel cluster, which results in a higher weight and thus a lower
likelihood of being the cluster that minimises eqn. (60). In the presence of low levels of noise variance, there is clearly a lower bound on the cardinality of the significant cluster that will be selected in favour of the background data, which while likely having larger intensity variance, will represent a larger cluster and more likely a connected cluster.

5.5.3. Reduction of Satellite Images. The results from reducing the test image highlights that the proposed block reduction operator does preserve fine structured detail in images, even when those pixels are in the minority within the local block. However, whether or not this is effective in the reduction of real images is a question of interest. Thus, to assess the preservation of fine details, the various reduction operators considered in this thesis have been applied to satellite images that contain a large variety of structures at different scales.

A US Geological Survey image showing a portion of the Port of Los Angeles is depicted in Figure 5.18. The original image is an RGB image with size $1000 \times 800 \times 3$. It should be noted that this Figure shows a reduced version of the original source satellite image (which was not available in raw form) and a close inspection reveals that it has already been reduced using a low pass filtering operation (which has produced aliasing in regions near sharp intensity changes). However, irrespective of this prior reduction, the image is sufficient for the purposes of evaluating image reduction operators due to the presence of a variety of fine scale image structures. These structures show interesting and significant intensity variations that should ideally be preserved in any reduction operation in which visual quality is to be preserved. Some specific regions of interest to consider before and after reduction are:

1. The housing in the upper left corner, denoted as (A);
2. The boats in the marina on the left side (B);
3. The vessels within the central channel and the harbour area (C);
4. The shipping containers on the docks on the right side (D); and,
5. The port infrastructure, such as the cranes on the right side and the fuel tanks on the left side of the central channel (E).
Figure 5.18. Port of Los Angeles, March 20, 2004 (Image courtesy U.S. Geological Survey - used with permission)
A modification of the local block-based reduction algorithm was applied. In this scheme, depicted in Figure 5.19, disjoint $3 \times 3$ regions are to be replaced by a single pixel representing the significant cluster in the local data. However, the surrounding $5 \times 5$ neighbourhood is used as input into the averaging function. As these larger neighbourhoods overlap they provide both enhanced robustness to noise and better preservation of structures across pixels representing the disjoint inner $3 \times 3$ blocks. The resulting images represent a $\frac{1}{9}$ scale version of the original image.

![Figure 5.19. Overlapped local block reduction operation.](image)

Overlapped local reduction operators were constructed using several functions evaluated in Chapter 4 (specifically the arithmetic mean, median, Shorth and P-MODE) and each operator was applied to the image in Figure 5.18. Two versions of the P-MODE average were evaluated; one using the distance based weights considered previously and one using cluster-based weights derived from an MST measure of cluster compactness in $5 \times 5$ blocks. The Shorth was selected as a representative robust estimator of location as it gave the best results on this image.

The results of reduction operations using the arithmetic mean and median are given in Figure 5.20. It is clear from visual inspection of these reduced images that both functions act as a low pass filter and produce images with significant loss of high frequency information, appearing blurred and fuzzy. In particular, considering the regions described above, a considerable loss of detail has occurred.
This is clearly evident in regions B and D, where individual boats in the marina are no longer distinguishable and the shipping containers appear more like long buildings than individual boxes. Additionally, the port infrastructure in region E is no longer evident. While the vessels in region C are still evident, the loss of high frequency information has softened the intensity of the wakes relative to the background water and made them less obvious in the final image. This loss of information is expected from reduction using the arithmetic mean, however it is perhaps surprising at the poor performance of the median-based reduction operator.

The Shorth was selected as a representative function of the robust estimators of location. These functions, which performed well in experiments described in Chapter 4, are robust to noise and outliers and should fair significantly better than the monotonic operators. Figure 5.21 shows the satellite image reduced using the Shorth-based reduction operator.

To assess the capacity of the weakly monotonic CMODE average to preserve the small scale, structured detail within the image, it is compared against the monotonic averages (arithmetic mean and median) by computing the difference between the respective images (given in Figure 5.23), whereby the respective image is subtracted from the CMODE-reduced image. Figure 5.23(a) shows the difference between the image reduced using the arithmetic mean and that reduced using the mode-like average with cluster-based weights. Figure 5.23(b) similarly shows the difference between the image reduced with the median operator and that reduced using the mode-like average with cluster-based weights. It is apparent in both images that there is a significant amount of structured information present in these image, indicating that the CMODE operator preserved this information where neither of the monotonic averages did. It is also apparent that there is very little noise signal in these images, indicating that all three operators effectively removed high frequency random variations in the images. The CMODE operator (the weakly monotonic mode-like operator with cluster-based weights) appear to have met the requirements laid out in Chapter 1, as being a
5.5. EVALUATION

Figure 5.20. Port of Los Angeles satellite image reduced using monotonic reduction operators.
robust averaging function suitable for preserving structured information at pixel scales, while reducing variations at these scales due to random noise.

5.6. Conclusions

In this chapter the problem of quantifying the compactness of a single geometrical cluster of spatially organised pixels was considered, with the proposed solution being a fuzzy measure with specific reasonable properties. This problem arose in the context of preserving fine image detail in an image reduction task where a weakly monotonic averaging function was used to compute the representative value of each block. In this context it is required to discriminate between compact groups of pixels (of similar intensity) representing an object, from scattered groups likely to be noise, even if the cluster is a small subset of the local block being reduced.
Figure 5.22. Port of Los Angeles satellite image reduced using mode-like functions.
Figure 5.23. Difference images for reduced Port of Los Angeles images.
Three alternative approaches to defining a suitable fuzzy measure of cluster compactness were proposed. Each of the presented approaches has shown potential in characterising the compactness of geometrical clusters and each has its own merits. The approach based on the Minimum Spanning Tree computed from the cluster graph is the simplest method, with the fuzzy measure values computed directly from the weights of the edges without user input. This saves computational resources and can thus be applied to compute measures for large image blocks. However, this approach does not allow a user to provide reference points or constraints. Thus, it is not suggested for applications where users might want to flag clusters with specific shapes or impose other domain dependent constraints.

The approach based on Sugeno-type fuzzy measures is more flexible than the MST approach as it allows for the specification of reference points and constraints. This is due to the optimization process for estimating the parameter $\lambda$, together with the value $\varepsilon$, or using a prescribed value of $\varepsilon$. A shortcoming of this approach is its computational cost for generating all fuzzy measure values at every iteration of the optimization process. CPU time will become prohibitive for computing the measures over large image blocks.

A special feature of these two approaches is that the estimated fuzzy measure values are not only invariant with respect to translation, reflection and rotation, but also invariant with respect to the shape of clusters with connected elements. On the other hand, if a user wishes to impose preferences with respect to cluster shapes, the Decomposing Fuzzy Measure is a suitable alternative due to its computational flexibility derived from identifying suitable basic components of clusters. In all three approaches it was demonstrated that the fuzzy measure was monotonic in set cardinality and discriminated between more and less compact subsets. By construction, the resulting values were invariant with respect to basic geometrical manipulations of the clusters, such as translation, rotation and reflection.

These measures were used to weight the penalty terms of tonally compact clusters in a penalty-based mode-like weakly monotonic averaging function, which was applied as an image reduction operator. These cluster-based weights were
found to be effective at identifying local pixel clusters representing fine details at the pixel scales, ensuring preservation of these details under reduction, while also reducing random variations at pixel scales. It remains to be seen how broadly measures of cluster compactness can be applied within other image processing tasks, such as filtering or smoothing using spatial-tonal filters and this is a topic for future research.
CHAPTER 6

Conclusions and Future Work
6.1. Summary of Key Results

The problems of image reduction and image denoising were described as averaging problems, and the former problem was considered with regard to recent literature that has formulated image reduction as an averaging procedure over local image blocks. Recent results by other authors regarding monotonic averages applied in local block-based image reduction operators did not sufficiently address the issues surrounding the preservation of fine scale detail in images, or the capacity to deal with non-Gaussian and non-additive noise. Given the prevalence of robust non-monotonic averaging functions within the literature, this thesis proposed an investigation of non-monotonic averaging functions within the context of image processing problems, with the view to establishing an understanding of properties of these functions that made them suitable as image reduction operators. Several specific questions were highlighted in Chapter 1 and herein these will be reconsidered in light of the results from the previous chapters and the conclusions drawn therein.

The first question asked,

**What general property or properties make non-monotonic averaging functions robust and effective for averaging problems in image processing?**

In Chapter 3 a definition of weak monotonicity was given and used as the basis for defining the class of weakly monotonic averaging functions. This relaxation of the monotonicity constraint, fundamental to the definition of aggregation functions, was motivated by two observations. First, that it is reasonable to expect that a representative value of a set of data does not decrease if all inputs are increased by the same amount (or shifted uniformly); and, second, that in many domains, a sufficient increase in one input value (or a subset of input values) should lead to a decrease in the representative value, as these values may be indicative of outliers corrupting the input data. This lead to Definition 70, which is a fundamental and significant contribution of this thesis. Some properties of
weakly monotonic averaging functions were established, particularly with respect to composition and transformations of these functions.

Using this new concept, several important classes of non-monotonic means reported in the literature were investigated and sufficient conditions for weak monotonicity of the Gini means, Lehmer means and generalised mixture operators were established. Additionally, several measures of central tendency (namely the Least Median of Squares, the Least Trimmed Squared, the Shorth and the mode-like average) were shown to be weakly monotonic averaging functions. The density-based means were also found to be weakly monotonic and several generalisations to this class of functions were produced using weak monotonicity as the over-arching constraint. Further to these results, the important edge-preserving spatial tonal filters were established as weakly monotonic mixture operators and conditions for weak monotonicity of generalisations of these functions were established, opening up a larger class of possible denoising and smoothing functions for future investigation. The results in this chapter clearly established a firm theoretical basis for understanding a common property of many practical, robust, non-monotonic averages; that of weak monotonicity.

An important difference between images and other data types is the geometric organisation of the intensity values. In other words, the relative location of the data is of central importance and cannot be ignored. While block-based reduction algorithms do take location into account by the fact that they operator on a limited, compact subset of the image, when computing the average using a monotonic function, no account is given for the spatial location of a pixel within the block. In Chapter 4 the mode-like average and several robust estimators of location were proposed as candidate weakly monotonic averaging functions that could be applied as image reduction operators. Beliakov’s mode-like average with distance-based penalty weights was assessed against the Shorth, Least Median of Squares, Least Trimmed Squares, the arithmetic mean and median. The ability of each function to preserve structured intensity information under varying levels of non-Gaussian noise corruption was assessed by measuring performance on a facial recognition task, where reduction was used as a pre-processing step.
Comparisons to baseline performance on non-reduced, clean images were made, as well as comparison against non-reduced noisy images. All of the weakly monotonic averages were shown to perform consistently well on both speckle noise and impulse noise. The median operator performed well on impulse noise, while the arithmetic mean performed poorly under all circumstances. The mode-like average performed better than all other operators. From these experiments it was clear that weakly monotonic averages that account for location of the data performed better in the image reduction task than did the monotonic averages. Furthermore, the mode-like average, which determined a representative value of the principle tonal cluster (weighted by interpixel distances), was better able to preserve information relevant to the recognition task. From these results it is concluded that while weak monotonicity is important for robustness to outliers, that alone is not necessarily the only desirable property of an averaging function used for image reduction. Averages that account for the relative location of the data are also important. These results support an affirmative answer to the second question, which asked:

**Are mode-like averages suitable for representing the principle cluster within a local image block and do such averages preserve information content during image reduction operations?**

Evidently mode-like averages do represent the principle cluster within a local image block and subsequently account for location of pixel intensity values during image reduction. Further support for this conclusion is provided by the results in Chapter 5, where the mode-like averages constructed using distance-based and cluster-based penalty weights were evaluated against the robust estimators of location and the monotonic averages, on synthetic test images that contained pixel scale structured curves. Only the mode-like averages were able to preserve both continuity and spatial intensity gradients. The robust estimators and the median performed particularly poorly when applied to the reduction of images containing circles having one pixel thickness. While the arithmetic mean preserved line continuity, as expected, it significantly distorted radial intensity gradients, as it
acts as a low pass filter. With regards to these results on synthetic images, the question

*Can fine image details, that would otherwise be filtered out by a monotonic averaging procedure, be preserved using non-monotonic averages?*

is answered with respect to weakly monotonic averages and the mode-like average. Yes, non-monotonic averages do preserve image details, however as the relative size of the image feature within a local block drops in comparison to background or noise pixels, the difficulty of this preservation increases. The mode-like average with cluster-based weights, constructed using several different fuzzy measures of cluster compactness, performed better than all other averaging functions on synthetic images. This confirmed that the novel cluster compactness measures derived from fuzzy measures were an appropriate and useful method of accounting for the geometric organisation of pixels representing image features within a local block. This provides one answer to the final question posed in Chapter 1,

*How do we account for the geometric arrangement of pixels in the design of averaging functions suitable for image processing tasks?*

The most striking results supporting the claim that weakly monotonic averages are suitable for image reduction tasks and that the mode-like average is effective in preserving fine detail, comes from the reductions of satellite images considered in Chapter 5. It was evident that the monotonic averages acted as low pass filters and degraded the visual quality of the image under reduction, as well as destroying the fine detailed features within the image. This confirms that in practical scenarios, monotonic averages are not as useful as weakly monotonic averages in image reduction problems. Additionally, while the Shorth operator performed better than the monotonic averages, it too causes degradation of fine features and removed some high frequency signal. While some of this omitted signal would certainly be noise, it is clear that structured information was also lost.
Both versions of the mode-like average - one built using distance based weights and one using the fuzzy measure derived from the Minimum Spanning Tree graph of the local cluster elements - performed exceptionally well in this reduction task. Visually the reduced images maintained significant detail, even at pixel scales, while not apparently preserving noise. This was evident from the difference images obtained by subtracting away the image formed by the arithmetic mean reduction operator and separately, that formed by the median reduction operator. In both instances the difference image showed significant amounts of structured information, indicating that the CMODE-reduced image had preserved structured high frequency information whereas the mean and median had not. Clearly the CMODE averaging function passes higher frequencies than the monotonic averages, indicating it is not necessarily a low-pass filter (at least not with the same cutoff frequency). Additionally, these difference images did not suggest that the CMODE-reduced image contained noise that the other images did not, since there was not significant levels of unstructured pixels in the difference image. This indicates that the CMODE operator was able to discriminate between high frequency signal corresponding to structured information and high frequency signal corresponding to random noise, preserving the former and omitting the latter. This is exactly the objective laid out in Chapter 1 of this thesis:

*Preserve structured information at pixel scales while reducing random noise in image processing tasks*

6.2. Future Research

This thesis has established that weakly monotonic averaging functions are suitable in image reduction problems formulated as averaging problems over local image blocks. They are capable of rejecting outliers and noise, while offering essential properties of idempotence and monotonicity with respect to a constant increase in all input values. Methods for accounting for the location and structured organisation of pixel intensity data within a local block were proposed and
evaluated, showing the benefits of using penalty weights that account for this structure in the selection of a representative pixel within a local block.

This approach to averaging can naturally be applied to image denoising, with the first obvious procedure being to compute the average of a block and use that to replace a single pixel within that block (typically the central pixel). Alternatively, each pixel within a block could be replaced with a weighted sum of the average value of every block that the pixel is a member of. Which averaging function works best in such procedures must be established through consideration of the constraints and requirements that the application imposes, as well as verified through experimentation on synthetic and real images.

Further investigation of image denoising and smoothing can be investigated by considering generalisations of the spatial tonal filters, as proposed in Section 3.3.6. For example, the first obvious example would be to take, for the function $f(x)$ in equation (31) (or (32)), the median of the input pixels. Such a choice would likely produce a filter similar to the spatial and rank filters discussed in Section 2.6.3 and evaluation against these filters and the bilateral filter (and similar variants) would be appropriate.

Alternatively, taking as $f(x)$ the value of the mode-like average, or any other weakly monotonic average that accounted for the spatial structure within the block used for filtering, would permit the filtering to be performed with respect to the principle cluster within the block. Furthermore, applying the results obtained in Section 3.3.5, alternative kernel functions - which act as weight functions in the mixture function formulation of this class of filters - could be investigated and compared to the traditional Gaussian kernel.

The benefit of such choices over the current form of the spatial tonal filter are that we can effectively account for the possibility that the pixel being filtered is itself an outlier. In the traditional implementation of convolution-based filters (not just in images, but also in general signals), the tonal difference between the central pixel of the block and every other pixel is used, weighted by the nonlinearly scaled spatial distance between them. If the central pixel is the outlier, the output pixel is subsequently corrupted. Using a more robust estimate of the intensity
of the true imaged surface at that pixel location should lead to a better quality output image with lower noise variation. Careful selection of this robust estimate would permit the possibility of accounting for the visual structure appropriately, as was achieved in Chapter 5 of this thesis.

Other directions in future research naturally suggest themselves as extensions on this current work. For example, computing weakly monotonic averages on lattices, for the reduction of coloured images. The results in Chapter 5 regarding the reduction of the colour image of the Port of Los Angeles suggest that it may not be necessary to obtain the mode-like average on a lattice, since the reduction of the individual colour channels produced an image with low chroma noise. Further investigation of the level of chroma noise generated or eliminated is required and comparison to reduction on the lattice performed, to establish the most appropriate and practical procedure.

Perhaps the most exciting direction for new research opened up by this thesis is the investigation of the generalisations of the density based averages. These averages offer significant advantages over traditional methods for maintaining aggregate statistics of streamed data and enable the computation of a weakly monotonic average given a new data point and the previously computed average prior to obtaining the new data. In the age of “big data” analysis problems, where streams such as stock prices, videos and even changing user preferences over time are being mined for “new knowledge”, the possibility that density based averages could contribute new useful statistics motivates further investigation.

Along these lines, the representation of stream data using feature variables derived from summary statistics arises in applications such as activity recognition from accelerometer sensor data. The use of density based averages as feature variables offers a computationally efficient method for maintaining the feature vector at the current time given new sensor observations, which promotes efficiency improvements on small devices. The possibility of embedding these algorithms into microprocessors running both the data aggregation and machine learning and deploying these devices into novel environments, is opening up new research areas for the application of non-monotonic averages.
This research has met its stated objectives and created a significant opportunity for many future research investigations that will build upon the unification of monotonic and non-monotonic averaging established by this thesis. There are many domains beyond images in which the location of data is relevant to its analysis and new problems in this area are arising regularly within the context of “big data”; particularly those related to location-based, or geo-tagged data. Further research is certainly needed to determine the most appropriate methods for summarising these massive data sets, however it is certain that averaging functions will play a definitive role in future research investigations. Through this thesis a far broader class of functions can now be studied from the perspective of averaging aggregation and assuredly, many new and interesting results will follow as a result.
References


