Discrete pulse transform of images and applications

by
Inger Nicolette Fabris-Rotelli

Submitted in fulfilment of the requirements for the degree

Philosophiae Doctor

in the Department of Mathematics and Applied Mathematics
in the Faculty of Natural and Agricultural Sciences
University of Pretoria
Pretoria

May 2012

I, Inger Nicolette Fabris-Rotelli declare that the dissertation, which I hereby submit for the degree PhD Mathematical Sciences at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

SIGNATURE:

DATE:

Summary

The LULU operators L_n and U_n operate on neighbourhoods of size n. The Discrete Pulse Transform (DPT) of images is obtained via recursive peeling of so-called local maximum and minimum sets with the LULU operators as n increases from 1 to the maximum number of elements in the array. The DPT provides a new nonlinear decomposition of a multidimensional array. This thesis investigates the theoretical and practical soundness of the decomposition for image analysis. Properties for the theoretical justification of the DPT are provided as consistency of the decomposition (a pseudo-linear property), and its setting as a nonlinear scale-space, namely the LULU scale-space. A formal axiomatic theory for scale-space operators and scale-spaces is also presented. The practical soundness of the DPT is investigated in image sharpening, best approximation of an image, noise removal in signals and images, feature point detection with ideas to extending work to object tracking in videos, and image segmentation. LULU theory on multidimensional arrays and the DPT is now at a point where concrete signal, image and video analysis algorithms can be developed for a wide variety of applications.

Acknowledgements

The most important thanks has to go to my supervisor Professor Roumen Anguelov. Throughout my Masters and now my PhD he has provided the perfect support and encouragement, and most importantly belief in my mathematical abilities. The provision by him of a mathematics topic which grabbed my interest enormously will forever be appreciated. I look forward to many more years of research together.

Secondly, thanks are in order to the CSIR for presenting their research opportunity to me. The annual PRISM meetings at the CSIR with other researchers to share ideas, collaborate and do some brainstorming, contributed greatly to my knowledge and ideas.

Thirdly, to my husband, Steven Fabris-Rotelli, for constant support and encouragement to keep going especially when I had spent days working on some idea but achieved nothing significant. Without him life would be incredibly hard.

My parents must also get credit for this achievement. Having the type of parents who never question the direction you move towards is truly a blessing. My announcement of deciding to study mathematics after high school was met with "What will you be able to do with that?" but even so complete support of my decision and what makes me happy. This is what you can do with mathematics... read on...

Collaborators throughout this work include Dr Stéfan van der Walt, Department of Applied Mathematics, Stellenbosch, whose amazing insight into coding LULU and the DPT provided me the way forward with all these applications. Certain aspects of some initial applications where done in collaboration with Stéfan. Also, Paul J van Staden, Department of Statistics, University of Pretoria, with whom to collaborate with is always an immensely exciting and thought-provoking experience. Thanks also to Dirk Laurie, De-

partment of Mathematics, Stellenbosch, for useful collaboration on the proof of the Highlight conjecture.

Thank you also to the Departments of Statistics as well as Mathematics and Applied Mathematics at the University of Pretoria for the stimulating environment in which to do this research.

Lastly, thanks must go to the anonymous examiners of this thesis. Their very valued input resulted in a vastly improved scientific write-up.

Contents

Chapter 1

Introduction

Gestalt theory sets a basis for vision perception [254, 130] but as it is non-computational it doesn't provide the required practical setting as well. David Marr, a British neuroscientist and psychologist, published the famous book known as the 'Vision of Marr' [138] in 1982 providing a modern theory for visual perception. Image processing is not a purely algorithmic field. The connection with artificial intelligence is undeniable and methods investigated inevitably require thought into how the human vision system operates and how it can be replicated. Marr realized this connection when writing his book and a number of earlier works [136, 137, 140, 139]. He defines vision as the 'process of discovering from images what is present in the world, and where it is', and defines three levels of the computer vision process (1) computational theory, (2) representation and algorithm, and (3) hardware implementation.

In [52] the computational theory of the LULU operators and the Discrete Pulse Transform (DPT) were presented in detail. Therein the algorithm was also presented in which the DPT is the successive slicing off of local maximum and minimum sets of increasing support size. Software implementation of the DPT has also progressed from MATLAB to Python, one dimension to two dimensions, and further implementation in three dimensions is envisioned. In [138, Chapter 1] it is described how advanced human vision uses both sides of the brain, the left side for shape vision and the right side to interpret the purpose of what is seen, dealing with these separately and then combining the interpretations. The Discrete Pulse Transform extracts discrete pulses of the image of every possible shape and size thereby setting the stage for an effective computer vision method.

As we move from the human vision system to develop a computer vision model, the first requirement is an operational definition of an image. Florack [61] provides this is some detail in the setting of algebras and groups, but the basic definition for practical purposes is the projection of a three dimensional environment into a two dimensional digitized (discrete) space. The process of acquiring an image is done by the capture of light onto a photographic film or more modernly as the conversion of a similarly obtained electric signal into a digital image. The most obvious acquirement process is with a normal digital camera but others such as infrared (long-wave), laser, night vision, and satellite cameras or capturing devices exist as well, including video capture which is simply sequences of images with the third dimension as time. The applications presented in this work are based on grey-scale images but can quite easily be extended to colour images as well as other forms. Methods presented would, however, need to be altered for the intrinsic manner in which different image types represent the image content but the basic ideas would still hold.

The main contributions of this thesis to the field of image analysis are as follows.

Development of the theory of Discrete Pulse Transform (DPT) for images
This follows the extension of the LULU operators to multiple dimensions which is presented in [52]. We prove that the properties if the DPT in one dimension can be generalised almost unchanged for the DPT on multidimensional arrays and in particular for images. The method of proof relies on the properties of morphological connectivity rather than that of finite sets of consecutive integers which makes proofs fundamentally different from the one dimension case. Further, they are applicable for any kind of connectivity since the only structure assumed for the domain is a morphological connection. The established properties of consistent decomposition and total variation preservation are applicable to any hierarchical decomposition and can be considered as a step towards a general formulation-independent theory of nonlinear hierarchical decompositions. Here we need to acknowledge that the property of strong consistency or the so-called highlight conjecture, formulated as an open problem for the one-dimensional DPT in 2005, was first proved by Dirk Laurie [112]. However, the basic consistency property stated here in Theorem 23 was proved first in our paper [8]. Moreover, the method of analysis of the DPT in this paper also led to the Highlight theorem thus providing an alternative proof of the result based on a morphological approach rather than one based on graphs as in [112].

DPT Scale-Space

Scale-space theory is an important approach to most of the problems in image analysis, e.g. feature detection, segmentation, noise removal, etc. We show that the DPT has a natural scale-space associated with it, namely the space of discrete pulses, which can be used successfully in addressing the mentioned problems. We further derived an axiomatic definition of a scale-space, which is an attempt to develop a unified scale-space theory encompassing those scale-spaces defined by integral operators like the ever popular Gaussian scale-space, and the morphological scale-spaces of which the DPT scale-space is a part. This work was published in [7].

Implementation and Application

The presented theory is developed ultimately for practical goals, thus its implementation and demonstration of usefulness in applications are its essential partner. This is particularly important for the DPT since it is very computationally intensive. Using an efficient computer algorithm for derivation and storage of the DPT pulses, the practical soundness of the DPT is investigated in image sharpening, best approximation of an image, noise removal in signals as well as images, feature point detection with ideas to extending work to object tracking in videos, and lastly image segmentation.

In Chapter 2 we summarize the LULU theory thoroughly presented in [52]. A look at characterizing the resulting Discrete Pulse Transform amongst nonlinear decompositions is presented Chapter 3.4. The Discrete Pulse Transform is connected to the frequently investigated scale-space theory in Chapter 4. A thorough review of the original Gaussian scale-space, a linear scale-space, is also provided as well as alternative scale-spaces and the progress of scale-spaces after the Gaussian scale-space. A formal, much needed, definition of a general scale-space is introduced together with a proof for the connection of the LULU scale-space via the Discrete Pulse Transform to this definition. In Sections 4.8.2 and 4.8.3 as well as Chapter 5 the basic image analysis techniques, namely feature detection, image segmentation and cleaning up of an image (sharpening and noise removal) are investigated. These provide an indication of the ability of the LULU operators and the Discrete Pulse Transform to perform the basics in image analysis. As a completely new theory, the foundation methods need to be investigated before more precise and more detailed methods are progressed towards. This work does just this.

Chapter 2

LULU Theory Background

The LULU and Discrete Pulse Transform (DPT) theory on sequences and multidimensional arrays was presented in detail in [52]. We repeat the work here in a summarized manner for completeness.

2.1 Setting

Let Ω be an abelian group, so that commutativity always holds. Recall that an Abelian group is an algebraic structure with a set, say G, and set operation, say $*$, satisfying the five axioms of closure (for all $a, b \in G \Rightarrow a * b \in G$), associativity ($(a * b) * c = a * (b * c)$), identity element ($\exists\ e \in G$ such that $a * e = e = e * a$ for all $a \in G$; e is called the identity element), inverse element (for each $a \in G$ there exists $b \in G$ such that $a * b = e = b * a$) and commutativity ($a * b = b * a$ for all $a, b \in G$) [79]. Denote by $\mathcal{A}(\Omega)$ the vector lattice of all real functions defined on Ω with respect to the usual point-wise defined addition, scalar multiplication and partial order. Let us recall that

Definition 1 *A partially ordered set L is a* **lattice** *if any $\ell_1, \ell_2 \in L$ admit a least upper bound $\ell_1 \vee \ell_2$ and a largest lower bound $\ell_1 \wedge \ell_2$. For a* ***vector lattice*** *we have that for two sequences $x = (x_n), y = (y_n)$ that $x \leq y \iff x_n \leq y_n\ \forall\ n \in \mathbb{Z}$.*

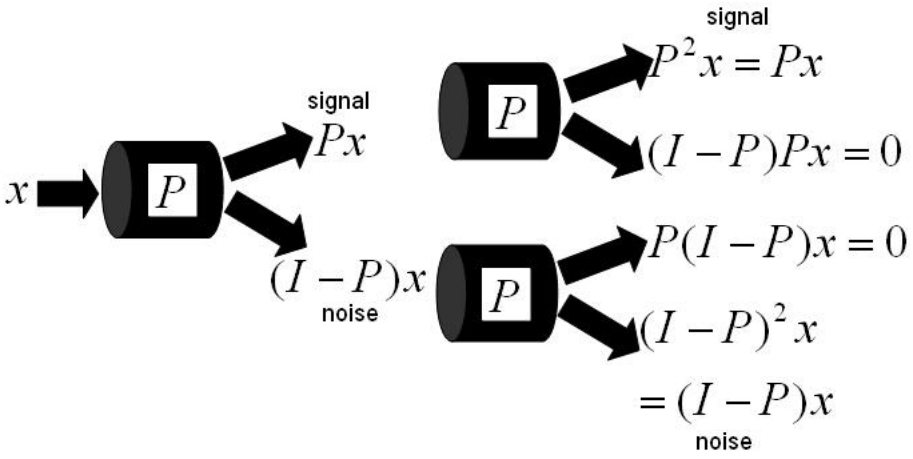


Figure 2.1: *The action of a separator P*

2.2 Separators

A common requirement for a filter P, linear or nonlinear, is its idempotence, i.e. $P \circ P = P$. For example, a morphological filter is by definition an increasing and idempotent operator. For linear operators the idempotence of P implies the idempotence of the complementary operator $id - P$, where id denoted the identity operator. For nonlinear filters this implication generally does not hold so the idempotence of $id - P$, also called co-idempotence, [243], can be considered as an essential measure of consistency.

For every $a \in \Omega$ the operator $E_a : \mathcal{A}(\Omega) \to \mathcal{A}(\Omega)$ given by $E_a(f)(x) = f(x-a),\ x \in \Omega$, is called a shift operator. We now define a separator which mimics the actions required of an operator P. The first three properties in Definition 2 define a **smoother**. More detail on smoothers can be found in [52].

Definition 2 *An operator $P : \mathcal{A}(\Omega) \to \mathcal{A}(\Omega)$ is called a* **separator** *if*

(i)	$P \circ E_a = E_a \circ P,\ a \in \Omega$	*(Horizontal shift invariance)*
(ii)	$P(f + c) = P(f) + c,\ f, c \in \mathcal{A}(\Omega),$	c *a constant function (Vertical shift invariance)*
(iii)	$P(\alpha f) = \alpha P(f),\ \alpha \in \mathbb{R},\ \alpha \geq 0,$	$f \in \mathcal{A}(\Omega)$ *(Scale invariance)*
(iv)	$P \circ P = P$	*(Idempotence)*
(v)	$(id - P) \circ (id - P) = id - P.$	*(Co-idempotence)*

Figure 2.1 illustrates the action of a separator P. It illustrates how a separator will separate the signal into noise and the true signal without the need for recursive smoothing, that is, it does the separation the first time completely so that there is no 'signal' left in the 'noise' nor any 'noise' left in the 'signal'. The median, for example, smoother requires recursive application and thus does not possess this desirable property.

2.3 One Dimensional LULU

The LULU operators and the associated Discrete Pulse Transform developed during the last three decades or so are an important contribution to the theory of the nonlinear multi-scale analysis of sequences. The basics of the theory as well as the most significant results until 2005 are published in the monograph [183]. For more recent developments and applications see [5], [38], [106], [113], [184]. This LULU theory was developed for sequences, that is, the case $\Omega = \mathbb{Z}$. Given a bi-infinite sequence $\xi = (\xi_i)_{i \in \mathbb{Z}}$ and $n \in \mathbb{N}$ the basic LULU operators L_n and U_n are defined as follows

$$(L_n\xi)_i = \max\{\min\{\xi_{i-n}, ..., \xi_i\}, ..., \min\{\xi_i, ..., \xi_{i+n}\}\},\ i \in \mathbb{Z}. \tag{2.1}$$

$$(U_n\xi)_i = \min\{\max\{\xi_{i-n}, ..., \xi_i\}, ..., \max\{\xi_i, ..., \xi_{i+n}\}\},\ i \in \mathbb{Z}. \tag{2.2}$$

Figure 2.2[1] illustrates how the operators L_1 and U_1 affect a sequence x, by respectively lowering or raising a local maximum or minimum point to the value of its nearest neighbour.

It is shown in [183] that for every $n \in \mathbb{N}$ the operators L_n and U_n as well as their compositions are increasing separators. Hence they are an appropriate

[1]Graphs are from collaborative work done with PJ van Staden and K van Oldenmark [56]

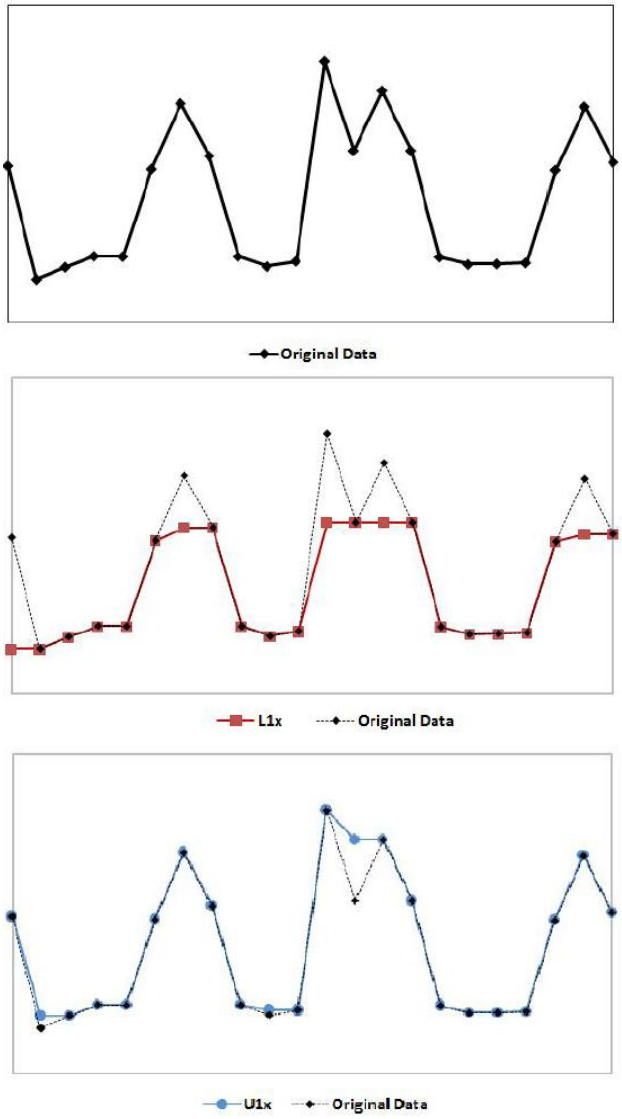


Figure 2.2: *An illustration of the effect of L_1 and U_1 on a sequence respectively*

tool for signal extraction. Furthermore they are fully trend preserving in the sense that both the operator and its complement preserve the monotonicity between consecutive terms in the input sequence. This implies that these operators are total variation preserving.

2.4 One Dimensions to n Dimensions

The definition of the operators L_n and U_n for sequences involves maxima and minima over sets of consecutive terms, thus, making essential use of the fact that $\mathbb{Z}$ is totally ordered. Since $\mathbb{Z}^d$, $d > 1$, is only partially ordered the concept of 'consecutive' does not make sense in this setting. Instead, we use the morphological concept of set connection, [207].

Definition 3 *Let B be an arbitrary non-empty set. A family $\mathcal{C}$ of subsets of B is called a connected class or a* **connection** *on B if*
(i) $\emptyset \in \mathcal{C}$
(ii) $\{x\} \in \mathcal{C}$ for all $x \in B$
(iii) for any family $\{C_i\} \subseteq \mathcal{C}$ we have $\bigcap_{i\in I} C_i \neq \emptyset \Longrightarrow \bigcup_{i\in I} C_i \in \mathcal{C}$.
If a set C belongs to a connection $\mathcal{C}$ then C is called ***connected****.*

This definition generalizes the topological concept of connectivity (i.e. a set is connected if it cannot be partitioned into two open disjoint sets) to arbitrary sets including discrete sets like $\mathbb{Z}^d$. It generalizes the concept of graph connectivity. If the underlying set B is a graph, then the graph connectivity also defines a connectivity.

2.5 n Dimensional LULU

Definition 4 *Given a point $x \in \mathbb{Z}^d$ and $n \in \mathbb{N}$ we denote by $\mathcal{N}_n(x)$ the set of all connected sets of size $n+1$ that contain point x, that is,*

$$\mathcal{N}_n(x) = \{V \in \mathcal{C} : x \in V,\ \text{card}(V) = n+1\}.$$

In addition to conditions assumed for the connection $\mathcal{C}$ we also assume that

$$\text{card}(\mathcal{N}_n(x)) < \infty,\ \forall n \in \mathbb{N},\ \forall x \in \Omega. \tag{2.3}$$

In image analysis ($d = 2$) the simplest and most obvious connectivity to make use of is a graph connectivity defined via a neighbour relation, e.g. 4-connectivity, 8-connectivity. However, in order to have maximum generality, we adopt the present axiomatic approach. Let us also mention that LULU operators on a continuous domain ($\Omega = \mathbb{R}$) are discussed in [9] and [5].

Now the operators L_n and U_n are defined on $\mathcal{A}(\mathbb{Z}^d)$ as follows.

Definition 5 *Let $f \in \mathcal{A}(\mathbb{Z}^d)$ and $n \in \mathbb{N}$. Then*

$$L_n(f)(x) = \max_{V \in \mathcal{N}_n(x)} \min_{y \in V} f(y), \ x \in \mathbb{Z}^d, \tag{2.4}$$

$$U_n(f)(x) = \min_{V \in \mathcal{N}_n(x)} \max_{y \in V} f(y), \ x \in \mathbb{Z}^d. \tag{2.5}$$

Let us confirm that Definition 5 generalizes the definition of L_n and U_n for sequences. Suppose $d = 1$ and let $\mathcal{C}$ be the connection on $\mathbb{Z}$ generated by the pairs of consecutive numbers. Then all connected sets on $\mathbb{Z}$ are sequences of consecutive integers and for any $i \in \mathbb{Z}$ we have

$$\mathcal{N}_n(i) = \{\{i-n, i-n+1, ..., i\}, \{i-n+1, i-n+2, ..., i+1\}, ..., \{i, i+1, ..., i+n\}\}.$$

Hence for an arbitrary sequence ξ considered as a function on $\mathbb{Z}$ the formulas (2.4) and (2.5) are reduced to (2.1) and (2.2), respectively.

2.6 Properties

Matheron Pair

An essential property of L_n and U_n is that they form a Matheron pair [8], that is we have

$$L_n \circ U_n \circ L_n = U_n \circ L_n \text{ and } U_n \circ L_n \circ U_n = L_n \circ U_n. \tag{2.6}$$

Area Opening and Closing

The operators L_n and U_n are an area opening and area closing respectively. It is well known that the area opening (closing) is an algebraic opening (closing). We may recall that a map is called an algebraic opening (closing) if it is

increasing, idempotent and anti-extensive (extensive). Then the following holds,

$$f \leq g \Longrightarrow (L_n(f) \leq L_n(g),\ U_n(f) \leq U_n(g)) \tag{2.7}$$
$$L_n \circ L_n = L_n, \quad U_n \circ U_n = U_n \tag{2.8}$$
$$L_n(f) \leq f \leq U_n(f) \tag{2.9}$$

Monotonicity

The operators are monotone with respect to n in the following sense,

$$n_1 < n_2 \Longrightarrow (L_{n_1} \geq L_{n_2},\ U_{n_1} \leq U_{n_2}). \tag{2.10}$$

Semigroup

The operators L_n, U_n and all their compositions form a four element semi-group with respect to composition. Moreover, this semi-group is fully ordered as follows,

$$L_n \leq U_n \circ L_n \leq L_n \circ U_n \leq U_n. \tag{2.11}$$

The semi-group is also a *band* which means that all elements are idempotent.

Separators

The operators L_n, U_n are separators for every $n \in \mathbb{N}$.

Action of the Operators

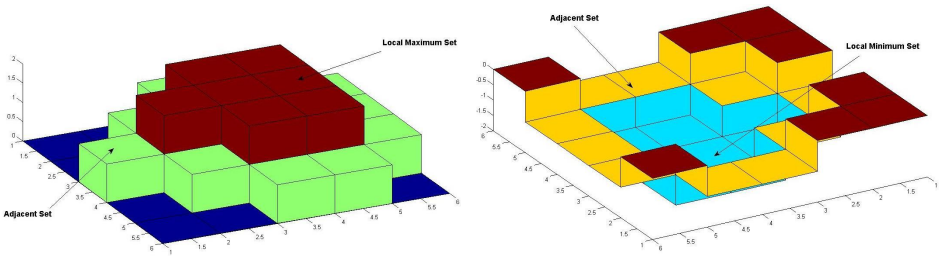
Similar to their counterparts for sequences the operators L_n and U_n defined for multidimensional arrays above smooth the input function by removing peaks (the application of L_n) and pits (the application of U_n). The smoothing effect of these operators is made more precise by using the concepts of a local maximum set and a local minimum set defined below.

Definition 6 *Let $V \in \mathcal{C}$. A point $x \notin V$ is called* **adjacent** *to V if $V \cup \{x\} \in \mathcal{C}$. The set of all points adjacent to V is denoted by* $\mathrm{adj}(V)$*, that is,*

$$\mathrm{adj}(V) = \{x \in \mathbb{Z}^d : x \notin V, V \cup \{x\} \in \mathcal{C}\}.$$

Definition 7 *A connected subset V of $\mathbb{Z}^d$ is called a* **local maximum set** *of $f \in \mathcal{A}(\mathbb{Z}^d)$ if*

$$\sup_{y \in \mathrm{adj}(V)} f(y) < \inf_{x \in V} f(x).$$



(a) (b)

Figure 2.3: *(a) A Local Maximum Set (b) A Local Minimum Set*

Similarly V is a **local minimum set** *if*

$$\inf_{y \in \mathrm{adj}(V)} f(y) > \sup_{x \in V} f(x).$$

Figure 2.3 provides an illustration for the concepts introduced in Definition 41. In this figure the case of constant sets is presented. Although this is not required by Definition 41 it is illustrated as such because the Discrete Pulse Transform acts on such sets due to the mechanism employed in its application. The following theorem illustrates this.

Theorem 8 *For $f \in \mathcal{A}(\mathbb{Z}^d)$,*

a) $L_n(f)$ is constant on any local maximum set W of f with $card(W) \leq n+1$

b) $U_n(f)$ is constant on any local minimum set W of f with $card(W) \leq n+1$

We present two more theorems which illustrate the relationship between the LULU operators and local maximum and minimum sets.

Theorem 9 *Let $f \in \mathcal{A}(\mathbb{Z}^d)$ and $x \in \mathbb{Z}^d$. Then we have*

a) $L_n(f)(x) < f(x)$ if and only if there exists a local maximum set V of f such that $x \in V$ and $\mathrm{card}(V) \leq n$;

b) $U_n(f)(x) > f(x)$ *if and only if there exists local minimum set* V *of* f *such that* $x \in V$ *and* $\text{card}(V) \leq n$.

Theorem 10 *Let* $f \in \mathcal{A}(\mathbb{Z}^d)$. *Then*

a) the size of any local maximum set of the function $L_n(f)$ *is larger than* n;

b) the size of any local minimum set of the function $U_n(f)$ *is larger than* n.

In summary the theorems provide the following characterization of the effect of the operators L_n and U_n on a function $f \in \mathcal{A}(\mathbb{Z}^d)$:

- The application of L_n (U_n) removes local maximum (minimum) sets of size smaller or equal to n.
- The operator L_n (U_n) does not affect the local minimum (maximum) sets in the sense that such sets may be affected only as a result of the removal of local maximum (minimum) sets. However, no new local minimum (maximum) sets are created where there were none. This does not exclude the possibility that the action of L_n (U_n) may enlarge existing local minimum (maximum) sets or join two or more local minimum (maximum) sets of f into one local minimum (maximum) set of $L_n(f)$ ($U_n(f)$).
- $L_n(f) = f$ ($U_n(f) = f$) if and only if f does not have local maximum (minimum) sets of size n or less;

Furthermore, as a consequence of the preceding results we obtain the following corollary.

Corollary 11 *For every* $f \in \mathcal{A}(\mathbb{Z}^d)$ *the functions* $(L_n \circ U_n)(f)$ *and* $(U_n \circ L_n)(f)$ *have neither local maximum sets nor local minimum sets of size* n *or less. Furthermore,*

$$(L_n \circ U_n)(f) = (U_n \circ L_n)(f) = f$$

if and only if f *does not have local maximum sets or local minimum sets of size less than or equal to* n.

Neighbour Trend Preservation

Definition 12 *An operator P is* **neighbour trend preserving** *if for any points $p, q \in \Omega$, such that $\{p, q\} \in \mathcal{C}$, and for $f \in \mathcal{A}(\mathbb{Z}^d)$ we have*

$$f(p) \leq f(q) \implies P(f)(p) \leq P(f)(q).$$

The operator P is **fully trend preserving** *if both P and $id - P$ are neighbour trend preserving.*

In Definition 12, for P to be fully trend preserving the requirement on $id - P$, that is the neighbour trend preserving property, can be equivalently formulated as

$$|P(f)(p) - P(f)(q)| \leq |f(p) - f(q)|. \tag{2.12}$$

The property (2.12) is called *difference reducing*.

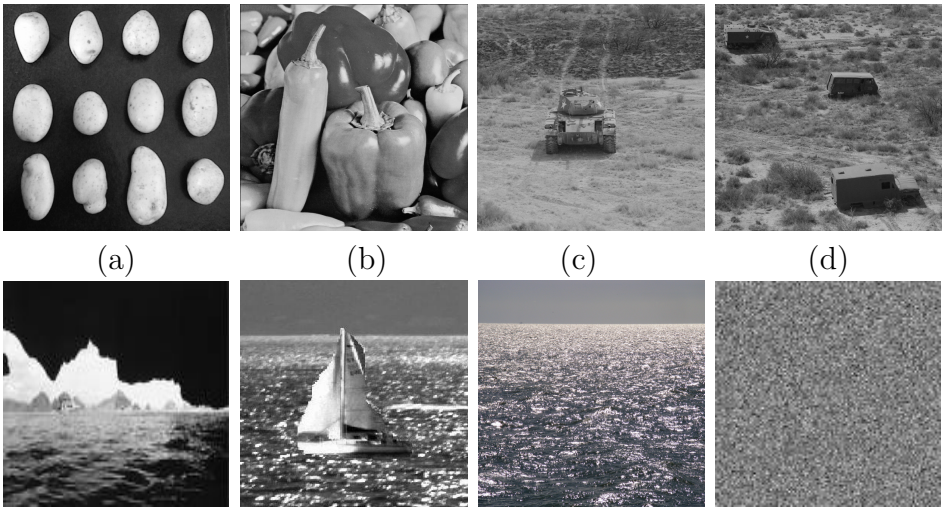
Theorem 13 *The operators L_n, U_n, $n = 1, 2, ...$, and their compositions, are all fully trend preserving.*

Total Variation Preserving

We assume for this section that the connection $\mathcal{C}$ on $\mathbb{Z}^d$ is defined via the so-called graph connectivity. More precisely, the points of $\mathbb{Z}^d$ are considered as vertices of a graph with edges connecting some of them. Equivalently, the connectivity of such a graph can be defined via a relation $r \subset \mathbb{Z}^d \times \mathbb{Z}^d$, where $p \in \mathbb{Z}^d$ is connected (by an edge) to $q \in \mathbb{Z}^d$ iff $(p, q) \in r$.

The relation r reflects what we consider neighbours of a point in the given context. For example, in image analysis ($d = 2$), it is common to use 4-connectivity (neighbours left, right, up and down) and 8-connectivity (in addition, the diagonal neighbours are considered). Let r be a relation on $\mathbb{Z}^d$. We call a set $C \subseteq \mathbb{Z}^d$ *connected*, with respect to the graph connectivity defined by r, if for any two points $p, q \in C$ there exists a set of points $\{p_1, p_2, ..., p_k\} \subseteq C$ such that each point is neighbour to the next one, p is neighbour to p_1 and p_k is neighbour to q. Here we assume that,

- r is reflexive, symmetric and shift invariant (2.13)
- $(p, p + e_k) \in r$, for all $k = 1, 2, ..., d$ and $p \in \mathbb{Z}^d$, (2.14)
  where $e_k \in \mathbb{Z}^d$ is defined by $(e_k)_i = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k \end{cases}$



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2.4: *Sample images*

Conditions (2.13) and (2.14) ensure that the set of connected sets $\mathcal{C}$ defined through this relation is a connection in terms of Definition 3. Condition (2.14) is essential to the definition of total variation as will be seen in the sequel.

Since the information in an image is in the contrast, the total variation of the luminosity function is an important measure of the quantity of this information. Image recovery and noise removal via total variation minimization are discussed in [34] and [193]. It should be noted that there are several definitions of total variation for functions of multi-dimensional argument (Arzelá variation, Vitali variation, Pierpont variation, Hardy variation, etc., see [2] [36] [164]). In the applications cited above the total variation is the L^1 norm of a vector norm of the gradient of the function. Here we consider a discrete analogue of this concept. Namely, the *Total Variation* of $f \in \mathcal{A}(\mathbb{Z}^d)$ if given by

$$TV(f) = \sum_{p\in\mathbb{Z}^d} \sum_{i=1}^{d} |f(p + (e_k)_i) - f(p)|. \tag{2.15}$$

If $TV(f) < \infty$, then f is said to be of *bounded variation*. Table 2.6 gives the total variation of a few sample images seen in Figure 2.4. Notice that the pure noise image, Figure 2.4(h), has the highest total variation and as the images contain more homogenous areas their total variation reduces.

Let us denote by $BV(\mathbb{Z}^d)$ the set of all functions of bounded variation in

Image in Figure 2.4	Total Variation (standardized)
(a)	109173
(b)	132527
(c)	167011
(d)	193650
(e)	235908
(f)	245480
(g)	386408
(h)	703707

Table 2.1: *Standardized Total Variation of Some Sample Images*

$\mathcal{A}(\mathbb{Z}^d)$. Clearly, all functions of finite support are in $BV(\mathbb{Z}^d)$. For example, the luminosity functions of images are in $BV(\mathbb{Z}^2)$. Note that when $d = 1$ equation 2.15 gives the total variation of sequences as discussed in [183, Chapter 6]. Similar to sequences the total variation in equation 2.15 is a semi-norm. An operator P on $BV(\mathbb{Z}^d)$ is called *total variation preserving* if

$$TV(f) = TV(P(f)) + TV((id - P)(f)). \tag{2.16}$$

It is natural to expect that a good separator P will not create new variation as this property requires. An operator P satisfying property 2.16 is called *total variation preserving* [185].

Theorem 14 *The operators L_n, U_n, $n = 1, 2, ...,$ and all their compositions, are total variation preserving.*

2.7 Conclusion

This chapter provided a summary on the theory of the LULU operators developed for multidimensional arrays. We now proceed with the resulting Discrete Pulse Transform.

Chapter 3

The Discrete Pulse Transform

3.1 The Discrete Pulse Transform

The Discrete Pulse Transform based on the LULU operators for sequences was derived in [183] [113] [184]. Using the extension of the LULU operators to functions on $\mathbb{Z}^d$ we present the DPT for functions in $\mathcal{A}(\mathbb{Z}^d)$. Similar to the case of sequences we obtain a decomposition of a function $f \in \mathcal{A}(\mathbb{Z}^d)$, with finite support. As usual $\text{supp}(f) = \{p \in \mathbb{Z}^d : f(p) \neq 0\}$. Let $N = \text{card}(\text{supp}(f))$. We derive the DPT of $f \in \mathcal{A}(\mathbb{Z}^d)$ by applying iteratively the operators L_n, U_n with n increasing from 1 to N as follows

$$DPT(f) = (D_1(f), D_2(f), ..., D_N(f)), \tag{3.1}$$

where the components of (3.1) are obtained through

$$D_1(f) = (id - P_1)(f) \tag{3.2}$$
$$D_n(f) = (id - P_n) \circ Q_{n-1}(f), \ n = 2, ..., N, \tag{3.3}$$

and $P_n = L_n \circ U_n$ or $P_n = U_n \circ L_n$ and $Q_n = P_n \circ ... \circ P_1$, $n \in \mathbb{N}$. We will show that this decomposition has the property that each component D_n in (3.1) is a sum of discrete pulses with pairwise disjoint supports of size n, where in this setting a discrete pulse is defined as follows.

Definition 15 *A function $\phi \in \mathcal{A}(\mathbb{Z}^d)$ is called a pulse if there exists a connected set V and a nonzero real number α such that*

$$\phi(x) = \begin{cases} \alpha & \text{if} \quad x \in V \\ 0 & \text{if} \quad x \in \mathbb{Z}^d \setminus V. \end{cases}$$

The set V is the support of the pulse ϕ, that is $\text{supp}(\phi) = V$.

The concept of a pulse as defined in Definition 15 is similar to the idea of a flat zone from mathematical morphology. It should be remarked that the support of a pulse may generally have any shape, the only restriction being that it is connected. Note that the smoothing process ultimately results in the last component $D_N(f)$ being a constant image, that is, one pulse the size of the entire image, and the remaining image component is $Q_N(f) = 0$.

It follows from (3.2) and (3.3) that

$$f = \sum_{n=1}^{N} D_n(f). \tag{3.4}$$

The usefulness of the representation (3.4) of a function $f \in \mathcal{A}(\mathbb{Z}^d)$ is in the fact that all terms are sums of pulses as stated in the sequel. First we need to state a technical lemma.

Lemma 16 *Let $f \in \mathcal{A}(\mathbb{Z}^d)$, $\text{supp}(f) < \infty$, be such that f does not have local minimum sets or local maximum sets of size smaller than n, for some $n \in \mathbb{N}$. Then we have the following two results.*

a)

$$(id - P_n)f = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} + \sum_{j=1}^{\gamma^+(n)} \varphi_{nj}, \tag{3.5}$$

where $V_{ni} = \text{supp}(\phi_{ni}), i = 1, 2, ..., \gamma^-(n)$, are local minimum sets of f of size n, $W_{nj} = \text{supp}(\varphi_{nj}), j = 1, 2, ..., \gamma^+(n)$, are local maximum sets of f of size n, ϕ_{ni} and φ_{nj} are negative and positive discrete pulses respectively, and we also have that

- $V_{ni} \cap V_{nj} = \emptyset$ *and* $\text{adj}(V_{ni}) \cap V_{nj} = \emptyset$,
  $i, j = 1, ..., \gamma^-(n),\ i \neq j,$ (3.6)
- $W_{ni} \cap W_{nj} = \emptyset$ *and* $\text{adj}(W_{ni}) \cap W_{nj} = \emptyset$,
  $i, j = 1, ..., \gamma^+(n), i \neq j,$ (3.7)
- $V_{ni} \cap W_{nj} = \emptyset$
  $i = 1, ..., \gamma^-(n)\ , j = 1, ..., \gamma^+(n).$ (3.8)

b) For every fully trend preserving operator A

$$U_n(id - AU_n) = U_n - AU_n,$$
$$L_n(id - AL_n) = L_n - AL_n.$$

Proof
a) Let $V_{n1}, V_{n2}, ..., V_{n\gamma^-(n)}$ be all local minimum sets of size n of the function f. Since f does not have local minimum sets of size smaller than n, then f is a constant on each of these sets, by Theorem 8. Hence, the sets are disjoint, that is $V_{ni} \cap V_{nj} = \emptyset$, $i \neq j$. Moreover, we also have

$$\text{adj}(V_{ni}) \cap V_{nj} = \emptyset, \ i, j = 1, ..., \gamma^-(n). \tag{3.9}$$

Indeed, let $x \in \text{adj}(V_{ni}) \cap V_{nj}$. Then there exists $y \in V_{ni}$ such that $(x, y) \in r$. Hence $y \in V_{ni} \cap \text{adj}(V_{nj})$. From the local minimality of the sets V_{ni} and V_{nj} we obtain respectively $f(y) < f(x)$ and $f(x) < f(y)$, which is clearly a contradiction. For every $i = 1, ..., \gamma^-(n)$ denote by y_{ni} the point in $\text{adj}(V_{ni})$ such that

$$f(y_{ni}) = \min_{y \in \text{adj}(V_{ni})} f(y). \tag{3.10}$$

Then we have

$$U_n f(x) = \begin{cases} f(y_{ni}) & \text{if } x \in V_{ni},\ i = 1, ..., \gamma^-(n) \\ \\ f(x) & \text{otherwise (by Theorem 9)} \end{cases}$$

Therefore

$$(id - U_n)f = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} \tag{3.11}$$

where ϕ_{ni} is a discrete pulse with support V_{ni} and negative value (down pulse).

Let $W_{n1}, W_{n2}, ..., W_{n\gamma^+(n)}$ be all local maximum sets of size n of the function $U_n f$. By [8, Theorem 12(b)] every local maximum set of $U_n f$ contains a local maximum set of f. Since f does not have local maximum sets of size smaller than n, this means that the sets W_{nj}, $j = 1, ..., \gamma^+(n)$, are all local maximum sets of f and f is constant on each of them. Similarly to the local minimum sets of f considered above we have $W_{ni} \cap W_{nj} = \emptyset$, $i \neq j$, and $\text{adj}(W_{ni}) \cap W_{nj} = \emptyset$, $i, j = 1, ..., \gamma^+(n)$. Moreover, since $U_n(f)$ is constant on any of the sets $V_{ni} \cup \{y_{ni}\}$, $i = 1, ..., \gamma^-(n)$, see Theorem 8, we also have

$$(V_{ni} \cup \{y_{ni}\}) \cap W_{nj} = \emptyset, \ i = 1, ..., \gamma^-(n), \ j = 1, ..., \gamma^+(n), \tag{3.12}$$

which implies (3.8).

Further we have

$$L_nU_nf(x) = \begin{cases} U_nf(z_{nj}) & \text{if } x \in W_{nj},\ j = 1, ..., \gamma^+(n) \\ U_nf(x) & \text{otherwise} \end{cases}$$

where $z_{nj} \in \text{adj}(W_{nj})$, $j = 1, ..., \gamma^+(n)$, are such that $U_nf(z_{nj}) = \max\limits_{z \in \text{adj}(W_{nj})} U_nf(z)$. Hence

$$(id - L_n)U_nf = \sum_{j=1}^{\gamma^+(n)} \varphi_{nj} \tag{3.13}$$

where φ_{nj} is a discrete pulse with support W_{nj} and positive value (up pulse). Thus we have shown that

$$(id - P_n)f = (id - U_n)f + (id - L_n)U_nf = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} + \sum_{j=1}^{\gamma^+(n)} \varphi_{nj}.$$

b) Let the function $f \in \mathcal{A}(\mathbb{Z}^d)$ be such that it does not have any local minimum or local maximum sets of size less than n. Denote $g = (id - AU_n)(f)$. We have

$$g = (id - AU_n)(f) = (id - U_n)(f) + ((id - A)U_n)(f). \tag{3.14}$$

As in (a) we have that (3.11) holds, that is we have

$$(id - U_n)(f) = \sum_{i=1}^{\gamma^-(n)} \phi_{ni}, \tag{3.15}$$

where the sets $V_{ni} = \text{supp}(\phi_{ni})$, $i = 1, ..., \gamma^-(n)$, are all the local minimum sets of f of size n and satisfy (3.6). Therefore

$$g = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} + ((id - A)U_n)(f). \tag{3.16}$$

Furthermore,

$$U_n(f)(x) = \begin{cases} f(x) & \text{if} \quad x \in \mathbb{Z}^d \setminus \bigcup\limits_{i=1}^{\gamma^-(n)} V_{ni} \\ v_i & \text{if} \quad x \in V_{ni} \cup \{y_{ni}\},\ i = 1, ..., \gamma^-(n), \end{cases}$$

where $v_i = f(y_{ni}) = \min_{y \in \text{adj}(V_{ni})} f(y)$. Using that A is fully trend preserving, for every $i = 1, ..., \gamma^-(n)$ there exists w_i such that $((id - A)U_n)(f)(x) = w_i$, $x \in V_{ni} \cup \{y_{ni}\}$. Moreover, using that every adjacent point has a neighbor in V_{ni} we have that $\min_{y \in \text{adj}(V_{ni})} ((id - A)U_n)(f)(y) = w_i$. Considering that the value of the pulse ϕ_{ni} is negative, we obtain through the representation (3.16) that V_{ni}, $i = 1, ..., \gamma^-(n)$, are local minimum sets of g.

Next we show that g does not have any other local minimum sets of size n or less. Indeed, assume that V_0 is a local minimum set of g such that $\text{card}(V_0) \leq n$. Since $V_0 \cup \text{adj}(V_0) \subset \mathbb{Z}^d \setminus \bigcup_{i=1}^{\gamma^-(n)} V_{ni}$ it follows from (3.16) that V_0 is a local minimum set of $((id - A)U_n)(f)$. Then using that $(id - A)$ is neighbor trend preserving and using [8, Theorem 17] we obtain that there exists a local minimum set W_0 of $U_n(f)$ such that $W_0 \subseteq V_0$. Then applying again [8, Theorem 17] or [8, Theorem 12] we obtain that there exists a local minimum set $\tilde{W}_0$ of f such that $\tilde{W}_0 \subseteq W_0 \subseteq V_0$. This inclusion implies that $\text{card}(\tilde{W}_0) \leq n$. Given that f does not have local minimum sets of size less than n we have $\text{card}(\tilde{W}_0) = n$, that is $\tilde{W}_0$ is one of the sets V_{ni} - a contradiction. Therefore, V_{ni}, $i = 1, ..., \gamma^-(n)$, are all the local minimum sets of g of size n or less. Then using again (3.11) we have

$$(id - U_n)(g) = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} \tag{3.17}$$

Using (3.15) and (3.17) we obtain

$$(id - U_n)(g) = (id - U_n)(f)$$

Therefore

$$\begin{aligned} (U_n(id - AU_n))(f) &= U_n(g) = g - (id - U_n)(f) \\ &= (id - AU_n)(f) - (id - U_n)(f) \\ &= (U_n - AU_n)(f). \end{aligned}$$

This proves the first identity. The second one is proved in a similar manner. ■

Theorem 17 *Let $f \in \mathcal{A}(\mathbb{Z}^d)$. For every $n \in \mathbb{N}$ the function $D_n(f)$ derived through (3.2) and (3.3) is a sum of discrete pulses with pairwise disjoint*

support, that is, there exist $\gamma(n) \in \mathbb{N}$ *and discrete pulses* ψ_{ns}, $s = 1, ..., \gamma(n)$, *such that*

$$D_n(f) = \sum_{s=1}^{\gamma(n)} \psi_{ns}, \tag{3.18}$$

and

$$\text{supp}(\psi_{ns_1}) \cap \text{supp}(\psi_{ns_2}) = \emptyset \text{ for } s_1 \neq s_2. \tag{3.19}$$

Moreover, if $n_1, n_2, s_1, s_2 \in \mathbb{N}$ *are such that* $n_1 < n_2$, $1 \leq s_1 \leq \gamma(n_1)$ *and* $1 \leq s_2 \leq \gamma(n_2)$. *Then*

$$\text{supp}(\psi_{n_1 s_1}) \cap \text{supp}(\psi_{n_2 s_2}) \neq \emptyset \Longrightarrow \text{supp}(\psi_{n_1 s_1}) \subset \text{supp}(\psi_{n_2 s_2}) \tag{3.20}$$

Proof
According to (3.3), D_n is obtained by applying $id - P_n$ to the function $Q_{n-1}(f)$ which, by the results in Chapter 1, does not have local maximum or minimum sets of size less than n. Thus by Lemma 16(a) we have that $D_n(f) = (id - P_n)Q_{n-1}(f)$ is a sum of pairwise disjoint discrete pulses as given in (3.5). More precisely,

$$D_n(f) = \sum_{s=1}^{\gamma(n)} \psi_{ns} = \sum_{i=1}^{\gamma^-(n)} \phi_{ni} + \sum_{j=1}^{\gamma^+(n)} \varphi_{nj},$$

where $\gamma(n) = \gamma^-(n) + \gamma^+(n)$. Property (3.19) follows from (3.6)–(3.8).

Let $\text{supp}(\psi_{n_1 s_1}) \cap \text{supp}(\psi_{n_2 s_2}) \neq \emptyset$. It follows from the construction of (3.18) derived above that the functions $Q_n(f)$ and $L_{n+1}(Q_n(f))$, $n \geq n_1$, are constants on the set $\text{supp}(\psi_{n_1 s_1})$. Furthermore, the set $\text{supp}(\psi_{n_2 s_2})$ is a local maximum set of $Q_{n_2-1}(f)$ or a local minimum set of $L_{n_2}(Q_{n_2-1}(f))$. From the definition of local maximum set and local minimum set it follows that $\text{supp}(\psi_{n_1 s_1}) \subset \text{supp}(\psi_{n_2 s_2})$. ■

Using Theorem 17, the identity (3.4) can be written in the form

$$f = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \psi_{ns}. \tag{3.21}$$

The equality above is the discrete pulse decomposition of f, where the pulses ψ_{ns} have the properties (3.19) and (3.20).

3.2 Connectivity

Besides the formal connectivity definition presented in [207] and Definition 3, various examples have been investigated to deal with certain applications as well as specific problems encountered when using standard connectivity. (See [186] for a good summary on connectivity.) The most active researchers in the development of connectivity have been Serra [207, 214, 215, 42, 200, 208, 209, 210, 187, 212] and Braga-Neto [22, 23, 25, 24, 26, 20, 21, 27], as well as their collaborators.

The usefulness of Definition 3 in image processing arises from a further result of Serra's, namely, that the sets of a connection $\mathcal{C}$ of subsets of B is equivalent to the family of openings $\{\gamma_x : x \in B\}$ such that (i) $\forall\ x \in B, \gamma_x(x) = \{x\}$, (ii) $\forall\ A \subset B, x, y \in B, \gamma_x(A)$ and $\gamma_y(A)$ are equal or disjoint, and (iii) $\forall\ A \subset B, \forall\ x \in B, x \notin A \Rightarrow \gamma_x(A)\emptyset$. The class

$$\mathcal{C} = \{\gamma_x(A) : x \in B, A \subset B\}$$

is a connection such that each A is partitioned into the smallest possible number of components belonging to the class $\mathcal{C}$ and if $A_1 \subset A_2$ then any connected component of A_1 is included in a connected component of A_2. Note that as mentioned in [25] if one has a means of extracting connected components, we have unambiguously defined a connection and vice versa. We discuss variations of Serra's connectivity.

λ-connectivity

Morphological filters by reconstruction have a leakage problem, that is two connected blobs connected by a thin pixel-sized filament is deemed connected although they should more realistically be considered two separate connected components. Serra [212] introduced a new underlying lattice called the viscous lattice, obtained as all the dilated (with a circular structuring element with radius λ) subsets of B instead of simply the subsets, namely

$$\mathcal{L} = \{\delta_\lambda(A), A \subset B, \lambda > 0\}.$$

Santillán et al [203] define the lattice

$$\mathcal{L}_\lambda = \{\delta_\lambda(A), A \subset B\}$$

for a specific viscosity λ and the connected components of $\mathcal{L}_\lambda$ as λ-connected components. It requires that a connected set be made up of at the very least

continuous paths made by disks with radius λ whose centers move along the path.

Pseudo-connectivity

To combat the leakage problem pseudo-connections are introduced in [198] to improve segmentation. The minima of the negative distance function, namely $-\text{dist}(f)$, are ultimate erosions obtained via the watershed transform. Instead a thresholded version of the distance function is used, $\mathcal{D}_\ell(f) = -\left(\text{dist}(f) \bigwedge \ell\right)$. The case $\ell = 0$ corresponds to the classical connection by an erosion of size ℓ and $\ell = \infty$ corresponds to the case of ultimate erosions. Choosing a value of $\ell > 0$ results in a pseudo-connectivity. This solves the leakage problem but does introduce false contours for larger values of ℓ, so to combat this instead of applying the watershed transform to $-\text{dist}(f)$ it is applied to the distance transform of the closing by reconstruction which fills the holes which cause the false contours.

Hyperconnectivity Filters

In his 2009 PhD thesis, Ouzounis [162] describes hyperconnectivity introduced by Serra [208] for the improvement of attribute filters due to their leakage and overlap issues. Hyperconnectivity is a relaxation of the connection defined in Definition 3.

Definition 18 *Operator* $\perp\colon \mathcal{P}(\mathcal{P}(B)) \mapsto \{0,1\}$ *is an* overlap criterion *such that* $\perp$ *is decreasing* $A_1 \subset A_2 \Rightarrow \perp(A_1) \leq \perp(A_2)$ *and* $\perp(A) = 1$ *means that* A *is overlapping and non-overlapping otherwise.*

Definition 19 *Let* B *be an arbitrary non-empty set. A family* $\mathcal{H}$ *of subsets of* B *is called a hyperconnected class or a* hyperconnection *on* B *if*
i) $\emptyset \in \mathcal{H}$
(ii) $\{x\} \in \mathcal{H}$ *for all* $x \in B$
(iii) for any family $\{C_i\} \subseteq \mathcal{H}$ *for which* $\perp(\{C_i\}) = 1$ *we have* $\bigcup_{i \in I} C_i \in \mathcal{H}$.
If a set C *belongs to a hyperconnection* $\mathcal{H}$ *then* C *is called* hyperconnected.

All connectivity classes are special cases of hyperconnectivity with

$$\perp(\{C_i\}) = \begin{cases} 1 & \text{if } \bigcap C_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases}.$$

Constrained Connectivity

Soille [217] restricts Definition 3 by requiring that the intensity value difference between neighbouring pixels in a connected set do not exceed a specified value α. He calls this constrained connectivity α-connectivity and based on this develops a connectivity index for each pixel.

Jump Connection

Serra [211] defines a *jump connection* which provides good segmentations in that the are fewer point zones (regions of only a few pixels), visually significant clusters and can be computed fast. The connected components

$$A(m) = \{x : x \in B, 0 < f(x) - m \leq k\}$$

where m if the minimum of f form a connection.

Multiscale Connectivity

As Braga-Neto and Goutsias [24, 22] state, important information is not confined to a single scale, but rather is spread out over several scales. The connectivity should also thus depend on the scale. Let Σ be the set of possible scales. We provide a brief definition of multiscale connectivity.

Definition 20 *For a connectivity measure ρ defining the degree of connectivity of a set A on a lattice, the set A is* σ-connected *if $\rho(A) \geq \sigma$ for $\sigma \in \Sigma$. A set is then* fully connected *if it is σ-connected for all $\sigma \in \Sigma$ or* fully disconnected *otherwise.*

In [27] Braga-Neto and Goutsias discuss three techniques for constructing multiscale connectivities, namely, pyramids of clustering, granulometries and clustering of openings. In [20] they apply multiscale connectivity to determine a scale-space representation for automatic target detection. They also describe connectivity for greyscale images [26]. A greyscale image is connected if all level sets below a pre-specified threshold are connected, namely level-k connectivity requires all level sets at level k or below are non-empty and connected. They investigate applications in object extraction, segmentation, object-based filtering and hierarchical image representations.

Other Connections

Ronse and Serra [187] define geodesic connectivity. Braga-Neto and Goutsias [25] elaborate on fuzzy connectivity first introduced by Rosenfeld [188, 189, 190].

Connected Operators

Salembier and Wilkinson [201] describe how filters based on shapes not related to the input result in distortion and thus must be adapted to local structures as well as connected operators made use of. Connected operators preserve edge, and thus shape, information in an image [26, 42].

Definition 21 *An operator P is* connected *[26] if for any $f \in \mathcal{A}(\Omega)$ the partition $z_{P(f)}$ is coarser than the partition z_f, that is, $z_f(x) \subseteq z_{P(f)}(x)\ \forall\ x \in \Omega$, or equivalently $P(f)$ is constant over any flat zone of f. Here, $z_f(x) = \gamma_x(f(x)), x \in \Omega$ is the connectivity opening associated with the connection $\mathcal{C}$ discussed at the beginning of the section and a flat zone is that largest connected region where the value of f is constant.*

Connected operators have been widely made use of due to their preservation properties. Salembier et al [199, 197] state that the first reported connected operators were the binary openings by reconstruction done in 1976 and generalized to gray-level functions by reconstruction in 1993. They mention other connected operators as the λ-max operator, the area opening, dynamic filtering, volumic operator, complexity operator, motion operator, and moment-oriented operators which all preserve contours. Jones [102] introduces non-flat gray-level connected filters. Crespo and Schafer [41] specify two constraints for connected operators which morphological operators do satisfy, but the median operator, for example, does not.

Wilkinson and Ouzounis [244] state the obvious about standard morphological connected operators, that is, each type of structuring element has a limited power to represent an image due to its specific shape. The LULU operators act like the morphological area operators in that the shape is not specified, only the size, thus allowing far more flexibility. The fact that the LULU operators are connected according to Definition 21 adds further strength to their capabilities. We discuss these capabilities later in this chapter.

3.3 LULU Implementation

In [52] a `MATLAB` implementation of the DPT was presented. The algorithm used was by no means optimal nor real time. Since then collaboration with

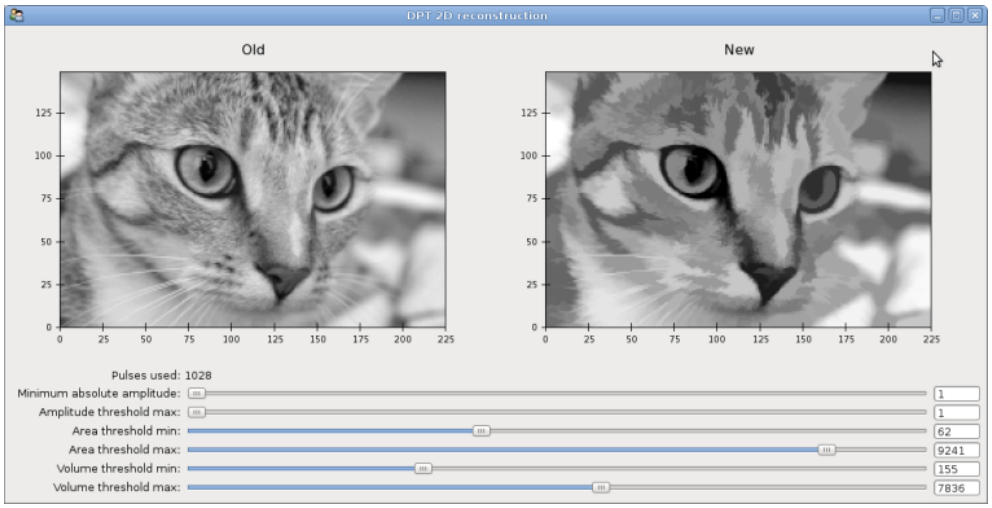


Figure 3.1: *Snapshot of the DPT User Interface*

Dr S van der Walt from Stellenbosch University has resulted in a `Python` implementation of the DPT in two dimensions (for image analysis) which provides a user interface, see Figure 3.1, with which to choose scale levels and other parameters and view the transformed image, in addition to significantly faster processing times [55, 231]. We discuss this implementation now which is collaborative work presented in [55].

The Discrete Pulse Transform decomposes a signal into a collection of pulses. In one dimension, a pulse is characterised by its start and end position, as well as by its amplitude. In two dimensions, a pulse describes a connected region over which function values are constant (for simplicity, we restrict ourselves here to 4-connectivity - where two function values are equal in the North-South or East-West directions). The number of pulses may vary from approximately 30,000 for a typical 300×300 image to over a hundred thousand for a 500×500 image. Since the decomposition produces such a large number of pulses, we need an efficient storage scheme to represent these in memory. Furthermore, we need to be able to calculate certain attributes of the pulses (such as the area and the boundary values) rapidly.

3.3.1 Storage

The storage scheme used is based on the popular Compressed Sparse Row (CSR) format, [49, 14], for representing sparse matrices. Using this scheme, the matrix

$$\begin{bmatrix} 5 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 9 & 0 \end{bmatrix}$$ is written as

- `values` $= \begin{bmatrix} 5 & 1 & 2 & 3 & 6 & 9 \end{bmatrix}$
- `columns` $= \begin{bmatrix} 0 & 2 & 3 & 3 & 1 & 3 \end{bmatrix}$
- `row_offset` $= \begin{bmatrix} 0 & 3 & 4 & 5 \end{bmatrix}$

.

The values of the non-zero elements are stored in `values`, and their column-positions given by `columns`. Each entry of `row_offset` specifies an offset into `columns`, indicating the starting position of a new row. In the example above, we see that the second row (second element of `row_offset`) starts at position 3 of `columns`. The number of elements in row j is given by `row_offset`$[j+1] -$ `row_offset`$[j]$.

When storing 2-dimensional pulses, we know that the pulse may only occupy a small portion of the image, has a single value across the pulse and consists of regions connected horizontally or vertically. We therefore modify the storage structure, so that the pulse $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$ is written as

- `value` $= 1$
- `columns` $= \begin{bmatrix} 0 & 5 & 1 & 3 & 4 & 5 & 1 & 4 \end{bmatrix}$
- `start_row` $= 1$
- `row_offset` $= \begin{bmatrix} 0 & 2 & 6 & 8 \end{bmatrix}$

.

Instead of specifying column values, `columns` now indicates the start and past-end indices of the one-dimensional pulses that comprise the rows. The values of `row_offset`, as in the previous example, specify where in `columns` each new row starts. The pulse may only cover a few rows of the entire image, therefore we use `start_row` to indicate the first occurrence, saving us from storing every single row.

As an example, consider the third row of the two-dimensional pulse above, which consists of two one-dimensional pulses: the first stretching from column 1 up to (but excluding) 3, the other from 4 up to 5. Since we are inter-

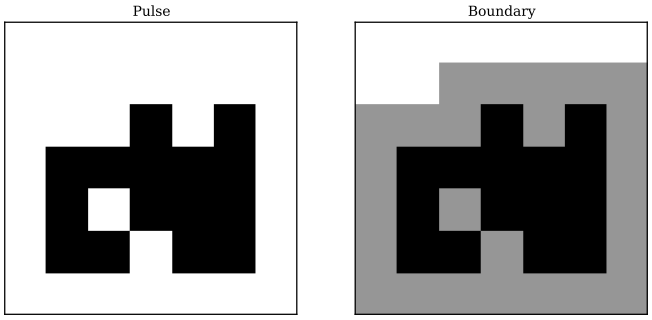


Figure 3.2: *Boundary positions of a pulse.*

ested in the third row (row number 2), and we only start recording rows at `start_row` $= 1$, we find the corresponding column indices in `row_offset`$[2-1] = 2$. At position 2, `columns` contains 1, 3 and 4, 5 as expected.

An advantage of this storage scheme is that it can also be used to store connected regions, a capability we exploit later to initialize the algorithm.

3.3.2 Queries

Given a pulse in the above format, we'd like to compute the following queries rapidly:

Area/number of non-zeros The area of the pulse is the sum of the lengths of the one-dimensional pulses comprising its rows. Each such length is given as the corresponding difference between the pulse start-end positions in `columns`. In the example above, the area would be $(5-0)+(3-1)+(5-4)+(4-1) = 5+2+1+3 = 11$.

Adjacent Set/Boundary positions Each pulse has four or more boundary positions – connected to the pulse in a 4-connected sense (see Fig 3.2) – that form the adjacent set. To find the boundary positions, we follow a scanline approach, with three scanlines moving from the top of the pulse to the bottom (see Fig. 3.3). Here, we describe the operation once the scanlines have entered the pulse (in other words, neglecting top and bottom boundaries, which need to be handled separately):

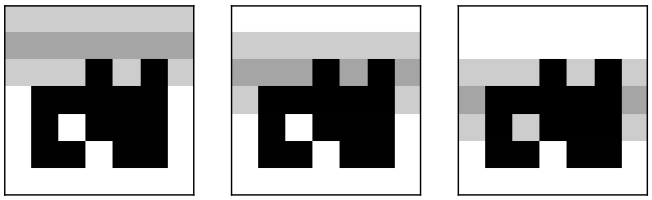


Figure 3.3: *Scanlines used to find boundary positions.*

1. The scanlines are centred around row j and are formed by constructing the pulse at rows $j-1$, j and $j+1$.

2. For each element of the central scanline that does *not* belong to the pulse, determine whether any of its neighbours (above, below, left or right) belong to the pulse. If they do, then that element lies on the boundary.

3. Move the scanlines one row down and repeat (it is only necessary to recalculate the bottom scanline at each step).

3.3.3 Operations

Merging Two Pulses Later on, when performing the Discrete Pulse Transform, we shall be required to merge two pulses that touch. This is done on a row-by-row basis. In the trivial case where a row is contained in only one of the two pulses, we simply include that row in the output. Otherwise, we need to sort and join the one-dimensional pulses that comprise the row carefully. Note, however, that these one-dimensional pulses cannot overlap in our problem description. We therefore:

1. Extract the stop-start intervals that form the one-dimensional pulses in row j.

2. Sort the intervals according to their starting position.

3. Step over the intervals and link them if they touch.

4. Save the linked intervals as the representation of row j.

5. Proceed to row $j+1$ and repeat.

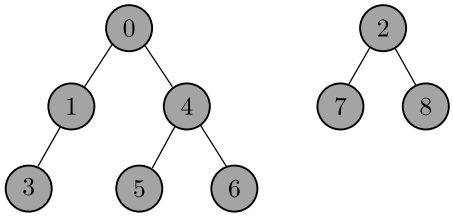


Figure 3.4: Two trees with labeled nodes.

3.3.4 Algorithm Overview

Each step of the Discrete Pulse Transform is now described in more detail. We'll use the following terms:

Input image The input image or data – an $M \times N$ matrix of integer values between 0 and 255.

Label image An $M \times N$ array of integer values that indicate the connectivity of pixels in an image. If neighbouring pixels have the same value (i.e., are 4-connected), then they are assigned the same label value.

Intermediate reconstruction An $M \times N$ image can be decomposed into pulses with areas ranging from 1 through MN . When summed, these pulses reconstruct the **input image**. It is also possible to only sum pulses with area $> k$. We call this an intermediate reconstruction, as it approximates the image up to a certain level only.

Finding Connection Regions First, we identify all 4-connected regions in the image (these are the initial pulses that are processed to yield the Discrete Pulse Transform), a pre-processing step also suggested by Lindeberg [123, Chapter 9.1] for comparing properties of constant grey-level regions. Our implementation uses the the Union-Find connected component algorithm of Fiorio and Gustedt [60], with the connectivity tree stored in an array as suggested by Wu et al. in [248]. It is shown in [248] that this algorithm executes in an optimal $\mathcal{O}(N)$, and we give a brief overview of its functioning:

Representing a tree using an array One or more trees consisting of N nodes can be stored in an array of length N. Examine the trees shown in

Figure 3.4 with nodes labeled $n = 0, \ldots, 8$. These trees can be represented as the array

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 2 & 1 & 0 & 4 & 4 & 2 & 2 \end{bmatrix}$$

where $\mathbf{x}_n$ gives the parent of node n. For example, $x_3 = 1$, which tells us that the parent of node three is node one. Similarly, $x_2 = 2$ implies that node two has no parent–it is the root of a tree.

Labeling connected regions as trees The goal of the connected components algorithm is to assign unique labels to each connected region in an $M \times N$ image I. An array, L, of length MN is used to store trees as indicated in the paragraph above.

The image is traversed in raster scan order (i.e. along rows). A region counter, k, is initialized to zero. At each pixel position (r, c):

1. Calculate the offset into the tree array as $t = rN + c$.
2. If the pixel is not connected to (does not have the same value as) the pixel above it or to the left, assign $L_t = t$, effectively creating a new tree.
3. If the pixel is connected to the pixel above, assign $L_t = L_{t-N}$, joining node t to its parent in the previous row.
4. If, in addition, the pixel is connected to the left, assign $L_{t-1} = L_{t-N}$.
5. If the pixel is only connected to the left, assign $L_t = L_{t-1}$.

Appropriate care needs to be taken in the first row and column to prevent indexing errors on the image boundary.

The label vector, L, can also be seen as the flattened version of a *label image* so that $L_{r,c} = L_{rN+c}$. From this image, all connected regions are extracted as pulses and stored in the format discussed in Section 3.3.1. We then proceed to perform the Discrete Pulse Transform as discussed next.

Identifying Pulses to Merge The Discrete Pulse Transform is performed by alternately executing the L_k (lower) and U_k (upper) operators on pulses of area k. Thinking of the image as a height-map, the U_1-operator removes

all valleys of area one. Here, a valley is defined as a connected area that is surrounded only by higher values. Similarly, the L_1-operator removes peaks of area one, where peaks are connected areas surrounded only by lower values.

After applying the L_1 and U_1 operators and *storing the removed peaks and valleys* (those form the first level of the DPT), we need to merge pulses that were joined in the process. Note that, at each decomposition level, we have the *intermediate reconstruction* available. It is obtained by setting the image values corresponding to the removed positive (negative) pulses equal to the maximum (minimum) value on the adjacent set.

For each pulse, we calculate its boundary positions using the method described in Section 3.3.2. We then examine the boundary values on the *intermediate reconstruction*, and if any of those values are equal to the pulse value, a merge is required. After examining all boundary positions, a list is drawn up of all coordinates that fall on merge boundaries. At each of those positions, a merge is performed as described in Section 3.3.3, after which the *label image* is updated. The L_{k+1} and U_{k+1} operators are now repeatedly applied, until the image has been entirely decomposed (in other words, until the final MN-sized pulse has been removed). All the removed pulses together from the Discrete Pulse Transform or decomposition.

3.3.5 Algorithm Optimizations

Area Histogram For an $M \times N$ image, the discrete pulse decomposition has pulses with areas ranging from $L = 0, \ldots, MN$. In practice, however, many values of L have no corresponding pulses. When applying the L and U operators, time is saved by skipping these cases entirely. We can track these cases by constructing a histogram, $H[k]$, of the pulse sizes during the initial connected component search. Thereafter, whenever merging two regions, the histogram is updated. Then, when U_k or L_k is executed, we simply verify that $H[k] > 0$ before proceeding.

A Benchmark of Accidental Recombinations The decomposition was implemented and executed on a Intel Core Duo 3.16 GHz processor. Memory utilisation peaked at less that 150MB during decomposition of the 512×512 *Airplane* and roughly 60MB while processing the 300×451 *Chelsea* (including the memory required to store the decomposition itself). Computation times were 3.73s (*Airplane*) and 1.51s (*Chelsea).* Reconstruction executed in a few

Figure 3.5: *Two test images used for benchmarking the 2D DPT. On the left is* Chelsea the Cat *(*300×351*) and on the right is* Airplane *(*512×512*).*

milliseconds.

Figure 3.6 shows benchmark times for the Discrete Pulse Transform applied to random matrices. Random matrices with a large number of discrete values seem to be the worst case scenario—execution times are much lower for real photographs and for signals limited to, say, 255 discrete values. Both these observations are explained intuitively: a random matrix has many more pulses than a typical photograph and limited discrete values cause merging of pulses that would otherwise remain separated. It would be interesting to investigate whether a link exists between image entropy and the number of pulses generated.

3.3.6 Reproducibility and Code

The code for this `Python` implementation is available under the open source BSD licence at `http://dip.sun.ac.za/~stefan/lulu`.

Further collaborative work into a parallel implementation as well as smarter storage methods for the pulses are being looked into to further improve the two dimensional implementation and provide a three dimensional implementation for video analysis.

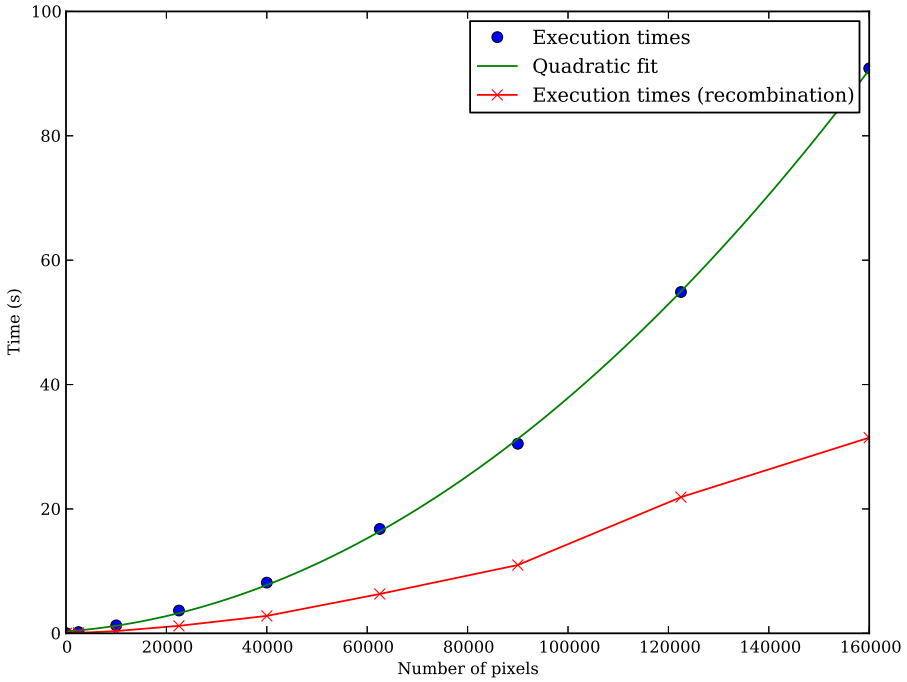


Figure 3.6: *Benchmark of the Discrete Pulse Transform on random images of varying size. Values on the x-axis indicate the total number of pixels, i.e., N^2 for an $N \times N$ matrix. In the bottom curve, labeled "recombination", the number of discrete input values were limited to 255. For large images, this quantisation causes the algorithm to execute more quickly than expected due to an increased number of merges.*

3.4 Properties of the Discrete Pulse Transform

3.4.1 Linear versus Nonlinear

As discussed in [52], the nonlinearity of the LULU smoothers, as mentioned in Chapter 2, make theoretical development more complicated than for linear operators. However, taking on the additional complexity is justified since in two dimensions an image is basically the transformation of data by a human eye or measuring instrument. This transformation is significantly complicated to be considered nonlinear [184]. Thus taking this stance the analysis of images via nonlinear operators is more logical than that of linear.

Linear processing techniques are however a natural starting point for analysis due to the simplicity of their application and theoretical backbone available. Examples of linear filters are the Fourier transform, Hadamard transform, the discrete cosine transform, and wavelets. They also provide sufficient results in most applications, but there are problems in which a nonlinear process would prove more viable and efficient [172]. Pitas and Venetsanopoulos [172] provide examples of such cases, such as signal dependent noise filtering e.g. photoelectron noise of photosensing devices; multiplicative noise appearing as speckle noise in ultrasonic imaging and laser imaging; and nonlinear image degradations e.g. when transmission occurs through nonlinear channels. Advantages of nonlinear filters are 1) the ability to handle various noise types, 2) edge preservation, 3) fine detail preservation, 4) unbiasedness (directional and illumination based) or invariance, and 5) computational complexity [172].

Nonlinear filtering techniques can be broadly classified accordingly in the following areas: order statistic filters, homomorphic filters, polynomial filters, mathematical morphology, neural networks, and nonlinear image restoration [172]. The LULU operators fit nicely into the areas of mathematical morphology as well as order statistics, two areas which have been integrated quite effectively in literature [172]. Examples of order statistics, discussed in detail in [172], are the median, rank-order filters, max-min filters, L_p-mean filters, and α-trimmed mean filters. The LULU operators are examples of max-min filters but with the disadvantages listed in [172] improved upon. The basic filters of mathematical morphology are the erosion and dilation, and subsequently the morphological opening and closing, to which the LULU filters are again closely related as area openings and closings.

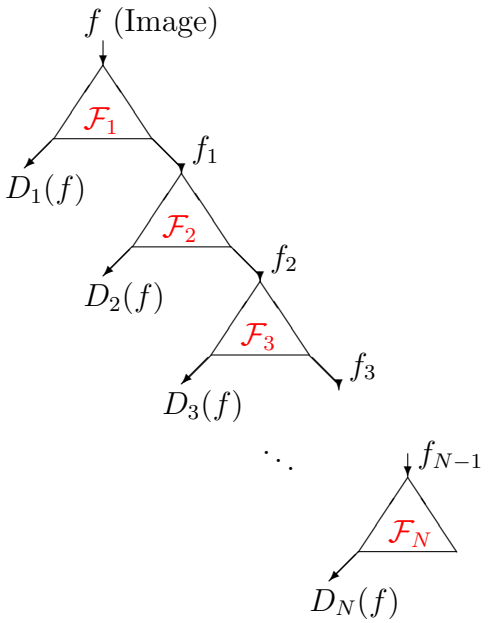


Figure 3.7: *A typical hierarchical decomposition*

3.4.2 Nonlinear Decompositions

Figure 3.7 presents the structure of a hierarchical decomposition. The operator $\mathcal{F}_1$ is applied to the input image f to obtain a decomposition of f into f_1, the smoother image, and D_1, the noise component removed. This process is repeated with $\mathcal{F}_2$, $\mathcal{F}_3$,...,$\mathcal{F}_N$ until there is nothing left to remove except the constant image D_N. The decomposition then has the form

$$f = D_1(f) + D_2(f) + ... + D_N(f). \tag{3.22}$$

Such a hierarchical decomposition has been investigated intensively in literature, see [223, 65, 247] for some nonlinear cases. However, in no literature we have come across have we found a unified theoretical backbone to connect such nonlinear hierarchical decompositions and provide methods of comparison nor methods of testing the capability of the structure of the decomposition. In Tadmor et al [223], for example, a decomposition $f = \sum_{j=1}^{k} u_j + v_k$ is obtained, where v_k is the noise component and the u_j's the decomposition components, by functional minimization. Tadmor et al discuss convergence of the minimizer, localization and adaptability, but nothing to indicate the strength of the decomposition save numerical visual examples. Wong et al [247] similarly do not provide a theoretical indication of the strength of their decomposition obtained as a probabilistic scale-space derived from the non-

linear diffusion equation of Perona and Malik [170]. In [65], Florack et al even state that comparisons with their proposed nonlinear scale-space and other nonlinear hierarchical decompositions 'are to be made with care'. It is thus clear that a unified theoretical setting for nonlinear decomposition escapes us.

A major advantage of the Discrete Pulse Transform is that it comes with its own theory comprising properties like consistent decomposition and total variation preservation, as is discussed in the next sections.

3.4.3 Consistent Decompositions

Definition 22 *A decomposition of the form (3.22) is called* consistent *if for every f in the considered domain and a set of nonnegative number $\alpha_1, \alpha_2, ..., \alpha_N$ we have*

$$D_j\left(\sum_{i=1}^{N} \alpha_i D_i(f)\right) = \alpha_j D_j(f),\ \ j = 1, 2, ..., N.$$

Definition 22 essentially means that the function

$$g = \sum_{i=1}^{N} \alpha_i D_i(f)$$

decomposes into its summands as is illustrated in Figure 3.8. This property is trivially true if D_j is a linear operator. However, it is also a desirable property for nonlinear decompositions. It implies that the components of the decomposition are in some sense 'quantitatively' different from each other, for example, multiplying one component by a positive number will not change it to another. This can be considered as an indication that each component extracts a feature or information which is to some extent unaltered and independent of the remaining part of f.

The property of consistency has a long history within the development of the Discrete Pulse Transform. Recall that the DPT of $f \in \mathcal{A}(\mathbb{Z}^2)$ is given by (3.1)-(3.3). In the one-dimensional case, i.e. $f \in \mathcal{A}(\mathbb{Z})$, the first result is the consistent decomposition of sums of the form

$$g = \sum_{i=m}^{n} D_i(f), 1 \leq m < n \leq N.$$

This property is called *basic decomposition*, see [183, Theorem 8.3]. In the same reference the full consistency is also proved [183, Theorem 8.4].

Note that for the DPT we have

$$D_n(f) = \sum_{s=1}^{\gamma(n)} \psi_{ns}$$

so that

$$f = \sum_{i=1}^{N} \sum_{s=1}^{\gamma(n)} \psi_{ns}.$$

The success with the consistency property made the author believe that a stronger property might hold, namely that the sum

$$g = \sum_{s=1}^{\gamma(n)} \alpha_{ns}\psi_{ns}, \ \alpha_{ns} \geq 0$$

decomposes consistently, that is,

$$D_n(g) = \sum_{i=1}^{\gamma(n)} \alpha_{ns}\psi_{ns}, n = 1, 2, ..., N.$$

This was formulated as the Highlight conjecture in [183, p. 100] and restated later in [113]. The proof in the one-dimensional case was claimed in [114] but not shown. In our work on the multidimensional DPT, we proved the basis consistency of the decomposition, published in [8]. This was followed by a publication of D Laurie [112] where he proved the Highlight conjecture for functions derived on graphs, hence also applicable to $\mathcal{A}(\mathbb{Z}^2)$.

Here we present theorems proving the basic consistency as in [8] and the Highlight conjecture in the setting of $\mathcal{A}(\mathbb{Z}^d)$ since they each use different methods. In particular, the method of proof of the Highlight theorem shows the power of the morphological approach adopted here and applies the techniques in Theorem 17 which was originally presented in [8].

Theorem 23 *Let $f \in \mathcal{A}(\mathbb{Z}^d)$. For any two integers m and n such that $m < n$ the function $g = D_m(f) + D_{m+1}(f) + ... + D_n(f)$ decomposes consistently, that is*

$$D_j(g) = \begin{cases} D_j(f) & \text{for } m \leq j \leq n \\ 0 & \text{otherwise} \end{cases}$$

The proof uses the following lemmas.

Lemma 24 *a) Given* $V, W \in \mathcal{C}$ *with* $W \subset V$. *Then for* $x \notin V$ *but* $x \in \text{adj}(W)$, *we have* $x \in \text{adj}(V)$.
b) Given distinct $V, W \in \mathcal{C}$ *with* $V \cap W \neq \emptyset$, *there exists an* $x \in V \backslash W$ *such that* $x \in \text{adj}(W)$.
c) For any $V \in \mathcal{C}$, *we have* $\text{card}(V) < \infty \implies \text{card}(\text{adj}(V)) < \infty$.

Proof
a) V and $W + \{x\}$ are connected and have a nonempty intersection thus their union $V \cup \{x\}$ is also connected. Then by Definition 40, $x \in \text{adj}(V)$.
b) Note that the following is true:

$$V, W \in \mathcal{C},\ W \subsetneq V \implies \text{adj}(W) \cap V \neq \emptyset. \tag{3.23}$$

Applying (3.23) to $V \cup W$, since $V \subsetneq V \cup W$, we get $\text{adj}(W) \cap V = \text{adj}(W) \cap (V \cup W) \neq \emptyset$. Thus there exists $x \in V \backslash W$ such that $x \in \text{adj}(W)$.
c) This follows from condition (2.3). Let $\text{card}(V) = n$ then for an arbitrary $x \in V$ we have $\{\{a\} \cup V : a \in \text{adj}(V)\} \subset \mathcal{N}_{n+1}(x)$, so that $\text{card}(\text{adj}(V)) \leq \text{card}(\mathcal{N}_{n+1}(x)) < \infty$. ■

Lemma 25 *Let* $Q_n = P_n P_{n-1} ... P_1$ *where* $P_k = L_k U_k$ *or* $P_k = U_k L_k$. *We have*

a) $Q_n Q_m = Q_{\max\{n,m\}}$

b) $Q_m(id - Q_n) = Q_m - Q_n = (id - Q_n)Q_m$ *for all integers* m, n *such that* $m \leq n$.

Proof
We consider only $P_k = L_k U_k$ as the other case is dealt with by symmetry. Let $f \in \mathcal{A}(\mathbb{Z}^d)$.
a) It follows from Corollary 11 in Chapter 2 that $Q_n(f)$ does not have any local minimum or local maximum sets of size n or less. Hence $P_k(Q_n(f)) = Q_n(f)$ for $k = 1, ..., n$. For $m \leq n$ this implies that $Q_m(Q_n(f)) = Q_n(f)$. If $m > n$ then we have
$(Q_m Q_n)(f) = (P_m ... P_{n+1} P_n ... P_1)(Q_n(f)) = (P_m ... P_{n+1})(Q_n(f)) = C_m(f)$.
b) We use induction on j as in the proof of this property in the one dimensional case, see [182]. Let $j = 1$. Using the result in Lemma 16(b), the full

trend preservation property of the LULU operators established in Chapter 2 and the absorbtion property in a) we have

$$\begin{aligned} Q_1(id - Q_n) &= L_1(U_1(id - Q_n L_1 U_1)) = L_1(U_1 - Q_n L_1 U_1) \\ &= L_1(id - Q_n L_1)U_1 = (L_1 - Q_n L_1)U_1 = Q_1 - Q_n = (id - Q_n)Q_1. \end{aligned}$$

Assume now that the statement is true for some $m = j < n$. From the inductive assumption we have

$$\begin{aligned} Q_{j+1}(id - Q_n) &= P_{j+1}Q_j(id - Q_n) = P_{j+1}(Q_j - Q_n) \\ &= P_{j+1}(Q_j - Q_n Q_j) = P_{j+1}(id - Q_n)Q_j. \end{aligned}$$

Using Lemma 16(b), the fully trend preserving property and a) as for $j = 1$ we obtain further

$$\begin{aligned} P_{j+1}(id - Q_n)Q_j &= L_{j+1}U_{j+1}(id - Q_n L_{j+1}U_{j+1})Q_j \\ &= (L_{j+1}U_{j+1} - Q_n L_{j+1}U_{j+1})Q_j = Q_{j+1} - Q_n = (id - Q_n)Q_{j+1}. \end{aligned}$$

■

Proof of Theorem 23

Using Lemma 25, function g can be written in the following equivalent forms

$$\begin{aligned} g &= ((id - P_m)Q_{m-1} + (id - P_{m+1})Q_m + ... + (id - P_n)Q_{n-1})(f) \\ &= (Q_{m-1} - Q_n)(f) = (id - Q_n)Q_{m-1} = Q_{m-1}(id - Q_n). \end{aligned} \tag{3.24}$$

It follows from the fact that since for every $f \in \mathcal{A}(\mathbb{Z}^d)$ the functions $(L_n \circ U_n)(f)$ and $(U_n \circ L_n)(f)$ have neither local maximum sets nor local minimum sets of size n or less, as well as the neighbour trend preserving property of the LULU operators that g does not have any local maximum or local minimum sets of size less than m. Hence $P_k(g) = g$ for $k = 1, ..., m-1$ and therefore $Q_k(g) = g$ for $k = 1, ..., m-1$. Then it follows from (3.24) that $D_j(g) = (id - P_j)(g) = 0$ for $j < m$. Let $m \leq j \leq n$. Then using again Lemma 25 we obtain

$$\begin{aligned} D_j(g) &= (Q_{j-1} - Q_j)(g) = (Q_{j-1}(id - Q_n)Q_{m-1} - Q_j(id - Q_n)Q_{m-1})(f) \\ &= ((id - Q_n)Q_{j-1}Q_{m-1} - (id - Q_n)Q_j Q_{m-1})(f) \\ &= ((id - Q_n)Q_{j-1} - (id - Q_n)Q_j)(f) = (Q_{j-1} - Q_n - Q_j + Q_n)(f) \\ &= (Q_{j-1} - Q_j)(f) = D_j(f). \end{aligned}$$

Finally, for $k \geq n$ we have

$$Q_k(g) = (Q_k(id - Q_n)Q_{m-1})(f) = (Q_k Q_n(id - Q_n)Q_{m-1})(f) = 0,$$

which implies that $D_j(g) = 0$ for $j > n$. ■

Theorem 26 *For a DPT decomposition of f*

$$DPT(f) = (D_1(f), D_2(f), ..., D_N(f)) \text{ where } D_n(f) = \sum_{s=1}^{\gamma(n)} \psi_{ns},\ n = 1, 2, ..., N$$

let $g = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \alpha_{ns}\psi_{ns}$ where the constants α_{ns} are positive. Then the DPT decomposition of g is obtained as

$$DPT(g) = (D_1(g), D_2(g), ..., D_N(g)) \text{ where } D_n(g) = \sum_{s=1}^{\gamma(n)} \alpha_{ns}\psi_{ns},\ n = 1, 2, .., N$$

so that the pulses of g are obtained as $\alpha_{ns}\psi_{ns}$. If $\alpha_{ns} = \alpha_n$ for each n then $DPT(g) = \sum_{n=1}^{N} \alpha_n D_n(f)$, so that $D_n(g) = \alpha_n D_n(f)$.

Proof
We carry out the proof by using mathematical induction. Denote $g_m = \sum_{n=N-m+1}^{N} \sum_{s=1}^{\gamma(n)} \alpha_{ns}\psi_{ns}$. Note that $D_N(f)$ consists of only a single pulse ψ_N. Hence, the statement is trivially true for $g_1 = \alpha_N\psi_N$.

Assume that it holds for some $m < N$ and consider $g_{m+1} = \sum_{n=N-m}^{N} \alpha_{ns}\psi_{ns}$. Considering the properties of the supports of the DPT pulses as stated in (3.19) and (3.20) we deduce that

(i) g_{m+1} does not have local maximum or minimum sets of size less than $N - m$,

(ii) the local maximum and local minimum sets of g_{m+1} of size $N - m$ are exactly $\{\text{supp}\{\psi_{N-m,s}\}, s = 1, 2, ..., \gamma(N - m)\}$,

(iii) for every $s = 1, 2, ..., \gamma(N - m)$ the function g_{m+1} is constant on $\text{supp}\{\psi_{N-m,s}\}$ and the difference with the nearest value on the adjacent set is $\alpha_{N-m,s}$ multiplied by the height of $\psi_{N-m,s}$.

Then by Corollary 11

$$D_i(g_{m+1}) = (id - P_i)P_{i-1}...P_2P_1(g_{m+1}) = 0,\ i = 1, 2, ..., N - m - 1.$$

Further, it follows from Lemma 16 that

$$\begin{aligned} D_{N-m}(g_{m+1}) &= (id - P_{N-m})P_{N-m-1}...P_2P_1(g_{m+1}) \\ &= (id - P_{N-m})g_{m+1} \\ &= \sum_{s=1}^{\gamma(N-m)} \alpha_{N-m,s}\psi_{N-m,s}. \end{aligned}$$

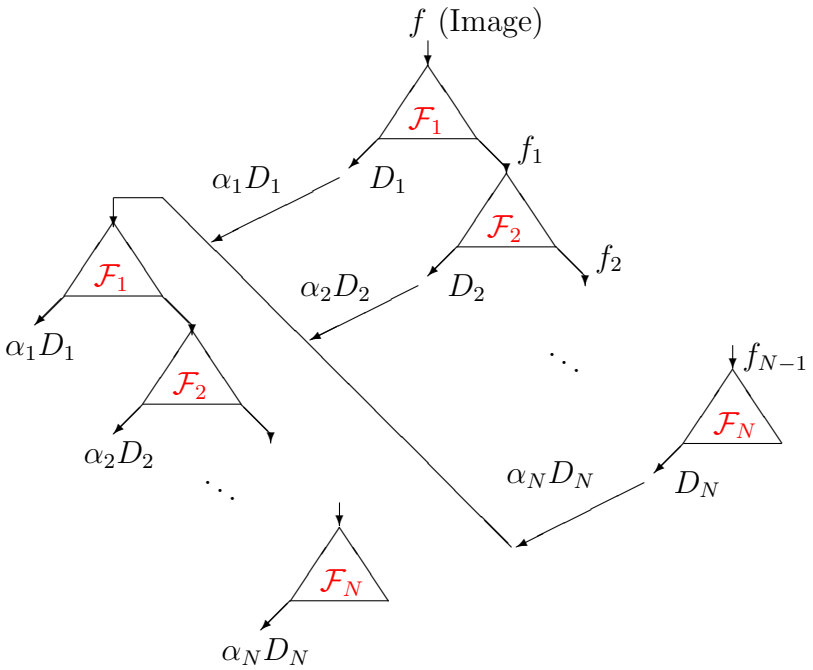


Figure 3.8: *Illustration of consistent decomposition*

By the inductive assumption the function $g_m = g_{m+1} - D_{N-m}(g_{m+1})$ decomposes consistently. Hence g_{m+1} also decomposes consistently. The statement of the theorem is the obtained for $m = N$.

■

3.4.4 Total Variation Preservation

Although the importance of total variation preservation for separators cannot be doubted, it is even more so for hierarchical decompositions like the Discrete Pulse Transform, due to the fact that they involve iterative applications of separators. Since the operators L_n, U_n, $n = 1, 2, ...$, and all their compositions, are total variation preserving, it is easy to obtain the statement of the following theorem, which shows that, irrespective of the length of the vector in (3.1) or the number of terms in the sum (3.21), no additional total variation, or noise, is created via the decomposition.

Theorem 27 *The discrete pulse decomposition in (3.1) is total variation*

preserving, that is

$$TV(f) = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} TV(\psi_{ns}). \tag{3.25}$$

The proof of Theorem 27 can be found in [8, 52]. We should remark that representing a function as a sum of pulses can be done in many different ways. However, in general, such decompositions increase the total variation, that is, we might have strict inequality in (3.25) instead of equality. Based on Theorem 27 we can construct the total variation distribution of images. More precisely, this is the distribution of the total variation of an image among the different layers of the DPT. That is, essentially the plot of $TV(D_n(f))$ vs. n. In Figure 3.9 we present the total variation distributions of some images, where one can observe how the total variation of each image as given in Table 2.6 is distributed over the pulse size. A log scale is used on the vertical axis and the pulse size values are grouped to form a histogram. The different character of the images naturally manifests through different kinds of total variation distributions.

3.4.5 Measuring the Smoothing Ability of the LULU Operators

The ability of an operator to effectively remove noise and smooth the signal is usually measured by its output variance or the rate of success in the noise removal [172]. Other measures used to assess the performance are the mean square error (MSE) and signal-noise-ratio (SNR) [172]. We investigate the noise removal Chapter 5 in detail. In this section we shall present a method in which to measure the quality of smoother or equivalently the resulting smoothed image. In [131] an operator $E_\gamma : X \to X$, for a Hilbert space X, is a smoother in the sense that

$$w - \lim_{\gamma \to 0} E_\gamma f = f.$$

Velleman [234] states that it is required from a smoother to separate the signal into noise and a smooth signal and suggests measuring the success of a smoother by obtaining a regression coefficient near one for the fit of a least squares regression of the smoothed signal compared to the original signal. In [175], a smoothing function $\mathcal{G}_\epsilon$ of f is defined such that for $\epsilon > 0, \mathcal{G}_\epsilon : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable on $\mathbb{R}^n$ and for all $x \in \mathbb{R}^n$, $\|f(z) - \mathcal{G}_\epsilon(z)\| \to 0$ as $\epsilon \to 0, z \to x$.

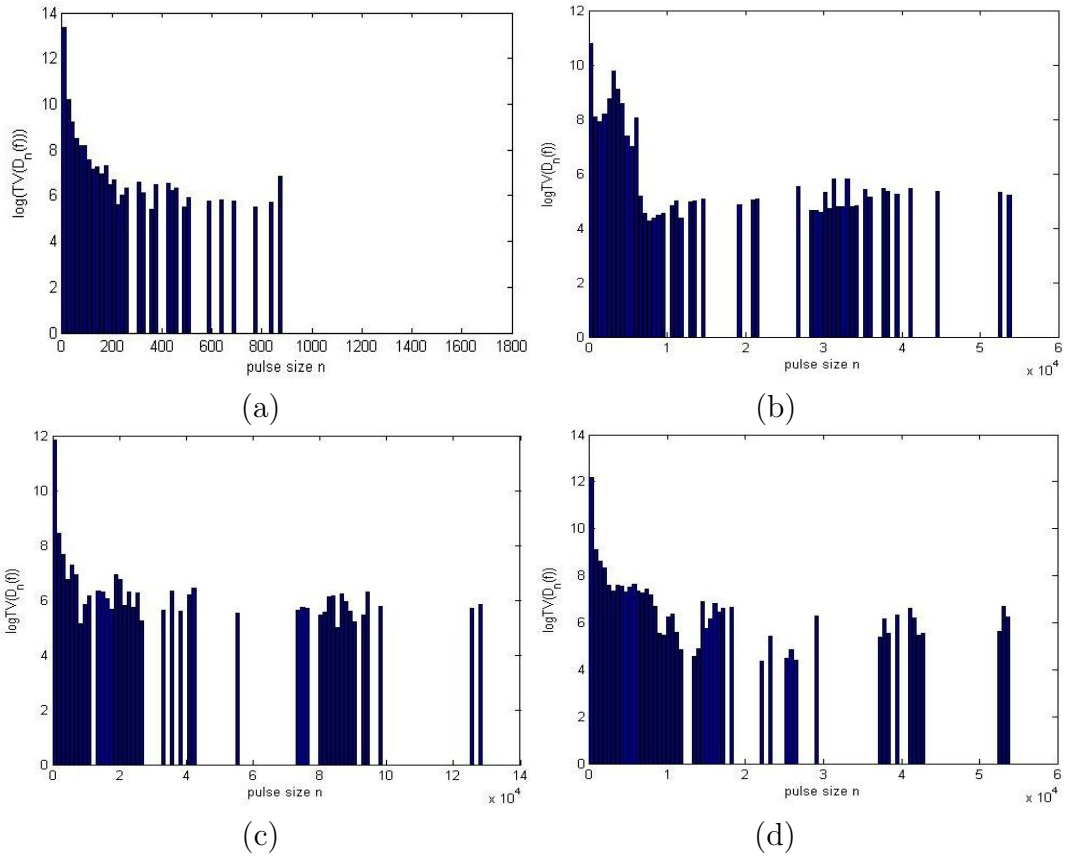


Figure 3.9: *Total Variation Distributions of Images in: (a) Figure 2.4(h)-Noise, (b) Figure 2.4(a)-Potatoes, (c) Figure 2.4(c)-Tank, (d) Figure 2.4(f)-Boat with Glint*

The question of measuring the smoothing ability of the DPT arises. The aim of a smoother primarily to remove the noise element present. The noise can be due to a number of factors, for example, acquisition, processing, compression, storage, transmission and reproduction of the image [235]. The easiest method of evaluation is purely subjective - namely, human visual investigation. In order for evaluation to be objective, quantitative methods need to be used instead. Quantitative methods can be divided into three categories [235]. First, *full-reference*, where the complete reference (undistorted) image is known with certainty, secondly, *no-reference*, where this reference image is not known at all, and third, *reduced-reference*, where only part of the original reference image is known, for example, a set of extracted features. We measure the similarity of the smoothed images $P_n(f)$ to the original unsmoothed image f with the structural similarity index [235],

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

for two corresponding sets of pixels, x and y, in each image, where $\mu_i, i = x, y$ is the mean of the pixel values in i; $\sigma_i^2, i = x, y$ is the variance of the pixel values in i; σ_{xy} the covariance between x and y; $c_j = (k_j L)^2, j = 1, 2$ constants to stabilize the division by the weak denominator; $L = 255$ for greyscale images and $k_j \ll 1$ constants (we used $k_j = 0.05$). This measure is a full-reference measure which provides a useful framework since we are comparing a smoothed version of the original distorted image with the original distorted image. The most widely used such measures are the mean-square-error (MSE) and the peak signal-noise-ratio (PSNR), but these measures do not compare well with the perceived visual quality of the human visual system [235]. Wang et al [235] introduce the SSIM measure in order to penalize errors based on their visibility, that is, to simulate the HVS as much as possible. This measure is applied to 8×8 windows in the image for each pixel and a final mean structural similarity index is calculated as the average of these SSIM values, called the MSSIM. An MSSIM value closer to 1 indicates stronger similarity. Wang et al provide MATLAB code for the implementation of the SSIM at `www.cns.nyu.edu/~lcv/ssim` which was made use of.

Figure 3.10 provides MSSIM values for various images as the LULU smoothing progresses through the DPT from scale $n = 1$ up to $n = N$. Notice how, based on the content of the images, the reduction in the MSSIM values as the DPT progresses varies from image to image. The graphs provide a mechanism to determine where visual structure is in the image, that is, when the HVS would pick out structures of significance. Figure 3.11 plots the differ-

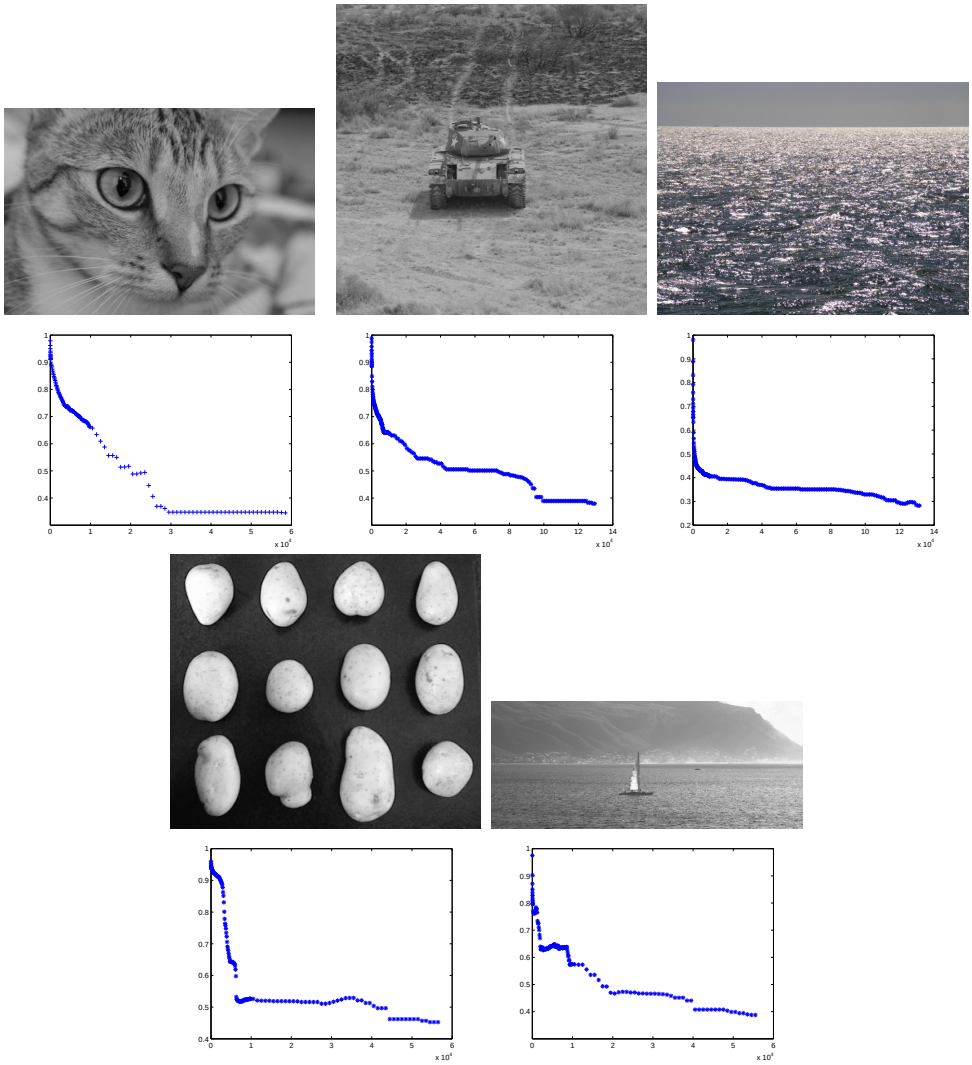


Figure 3.10: *MSSIM values comparing* $P_n(f)$ *with* f *values plotted against scale for* Chelsea, Tank, Potatoes, Ocean *and* Jetski *(values indicated are for increments of 10 up to scale 100, then increments of 100 up to scale 10000, and then increments of 1000 up to the maximum scale)*

ences between the smoothed images as the DPT progresses. The graphs also give an indication of how the information is removed at each step. Figures 3.12 and 3.13 provide an indication of the structure found at big jumps in Figures 3.10 and 3.11.

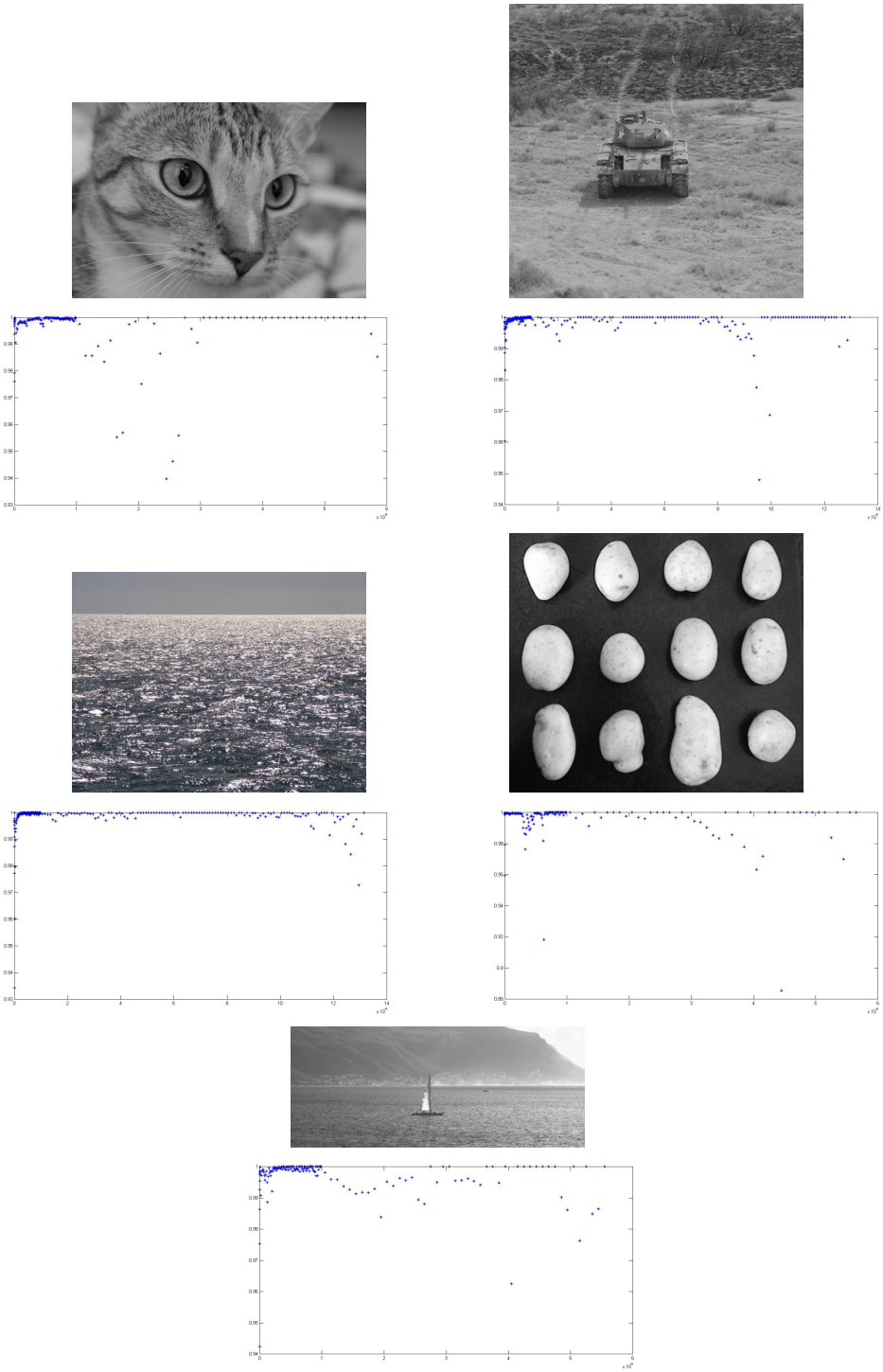


Figure 3.11: *MSSIM values comparing* $P_n(f)$ *with* $P_{n-1}(f)$ *plotted against scale for* Chelsea, Tank, Potatoes, Ocean *and* Jetski *(values indicated are for increments of 10 up to scale 100, then increments of 100 up to scale 10000, and then increments of 1000 up to the maximum scale)*

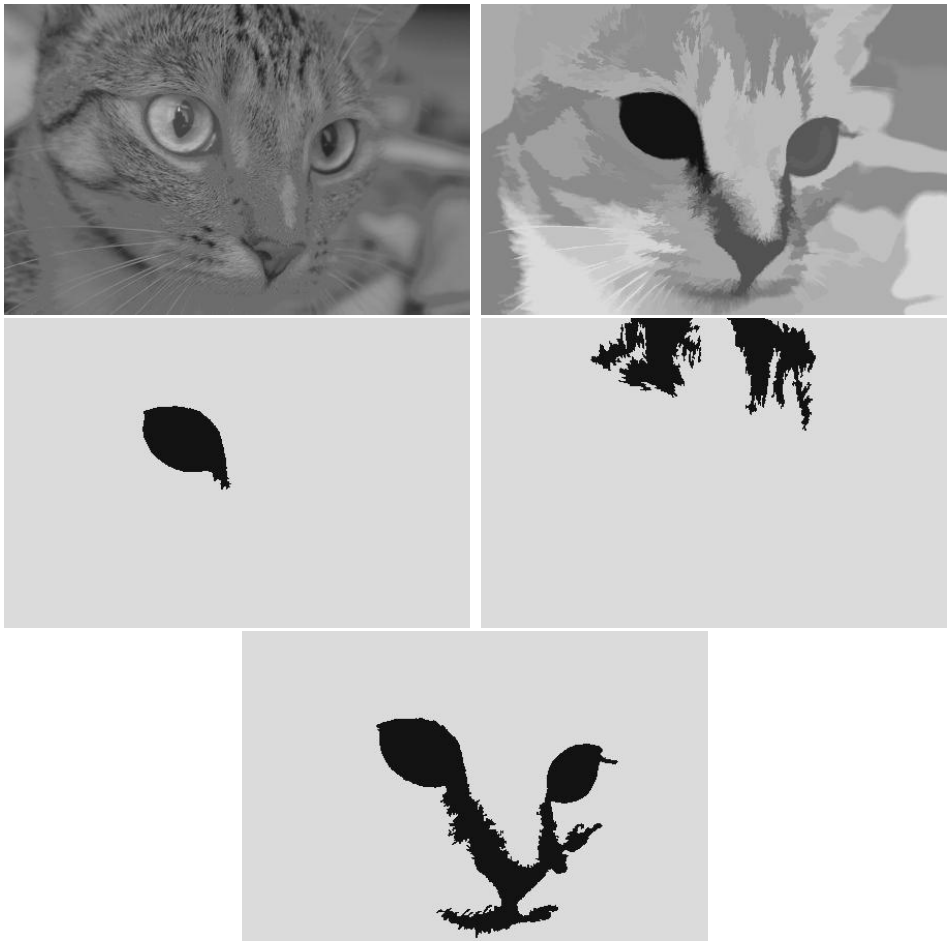


Figure 3.12: *Specific scales of* Chelsea *picked out using Figures 3.10 and 3.11: 1 to 4030 (detail), 4031 to 58571 (big background pulses), 4234 to 4236 (left eye), 4325 to 4335 (forehead), 14565 to 14575 (facial features)*

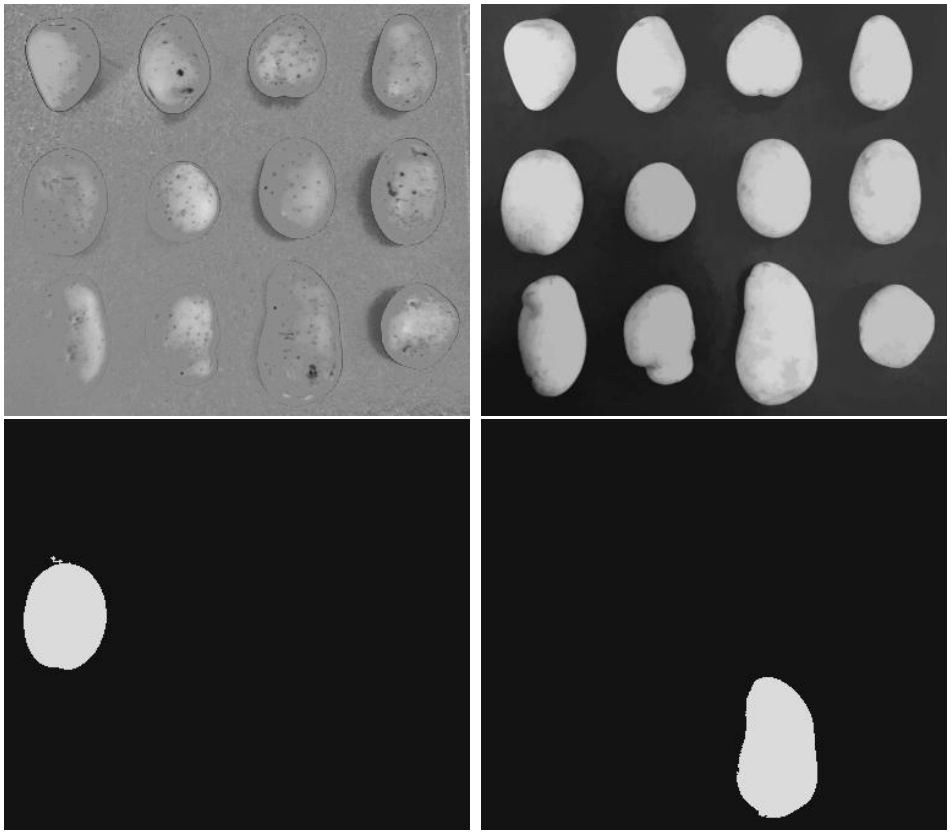


Figure 3.13: *Specific scales of* Potatoes *picked out using Figures 3.10 and 3.11: 1 to 2002 (detail), 2003 to 57478 (big background pulses), 4712 (far left, middle row potato), 5900 (right, last row potato)*

3.4.6 Pulse Shape

Lastly, an important aspect of the DPT is the provision of pulses without restricting their shape in any way. Recall they we only require the pulses to be connected, most commonly with 4-connectivity. This provides a vast area of investigation into the nature of the pulse shapes and what the shapes mean relative to the image structure, image patterns and image content. We look into exploiting this information in Section 4.8.2.

3.5 Conclusion

We have presented the Discrete Pulse Transform resulting from the alternative and recursive application of the LULU operators L_n and U_n, known as the DPT. In particular we looked at the characterization of the DPT with respect to its nonlinearity, consistency, shape preservation properties such as total variation preservation, as well as its ability as a smoother.

Chapter 4

Multi-scale Analysis

4.1 Introduction

At its core, image processing is simply a computer manipulation of a matrix of luminosity values. The manipulation may be complex, for example the reconstruction of a full image using only a small number of pixels from the original image, or quite simple, for example reducing the resolution or size of an image. Due to the development of digital technology and the vast amount of image data readily available there is an increasing need for computer based image processing and analysis. It may be a manipulation to improve a photograph for viewing purposes, such as deblurring, sharpening, colour manipulation or red eye reduction. These are examples of our most common requirements as humans from image editing. The relatively recently developed field of computer vision has seen the addition of other requirements. This field involves developing methods which allow a computer or robot to analyze an image automatically, similar to the way in which the human vision system (HVS), that is the eyes plus the brain or visual cortex, analyzes its surroundings. With the computer representing the brain, the camera the eyes, and the video or image captured the surroundings, the ultimate aim is to eliminate human involvement in the process. For example, consider applications in security such as target detection, identification and tracking, applications in medical imaging to automatically detect anomalies thereby enforcing a doctor's findings, or applications in industry to detect when a production plant is producing reduced quality products.

Obviously the human vision system is highly complicated. The introductions

of [252] and [84] provide insight into what little we do know about the brain and the visual cortex. Research into the brain has provided only glimpses of how subprocesses work and how we could relate the brain to our minds or consciousness. The question of the fruitfulness of computer vision then arises. If we are attempting to model a human action, but we do not understand how it works, how can we hope to replicate this action? It is obvious that the human brain learns as it progresses through life thus offering the solution to a complicated process - we need to provide our techniques with useful extracted information and then teach our algorithms to learn from this data. It is human nature to be inquisitive and research in computer vision can only advance. At some point brain research will have reached heights we cannot now fathom, as with scientific research over the centuries. It only makes sense then to attempt to keep up in the field of computer vision, and to model our computer visual system as closely as possible to the HVS for now, and as advances are made in brain research this can replicated in computer vision. This said, methods which do not attempt to model the HVS should not be discredited at all. The possibility of a smarter method is worth the departure from modeling reality and a computer is obviously not a brain so we cannot hope to treat them in the same manner. The field of super resolution imaging is such an example. It provides the creation of a scene at a higher resolution using a number of lower resolution captures of the same scene.

This chapter deals with changing the representation of an image from two to three dimensions by adding a scale parameter. The aim of this is to provide a more efficient representation in the sense that certain aspects of the information contained in the image is immediately accessible [129]. The original two-dimensional matrix representation of an image is implicit in the sense that the information is contained therein but is not directly accessible. The idea of adding a scale parameter stems from the observation that objects in our surroundings occur at different scales, either due to their size or their resolution with respect to the observer. All around us we view objects of different sizes and this is transferred into an image when a scene is captured on camera. Thus an image is made up of objects of varying sizes, or specifically varying scales. The content of an image can each be present at more than one scale with each scale representing information of varying importance or detail. Consider an image of a face brick house. The wall can be identified as a relatively large flat structuring element consisting of small rectangular elements each porous in texture and varying shading, that is, the presence of at least three different scales can be seen. This illustrates the importance of being able to extract information at various scales. Thus the natural multi-

scale nature of our surroundings encourages a model in which we include the scale as a parameter. To connect the concepts of scale and the DPT pulses, we consider the number of pixels included in a connected region , that is the area of the connected region, [213] as its scale.

The next obvious question is which scale(s) to focus on. A specific object will only be present in a certain range of scales, but this range could differ for another object or even for the same object is a different scene. In addition, as described in [109] there exists 'outer' and 'inner' scale restrictions in any image we analyze, which refer to the maximum extent of the image window or frame, that is, the coarsest detail, and the maximum resolution of the image, that is, the smallest detail which can be observed, respectively. These scale restrictions will differ from image to image as well, making matters more complicated. For example, an object in a lower resolution as well as smaller image will be present in a different range of scales to the same object in an image with higher resolution and larger size, since the base scale, namely 1, is measured at the individual pixel level. The 'inner' scale also poses an additional restriction compared to traditional numerical methods. The maximum resolution is restricted so that we cannot increase it further to improve our results as we do not have data at a higher resolution, as one can do in approximation theory by making the approximation points gradually closer and closer together [123]. There are also numerous operators available which we can apply to images. Thus one may ask, what operator should I use?, where should I use it exactly? and, what size scale should it act on? [129]. This all depends on where the meaningful information in the image. As discussed, this differs from image to image and also depends on the interest of the observer. The nature of the problem at hand is also ill-posed [123]. Recall that according to Hadamard [74, 174] a problem is well-posed if a solution exists, is unique and the solution depends continuously on the data. The projection on a two dimensional image of any three dimensional object or scene, except for simple cases such as a solid smooth sphere, may result in an infinite number of different possible shapes.

The above thus presents a strong case for a representation which treats *all* scales equally at first without any a priori information about the scene in the image. This is in alignment with the universal physical law of scale invariance, the Pi-theorem, namely that physical laws must be independent of the choice of fundamental parameters or in other words a function relating physical observables must be independent of the choice of dimensional units, i.e. no change over scale [64]. Thus we assume nothing for our computer visual front-end and initially consider all possible scales. This also allows

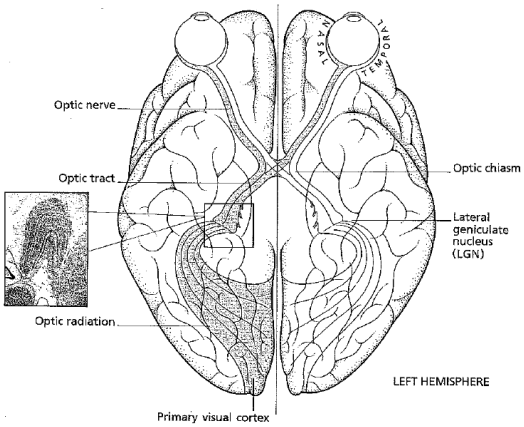


Figure 4.1: *Path of impulses received by the retina to the visual cortex (from [252, figure 3.3, page 23])*

for a large degree of generality in order for the model to be able to solve a large number of problems [123]. Making use of scale-spaces or multiscale methods to analyze an image allows the use of more information than the pixel luminosity only. This allows for providing our algorithms with all the necessary information with which to learn, and not restricting the view to those scales we a priori assume are important. A scale-space is formally (and in its original Gaussian form) the representation of an image f as a continuous family $\{T_t(f) : t \geq 0\}$ of gradually smoother versions of it [238]. The original image is represented at $t = 0$. As described by [168] it is 'an ordered stack of pictures each representing the same scene but at a different level of detail'. This representation provides us with more in-depth information than when the image is represented in its original form so this calls for an effective way to manage this new data and an effective way in which to reduce it to the significant information it is providing us with [246].

How does the human visual system operate? It is interesting to note that the HVS is the best understood part of our complicated brains [84]. As seen in Figure 4.1 the optic nerve carries the impulses received from the retina in the eyes across the optic chiasm to the visual cortex. Note the interesting routes the impulses travel.

There is strong evidence for a 'perceptive' cortex, the striate cortex, and an 'association' cortex, the cortex surrounding the striate cortex [252, Chapter 6]. This is interpreted as the HVS consisting of two stages [105, 156]: a pre-attentive stage in which 'pop-out' [104] features are detected, and then an attentive stage in which relationships between the features are detected and

grouping takes place. The 'pop-out' features are considered salient, that is more discriminating in some way. This idea of two-stage vision also influenced Marr in his book *Vision* [138].

In addition there is strong evidence for a hierarchical process in the HVS [252, Chapter 9]. An image of the surroundings is not projected into our eyes and transmitted as is to the visual cortex. Experiments by Hubel and Wiesel have proven the existence of a process at work which analyzes what we see in a hierarchical or layered manner, that is, takes in the surroundings as a number of building blocks which then make up the entire scene. Interestingly there is also evidence of the HVS working in a parallel manner [252, Chapters 11, 13] with multiple visual areas processing separate things individually.

So our argument for the use of scale-spaces, a representation of an image at all its scales, is strong if we prefer to make no prior assumptions about the scene and allow the algorithm every possible piece of information to work with. We now give a short overview of the scale-spaces researched before the most famous linear scale-space of Witkin was introduced.

The pyramid was the first approach to strictly treating an image in a hierarchical manner, but some pioneers did investigate looking at multiple scales at a time [129], namely, Rosenfeld and Thurston in 1971 who used operators of different sizes for edge detection [191], and [108, 226, 76, 224] who investigated sub-sampling by different amounts.

The basic idea of a pyramid involves the concept of a quadtree. This is obtained via recursive decomposition [202, 129, 123]. More specifically it is the successive subdivision of an image into four equally sized quadrants until the blocks obtained at some subdivision are homogeneous, for example, consist only of 1's and 0's in a binary image. For a greyscale image a measure Σ is defined to measure the homogeneity of the quadrants. It could be for example standard deviation or a thresholding between maximum and minimum pixel luminosity values. Consider an image f of size $2^K \times 2^K, K \in \mathbb{Z}$ and some subdivision $f^{(k)}$. If $H(f^{(k)})$, the measure of homogeneity of $f^{(k)}$, is too large according to some specific value, $f^{(k)}$ is split into $f_j^{(k-1)}, j = 1, 2, ..., p$ according to some rule. Generally p is taken as 4, thus referring to the resulting tree as a quadtree where the leaves $f_j^{(k-1)}$ are homogeneous. This is applied recursively to each subimage $f_j^{(k-1)}$ until the homogeneity of each subimage is satisfied.

This method can in fact be viewed as a simple segmentation algorithm and has been adapted into the 'split-and-merge' algorithm in which adjacent ho-

mogeneous regions (or quadrants) are merged if their measures of homogeneity are similar.

The pyramid is a version of the quadtree but includes a smoothing step at each subdivision as well, see [202, 129, 128, 123], and can be credited to Burt [31] and Crowley [43] individually. The subdivision is in fact a size reduction step so that the image size decreases exponentially with the scale level. The main advantage of a pyramid is that the reduction in image size leads to reduced computational work. For example, consider a low-pass pyramid of Burt and Crowley for a discrete one-dimensional signal f,

$$f^{(k-1)}(x) = \sum_{n=-N}^{N} c(n) f^{(k)}(2x-n)$$

with filter coefficients $c(n), n = -N, ..., N$. Criteria with respect to the coefficients include positivity $c(n) \geq 0$, unimodality $c(|n|) \geq c(|n+1|)$, symmetry $c(-n) = c(n)$, normalization $\sum_{n=-N}^{N} c(n) = 1$, and equal contribution. The *equal contribution* criterion ensures that all pixels contribute equal amounts to all levels by requiring the sum of the weights remains constant over the levels.

In choosing coefficients [143] proposed that an ideal low-pass filter should be approximated as best as possible. A low-pass pyramid involves a smoothing filter first and then a subsampling of the image at each step. Examples include the Gaussian pyramid and Laplacian pyramid, the latter of which is a bandpass pyramid obtained as the difference between two adjacent levels of a low-pass pyramid like the Gaussian pyramid. These have been used in feature detection and image compression.

Wavelets are another early example of incorporating scale into the analysis. The wavelet transform [133, 45, 147] was developed as an improvement over the window Fourier transform.

The continuous wavelet transform [71, 229, 132, 45, 180] decomposes a signal over a set of translated and dilated versions of a 'mother wavelet' $\psi \in \mathcal{L}^2(\mathbb{R})$ which has zero mean $\int_{\mathbb{R}} \psi(t)dt = 0$, is normalized $\|\psi\|_{\mathcal{L}^2} = 1$ and is centered at 0. The fact that ψ has zero mean also implies that the function must be oscillatory and therefore is a wave. For various dilation and translation parameters a and b a set of wavelets

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi \left(\frac{t-b}{a} \right)$$

is obtained. The simplest example of a wavelet is the Haar wavelet, see [45, 52]. The continuous wavelet transform is then a function of the two new parameters a and b,

$$CWTf(a,b) = \langle f, \psi_{a,b} \rangle_{\mathcal{L}^2} = \int_{\mathbb{R}} f(t) \frac{1}{\sqrt{a}} \psi^* \left(\frac{t-b}{a} \right) dt$$

where ψ^* indicates the complex conjugate of ψ, and it decomposes f with respect to wavelet basis set. The function can be fully recovered via the inverse wavelet transform

$$f(t) = \int \int CWTf(a,b)\psi_{a,b}(t) da db.$$

If $\psi(t)$ satisfies the following admissibility criterion,

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty$$

where Ψ is the Fourier transform of ψ, then ψ can be used to analyze and reconstruct the signal without loss of information [229]. Additional regularity conditions are also imposed by [147], namely that the wavelet transform decreases quickly with scale. The family of wavelets is considered redundant [45] thus an orthogonal basis of wavelets is preferred [133, 132, 45]. In higher dimensions the wavelet transform is simply the combination on a product space of a number of separable one dimensional transforms [78].

In order to apply the wavelet transform to digital signals a discrete theory is needed. This is simply attained by discretizing the continuous wavelet transform. The set of wavelets become

$$\psi_{j,n}(m) = \frac{1}{\sqrt{s_0^j}} \psi(s_0^{-j} m - n)$$

where ψ is the original continuous mother wavelet, $k \in \mathbb{Z}$, and s_0^j indicates a dilation of resolution s_0^j ($s_0 = 2$ corresponds to dyadic sampling). The signal is then also discretized by sampling it at points $m = 1, ..., N$. The discrete wavelet transform is then

$$DWTf(n,j) = \sum_m f(m) \psi_{j,n}^*(m).$$

The discrete signal can similarly be fully recovered here,

$$f(m) = \sum_{n,j} DWTf(n,j) \psi_{j,n}(m).$$

The wavelet transform is useful for compression by efficiently and effectively sampling from the parameters a and b [180]. In [180] a discrete time wavelet theory is developed by redefining what is meant by discrete scale and resolution through the sampling rate - they do not simply discretize the continuous theory.

4.2 Background Theory

For simplicity we present some background theory here which is needed later in this chapter.

The Gaussian

The univariate normal distribution [101] is given by

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left((x-\mu)/\sigma\right)^2/2}$$

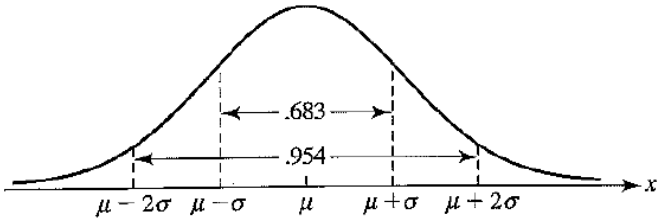
for $x \in \mathbb{R}$. More precisely, if a random variable X has a density function $f(x)$ as given above we say that X is distributed normally with mean μ and variance σ^2 (standard deviation σ), and we write $X \sim N(\mu, \sigma^2)$. The term $(x-\mu)^2/\sigma^2$ in the exponential exponent measures the distance from x to μ in standard deviation units.

The normal distribution in p dimensions [101], for a vector $\underline{X} = [X_1, X_2, ..., X_p]$, has the following form,

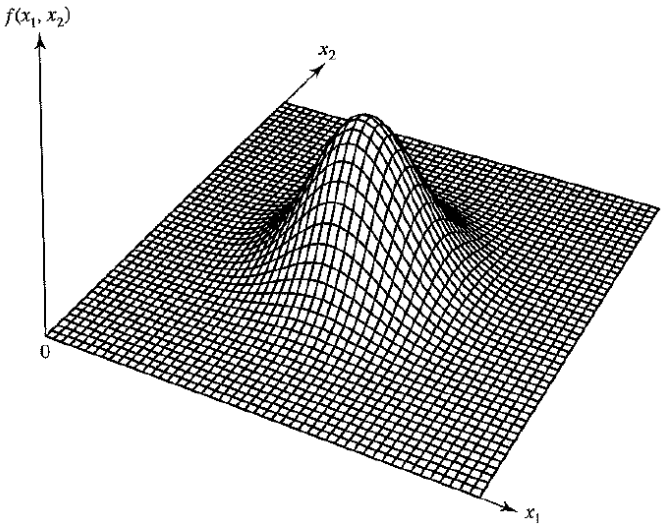
$$f(\underline{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu})} \text{ for } -\infty < x_i < \infty, i = 1, 2, ..., p.$$

The vector random variable is then normally distributed with mean $\underline{\mu} = [\mu_1, \mu_2, ..., \mu_p]$ and covariance matrix Σ where the covariance matrix is required to be positive definite. We write $\underline{X} \sim N(\underline{\mu}, \Sigma)$. The term $(\underline{x} - \underline{\mu})'\Sigma^{-2}(\underline{x} - \underline{\mu})$ in the density function above is called the Mahalanobis distance and measures the square distance from the vector $\underline{x}$ to the mean $\underline{\mu}$ in the units of the covariances.

For the bivariate case $p = 2$ (for application in images) we consider X_1 and X_2 uncorrelated and with equal variances and means so that the correlation



(a)



(b)

Figure 4.2: *(a) One dimensional normal density function [101] with the areas under the curve indicated by the vertical lines (b) Two dimensional normal density function [101] which may be symmetric for the case of equal variances or spherical for the case of unequal variances*

matrix Σ has the form

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}.$$

Of course the uncorrelated case is also a logical choice, as well as the case of unequal variances.

Some properties of the Gaussian distribution are that any linear combination of the components of $\underline{X}$ is normally distributed, any subset of the components of $\underline{X}$ have a normal distribution and the conditional distributions of the components are normally distributed [101]. Figure 4.2 provides an illustration of the one and two dimensional normal densities.

Convolutions

A convolution [18] is an integral expression, involving two functions f and g, for the amount of correlation of g with f as g is shifted and flipped over f. In other words it blends the one function into the other. A beautiful moving illustration of the concept in shown on the webpage [240]. The convolution is defined as follows.

$$G(t) = (f * g)(t) = \int_0^t f(\tau)g(t-\tau)d\tau \text{ (for a finite range)} \quad (4.1)$$

$$= \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = \int_{-\infty}^{\infty} g(\tau)f(t-\tau)d\tau. \quad (4.2)$$

The following properties hold for a convolution of f and g.

- $f * g = g * f$ (commutativity)
- $f * (g * h) = (f * g) * h$ (associativity)
- $f * (g + h) = (f * g) + (f * h)$ (distributivity)
- $a(f * g) = (af) * g = f * (ag)$ for a constant a
- $(f * g)' = f' * g = f * g'$ where $'$ is the derivative
- $F(f * g) = F(f) * F(g)$ where F is the Fourier transform (Convolution Theorem) [166]

Kernels

Schölkopf and Smola [205] provide an excellent work on kernels in computer learning. The first use of the kernel arose as a function in the field of integral operators [80, 39, 144]. A function k giving rise to an operator T_k via $(T_k(f))(x) = \int_{\mathcal{X}} k(x, x')f(x')dx'$ is called the *kernel* of T_k. More specifically a kernel k is a dot product of a feature space $\mathcal{H}$ via a mapping $\Phi : \mathcal{X} \to \mathcal{H}$, that is $k(x, x') = \langle \Phi(x), \Phi(x') \rangle$. The standard requirement on a kernel is positive definiteness. When $\mathcal{H}$ is $\mathbb{R}$ or $\mathbb{C}$ the kernel k is positive definite if the kernel matrix (Gram matrix) $K = [K_{ij}] = [k(x_i, x_j)]$ is positive definite, that is $\sum_{i,j} c_i \bar{c}_j k_{ij} \geq 0 \ \forall c_i \in \mathcal{H}$.

Name	Formulation
Homogeneous Polynomial Kernel	$k(x, x') = \langle x, x' \rangle^d$
Gaussian Kernel	$k(x, x') = \exp^{-\frac{1}{2\sigma^2}\|x-x'\|^2}$
Inhomogeneous Polynomial Kernel	$k(x, x') = (\langle x, x' \rangle + c)^d$ for $d \in \mathbb{N}, c \geq 0$
Radial Basis Function Kernel	$k(x, x') = f(d(x, x'))$ where d is a metric on $\mathcal{X}$, and f a function on $\mathbb{R}_0^+$

Table 4.1: Examples of Positive Definite Kernels

Some positive definite kernels are presented in Table 4.2. Other kernels include the cosine, Hilbert, exponential, B_n spline, rational quadratic, Bartlett, Daniell and Parzen kernels. There also exist kernels which are not symmetric [116]

In addition, if the solution of a partial differential equation, namely f, can be written as $T_k(f)$ above, then the kernel becomes the Green's function. For the heat or diffusion equation, the kernel is the Green's function. The heat kernel in $\mathbb{R}^d$ is as follows,

$$k_t(x, y) = \frac{1}{(4\pi t)^{d/2}} e^{-(x-y)^T(x-y)/4t} \ \forall x \in \mathbb{R}^d \text{ and for any } y \in \mathbb{R}^d.$$

The reader will notice the heat kernel is in fact the Gaussian. The heat kernel represents the evolution of temperature in a region whose boundary is held fixed at a particular temperature and a initial heat source is placed at a point at time 0 [16].

Kernel methods in machine learning include kernel principal components for feature extraction, kernel Fisher discriminant for feature extraction and classification, and Bayesian kernel methods to name but a few.

Modified Bessel Functions of Integer Order

The derivation of the discrete Gaussian scale-space involves in the use of the modified Bessel function of integer order. We present them here for simplicity.

The solutions of the differential equation

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 + n^2)y = 0$$

are called the Bessel functions of order n [10, 237]. These were first defined by Daniel Bernoulli but then generalized by Friedrich Bessel. The two linearly dependent solutions, for an integer n, are given by $J_n(x)$ and $J_{-n}(x)$ where

$$J_n(x) = \left(\frac{1}{2}x\right)^n \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{4}x^2\right)^k}{k!\Gamma(n+k+1)}.$$

The function $Y_n(x)$ is defined as

$$Y_n(x) = \frac{J_n(x)\cos(n\pi) - J_{-n}(x)}{\sin(n\pi)}$$

so that $J_n(x)$ and $Y_n(x)$ are linearly independent, and is called the Bessel function of the second kind (also known as Weber's function and Neumann functions).

The modified Bessel functions of integer order, $I_n(z)$, are obtained when allowing x to be complex but the result real. The solutions of the differential equation are then $I_n(z)$ and $I_{-n}(z)$ when n is not an integer and $I_n(z)$ and $K_n(z) = \frac{1}{2}\pi\frac{I_{-n}(z)-I_n(z)}{\sin(n\pi)}$ when n is an integer. In terms of the original Bessel functions,

$$I_n(z) = i^{-n}J_n(iz) = \sum_{m=0}^{\infty} \frac{\left(\frac{1}{2}x\right)^{n+2m}}{m!\Gamma(n+m+1)} = \frac{1}{\pi}\int_0^{\pi} \exp\{x\cos\tau\}\cos(n\tau)d\tau.$$

For more details, properties and results see [237, 178, 142, 159, 10].

4.3 Scale-Space History

In 1983 Witkin published the first work on the Gaussian scale-space [246, 245]. There was also a technical report from MIT by Stanfield in 1980 [220] which describes a first thought on scale-spaces, as mentioned in [239]. In 1984 Koenderink published an equivalent formulation to Witkin's as the solution of the linear diffusion process [109]. These are considered the foremost work on the linear Gaussian scale-space, which has now grown into a very well

known topic in computer vision. Further pioneering work has been done by Lindeberg, Weickert, Koendrink, ter Haar Romeny, Florack and Viergever, to name the most prominent.

However, contrary to the timeline above, it seems that scale-spaces were independently developed in Japan by Iijima in 1959, [97] - [88]. The work remained undiscovered by the western world, probably because the majority of the works were in Japanese, until 1997 when the connection between the two independent developments was provided in [239, 238]. Weickert also describes in these works that perhaps the research field of scale-spaces wasn't developed enough, or its importance thought of, in 1959 for Iijima's work to be appreciated and thus his work flew under the radar of computer vision scientists. In addition there are three other Japanese linear scale-space approaches that were developed before 1983. All the work presented below by Japanese scientists is from [239] and [238] where it is comprehensively summarized.

The oldest is Taizo Iijima's work done from 1959 [97, 86, 98, 87, 88]. Iijima was working, at the time, at the Electrotechnical Laboratory on optical character recognition and realized the need for a general framework for extraction of characteristic information from patterns. This first work was developed for one dimensional signals from simplicity and relies on four axioms, namely, linearity, translation invariance, scale invariance and a semigroup property. Iijima chose these axioms to remain in line with requirements for object recognition, that is, it should be invariant under changes in the reflected light intensity, parallel shifts in position, and expansions or contractions of the object. He also assumes that the observation results in a blurry transformation Φ and calls this class of blurring transformation 'BOKE' (defocusing). More specifically, with an original image $g(x)$ the blurred version obtained via a convolution with a kernel ϕ has the structure

$$\Phi(g,\sigma)(x) = \int_{-\infty}^{\infty} \phi_\sigma(x,x')g(x')dx' \tag{4.3}$$

where σ is an observation parameter. Four axioms are assumed to be satisfied by the transformation 4.3. They are *linearity w.r.t scalar multiplication* (if the image intensity becomes a times more, then the transformed intensity is similarly a times more), *translation invariance*, *scale invariance* and a generalized *semigroup property* (if g is observed at scale σ_1, and this is in turn observed at scale σ_2, then the equivalent observation scale is $\sigma_3(\sigma_1,\sigma_2)$ for some σ_3) [97, 86, 98, 87, 88, 239]. Note that for uniqueness of the scale-space Iijima claims preservation of positivity is needed as an additional axiom,

namely

$$\Phi\big(g, x, \sigma\big) > 0 \; \forall \; g(x) > 0, \; \forall \; \sigma > 0.$$

However, in his 2002 PhD thesis Felsberg [58] shows that the Poisson kernel also satisfies Iijima's 5 axioms of linearity, scale and shift invariance, a semi-group property and positivity preservation thus disputing the uniqueness under these specific axioms.

Iijima derives the following from (4.3)

$$\Phi\big(g, x, \sigma\big) = \frac{1}{2\sqrt{\pi}\sigma} \int_{-\infty}^{\infty} g(x') \exp\left\{-\frac{(x-x')^2}{4\sigma^2}\right\} dx',$$

which is a convolution between g and a Gaussian with standard deviation $\sigma\sqrt{2}$. Iijima also argues for Gaussian blurring as our visual perception is carried out through a lens which has a Gaussian-like blurring profile [88].

Iijima next generalized this derivation to two dimensions [98, 87]. His blurring transformation is as follows,

$$\Phi\big(f, x, \Sigma\big) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi(f(x'), x, x', \Sigma) dx'_1 dx'_2,$$

where Σ is a 2×2 symmetric positive definite matrix, $x' = (x'_1, x'_2), x = (x_1, x_2)$, and the four axioms are, similar to the one-dimensional case, linearity w.r.t multiplications, translation invariance, scale invariance and closedness under affine transformations, and a generalized semigroup property [98, 87].

If, in addition, positivity preservation is assumed then the blurring is called the affine Gaussian scale-space:

$$\begin{aligned}
\Phi\big(f, x, \Sigma\big) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x'_1, x'_2) \phi(x_1 - x'_1, x_2 - x'_2, \Sigma) dx'_1 dx'_2 \\
\text{with } \phi(u_1, u_2, \Sigma) &= \frac{1}{4\pi\sigma^2} \exp\left(- \frac{\mu_{22} u_1^2 - 2\mu_{12} u_1 u_2 + \mu_{11} u_2^2}{4\sigma^2} \right) \\
\text{and } \Sigma &= \sigma^2 \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{12} & \mu_{22} \end{bmatrix}, \det \begin{pmatrix} \mu_{11} & \mu_{12} \\ \mu_{12} & \mu_{22} \end{pmatrix} = 1.
\end{aligned}$$

To obtain the modern (Witkin and later) isotropic Gaussian scale-space kernel an axiom of invariance under rotations is further needed. This provides

a scale-space which is invariant with respect to direction and results in the third axiom of scale invariance and closedness under affine transformations being equivalent to ordinary pure scale invariance.

Iijima then further re-derived his scale-space in 1971 in order to obtain a more physically consistent formulation [239]. This work of his appears in [89, 91, 92, 90, 93, 94, 95, 96]. His idea is to generalize the original figure (signal or image) $f(r)$ to $f(r,\tau)$ such that the method attempts to model the defocusing of the HVS or an optical system. He assumes two principles, namely the *conservation principle* and the *principle of maximum loss of figure compression*. The conservation principle requires the transformation not to change the total energy of the image function so that the image function satisfies the continuity equation

$$\frac{\partial f(r,\tau)}{\partial \tau} + \nabla \cdot I(r,\tau) = 0$$

where I is the flux (flow per unit) for the figure flow, r is the location, τ the blurring parameter, and ∇ indicates divergence operation in $\mathbb{R}^2$. The continuity equation states that the rate at which the image function energy decreases is proportional to the outward flux. His second principle involves maximizing the figure flow, that is, maximizing

$$J(I) = \frac{|I^T \nabla f|^2}{I^T R^{-1} I}$$

where $R(\tau)$ is a positive definite matrix denoting the medium constant of the blurring process. This is maximized for $I(r,\tau) = -R(\tau) \cdot \nabla f(r,\tau)$. These two principles result in the anisotropic linear diffusion equation

$$\frac{\partial f(r,\tau)}{\partial \tau} = \nabla \cdot (R(\tau) \cdot \nabla f(r,\tau))$$

which Iijima calls the *basic equation of figure*. This is simply the formulation of the affine linear Gaussian scale-space as a partial differential equation.

In 1981 another Japanese scientist, Nobuki Otsu, wrote his thesis entitled 'Mathematical Studies on Feature Extraction in Pattern Recognition' [161]. He modified Iijima's five one dimensional axioms to derive a two dimensional Gaussian scale-space. He derives a transformation $\tilde{f}$ of an image f such that the axioms in Table 3.2 hold.

Axiom 2 in Table 3.2 implies that the integral kernel is symmetric ($W(r, r' + a) = W(r - a, r')$) and thus it is a convolution kernel, namely,

$$W(r, r') = W(r - r'). \tag{4.4}$$

	Axiom	Formulation
1.	**Linear Integral Operator**	$\exists W : \mathbb{R}^2 \times \mathbb{R}^2 \mapsto \mathbb{R}^2$ such that $\tilde{f}(r) = \int_{\mathbb{R}^2} W(r, r')f(r')dr' \; \forall \; r \in \mathbb{R}^2$
2.	**Translation Invariance**	$\forall \; r \in \mathbb{R}^2, a \in \mathbb{R}^2$, $\tilde{f}(r-a) = \int_{\mathbb{R}^2} W(r, r')f(r'-a)dr'$
3.	**Rotation Invariance of the Kernel**	For any rotation matrix T_Θ, and $\forall \; r = (x, y)^T \in \mathbb{R}^2$, $W(T_\Theta r) = W(r) = W(x^2 + y^2)$
4.	**Separability**	$\exists u : \mathbb{R} \mapsto \mathbb{R}$ such that $W(r) = u(x)u(y)$
5.	**Normalization of Energy**	preservation of nonnegativity: $\tilde{f}(r) \geq 0 \; \forall \; f(r) \geq 0$ average grey level invariance: $\int_{\mathbb{R}^2} \tilde{f}(r)dr = \int_{\mathbb{R}^2} f(r)dr$

Table 4.2: Otsu's Two Dimensional Axioms [161]

From Axioms 3 and 4 in Table 3.2 $W(r) = k \exp\{c(x^2 + y^2)\}$ for some parameters $k, c \in \mathbb{R}$ can be easily derived. Axiom 5 in Table 3.2 implies that $W(r) \geq 0$ and $\int_{\mathbb{R}^2} W(r)dr = 1$ respectively. Using these results and the five axioms he shows that $k = \frac{1}{2\pi\sigma^2}$ and $c = -\frac{1}{2\sigma^2}$ and the Gaussian kernel is obtained,

$$W(r) = \frac{1}{2\pi\sigma^2} \exp\left\{ -\frac{x^2 + y^2}{2\sigma^2} \right\}.$$

Otsu also derives an N-dimensional Gaussian scale-space in his thesis [161]. Taking $\rho = \sigma^2/2$, he starts by defining

$$T(\rho)f(r) = \frac{1}{(4\pi\rho)^{N/2}} \exp\left\{ -\frac{|r|^2}{4\rho} \right\},$$

and then via Fourier techniques obtains

$$\frac{\partial \tilde{f}(r, \rho)}{\partial \rho} = \Delta\big(\exp(\rho\Delta)f(r) \big) = \Delta\tilde{f}(r, \rho)$$

so that $\tilde{f}$ satisfies the isotropic linear diffusion equation.

All these Japanese works only became known from 1997. Before this Witkin's 1983 work was believed to be the starting point of the Gaussian scale-space. We go into this is more detail in this next section.

4.4 The Gaussian Scale-Space

The Gaussian convolution was first represented as a scale-space by Witkin in 1983 [246]. In 1980, though, Stansfield also discusses a Gaussian scale-space idea but without the necessary axioms and mathematical backbone [220]. He applies the idea to designing a commodity expert. In addition, in 1980 Marr published his well-known work 'Theory of Edge Detection' [139], in which he describes using a Gaussian convolution as a smoother and tracks zero-crossings to aid edge detection. Zero-crossings are those points where significant intensity changes are detected. Additional work was done by Crowley in his PhD thesis [44]. He developed the DOLP transform, a class of reversible transforms, and uses the cascading property of the Gaussian (discussed later on) to speed up his algorithm from $O(N^2)$ to $O(NlogN)$. He makes use of a discretized Gaussian though by sampling the domain and constructs a tree-like representation for an image using his transform. Additional attempts have also been published [109, 250, 32, 77, 191, 136, 140].

Wikin's 1983 formulation is as follows. His first formulation is for one-dimensional signals in order to initially develop his ideas. By assuming linearity the integral operator to be used must then involve a family of kernels $\{k_t : t \geq 0\}$ such that $T_t(f)(x) = \int_{\mathbb{R}} k_t(x, x')f(x')dx'$ [239]. By additionally assuming translation invariance, so that $\tau_a T_t = T_t \tau_a \ \forall (a \in \mathbb{R}, t > 0)$ and for a shift operator τ_a, the kernel must be a convolution kernel (Equation 4.4) [239]. The Gaussian convolution is thus argued for based on its 'well-behavedness', namely that it is symmetrical about its mean and decreases away from the mean providing less weight to pixel values further away from the focus pixel. The additional assumption is that zero-crossings of the Gaussian and its derivatives may appear but not disappear as scale decreases. This assumption ensures that the Gaussian is the only convolution kernel which provides the 'well-behavedness' required. Gaussian smoothing is obtained for a continuous signal $f : \mathbb{R} \mapsto \mathbb{R}$ as follows:

- The Gaussian smoothed version of f at scale $t \in \mathbb{R}_+ \backslash \{0\}$ is obtained as the convolution

$$L_f(t)(x) = \int_{-\infty}^{\infty} g_t(\xi) f(x - \xi) d\xi = (g_t * f)(x) \ \forall \ x \in \mathbb{R}, t > 0$$

  where $g : \mathbb{R} \times \mathbb{R}_+ \backslash \{0\} \mapsto \mathbb{R}$ is the one-dimensional Gaussian kernel

$$g_t(x) = \frac{1}{\sqrt{2\pi t}} e^{-x^2/2t}.$$

- The original signal f is defined as the representation at scale 0:

$$L_f(0)(x) = f(x) \; \forall \; x \in \mathbb{R}. \tag{4.5}$$

For every $t > 0$, $L_f(t)$, is called the *scale-space image* of f at scale t. Successive smoothing gradually suppresses fine detail making the signal smoother and more blurred each time.

For an N-dimensional signal $f : \mathbb{R}^N \mapsto \mathbb{R}$ the scale-space representation is similarly obtained using the N-dimensional Gaussian kernel.

- The Gaussian smoothed version of f at scale $t \in \mathbb{R}_+ \backslash \{0\}$ is obtained as the convolution

$$L_f(t)(x) = \int_{\xi \in \mathbb{R}^N} g_t(\xi) f(x - \xi) d\xi = (g_t * f)(x)$$

$\forall \; x = (x_1, x_2, ..., x_N) \in \mathbb{R}^N$ where $g : \mathbb{R}^N \times \mathbb{R}_+ \backslash \{0\} \mapsto \mathbb{R}$ is the N-dimensional Gaussian kernel

$$g_t(x) = \frac{1}{(2\pi t)^{N/2}} e^{-\frac{1}{2t} x^T x}.$$

- The original signal f is defined as the representation at scale 0:

$$L_f(0)(x) = f(x) \; \forall \; x \in \mathbb{R}^N. \tag{4.6}$$

We see that Gaussian smoothing a simply a diffusion process by which the high frequencies are removed. This can be seen easily by applying the convolution theorem [166] as the Fourier transform of the Gaussian remains the Gaussian. The two-dimensional Gaussian scale-space can be derived as the solution of the diffusion equation

$$\frac{\partial L_f(t)(x)}{\partial t} = \frac{1}{2} \frac{\partial^2 L_f(t)(x)}{\partial x^2}$$

with initial condition $L_f(0)(\cdot) = f(\cdot)$, see [109]. As is well-known, this parabolic partial differential equation models the evolution over scale [67, 242, 221]. In its original form (see [30]) it models the flow of heat along a rod length ℓ, say, at time t with initial state $f(x)$ such that along each cross-section the temperature is uniform. The constant on the right hand side is

determined by the heat-conductive properties of the rod material. In higher dimensions the partial differential equation has the form

$$\frac{\partial L_f}{\partial t} = \kappa \bigtriangledown^2 L_f,$$

where κ is the thermal diffusivity as in one dimension and $\bigtriangledown^2$ is the Laplace operator. Koenderink's scale-space derivation, as described in [123], is done for two dimensions but can be reduced to one dimension as well.

The term *scale-space* is reserved for multi-scale representations for which similar theoretical properties can be proven, the most important being that of non-creation of 'new' or 'artificial' structures [123]. We provide a formal definition in Section 4.6. Note that there is a subtle difference between the terms multi-scale and multiresolution, however the terms are used freely and no exact difference is clear.

4.4.1 Gaussian Scale-Space Properties

How the Gaussian kernel smooths a signal

The scale parameter t is the standard deviation in the Gaussian kernel. Thus it acts by averaging the signal symmetrically in every direction with increasing window size as t increases. Structures with support smaller than t will then be suppressed [123].

The smoothness obtained is measured in different ways by different authors. For example, in [250] regularity appears as the convergence of the convolution kernels to the Dirac delta distribution and in [64] as the Fourier transform becoming 1 everywhere. In [12, 61] infinitely differentiable convolution kernels are assumed which are rapidly decreasing functions of x. In [120] the kernels are assumed to be Borel measurable and in [125] the kernels are assumed to converge for $t \to 0^+$ in the L^1 norm to the Dirac distribution. In [4] it is required that for smooth f and g

$$\|L_{f+hg}(t) - (L_f(t) + hg)\|_\infty \leq Cht, \ \forall h, t \in [0, 1]$$

where C may depend on f and g and $L_f(t)$ is the Gaussian convolution of f. In [168] the kernels are assumed to be separately continuous in x and t. More in line with signal and image processing, the LULU smoothers for sequences [183] create smoother versions of their input which are n-monotone if every window of length n is monotone non-increasing or non-decreasing. A similar

definition applies for the LULU smoothers for images, see Section 5.3. Other filters in signal and image processing have similar results.

Of all the possible probability density functions the Gaussian is the one with maximum entropy [12]. Entropy (known as Shannon entropy) is a measure of uncertainty associated with a random variable. With the Gaussian having maximum entropy we are thus making use of a kernel which applies the least amount of prior assumptions and structure onto the signal, as is desired, thereby further enabling smoothing with the Gaussian kernel.

Semigroup and Cascading Property

Since the Gaussian kernel exhibits the semigroup property $g(\cdot,t) * g(\cdot,s) = g(\cdot,t+s)$, a representation at a coarser scale t_2 can be computed from a representation at a finer scale t_1 by an additional convolution with parameter $t_2 - t_1 > 0$ i.e. $L_f(t_2)(\cdot) = (g_{t_2-t_1} * L_f(t_1))(\cdot)$, so that a cascade smoothing property is implied [123].

Separability

The N-dimensional Gaussian kernel $g : \mathbb{R}^N \mapsto \mathbb{R}$ can be written as the product of N one-dimensional Gaussian kernels $g_1 : \mathbb{R} \mapsto \mathbb{R}$,

$$g(x,t) = \prod_{i=1}^{N} g_1(x_i,t), x = (x_1, x_2, ..., x_N) \in \mathbb{R}^N$$

since

$$\frac{1}{(2\pi t)^{N/2}} e^{-\frac{1}{2t}x'x} = \prod_{i=1}^{N} \frac{1}{(2\pi t)^{1/2}} e^{-\frac{x_i^2}{2t}}.$$

This is a useful property, especially for decreasing computational complexity [123].

The Maximum Principle

This property is exactly the strong maximum principle of parabolic equations [158] which states that if a function attains its maximum on the interior of its domain the function is constant. In terms of the scale-space then if $x_0 \in \mathbb{R}$ is a local maximum of $x \mapsto L_f(t_0)(x)$ at a certain scale $t_0 \in \mathbb{R}_+$, then the Laplacian is negative $\bigtriangledown^2 L_f(t_0)(x_0) < 0$ i.e. $\partial_t L_f(t_0)(x_0) < 0$, and if this x_0 is a local minimum then $\bigtriangledown^2 L_f(t_0)(x_0) > 0$ i.e. $\partial_t L_f(t_0)(x_0) > 0$. This means that small local variations are suppressed so that a 'hot spot' will not become warmer and a 'cold spot' not cooler [12, 85, 125].

Scaling Property

If $f(x) = f'(sx)$, let $x' = sx$ and $t' = s^2t$. Then $L'(\cdot, t') = g(\cdot, t') * f'$ and it can be shown that the two representations are the same $L(x,t) = L'(x',t')$ i.e. stretching the parent kernel such that the areas remain the same [168] (see [123] for a proof). Also see [125] where it is shown that this scale invariance follows from the semi-group property when combined with isometry invariance (symmetry) and causality.

Scale-Space Derivatives and Infinite Differentiability

We recall the notation for multi-scale derivatives. Let $n = (n_1, n_2, ..., n_N) \in \mathbb{Z}_+^N, n_i \in \mathbb{Z}_+$, $x = (x_1, x_2, ..., x_N) \in \mathbb{R}^N$ and $x^n = x_1^{n_1} x_2^{n_2} ... x_N^{n_N}$. The

$$\partial_{x^n} = \partial_{x_1^{n_1}} \partial_{x_2^{n_2}} ... \partial_{x_N^{n_N}}$$

is the derivative of order $|n| = n_1 + n_2 + ... + n_N$. The multi-scale derivatives are the scale-space derivatives of f at scale t and are given by

$$\begin{aligned} L_f^{x^n}(t)(x) &= \partial_{x^n} L_f(t)(x) = \left(g_t^{x^n} * f\right)(x) \\ &\quad \text{where } g_t^{x^n} \text{ is the partial derivative of the Gaussian} \\ &\quad \text{kernel of order } |n| \\ &= \int_{x' \in \mathbb{R}^N} g_t^{x^n}(x - x') f(x') dx' \\ &= \int_{x' \in \mathbb{R}^N} g_t^{x^n}(x') f(x - x') dx'. \end{aligned}$$

The scale-space derivatives are guaranteed to converge for any $t > 0$ if f is bounded above by some polynomial. Since the Gaussian function decreases exponentially if there exists $c_1, c_2 \in \mathbb{R}_+$ such that $|f(x)| \leq c_1(1 + x'x)^{c_2}$ then even if f is not differentiable convergence is guaranteed. The convolution provides a strong regularizing property and for every $t > 0$ the scale-space derivatives can be treated as infinitely differentiable [123].

In addition the scale-space properties mentioned thus far transfer to the scale-space derivatives as well. Namely, they satisfy the diffusion equation, also get successively smoother, ensure non-enhancement of extrema and possess the cascading smoothing property. They satisfy a scaling property but one which is slightly different,

$$g_{x^n}(x,t) = s^{N+|n|} g_{x^n}(sx, s^2t).$$

We then have that $L_f^{x^n}(t)(x) = \partial_{x^n} L'_f(t')(x') = s^{|n|} L_f'^{x'^n}(t')(x')$. The coordinates can however by normalized to $\xi = x/\sqrt{t}$ and $\xi' = x'/\sqrt{t'}$ to make them dimensionless and then $L_f^{\xi^n}(t)(x) = L_f'^{\xi'^n}(t')(x')$.

Other Properties [123]

For a function h with Fourier transform $\hat{h}$ the normalized second moments Δx and $\Delta\omega$ in the spatial and Fourier domain, which describe the spread of the distribution of these two functions, are

$$\Delta x = \frac{\int_{x\in\mathbb{R}} x'x|h(x)|^2 dx}{\int_{x\in\mathbb{R}} |h(x)|^2} dx \text{ and } \Delta\omega = \frac{\int_{\omega\in\mathbb{R}} \omega'\omega|\hat{h}(\omega)|^2 d\omega}{\int_{\omega\in\mathbb{R}} |\hat{h}(\omega)|^2} d\omega.$$

The uncertainty relation states that $\Delta x \Delta\omega \geq \frac{1}{2}$ and the Gaussian kernel is the only real kernel that gives equality here. The Gaussian kernel is also the only rotationally symmetrical kernel that is separable in Cartesian coordinates.

4.4.2 Gaussian Scale-Space Axioms for Uniqueness

A number of authors have, since Witkin and Koenderink's work, made similar derivations of the Gaussian scale-space and its uniqueness based on various sets of axioms. The main idea throughout all the research done is that the smoothing mechanism does not allow creation of spurious structures. This idea has been formulated in various works. We discuss them now. A summary table is presented in Table 4.4.2 (replicated from [239]). Note, that the uniqueness referred refers to the Gaussian kernel in the convolution formula, not the unique existence of only the Gaussian scale-space.

Witkin 1983 [246]

Witkin introduced the theory for one-dimensional signals and observed that new local extrema were not created. This property extends to the scale-space derivatives and he thus tracked the zero-crossings across scale forming a tree data structure for the signal. He mentions a link between the length of the branches of the tree and the perceptual saliency of the viewer.

Axiom	I1	I2	I3	O	K	Y	B	L1	F1	A	P	N	L2	F2
Convolution Kernel	•	•		•		•	•	•	•	•	•		•	•
Semigroup Property	•	•						•	•	•	•	•	•	•
Locality										•				
Regularity						•	•	•	•	•	•		•	•
Infinitesimal Generator											•			
Maximum Loss Principle			•											
Causality					•	•	•	•					•	
Nonnegativity	•	•		•						•	•			•
Tikhonov Regularization												•		
Average Grey Level Invariance			•	•			•	•		•	•			
Flat Kernel for $t \to \infty$						•			•					
Isometry Invariance (symmetry)		•		•		•		•	•	•	•	•	•	•
Homogeneity and Isotropy					•									
Separability				•					•					
Scale Invariance	•	•				•	•		•		•	•		•
Valid for which dimensions?	1	2	2	2	1,2	1,2	1	1	> 1	N	1,2	N	N	N

Table 4.3: Comparison of the Gaussian Scale-Space Axioms [239]. (Key: I1 = [97][98][87][88], I2 = [98][87]), I3 = [89], O = [161], K = [109], Y = [250], B = [12], L1 = [120], F1 = [64], A = [4], P = [168], N = [157], L2 = [125], F2 = [61]

Koenderink 1984 [109]

Koenderink shows that the family generated by using the Gaussian convolution is unique when assuming axioms of causality, homogeneity and isotropy. These are, more specifically, that 'spurious events' may not be generated so that every feature at a coarse scale level must have a 'cause' at a finer scale level (every isophote - constant luminosity levels - in scale-space must be upwards convex), and that smoothing is both scale and spatially invariant. Using these he shows that the scale-space representation must satisfy the diffusion equation and since the Gaussian kernel is the Green's function of the diffusion equation the uniqueness of the solution follows. Green's function is a function used to show existence and uniqueness of the solution of inhomogeneous differential equations [10]. Since the scale-space derivatives also satisfy the diffusion equation the property of no new zero-crossings with increasing scale still holds.

Yuille and Poggio 1983 [250]

Yuille and Poggio impose their assumptions on the filter F used as boundary conditions in two dimensions. Their assumptions are as follows.

1. **The filter is shift-invariant:** The filter is therefore a convolution $F * f = \int F(\underline{x} - \underline{\xi}) f(\underline{\xi}) d\underline{\xi}$.
2. **The filter has no preferred length**
3. **The filter covers the entire image at sufficiently small scales:** $\lim_{t \to 0} F(\underline{x}, t) = \delta(\underline{x})$ where $\delta(\underline{x})$ is the Dirac delta function.
4. **The position of the center of the filter is independent of** t
5. **A Flat kernel as** $t \to \infty$**:** As $|\underline{x}| \to \infty$ and $t \to \infty$ we have that the filter goes to 0 and so $\lim_{t \to \infty} k_t(x) = 0$.

Note that symmetry is not one of their requirements. With these assumptions they are able to prove that in one and two dimensions the Gaussian filter is the only filter which doesn't create zero-crossings as scale increases, and in two dimensions, when using the directional operator along the gradient, there is no filter which obeys their assumptions and does not create zero-crossings as scale increases.

A related uniqueness formulation is also presented in [85].

Witkin et al 1986 [12]

Witkin et al prove the uniqueness of the Gaussian kernel in one dimension under a number of conditions, the main one being a monotonicity condition such that zero-crossings appear from coarse to fine scale but existing ones never disappear. This means that the local maxima (and minima) of the surface swept out by f always increase (and decrease) as scale increases so that peaks and valleys become more pronounced as scale increases. Their additional assumptions in order to prove the uniqueness are that the kernel g is infinitely differentiable and rapidly decreasing (Schwartz), there exists a kernel h such that $g(x,t) = th(xt)$ so that the scale parameter t stretches the kernel along the x-axis while keeping its area invariant, the kernel is symmetric, that is $g(x,t) = g(-x,t)$, the kernel is normalized so that $\int_{-\infty}^{\infty} g(u,t)du = \int_{-\infty}^{\infty} h(v)dv = 1$, and there exists a $p \in \mathbb{Z}$ such that $h^{(2p)}(0) \neq 0$, that is, not all derivatives of h vanish at 0. The normalization assumption insures that if f is a constant signal then it remains the same constant though the convolution. The authors also show that the diffusion equation is equivalent to requiring the monotonicity condition.

In two dimensions the zero-crossings are more complicated. They do not vanish as scale increases but can split and merge.

Florack et al 1992 [64]

Florack et al also prove that the Gaussian kernel is unique. They use the assumptions of linearity, spatial shift invariance, isotropy and scale invariance as the basic axioms, and then derive a weak semi-group property which ensures that several successive scalings is the same as performing a single equivalent scaling and combine it with a uniform scaling property over scales to finally show the uniqueness.

Lindeberg 1994 [123]

In Lindeberg's 1994 book all his work over the previous decade is nicely summarized. He uses non-creation of features as well as a semi-group structure to

prove the uniqueness of the Gaussian kernel (proven in his 1990 paper [120]). He also shows that the number of zero-crossings in the second derivative decreases monotonically with scale.

In a later paper by Lindeberg [124] the main results of his book are summarized.

Alvarez et al 1993 [4]

Alvarez et al present a very theoretical paper on the requirements of an image processing transform. They classify the requirements as either *architectural*, *stability* or *morphological*. The architectural axioms are those of recursivity (semi-group property, causality), existence of an infinitesimal generator (to remove the dependence on h, the sampling distance), regularity

$$\|L_{f+hg}(t) - (L_f(t) + hg)\|_\infty \leq Cht \ \forall h, t \in [0,1],$$

for smooth f, g where C depends on f, g, and locality, namely, for small t, $L_f(t)$ at any point x is determined by its vicinity, namely, for all $f, g \in C^\infty$ whose derivatives are equal at x,

$$(L_f(t) - L_g(t))(x) = o(t) \text{ as } t \to 0^+.$$

The stability axioms boil down to the comparison principle i.e. no enhancement can be made. This is also interpreted by [239] as nonnegativity, that is, $k_t(x) \geq 0 \ \forall x, \ \forall t > 0$, to ensure new level crossings do not appear. This is satisfied if we require monotonicity,

$$f \leq g \longrightarrow L_f(t) \leq L_g(t) \ \forall t > 0,$$

or preservation of non-negativity,

$$f \geq 0 \longrightarrow L_f(t) \geq 0 \ \forall t > 0.$$

The morphological axioms are average grey level invariance, translation invariance, isometry invariance and scale invariance:

AGLI: $\int_{\mathbb{R}^N} L_f(t)(x)dx = \int_{\mathbb{R}^N} f(x)dx \ \forall t > 0$. This requires that the kernels be normalized $\int_{\mathbb{R}^N} k_t(x)dx = 1$ or that grey level shift invariance is satisfied

$$L_f(t)(0) = 0, L_{f+c}(t) = L_f(t) + c.$$

TI: $L_{\tau_h f}(t) = \tau_h(L_f(t))$ where $\tau_h f = f(x+h)$.

II: $L_{Rf}(t) = RL_f(t)$ for all orthogonal transforms R defined by $(Rf)(x) = f(Rx)$.

SI: For any λ and t, there exists t' such that $D_\lambda L_f(t') = L_f(t)D_\lambda$ so that the result of $L_f(t)$ is independent of the size of the features involved.

They show that a sequence of multi-scale operators $L_f(t)(x) = u(t,x)$ is a solution of a second order partial differential equation

$$\frac{\partial u}{\partial t} F(D^2 u, Du) \text{ with } u(0,x) = f(x)$$

with certain requirements satisfied, namely recursivity, regularity, locality, translation and shift invariance. The heat equation is then the only linear isometrically invariant special case of this

They in addition combine the multi-scale ideas of Witkin et al (Gaussian scale-space and the heat equation) with the morphology scale-space ideas (structuring elements of differing sizes and the opening and closing operations) to obtain a 'class of morphological multi-scale analyses'. These satisfy

$$\frac{\partial u}{\partial t} \beta(t\text{curv}(u))|Du|$$

where β is an arbitrary non-decreasing real function and curv(u) is the curvature of the level set of u passing through x. This combination keeps the noise-elimination properties of the heat equation but is now shape-preserving due to the morphological operators.

Pauwels et al 1995 [168]

In this well-written 1995 paper by Pauwels, it is described how by assuming a semi-group property (what they and [4] call recursivity) and scale-invariance, and other more trivial assumptions, it is possible to derive a class of scale-space operators which depend on a parameter α for which the Gaussian is a special case when $\alpha = 2$.

They begin with assuming that the operators are linear, as all the other authors do as well, and are integral operators. This also allows operations to be run in parallel as as comparisons of neighbouring pixels are done.

So the operator has the form $(L_f(t)(x) = \int_{\mathbb{R}} k_t(x,\xi)f(\xi)d\xi$ where k_t is the integral kernel. By also assuming shift-invariance the kernel must then be a convolution kernel i.e. $k_t(x,t) = k_t(x-t)$, and so $K_t = k_t * f$. They impose the following conditions on the kernel k_t:

- k_t **is mass-preserving:** $\int k_t(x)dx = 1$ so that $k_t * 1 = 1$ and a constant signal is not changed.
- k_t **is even:** $k_t(x) = k_t(-x)$
- k_t **is integrable** ($k_t \in L^1$)**:** otherwise the convolution is not well-defined
- k_t **is a continuous function of** t **and** x

Then assuming an additive property, namely recursivity: $K_0(f) = f$ and $K_t K_s = K_{t+s}\ \forall t, s \geq 0$ (the kernel also forms such a semi-group: $k_t * k_s = k_{t+s}$) and scale-invariance they derive a rescaling of this kernel family from a fixed kernel ϕ which depends on a parameter α. Thus recursivity and scale-invariance are not sufficient to single out the Gaussian kernel as unique as it is a special case when $\alpha = 2$. They obtain this same result for two dimensions. They delve deeper and show that the Gaussian kernel is only unique if requiring the existence of an infinitesimal generator of differential form. Then the α's can only be even integers and only for $\alpha = 2$ do we obtain positivity everywhere.

Nielson et al 1996 [157]

In this paper scale-space, functional minimization and edge detection filters are compared. They show that the Gaussian scale-space can be obtained through Tikhonov regularization if requiring scale invariance and a semi-group constraint (recursivity). Regularization is the minimization of a signal with respect to an energy functional. A function u is a Tikhonov regularization of a signal $f \in L^2(\mathbb{R}^2)$ if it minimizes the energy function

$$E_f[u] = \int_{\mathbb{R}} \left[(f-u)^2 + \sum_{i=1}^{\infty} \lambda_i \left(\frac{d^i u}{dx^i} \right)^2 \right] dx$$

where $\lambda_i \geq 0$. They also show that this regularization then further more results in the heat equation. Their results are also proven for higher dimensions.

Florack 1996 [61]

In this paper Florack presents a formal theoretical definition of an image with an associated filter space as well as group structure. He also shows in this manner, that the Gaussian is the unique filter for a linear convolution integral operator.

Relation to the Japanese Gaussian Scale-Space Axioms [239]

The Japanese axioms for uniqueness differ from the more recent approaches in two ways. Firstly, the earlier Japanese approaches use less axioms than even recent approaches. Secondly, the axioms are simpler as they don't require any Fourier analysis, complex integrals nor functional analysis.

4.4.3 Discretizing the Gaussian Scale-Space

In practice signals are not continuous. We only have discrete data when a signal, image or video is captured. A signal is captured as a discrete sequence, an image as a matrix, and a video as a discrete sequence of matrices. The Gaussian scale-space theory presented up to now has assumed a continuous input f. The actual implementation of the continuous Gaussian scale-space thus proves difficult. There are two options presented in [120, 121, 123].

The first option is the obvious one, namely, the sampled application of the continuous theory. More specifically this involves discretizing the developed continuous theory and the equations therein via numerical methods. This can be done relatively effectively by using sampled values of the Gaussian kernel together with the rectangle rule of integration. This method, although it gives accurate numerical results, does not guarantee the non-creation of structure as scale increases, which is the most important requirement for a scale-space. The discretization of the diffusion equation is also an option. This is proposed and done with the ordinary 5-point Laplace operator thereby forming a set of ordinary differential equations. We will return to the discretized diffusion equation after we first deal with the second option. The scale, currently continuous, should also be discretized in a logical manner to enable the application. This will be discussed later in this chapter.

In order to maintain the desired theoretical structure of the continuous theory

through the discretization process, the second option is to develop an entirely new (discrete) theory based on the same axioms but modified for the discrete structure we must now work with. This method in fact gives a computational advantage over the first option as well. A discrete convolution of f by a kernel T, namely $T(\cdot,t) * f(\cdot)$, is obtained as

$$L_f(t)(x) = \sum_{n=-\infty}^{\infty} T(n,t) f(x-n), t > 0.$$

The scale parameter t will be kept continuous though to allow for the freedom of choosing any scale t greater than zero instead of only certain values.

Lindeberg first develops this new theory for one dimension. His main requirement for the discrete kernel, $T(n,t)$, is that the number of local extrema in the convolved signal does not exceed the number of local extrema in the original signal. This implies that the amount of structure in the signal will decrease as scale increases, as is the case with the continuous theory. He calls a kernel which satisfies this property as a *scale-space kernel*. He then derives the discrete scale-space as

$$\begin{aligned} L_f(0)(x) &= f(x) \\ L_f(t)(x) &= \sum_{n=-\infty}^{\infty} T(n,t) f(x-n), t > 0 \end{aligned}$$

where $T(n,t) = e^{-t} I_n(t)$ and I_n is the modified Bessel functions of integer order which was discussed in the earlier part of this chapter. This discrete scale-space satisfies the following properties:

- The amount of structure does not increase with scale so that for $t_2 > t_1$ the number of local extrema in $L_f(t_2)(x)$ is not more than the number in $L_f(t_1)(x)$.
- A semi-group property: $L_f(t_2)(\cdot) = T(\cdot, t_2 - t_1) * L_f(t_1)(\cdot)$.
- Normalization: $\sum_{n=-\infty}^{\infty} T(n,t) = 1$.
- Symmetry: $T(-n,t) = T(n,t)$.
- $var\,(T(\cdot,t)) = \sum_{n=-\infty}^{\infty} n^2 T(n,t) = t$.

There are a few points to consider for numerical implementation of this discrete scale-space. Firstly, the infinite sum needs to be replaced by a finite

one, that is, we sum from $n = -N$ to N for some finite N. This N can be chosen such that the absolute error in L due to the truncation does not exceed a given error limit ϵ_{trunc}. Secondly, the modified Bessel function need to be calculated with the recurrence relation,

$$I_{n-1}(t) - I_{n+1}(t) = \frac{2n}{t} I_n(t)$$

which is stable for backward iteration. In Lindeberg's work he states that built in routines are not available to evaluate the I_n's and hence his development of code making use of this recurrence relation. However, at present this is no longer true. In Mathematica the function *BesselI[n,t]* is available, in R the function *besselI[n,t,expon.scaled=FALSE]* is available, in SAS the function *IBESSEL(t,n,0)* is available, and in MATLAB the function *besseli(t,n)* is available, to name a few. This is thus no longer a major problem for the implementation.

We return to investigating the discretization of the diffusion equation. The convolution $L_f(t)(x)$ above is the solution of the following partial differential equation

$$\frac{\partial L_f(t)(x)}{\partial t} = \frac{1}{2}\left(L_f(t)(x+1) - 2L_f(t)(x) + L_f(t)(x-1) \text{ for } x \in \mathbb{Z}\right).$$

A two-dimensional discrete scale-space is more tricky to develop since the non-creation of structure as scale increases isn't always true in two dimensions. Lindeberg requires instead that local extrema must simply not be enhanced as scale increases, that is local maxima must not increase and local minima not decrease as scale increases. This reduces to the one dimensional axiom if the space is reduced to one dimension. He derives the scale-space operator as

$$L_f(t)(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T(m,n,t) f(x-m, y-n), t > 0$$

which satisfies the differential equation

$$\frac{\partial L_f(t)(x,y)}{\partial t} = \frac{1}{2}\left((1-\gamma) \bigtriangledown_5^2 L_f(t)(x,y) + \gamma \bigtriangledown_\times^2 L_f(t)(x,y)\right)$$

where $\bigtriangledown_5^2$ is the five-point operator

$$(\bigtriangledown_5^2 f)(x,y) = f(x-1,y) + f(x+1,y) + f(x,y-1) + f(x,y+1) - 4f(x,y)$$

and $\bigtriangledown^2_\times$ is the cross operator

$$\begin{aligned}(\bigtriangledown^2_\times f)(x,y) &= \frac{1}{2}\left(f(x-1,y-1)+f(x-1,y+1)+f(x+1,y-1)\right.\\ &\left.+f(x+1,y+1)-4f(x,y)\right),\end{aligned}$$

both approximations of the two-dimensional Laplace operator $\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$. The kernel T is symmetric, that is, $T(-x,y,t)=T(x,y,t)$ and $T(y,x,t)=T(x,y,t)$ and satisfies a continuity property

$$\|T(\cdot,\cdot,t)-\delta(\cdot,\cdot)\|_1 \to 0 \text{ as } t \to 0$$

where δ is the two-dimensional delta function which is 1 at $(0,0)$ and 0 elsewhere. The operator L_f is linear, shift-invariant and satisfies the semi-group property. If $\gamma = 0$ then $T(m,n,t)$ is a separable convolution kernel and so

$$L_f(t)(x,y) = \sum_{m=-\infty}^{\infty} T(m,t) \sum_{n=-\infty}^{\infty} f(x-m,y-n), t>0$$

where $T(n,t)=e^{-t}I_n(t)$. In addition if (x_0,y_0) is a local maximum (minimum) point then

$$\frac{\partial L_f(t)(x_0,y_0)}{\partial t} \leq (\geq)\ 0.$$

Lindeberg defines (x,y) as a local maximum (minimum) point if for $f:\mathbb{Z}^2 \mapsto \mathbb{R}$, we have $f(x,y) \geq (\leq) f(\xi,\eta)\ \forall\ (\xi,\eta) \in N_8(x,y)$ where $N_8(x,y)$ defines the eight vertical, horizontal and diagonal neighbours of the point (x,y). This two dimensional formulation can also be generalized to higher dimensions, see [123, Chapter 4].

4.4.4 Relating Scales

Having thus obtained a scale-space of the signal f the true question now is how do we use all these smoothed versions of the signal as one? How do we construct links between the scale levels? Witkin [245] presents a tree structure in this regard however as stated in [4] this method implies heavy implementation from the computational point of view and is unstable because of the follow-ups to check for edges (zero-crossings) at each scale. However, with today's computing power this statement may no longer be valid. Relating the scales is directly related to feature detection via the scale-space, so we'll return to this in Chapter 4.8.2.

4.5 Other Scale-Spaces Developed

The Gaussian scale-space provides a multi-scale representation of an image such that a full image is derived at each possible (or required) scale level [123]. This is in contrast to a pyramid representation in which the original image size is reduced at every step and provides then a multiresolution representation. Such a multiresolution technique provides reduced computational requirements but does not allow for explicit access to each of the scale levels. Even more importantly, it does not allow for a method of associating structures over the scale levels, and for which a scale-space does. Thus we focus on multi-scale methods.

Numerous researchers have introduced multi-scales methods different to the Gaussian scale-space. We discuss them now. Note the the uniqueness of the Gaussian scale-space is specific to the axioms imposed.

Scale-Space via the Gabor Functions

Daniel Gabor suggested the Gabor functions in 1946 [68] when Fourier analysis didn't provide him with the freedom to vary the frequency parameter through time. The Gabor functions are as follows,

$$g_{\ell,n}(x) = g(x - a\ell)e^{2\pi ibnx}, -\infty < \ell, n < \infty$$

where $g \in \mathcal{L}^2(\mathbb{R})$ and $\|g\| = 1$, i.e. they are a family of functions built from translations and modulations of a function g. By choosing the function g to be the Gaussian, a scale parameter is introduced and a hierarchical decomposition of a signal can be obtained.

In [134] the Gabor functions are used to develop a time-frequency dictionary of functions $g_\gamma(t)$ to yield an adaptive decomposition of a signal f, namely,

$$f(t) = \sum_{n=-\infty}^{\infty} a_n g_{\gamma_n}(t),$$

in which the functions are selected in order to best match the structure of the signal. The possibility of applying this to signal coding is discussed since it will provide a more efficient coding than orthogonal decompositions. In [72] Granlund designs a general parallel and hierarchical operator and bases it in the Gabor functions with g as the Gaussian. His basic idea is for the operator to describe the image locally as a vector with two components, direction and magnitude. In [149] a hierarchical model using multi-oriented, multi-scale

Gabor functions is presented which models the human visual cortex. The model is used for multi-class object recognition by extracting a feature set representing the salient characteristics of the objects.

Nonlinear Anisotropic Diffusion

In 1984 Cohen and Grossberg [37] discuss the diffusion of boundary feature information for a boundary-completion process in the HVS and provide a nonlinear diffusion equation to model the activity, but in 1990 Perona and Malik [170] presented the anisotropic diffusion scale-space in order to improve on the non-meaningful and blurred edges resulting from the Gaussian scale-space. Instead of the constant diffusion coefficient c in Koenderink's linear diffusion equation $I_t = c(I_{xx} + I_{yy})$ [109] they use a coefficient $c(x, y, t)$ dependent on the spatial and scale parameters thereby introducing the nonlinear equation $I_t = c(x, y, t)(I_{xx} + I_{yy}) + \nabla c \cdot \nabla I$. They apply the scale-space for improved edge detection. Whitaker and Pizer [241] combine the information over the scales effectively for edge detection. Shah [216] investigates using nonlinear diffusion for improved segmentation. Alvarez et al [4] discuss Perona and Malik's nonlinear anisotropic equation as well as their own adapted nonlinear approach which is linked with a morphology approach.

Mathematical Morphology

Scale-spaces are also prominent in mathematical morphology. They result from the recursive applications of morphological operators. Some examples follow.

Maragos [135] investigates the morphological scale space using morphological openings and closings which ensure the preservation of edges. Braga-Neto [20] defines a σ-connected operator, that is, an operator connected at scale σ. He uses these operators to obtain a morphological scale-space representation and applies it for automatic target detection. Braga-Neto and Goutsias [26] use greyscale connectivity, namely a grayscale image is connected if all level sets below a pre-specified threshold are connected, to build a morphological scale-space. The apply the scale-space to object extraction, segmentation, and object-based filtering. Brag-Neto [21] also investigates a nonlinear pyramidal image representation scheme via multiscale grain filters by gradually removing connected components from an image that fail to satisfy a given criterion.

4.6 Scale-Space: A Formal Definition

In [20] a scale-space is referred to as a representation which allows for the tracking of the evolution of image structures (e.g. regional maxima and minima) through a continuous range of scales, from fine to coarse, basically an ordered set of derived images which represent the original at alternative scales [123]. These descriptions are very vague although clear as to their intention. We proceed to define a scale-space formally. Let Ω be an infinite space (for example $\mathbb{R}^n$ or $\mathbb{Z}^n$) and $\mathcal{A}(\Omega)$ the set of all real functions defined on Ω. The space Ω is purposefully general so as to provide an axiomatic definition of a scale-space.

First we provide an axiomatic definition for a scaling operator. This definition makes allowance for any domain, discrete, continuous or otherwise.

Definition 28 *An operator $\varphi : \Omega \mapsto \Omega$ is called a* scaling operator *if it is 1) an order preserving mapping, and 2) $\forall\ x \in \Omega$ there exists $a_x \in \Omega$ such that $\varphi^{-1}(x) = \varphi^{-1}(0) + a_x$.*

We now define a measure of smoothness.

Definition 29 *A function $S : \mathcal{A}(\Omega) \mapsto \mathcal{A}(\Omega)$ is called a* measure of smoothness *if the following axioms hold for any $f, g \in \mathcal{A}(\Omega)$:*

A1 $Sf = 0 \iff f$ *is constant.*

A2 $S(\alpha f) = |\alpha| S(f)$

A3 $S(f + g) \leq S(f) + S(g)$

A4 $S(f \circ E_\alpha) = Sf$ *for* $\alpha \in \Omega$ *(translation invariance)*

A5 $S(f \circ \varphi) = Sf$ *(scale invariance)*

In Definition 29 for $\alpha \in \Omega$ the operator $E_\alpha : \mathcal{A}(\Omega) \mapsto \mathcal{A}(\Omega)$ is a shift operator, namely, $(E_\alpha f)(x) = f(x - \alpha)$, and the function $\varphi : \Omega \mapsto \Omega$ is a scaling operator as in Definition 28. Note that the first three axioms for the smoothing operator in Definition 29 are those for a semi-norm. Axioms 4 and 5 are invariance properties. Note also that the operator S is actually a measure of

'roughness' since a larger value indicates less 'smoothness'. As mentioned in [81], the choice of S in general depends on the requirements of the specific task so Definition 29 sets general axioms. A number of alternatives for S have been suggested in literature. For example, in [234] smoothness is considered as a measure of how each data point is similar to or well supported by the data points in its vicinity. Qi and Sun [175] consider a function smooth if it is continuously differentiable, that is the function as well as its first derivative are continuous. We could also consider a function smooth provided the derivatives up to a specific order are continuous, choosing the specific order based on the task at hand.

We now define a scale-space operator.

Definition 30 *Let $\Lambda \subset \mathbb{R}^+$ be the an unbounded set of scale parameters. An operator $\mathcal{L}(f, \lambda) : \mathcal{A}(\Omega) \times \Lambda \to \mathcal{A}(\Omega)$ where $f \in \mathcal{A}(\Omega)$ is a* scale-space operator *if it satisfies the following axioms:*

A1 $\mathcal{L}(f, 0) = f$

A2 *For every $\lambda_1, \lambda_2 \in \Lambda, \lambda_1 < \lambda_2$ we have $S(\mathcal{L}(f, \lambda_2)) \leq S(\mathcal{L}(f, \lambda_1))$. Moreover,*

$$\lim_{\lambda \to \infty} S(\mathcal{L}(f, \lambda)) = 0.$$

A3 $\mathcal{L}(\alpha f, \lambda) = \alpha \mathcal{L}(f, \lambda)\ \forall \alpha > 0$ *(***Positive Homogeneity***)*

A4 *For every $\lambda_1, \lambda_2 \in \Lambda, \lambda_1 < \lambda_2$, there exists an operator $\mathcal{M}(\lambda_1, \lambda_2) : \mathcal{A}(\Omega) \mapsto \mathcal{A}(\Omega)$ such that $\mathcal{M}(\lambda_1, \lambda_2) \circ \mathcal{L}(f, \lambda_1) = \mathcal{L}(f, \lambda_2)$. (***Cascading Property***)*

A5 $E_\alpha \circ \mathcal{L}(f, \lambda) = \mathcal{L}(f, \lambda) \circ E_\alpha$ *(***Translation Invariance***)*

A6 *For each $\lambda \in \Lambda$ there exists $\lambda' \in \Lambda$ such that $\mathcal{L}(f, \lambda') \circ \varphi = \mathcal{L}(f \circ \varphi, \lambda)$ (***Scale Invariance***)*

Some points to take note of. Axiom A1 ensures that the original image forms part of the scale-space. Axiom A4 enables the successive smoothing by $\mathcal{L}(f, \cdot)$ first at a scale λ_1 and then at scale $\lambda_2 > \lambda_1$ on the already smoothed $\mathcal{L}(f, \lambda_1)$. Notice also that $\mathcal{L}$ need not necessarily be linear. In [123, Chapter 3] some general axioms are presented for a linear scale-space ensuring the smoothing operation is a convolution.

For convenience we denote $\mathcal{L}(f, \lambda)$ as $\mathcal{L}_f(\lambda)$ since the first parameter f is fixed and the second parameter λ varied in applications.

Following Definition 30, we define a precise definition of a scale-space associated with a given function $f \in \mathcal{A}(\Omega)$ as the range of the operator $\mathcal{L}_f$.

Definition 31 *Let $f \in \mathcal{A}(\Omega)$. The set*

$$\mathcal{S}_{f,\Lambda} = \{(\lambda, \mathcal{L}_f(\lambda)) : \lambda \in \Lambda\}$$

is called a scale-space *of f generated by the operator $\mathcal{L}$ with respect to scale parameter set Λ and measure of smoothness $S \in \mathcal{A}(\Omega)$.*

In the literature the term scale-space is used with more broad meaning. In addition to the set in Definition 31 the term is also referred to its subsets or to the operator $\mathcal{L}$. As this may lead to confusion, we will use it here only with the meaning given in Definition 31.

We show that the Gaussian scale-space satisfies the axioms of Definition 30 and 31.

Theorem 32 *The Gaussian scale-space operator defined in Section 4.4 satisfies the axioms of Definition 30.*

Proof
For the Gaussian scale-space the scale parameter set Λ is continuous and is given by $\{t : t \geq 0\}$.

A1 This follows from Equations 4.5 and 4.6.

A2 For the Gaussian scale-space operator the measure of smoothness $S \in \mathcal{A}(\Omega)$ defined in Definition 29 is the continuous total variation, namely

$$TV(f) = \int_\Omega |\nabla f(x)| dx.$$

It is clear that $TV(f)$ satisfies the axioms of Definition 29. Since the derivative of the Gaussian scale-space operator is

$$\frac{\partial}{\partial x}\mathcal{L}_f(t)(x) = \left(\frac{\partial}{\partial x} g(x,t)\right) * f(x),$$

and it is well known that the Gaussian density function flattens out as the variance t increases, namely g satisfies for $t_1 < t_2$

$$\frac{\partial}{\partial x}g(x,t_2) \leq \frac{\partial}{\partial x}g(x,t_1),$$

we know that

$$|\nabla \mathcal{L}_f(t_2)(x)| \leq |\nabla \mathcal{L}_f(t_1)(x)|$$

so that $TV(\mathcal{L}_f(t_2)) \leq TV(\mathcal{L}_f(t_1))$. Also as $t \to \infty$ $\frac{\partial}{\partial x}g(x,t) \to 0$ so that $\frac{\partial}{\partial x}\mathcal{L}_f(t)(x) \to 0$ and so

$$\lim_{t\to\infty} TV(\mathcal{L}_f(t)) = 0.$$

A3 This follows immediately for a convolution.

A4 The cascading property of the Gaussian scale-space operator is as follows for $t_2 > t_1$, [123, Chapter 2.4.4]

$$\mathcal{L}(\cdot,t_1) = g(\cdot,t_2-t_1) * \mathcal{L}(\cdot,t_1).$$

So the operator $\mathcal{M}(t_1,t_2)$ is given by a convolution with a Gaussian kernel with parameter $t_2 - t_1$.

A5 Translation-invariance is a required property of the Gaussian scale-space operator [123].

A6 In [123, Chapter 2.4.8] it is verified that for each $t \in \Lambda$ there exists $t' \in \Lambda$ such that $\mathcal{L}(f,t)(x) = \mathcal{L}(f',t')(x')$ where $f \circ \varphi(x) = f'(sx)$, $t' = s^2t$ and $x' = sx$ for $s \in \mathbb{R}^+$.

■

4.7 The LULU Scale-Space

The DPT forms a scale-space in the sense of Definitions 30 and 31 when applied to a function f. We shall prove this. Firstly note that due to the idempotence of the LULU operators $\mathcal{L}_f(\lambda) = \mathcal{L}_{\mathcal{L}_f(\lambda_1)}(\lambda)$, indicating it doesn't really make sense to apply $\mathcal{L}_f(\lambda_1)$ first as the same is achieved by applying $\mathcal{L}_f(\lambda)$ for $\lambda > \lambda_1$. However, the information which is peeled off by first $\mathcal{L}_f(\lambda_1)$ and then $\mathcal{L}_f(\lambda)$ indicates the reason for applying them step by step. Total variation as defined in Definition 2.15 is a smoothing operator as described in Definition 29. The five axioms are proved in [52].

Theorem 33 *The Discrete Pulse Transform, when applied to* $f : \mathbb{Z}^d \to \mathbb{R}$*, derives a scale-space* $S_f^{LULU} = \{(n, P_n(f)) : n \in \Lambda_0 = \{0, 1, 2, ..., N\}\}$ *as described by Definition 31 which we call the* LULU scale-space.

Proof We proceed to show that the axioms in Definition 30 are satisfied by the LULU scale-space.

A1 Since the DPT is the result of P_n, $n = 1, 2, ..., N$, where N is the total number of data points, it is trivial then to have $P_0(f) \equiv f$.

A2 By Theorem 14 we know $P_n(f)$ is total variation preserving so for $n_2 > n_1$

$$\begin{aligned} TV(P_{n_1}) &= \sum_{n=n_1+1}^{N} TV(D_n(f)) \\ &\leq \sum_{n=n_2+1}^{N} TV(D_n(f)) \\ &= TV(P_{n_2}(f)). \end{aligned}$$

Since $D_N(f)$ is constant we know that

$$\lim_{n \to N^+} TV(P_n(f)) = 0.$$

A3 Axiom A3 holds as discussed in detail in Chapter 1 and presented in Theorem 26.

A4 Due to the idempotence of the LULU operators

$$\begin{aligned} \mathcal{L}_{\mathcal{L}_f(n_1)}(n_2) &= P_{n_2}(\mathcal{L}_{\mathcal{L}_f(n_1)}(n_2 - 1)) \\ &= P_{n_2} \circ ... \circ P_{n_1+1}(\mathcal{L}_{\mathcal{L}_f(n_1)}(n_1)) \\ &= P_{n_2} \circ ... \circ P_{n_1+1}(P_{n_1} \circ P_{n_1}(\mathcal{L}_f(n_1 - 1)) \\ &= P_{n_2} \circ ... \circ P_{n_1+1}(P_{n_1}(\mathcal{L}_f(n_1 - 1)) \text{ by idempotence} \\ &= P_{n_2} \circ ... \circ P_{n_1+1}(\mathcal{L}_f(n_1)) \\ &= \mathcal{L}_f(n_2) \end{aligned}$$

A5&A6 These axioms of translation and scale in variance follow immediately from the properties of a separator given in Definition 2 since the LULU operators are separators.

■

Definition 34 *The Discrete Pulse Transform, when applied to* $f : \mathbb{Z}^d \to \mathbb{R}$, *also derives a related scale-space* $S_f^{LULU^C} = \{(n, P_n(f) - P_{n-1}(f)) : n \in \Lambda_0 = \{0, 1, 2, ..., N\}\} = \{(n, D_n(f)) : n \in \Lambda_0 = \{0, 1, 2, ..., N\}\}$ *which we call the* complimentary LULU scale-space.

Theorem 35 *If* $\mathcal{L}_f : \mathcal{A}(\Omega) \mapsto \mathbb{R}$ *satisfies the cascading property in Axiom 4 of Definition 30, then for every* $\lambda_1, \lambda_2 \in \Lambda, \lambda_1 < \lambda_2$,

$$\mathcal{L}_f(\lambda_1) = \mathcal{L}_g(\lambda_2) \Rightarrow \mathcal{L}_f(\lambda_2) = \mathcal{L}_g(\lambda_2).$$

Proof
By the cascading property we have

$$\begin{aligned} \mathcal{L}_f(\lambda_1) &= \mathcal{M}(\lambda_1, \lambda_2) \circ \mathcal{L}_f(\lambda_1) \\ &= \mathcal{M}(\lambda_1, \lambda_2) \circ \mathcal{L}_g(\lambda_2) \end{aligned}$$

■

Definition 36 *Given a measure of smoothness* S, *a function* $g \in \mathcal{A}(\Omega)$ *is an* event *of* $f \in \mathcal{A}(\Omega)$ *if*

$$S(f - g) + S(g) = S(f).$$

Definition 36 indicates that by removing g from f the smoothness has increased (or roughness has reduced) as a part of f has been removed.

Definition 37 *An event* $g \in \mathcal{A}(\Omega)$ *of* $f \in \mathcal{A}(\Omega)$ *is present at scale* λ *if* $\mathcal{L}_{f-g}(\lambda) \neq \mathcal{L}_f(\lambda)$.

Note also that if $g \in \mathcal{A}(\Omega)$ is an *event* of $f \in \mathcal{A}(\Omega)$ we have

$$S(\mathcal{L}_{f-g}(\lambda)) < S(\mathcal{L}_f(\lambda)).$$

Theorem 38 *For every* $\lambda_1, \lambda_2 \in \Lambda, \lambda_1 < \lambda_2$, *if* $g \in \mathcal{A}(\Omega)$ *is an event of* $f \in \mathcal{A}(\Omega)$, *then*

$$\mathcal{L}_f(\lambda_1) = \mathcal{L}_{f-g}(\lambda_1) \Rightarrow \mathcal{L}_f(\lambda_2) = \mathcal{L}_{f-g}(\lambda_2).$$

Proof
Follows by Theorem 35. ■

Theorem 38 shows that if an event is present at scale λ_1, the same event is present at scale $\lambda_2 > \lambda_1$.

The local maximum and minimum sets (see Definition 41) derived by the DPT are exactly such events.

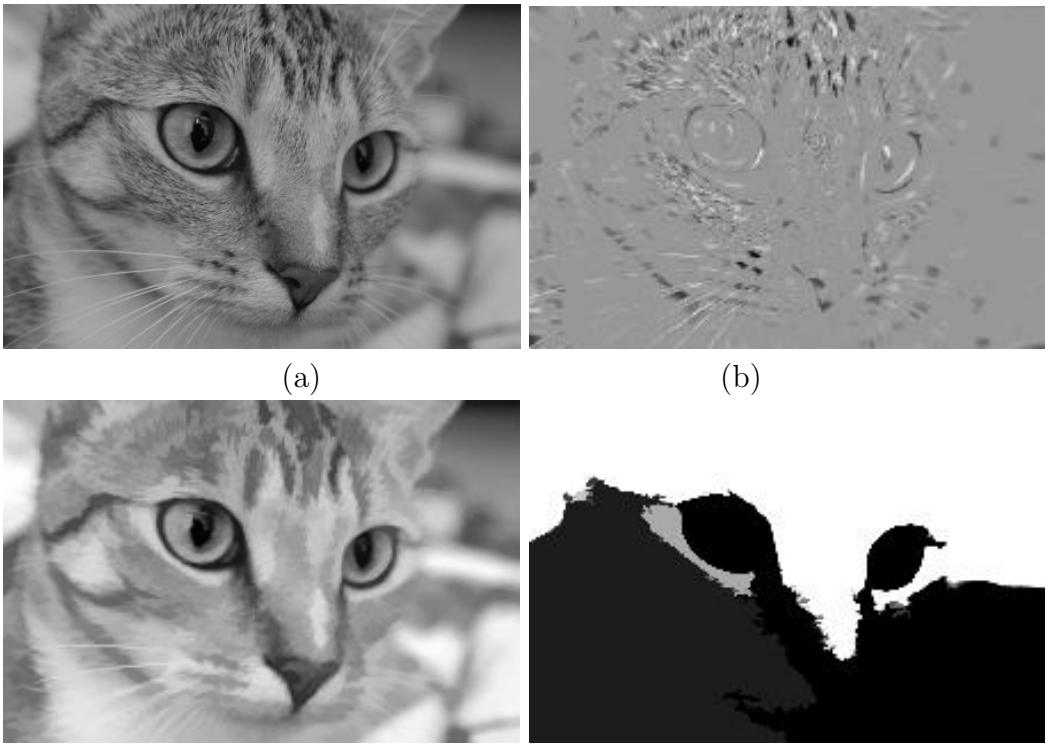
As mentioned before the Gaussian scale-space shifts and blurs edges through its scales and also does not correspond directly to object shapes at each scale [135]. The LULU scale-space does not suffer from this disadvantage. The LULU scale-space satisfies the axioms of the Gaussian scale-space as shown above but has the benefit of nonlinearity. This results in excellent shape and preservation properties, namely consistent separation, and total variation and shape preservation [8]. The Discrete Pulse Transform, $f = \sum_{n=1}^{N} D_n(f)$, forms a scale-space where the scaled image is $P_n(f)$ for discrete scales $n = 1, 2, 3, ..., N$. A second advantage of this LULU scale-space is then clear - it is already discrete and no approximations or sampling needs to be done, unlike with the Gaussian scale-space [12, 120, 109].

Often, a limited number of specific scales can sufficiently describe the important parts of an image, with discarded scales representing the background or noisy parts of the image. In addition, scales that repeat the representation of the same structures can be discarded or reduced, thereby reducing the amount of data but preserving the information contained in the image. Figure 4.3 gives an example of the break-down of an image into one possible LULU scale-space.

How do the individual pixel values change through the scale-space? We refer to this change over the scales as the *DPT pixel signatures*. Each pixel belongs to k pulses, $\phi_{n_1 s_{n_1}}, \phi_{n_2 s_{n_2}}, ..., \phi_{n_k s_{n_k}}$, at scales $\{n_1, n_2, ..., n_k\} \subseteq \{1, 2, 3, ..., N\}$. For each pixel x, we then have what we call a Discrete Pulse Vector (DPV) for a specified pixel $x \in \mathbb{Z}^2$,

$$\mathbf{p}_x = \begin{bmatrix} n_1 & n_2 & n_3 & \dots & n_k \\ \ell_1 & \ell_2 & \ell_3 & \dots & \ell_k \end{bmatrix}^T, x \in \mathbb{Z}^2 \tag{4.7}$$

where for each scale n_i, we have the corresponding relative luminosity ℓ_i of the pulse $\phi_{n_i s_{n_i}}$, that is, the height (positive) or depth (negative) of the local maximum or minimum set which pixel x belongs to at scale n_i. Figures 4.5 to



(a) (b)

(c) (d)

Figure 4.3: *A LULU scale-space for the Chelsea image (a) Original Image (b) Details - Scales* $n = 1$ *to* 35 *(c) Smoothed Image - Scales* $n = 36$ *to* 8000 *(d) Large Pulses - Scales* $n = 8001$ *to* $N = 33900$

Figure 4.4: *Original Canoeist image with the direction of the pixel indices indicated*

4.8 show the DPT pixel signatures of the canoeist, white water, dark water and 'normal' water areas of the image in Figure 4.4. The signatures indicated are similar for the same regions.

As a starting point in using the DPT for feature detection we investigate the DPV lengths. We detect the longest DPV's which represent pixels which are present over the most scales and reconstruct the image using only the discrete pulses that these pixels belong to. Figure 4.9 illustrates this idea. The method picks out the bottom left hand potato most likely because it has a background shadow as opposed to the rest of the potatoes. Figure 4.10 shows a similar result.

4.8 Scale-Space Applications

Scale-spaces have been used in a variety of applications namely image clustering and segmentation, deblurring and denoising of images, image enhancement, image compression, feature, corner and edge detection, as well as texture and shape analysis, to name a few.

In [118] the Gaussian scale-space is used to create a tree structure and then a stack approach used on this tree to segment the image. In [196] the hierarchical wavelet decomposition and Daubechieś four-tap filter are used to decompose the image into three detail images and a single approximate image. This is done recursively through the resulting pyramid to result in final improved segmentation via texture features. In [148] a hierarchical Markov Random Field (MRF) is used in segmenting high-resolution sonar images (in an unsupervised manner) using what they introduce as the scale casual

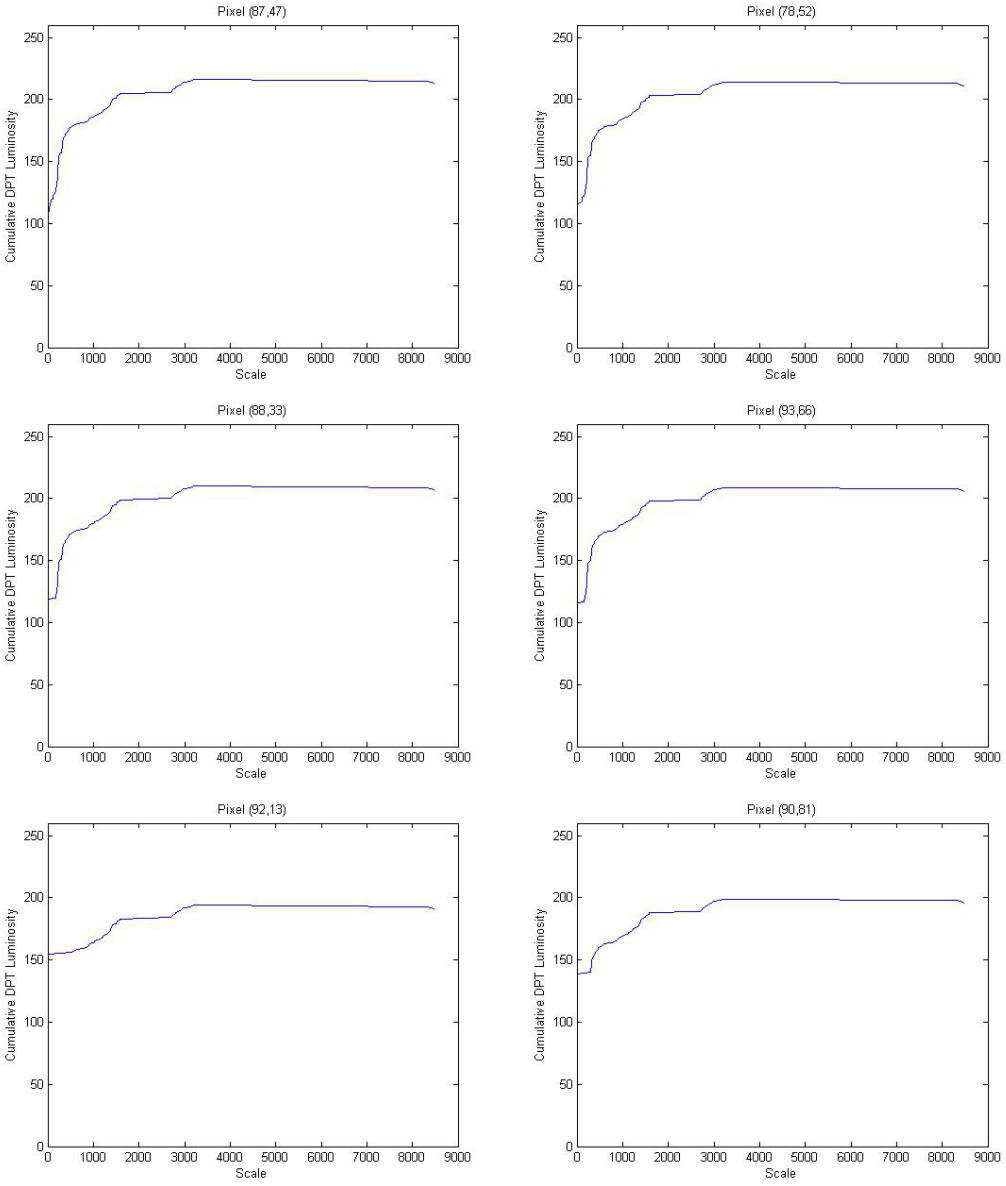


Figure 4.5: *DPT signatures of randomly selected pixels of the canoeist in the Canoeist Image*

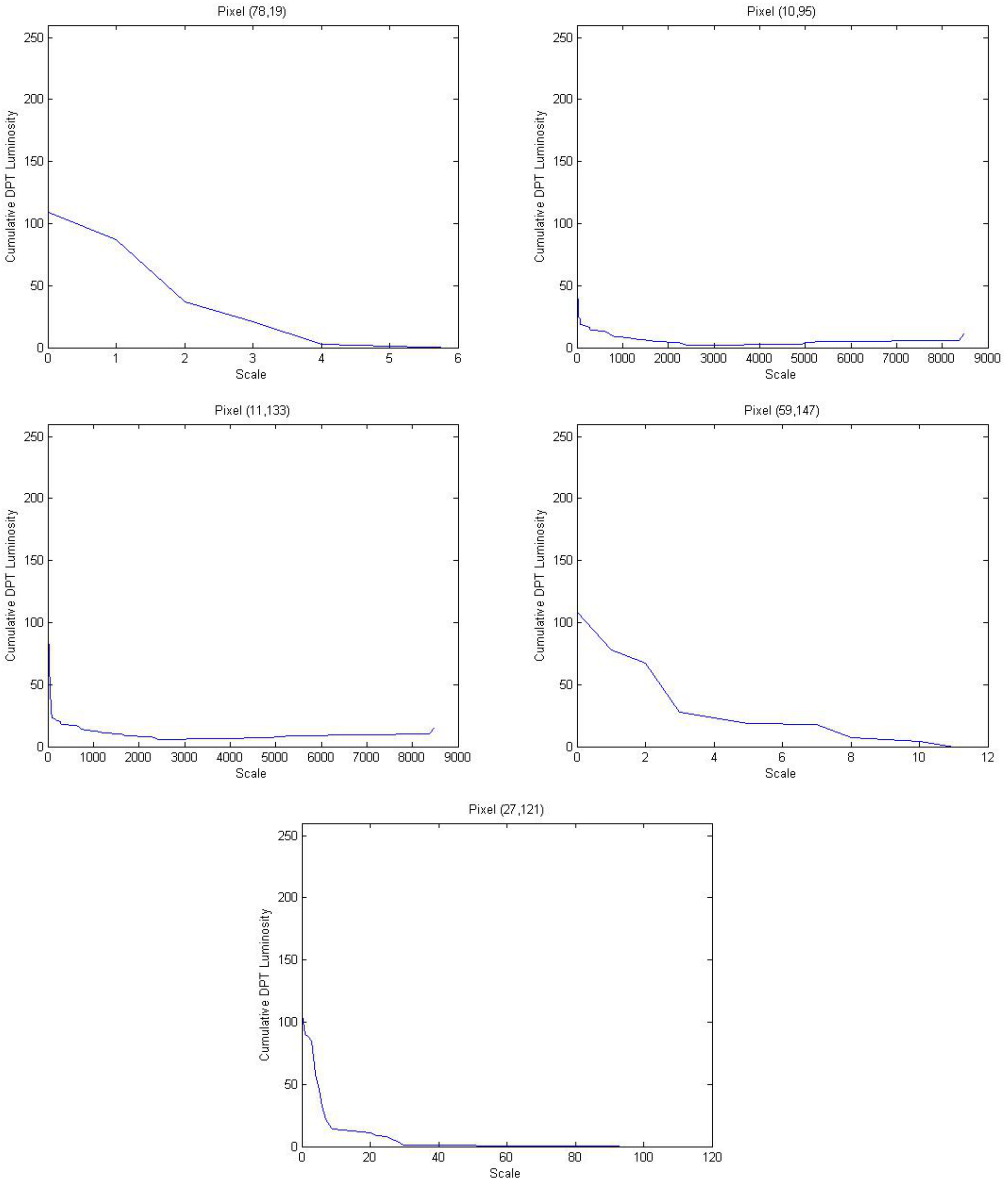


Figure 4.6: *DPT signatures of randomly selected pixels of the white water in the Canoeist Image*

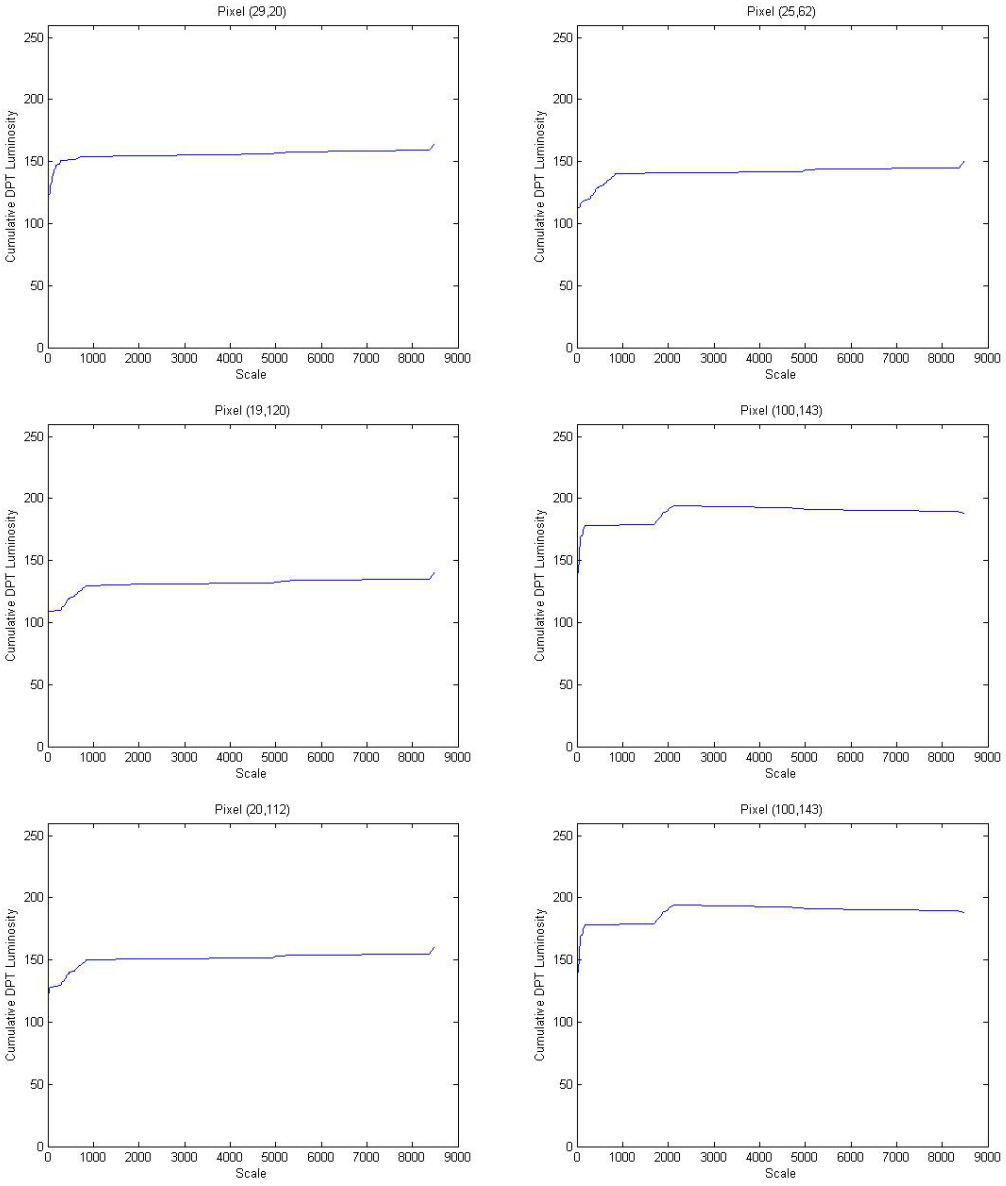


Figure 4.7: *DPT signatures of randomly selected pixels of the dark (shadowed) water in the Canoeist Image*

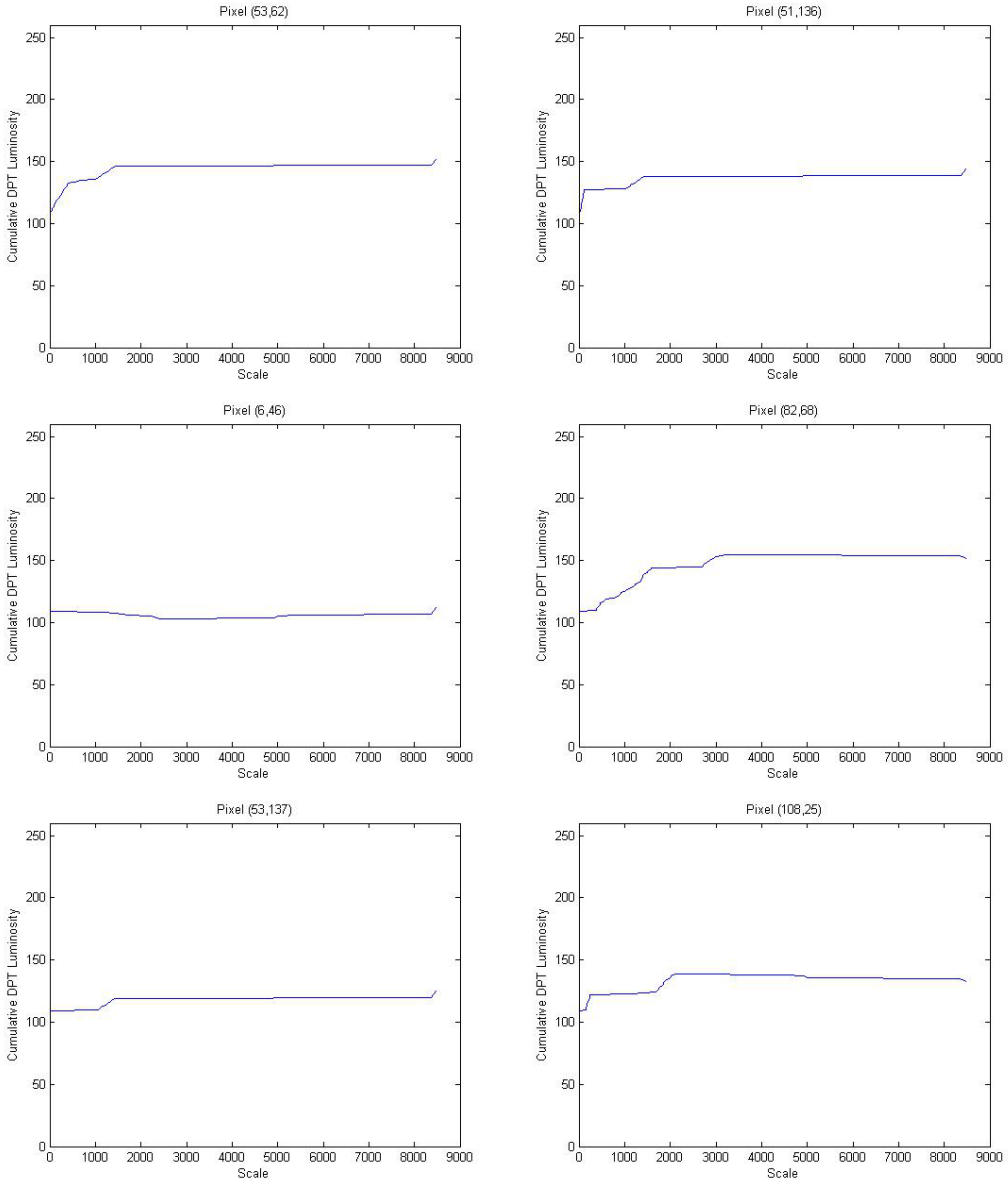
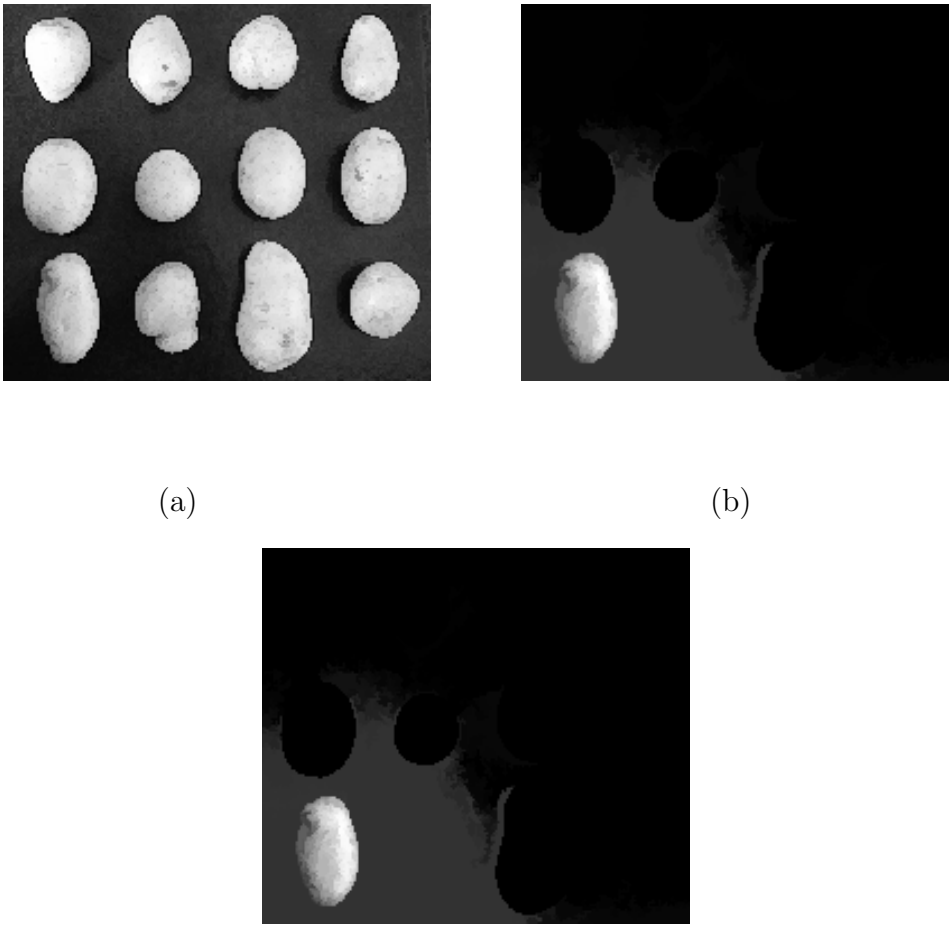


Figure 4.8: *DPT signatures of randomly selected pixels of the normal water in the Canoeist Image*



(a) (b)

(c)

Figure 4.9: *(a) Original (b) Ten largest DPV's (b) Largest DPV only*

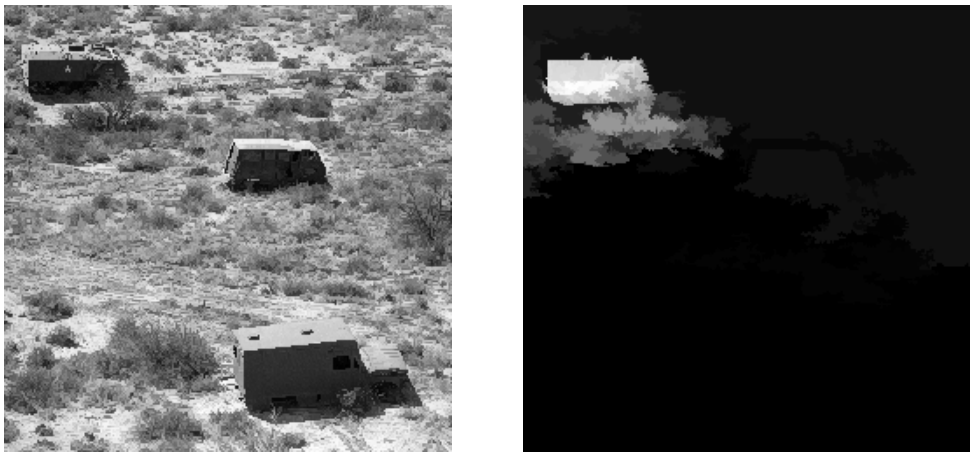


Figure 4.10: *(a) Original (b) 28 largest DPV's*

multigrid (SCM) algorithm. In [115] use of the Gaussian scale-space is justified as it simulates the action of the human visual system and a nested clustering dendrogram is produced such that data falling within the same region of the tree form a cluster in the segmentation. They also produce a non-nested hierarchical segmentation. In [146] a nonlinear scale-space via a general class of morphological levelings is presented and a brief description of how these levelings produce a segmentation is discussed. In [1] an improved segmentation is presented by using the morphological operators area open-close and close-open to produce a scale-space. In [111] use the Gaussian scale-space and its 'deep structure' to improve segmentation. In [171] a nonlinear scale-space is constructed via a diffusion equation, a tree is created and then unsupervised as well as supervised segmentation is presented via the edges through the scales of the scale-space. In [194] look at the spatial gradients between scales in the Gaussian scale-space and present a temporal segmentation of a sequence of images via the resulting scale-space tree. In [195] unsupervised discovery of valid clusters using statistics on the modes of the probability density function in the Gaussian scale space is shown.

In [192] multiscale total variation is introduced to improve their previous technique on textured regions for image recovery. In [51] a scale-adaption algorithm for reliable edge detection and blur estimation is presented. In [154] noise estimation is investigated via (i) multi-scale transforms, including wavelet transforms; (ii) a data structure termed the multiresolution support; and (iii) multiple scale significance testing. In [223, 222] an iterative varia-

tional decomposition via a minimizer functional is presented for deblurring, denoising and segmentation. In [247] a scale-space is developed via Markov Random Fields for the application of restoring degraded images.

In [19] an image compression scheme is introduced which involves a multiresolution decomposition derived from the wavelet transform. In [145] a cascade of compressions are produced via wavelet packets by coding the residual parts of each layer in a lossy manner which provides a sparse representation.

In [127] feature detection is determined via automatic scale selection in the Gaussian scale-space. In [228] a scale-space is created via pyramids of morphological operators and features are measured according to their persistence through the scales. In [149] multi-oriented, multi-scale Gabor filters are used to build a hierarchical model based on the visual cortex.

In [139] the raw primal sketch obtained with the Gaussian scale-space and its applicability for edge detection is presented.

In [177] the Gaussian scale-space is also made use of for corner detection. In [122] edge detection is investigated via automatic scale selection in the Gaussian scale-space. In [126] an edges strength is measured via the zero-crossings in the Gaussian scale-space and thereby enables edge detection. In [59] a thermodynamic model is employed for scale-space generation and significant edges (thin regions) are detected via this.

In [153] the curvature scale-space is presented (together with two additional versions of it) for shape representation at arbitrary scales and orientations. This provides insight into texture analysis. The author continues his work in [151, 152, 150] discussing shape matching similarity retrieval.

In [219] the author combines Shannons entropy and Witkin and Koenderinks scale-space to establish a precise connection between the heat equation and the thermodynamic entropy in Scale-Space. Experimentally the entropy function is used to study global textures.

Other applications of scale-spaces involve image fusion [225], image watermarking [173, 83], road extraction [141], astronomy [154], fingerprint enhancement [3], object tracking and recognition as well as image retrieval [105], surface editing in images [17] and palm print verification [119]. In [4] a multi-scale video analysis is described, an extension of their work for images, in which they introduce a new axiom namely that of Galilean invariance. This requires that the motion of the whole picture at constant velocity does not alter the analysis.

In [160] an imaged is decomposed into two images namely cartoon and the texture or noise and image deblurring denoising are presented as applications.

This is by no means an exhaustive list but simply an indication of the wide variety of applications in which scale-spaces are made use of.

4.8.1 Feature Detection in the Gaussian Scale-Space

One drawback of the Gaussian scale-space is its linearity. It removes small scale features (noise) very well but results in spatial distortions as scale increases, i.e. reduced sharpness of edges and shapes [129, 62]. To prevent this a nonlinear smoothing step is introduced in the literature, see [124, 170, 99, 63, 28, 230], and the LULU scale-space, obtained as the DPT, does the same (see [52] and [8] for the edge preserving properties of the DPT). Nonlinear filtering needs to be introduced into image analysis if realistic structures are the aim of the detection [123]. Koenderink and collaborators introduce the idea of using nonlinear, possibly, combinations of derivatives i.e. differential geometric descriptors, to introduce nonlinearity.

In [123, Chapter 6] a basic introduction into the use of the Gaussian scale-space and its scale-space derivatives for edge detection, junction (corner) detection and feature detection is presented. For edge detection the local directional derivatives are used to detect maximum gradient changes. Junction detection is obtained at high curvature combined with high gradient points . Feature detection is obtained by detecting zero-crossings and/or local extrema. Weickert et al [239] detect regions of interest as the stable stationary points in the Gaussian scale-space tree within a surrounding circular radius of appropriate radius.

4.8.2 Feature Detection in the LULU Scale-Space

With the availability of all the pulses of the Discrete Pulse Transform, the question arises as to how we can utilize all the obtained pulses to solve some of the problems encountered in image analysis? From the DPT, we no longer only have the original luminosity at each pixel, but instead have an otherwise invisible insight into the make-up and content of the image and the pixels within.

The additional image structure provided by the DPT provides improved fea-

ture detection over standard approaches using only luminosity values. This chapter will first look at the philosophy of feature detection in images and then introduce a number of techniques which utilize the DPT. The techniques investigated are rudimentary as we investigate the basis for using the DPT for feature point detection and feature detection. Advancements can be made once ability of the DPT in feature point detection as well as feature detection has been explored.

Philosophy of Feature Detection

In 1978 the ability of the human visual system (HVS) to discriminate an object in a random dot display was investigated by Barlow [13]. The aim was to determine estimates for the efficiency of the HVS to achieve this task, which Barlow refers to as absolute measures of sensory performance. He determined, albeit with a sample size of only 2, that the efficiency limit is 50% i.e. the HVS uses 50% of the data available for recognition tasks. In more obvious discrimination tasks this measured efficiency was less. Other experimenters determined similar results. Also in 1978, according to Barrow and Tenenbaum [15], the HVS easily characterizes a scene with respect to range, orientation, reflectance and incident illumination on first the first view. It contains cells which measure these individual characteristics and in a manner sums them to estimate the shape information [169]. Mishra and Jenkins [149] also designed feature extractors based on Gabor filters and motivate them with their link to detecting natural stimuli i.e. they are biologically inspired.

A feature extraction method needs to, in some manner, extract the signature of the objects in the scene. This should be done as accurately and uniquely as possible, as emphasized in [149], and the salient characteristics of the object should be measured. Salient is defined very nicely as *prominent* or *conspicuous* in the Oxford English Dictionary. A high profile paper in *Nature* [103], describes textons. These are local conspicuous features. The pre-attentive texture probing by the HVS uses these textons and first order moments for discrimination rather than higher order moments, that is, the simplest and most obvious is the most useful the descriptor. We can in fact go one step further than feature extraction and look into recognition of the feature detected. This then requires a type of classification based on the salient features we can extract from the object.

The aim of our feature detection using the DPT is to determine salient feature points of the image using the pulses in the DPT, as opposed to full features

extracted as objects (i.e. targets). Very importantly, we do not make any initial assumptions for the image regarding luminosity, amount of variation, size of objects present, texture etc., so that we can process any possible image. There does exist literature which assume a model for the image data, for example, a normal distribution [11], or a model for the way in which the image was obtained, for example when making use of the camera technique to remove illumination. In order to make our method applicable in all situations we shall ignore such ideas (although they are very useful when such assumptions are indeed true for a sample of images and thus improve the processing of the image). Of course, the interested reader could make improvements on our ideas if such assumptions are valid for their case.

How do we decide how salient a feature is? The most obvious is that large, high-contrast objects will naturally be more salient than small, low-contrast objects, in the absence of complicated backgrounds, but then at what size and contrast does the required saliency begin? In [107] this is determined by measuring the ability of the agent to draw a line around the target distinctively and they present a theory of optimal linear edge detection. According to Chi and Leung [35] humans recognize line drawings as quickly and almost as accurately as full detailed images. In addition they follow the five laws of Gestalt theory, which describe human perception of significant shape features, to set up good edge detectors. These five laws are *focal point*, that is select the top $a\%$ longest lines and arcs, *proximity*, *continuity*, *similarity* and *symmetry*, the latter four which choose the neighbours of the focal features concurrently according to these properties.

By using the edges or boundaries of an object we can also enter the field of shape analysis. Colour alone will not provide enough concrete data for detection as two vastly different objects present in an image may have the same luminosity. It is shape that represents the inherent structure of the image [50]. However, as mentioned in [69] by storing all the shape information we extract a huge amount of data from the image. There are measures available which can accurately describe a shape in a simple manner avoiding additional storage memory. This is also a very important strategy to be considered since the DPT produces a large number of pulses and thier analysis can require significant computational effort. The idea is to represent each object in the image with a feature vector, and not each pixel, thereby reducing the information that would need to be processed. Urdiales et al [227] describe some ideal properties of a feature vector, namely, uniqueness for each object, resistant to noise and as compact as possible for storage. Loncaric [130] gives a nice summary of shape analysis techniques.

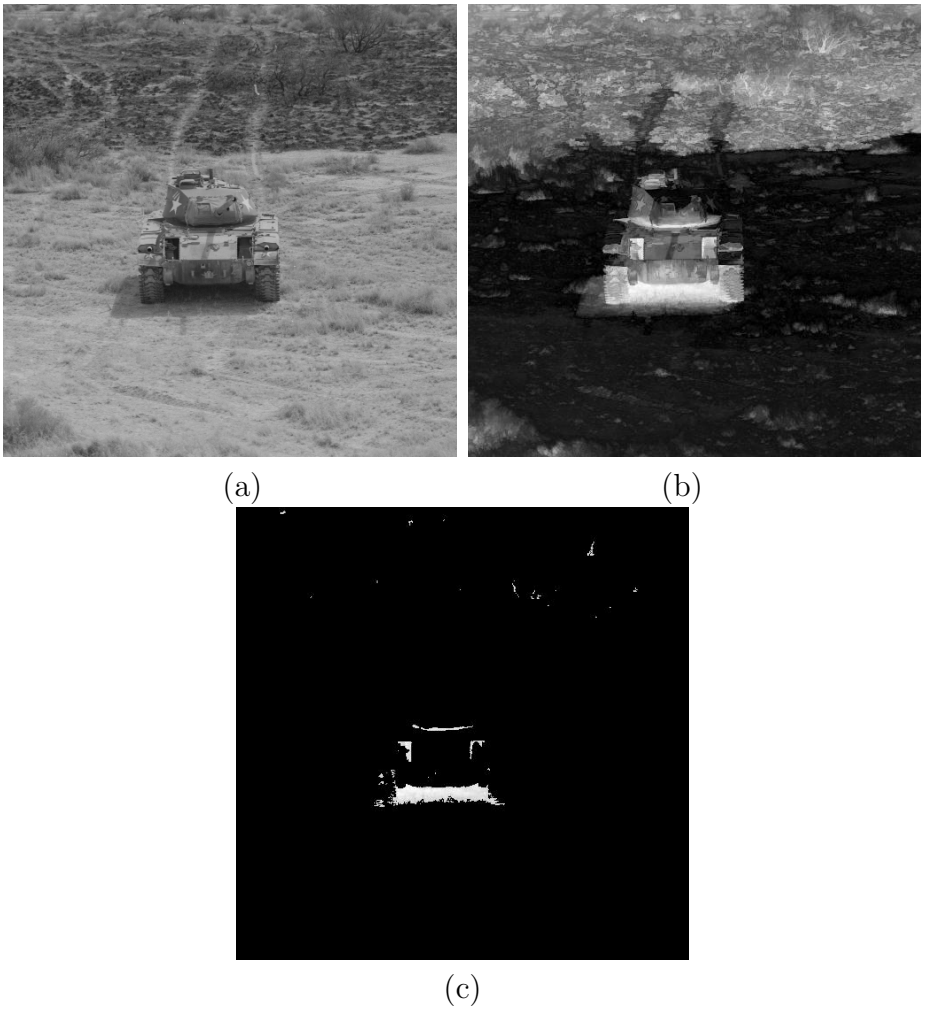
Feature Extraction

In the absence of prior knowledge about the feature characteristics and size one has to keep every scale. Reflecting on how a human eye picks out features in an image the Human Vision System (HVS) model [105] provides some insight. It consists of a first stage, the *Pre-Attentive Stage*, in which the features are detected and then a second stage, the *Attentive Stage*, in which matching takes place between the detected features of the first stage and the rest of the image. It is clear that the HVS possesses a degree of saliency to detect the 'pop-out' features. We will show how the LULU operators can be used to detect these 'pop-out' features. Following the HVS model, the features are those areas in the image that are stable, that is, the areas that survive over a wide range of scales, [129, 122]. This is simple to apply to the LULU scale-space. Indeed, each pixel belongs to k pulses, $\phi_{n_1 s_{n_1}}, \phi_{n_2 s_{n_2}}, ..., \phi_{n_k s_{n_k}}$, at scales $\{n_1, n_2, ..., n_k\} \subseteq \{1, 2, 3, ..., N\}$. For each pixel x, we then have what we call a Discrete Pulse Vector (DPV) for a specified pixel $x \in \mathbb{Z}^2$,

$$\mathbf{p}_x = \begin{bmatrix} n_1 & n_2 & n_3 & \dots & n_k \\ \ell_1 & \ell_2 & \ell_3 & \dots & \ell_k \end{bmatrix}^T, \; x \in \mathbb{Z}^2 \tag{4.8}$$

where for each scale n_i, we have the corresponding relative luminosity ℓ_i of the pulse $\phi_{n_i s_{n_i}}$, that is, the height (positive) or depth (negative) of the local maximum or minimum set which pixel x belongs to at scale n_i. The simplest and most obvious way of using these DPV's for feature detection is by keeping only those pixels belonging to DPV's that contain a large number of scales, i.e. large values of k. This is illustrated in Figure 4.11. Whiter values (higher luminosity) indicate larger values of k. In the last image only the top 20% proportion of the largest values of k and their respective pixels are kept. Notice how the front of the tank is a strong feature. We refer to the value k as the *impulse strength*. Van der Walt refers to this as the *strength of the pixel* [231].

An alternative is to consider the ranges $n_k - n_1$, referred to as the *scale-space lifetime* at the pixel [122], but this method does not differentiate between features as effectively as the first. Compare Figure 4.11 with Figure 4.12. We clearly see this measure does not pick out dominant features as well. For this method one needs to probably do some removal of outliers and cleaning of the data first. It still picks out the tank as a feature (seen in white) but the background also gets picked out and remains even if we threshold. This



(a) (b)

(c)

Figure 4.11: *(a) The original tank image with (b) its impulse strength shown, as well as (c) only the top* 20% *largest impulse strength pixels*

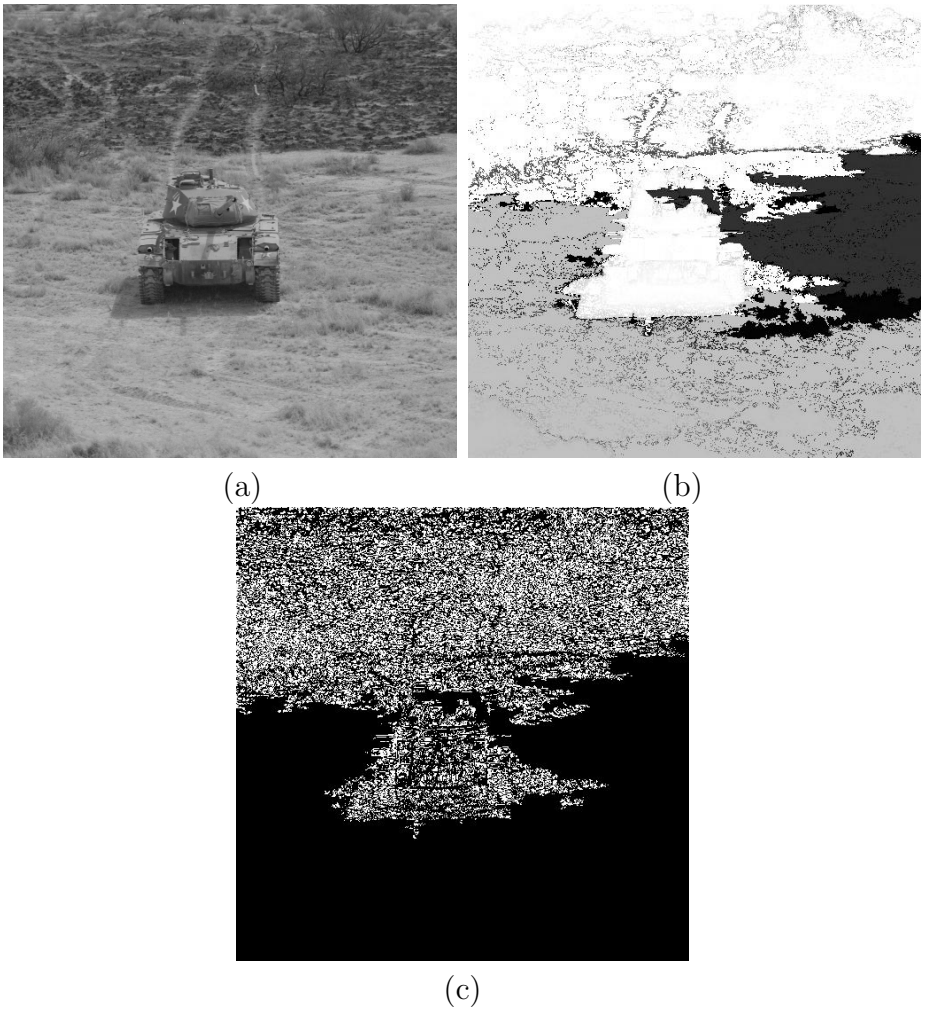


Figure 4.12: *(a) The original tank image with (b) its scale-space lifetimes shown, and (c) only the top* 0.01% *largest scale-space lifetime pixels*

can be understood logically, however, since a discrete pulse vector may be for example only of length 2 and have only a small scale and a very large scale, giving a large value for $n_k - n_1$. The pixel however does not exist over a large range of scales and should not be classified as such. The scale-space life-time may however provide an indication of whether a pixel is noise, texture, small detail, large detail etc.

As mentioned in [122] a two stage approach may be better, thus we present a method in which first the feature are detected via impulse strength mentioned above and then fine-tuned using finer scale data and shape descriptors. We present three examples to illustrate this idea. Further research is currently

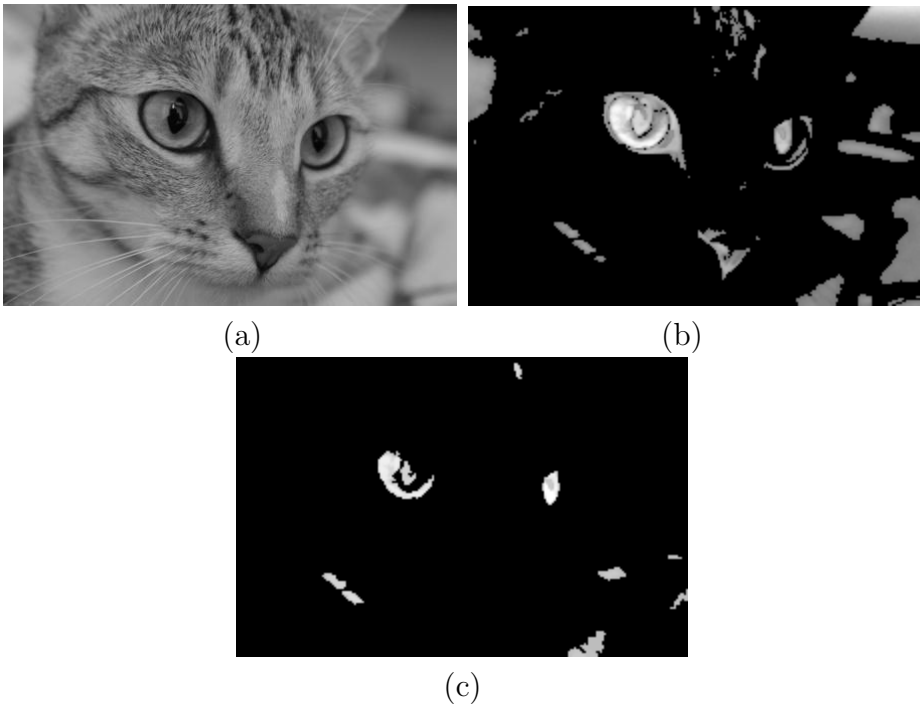


Figure 4.13: *Pulse strength illustrated on (a). (b) Only pixels included in at least 59 out of the 126 possible pulse scales are shown. (c) The circularity of the pulses used in (b) is restricted to between 0.3 and 0.6 to extract the eyes of the cat.*

being conducted to perfect this technique.

In Figure 4.13, we keep only the pixels which have discrete pulse vectors with at least 59 scales out of the maximum of 126 over all the pixels. We can see that the cat's two eyes and nose are picked out as features. We also see that some large background pulses are detected as features. These large noise pulses can be filtered out with a circularity shape descriptor. A circularity value close to 1 then indicates higher circularity than a value closer to 0.

In Figure 4.14, we keep only the pixels that have discrete pulse vectors with at least 65 scales out of a maximum of 105 over all the pixels. The three vehicles are detected as features. In Figure 4.15, we first remove the glint on the surface of the ocean by limiting the luminosity of the individual pulses. The third image in Figure 4.15 indicates the impulse strength of the image with the glint removed. The two main features in this image are the yacht and surprisingly the atmospheric mist effect on the land sea border. Atmospheric conditions often affect feature detection in marine images. In addition, when

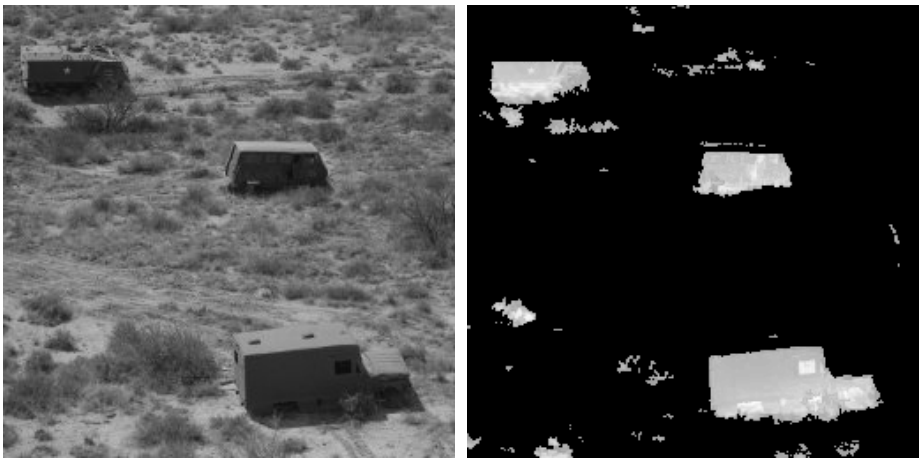


Figure 4.14: *Impulse strengths illustrated on* Trucks & APCs.

we talk about features we refer to those 'objects' which 'pop-out' first, and indeed when looking at the image in Figure 4.15 the yacht and the mist stand out first. Notice also though that the small boat in the background is also picked out via the impulse strength, even though it is very dark and almost camouflaged into the water. We can filter out the effect of the mist, and other effects, by using only pulses with specific areas, see the fourth image in Figure 4.15. The small boat is picked out in a similar manner.

These examples give an overview of the capability of the DPT for feature detection.

4.8.3 Segmentation in the LULU Scale-Space

Segmentation is the process of partitioning an image or signal into segments which provide a simpler representation more indicative of the image content with respect to visual characteristics. Serra provides a formal definition in terms of partitions and connectivity in [213]. It is immediately obvious that different segmentations could be obtained by using different measures for the similarity of image content. An obtained segmentation may be over-segmented meaning there exist some pairs of regions for which the between-region variation is small compared to the within-region variation, so that there are too many regions in the segmentation, [57]. In [163] a connection is presented which can be used in place of the usual image connectivity to avoid over-segmentation. An image may also be under-segmented meaning there

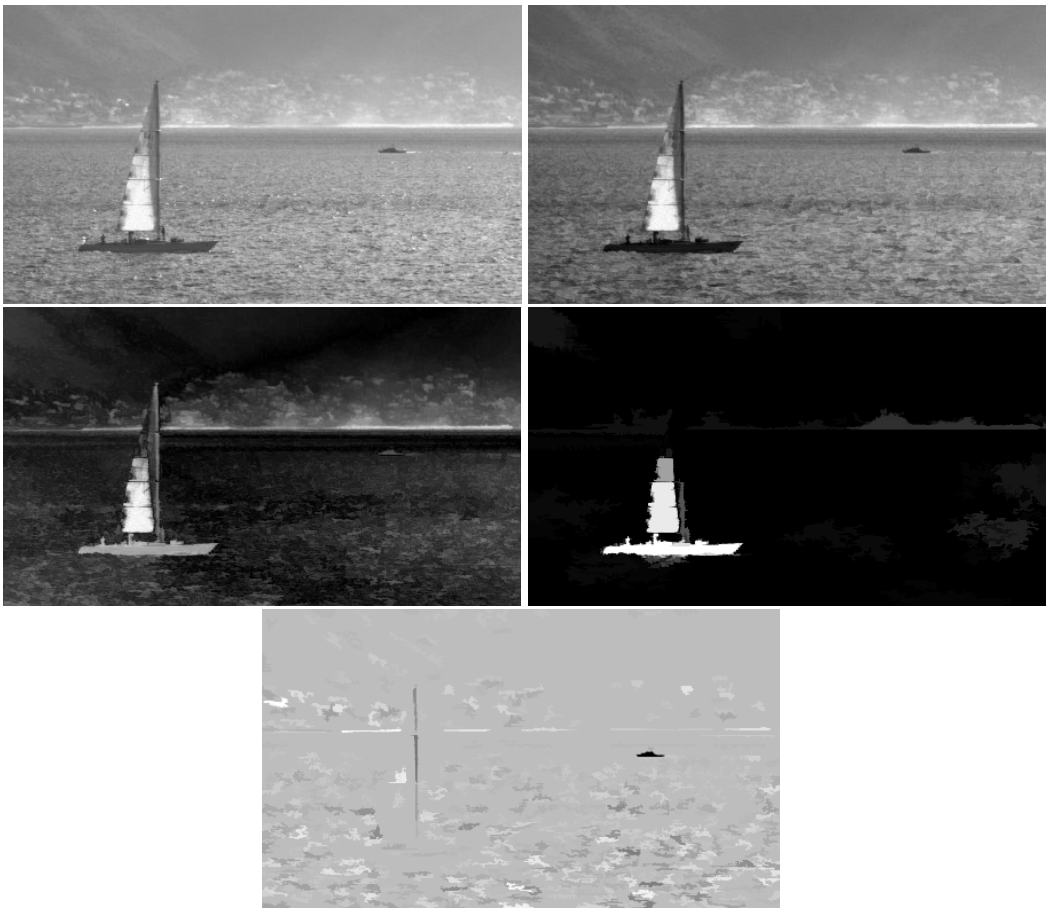


Figure 4.15: *Impulse strengths illustrated on* yacht.

exists a way to further segment regions without causing over-segmentation so that there are too few regions in the segmentation [57]. Felzenszwalb and Huttenlocher [57] and Hoover et al [82] provide examples of graph-theoretical and set-theoretic definitions for over- and under-segmentation respectively. Zhang discusses techniques to measure the quality of a segmentation, namely analytical, which involves looking at the actual algorithm and its properties, most importantly convergence properties; empirical goodness, which is based on human perception of a good segmentation; and empirical discrepancy, which involves a comparison with the ground truth segmentation, if available [253].

Algorithms for segmentation may take into account a priori information about an image. Methods like this, when the features in the image are known, are called supervised segmentation . Methods may also be semi-supervised in which case features are only partially known. Unsupervised methods assume no knowledge of the image features and learn as the algorithm proceeds. Song and Fan [218] present a study on the different techniques based on the availability of image features. The number of segments an algorithm should aim for is also a problem which has been given attention. If this is known before hand it can be specified. Other methods determine this as the algorithm proceeds. For example, Nakamura and Kehtarnavaz [155] provide a method to determine the appropriate number of clusters by making use of scale-space theory in which a prominent data structure is one which survives over many scales and Sakai and Imiya [195] use the modes of a probability density function obtained via the Gaussian scale-space for cluster discovery.

As discussed in detail in Chapter 4, the human vision system has a big effect on the philosophy of imaging techniques, and this is true for segmentation as well. Zahn segments into Gestalt clusters which are those perceived by humans [251]. Ramos et al base a segmentation into strong edges, smooth regions and textured regions on psychophysical studies [176]. Leung et al present a clustering by using scale-space's to simulate the human visual system [115].

Segmentation using connected operators has proved very effective. Salembier and Serra [200] argue for the use of filters by reconstruction since they simplify the image while preserving contours and are thus good for noise cancelation and improved segmentation. In [215] the same authors use pyramids of nested flat zones based on connected operators. This also provides good segmentation since simplified into flat zones and preserves contour information. Soille [217] goes further and deals with a constrained connectivity for

which 2 pixels are connected if they satisfy a series of constraints in terms of simple measures. He uses this connectivity for segmentation.

Using the pulses of the DPT we obtain improved segmentation. Each pixel in the image belongs to a number of pulses in the DPT but not at every scale. We associate a *Discrete Pulse Vector* (DPV) with each pixel, namely

$$DPV(x) = \begin{pmatrix} s_1^{(x)} & s_2^{(x)} & ... & s_m^{(x)} \\ \ell_1^{(x)} & \ell_2^{(x)} & ... & \ell_m^{(x)} \end{pmatrix},$$

where the $s_i^{(x)}$'s for $i = 1, 2, ..., m$ are the scales at which pixel x appears in a pulse and the the relative luminosities $\ell_i^{(x)}$'s for $i = 1, 2, ..., m$ are the respective heights or depths of the pulse at that scale containing x. Various pixels may be present in a large number of scales resulting in very large DPV's as well as DPV's of different lengths so the DPV's cannot be clustered directly. This information needs to be summarized into only a few values in order for each pixel to be clustered using the algorithm. We investigated using the following possible summarizing measures,

- $\sum_{i=1}^{m} |\ell_i|$
- $\sum_{i=1}^{m} |\ell_i| s_i$
- $\sum_{i=1}^{m} \ell_i s_i$
- $\sum_{i=1}^{m} \ell_i \sqrt{s_i}$
- $\sum_{i=1}^{m} (\ell_i)^2 s_i$.

The investigations indicate that $\sum_{i=1}^{m} |\ell_i|$ performs best in representing the content of the image obtained from the DPT and we use this measure throughout. The measure can in addition be broken into bands

$$\sum_{i=1}^{m_1} |\ell_i|, \sum_{i=m_1+1}^{m_2} |\ell_i|, ..., \sum_{i=m_n+1}^{m} |\ell_i| \tag{4.9}$$

and a vector clustering algorithm applied. As long as the number of bands is not too large this is fairly simple and provides better segmentations.

We make use of the FCM algorithm for initial illustrations. We present some examples in Figures 4.16 to 4.20. FCM is an alternative to the standard k-means algorithm and incorporates a degree of fuzziness with respect to

the cluster assignments, as opposed to the hard clustering of the k-means algorithm where each observation can belong to only one cluster. Duda and Canty [48] compare a number of algorithms and conclude that fuzzy association works the best. Each observation x_p is assigned a coefficient $w_i(x_p)$ representing the degree of association of x_p with cluster i such that $\sum_{i=1}^{c} w_i(x_p) = 1$ for each x_p. A larger coefficient indicates a better strength of association with that respective cluster. The centers are calculated as

$$\mu_i = \frac{\sum_{x_p} w_i(x_p)^m x_p}{\sum_{x_p} w_i(x_p)^m}$$

where m is the fuzzy exponent (usually taken as 2) and the coefficients updated as the inverse distance from the observation to the cluster

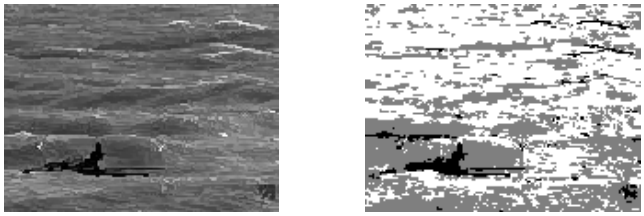
$$w_i(x_p) = \left(\sum_{j=1}^{c} \left(\frac{\|\mu_i - x_p\|}{\|\mu_j - x_p\|} \right)^{2/(m-1)} \right)^{-1}.$$

Convergence of the fuzzy c-means (FCM) algorithm is obtained when the coefficients no longer change significantly. The final segmentation is obtained by assigning observations to the cluster i for which w_i is the largest of the coefficients for that observation. The FCM algorithm results in similar poor segmentation sometimes. Gath and Geva [70] provide an unsupervised FCM algorithm which determines the number of clusters as it proceeds. Xie and Beni [249] introduce a validity measure for the FCM clusters. Krishnapuram and Keller [110] compare fuzzy and hard k-means with possibilistic clustering since the former two encounter trouble in noisy environments. Possibilistic clustering softens the requirements on the fuzzy coefficients such that $\sum_{i=1}^{k} w_i(x_p) \leq 1$ for each x_p. Pal et al [165] also include a possibilistic element to the algorithm to improve its effect on noise. Hammah and Currah [75] look at using different distance measures and how they affect the algorithm. They also introduce a new measure based on the Kent probability distribution.

In Figure 4.16 the improved segmentation using the LULU scale-space is shown. In Figure 4.17 the same is illustrated on the sharpened image (see Section 5.2) giving similar results, but in fact the segmentation appears worse. This can be attributed to the low resolution of the image. We include it none-the-less as it does provide insight into the cluster scale distributions. Figure 4.18 shows the distributions of the scales represented by the three clusters of Figure 4.16(b). Although similar there are distinct difference too. The black cluster, for example, represents more smaller scales than larger scales, as opposed to the white cluster.

(a) (b) (c)

Figure 4.16: *(a) Original Image, (b) Clustering the DPT into 3 clusters, (c) Ordinary FCM with 3 clusters*



(a) (b)

Figure 4.17: *(a) Sharpened Original Image, (b) Clustering the DPT into 3 clusters*

The fact that the DPT provides us with pulses of every size also allows us to remove certain scales before segmentation. Figure 4.19 illustrates this idea on the previous example. Only pulses larger then 150 are used in the segmentation. The result is a very sound segmentation. The scale distributions amongst the 3 clusters are given in Figure 4.20 yielding similar results.

The ICM clustering algorithm presented in [46] is effective yet simple enough to illustrate improved segmentation as well. The ICM algorithm follows. Notice that k-means is used as an initial step for the algorithm providing even better segmentation.

ITERATED CONDITIONAL MODES ALGORITHM

For a segmentation of an image I with N pixels represented by (i, j), and given feature vectors f_{ij} for each pixel, into K clusters $C_1^{(\alpha)}, C_2^{(\alpha)}, ..., C_K^{(\alpha)}$ where α is the number of iterations the steps proceed as follows:

1. Use the k-means algorithm to obtain initial cluster mean vectors $\mu_k^{(0)}$ for clusters $k = 1, 2, 3, ..., K$.

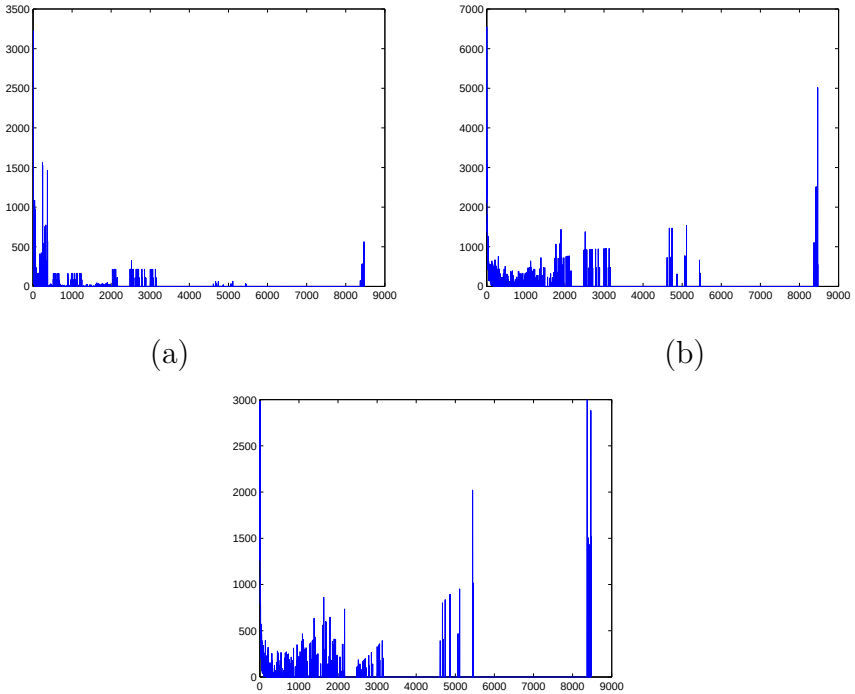


Figure 4.18: *Scale distributions within each cluster of Figure 4.16(b) (a) Black Cluster, (b) Grey Cluster, (c) White Cluster*

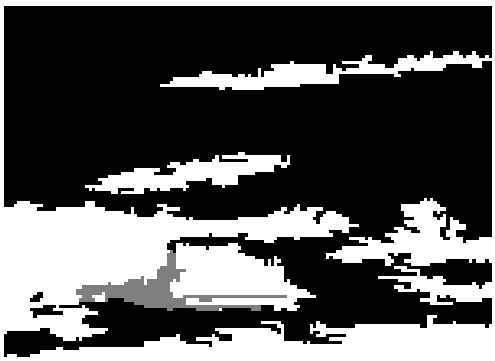
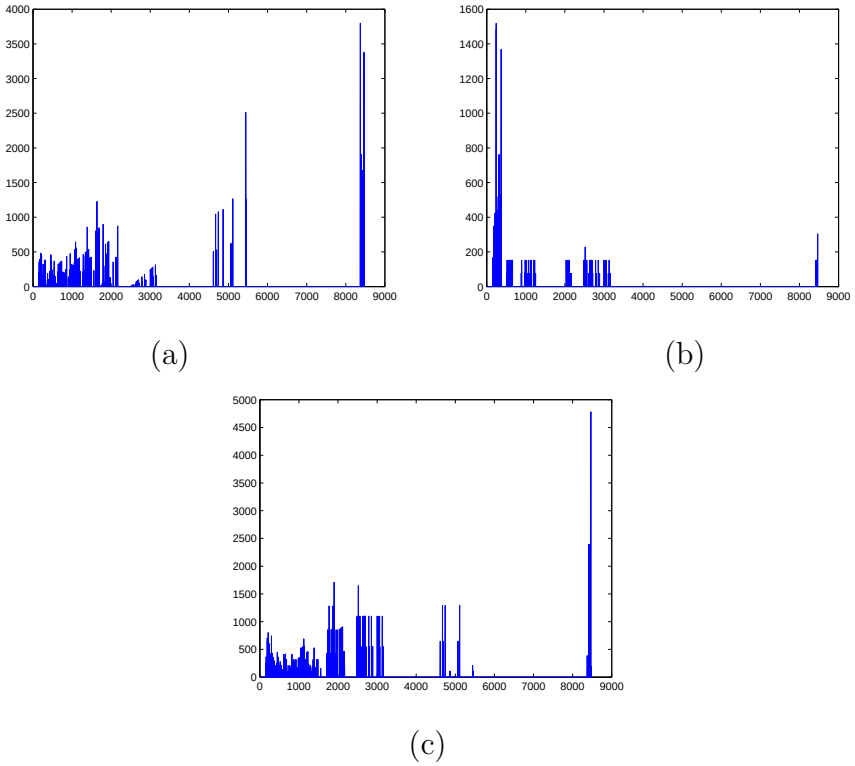


Figure 4.19: *Clustering of the sharp DPT into 3 clusters using only pulses larger than 150 i.e.* $f_{new} = \sum_{n=150}^{N} D_n(f)$



(a) (b)

(c)

Figure 4.20: *Scale distributions within each cluster of Figure 4.19 (a) Black Cluster, (b) Grey Cluster, (c) White Cluster*

2. Assign pixel (i, j) to cluster k for which the minimum of

$$\left(f_{ij} - \mu_k^{(\alpha)}\right)^T \left(f_{ij} - \mu_k^{(\alpha)}\right) - \beta\nu^{(\alpha)} N_{ij}^{(\alpha)}(k)$$

is obtained, where

- β is a spatial penalization parameter (suggested as 1.5 in [46]),
- $\nu^{(\alpha)} = \frac{1}{N}\sum_{k=1}^{N}\sum_{(i,j)\in C_k^{(\alpha)}} \left(f_{ij} - \mu_k^{(\alpha)}\right)^T \left(f_{ij} - \mu_k^{(\alpha)}\right)$ is the within cluster variance, and
- $N_{ij}^{(\alpha)}(k)$ is the number of neighbours of pixel (i, j) currently classified in cluster k at iteration α.

3. Recalculate the cluster mean vectors

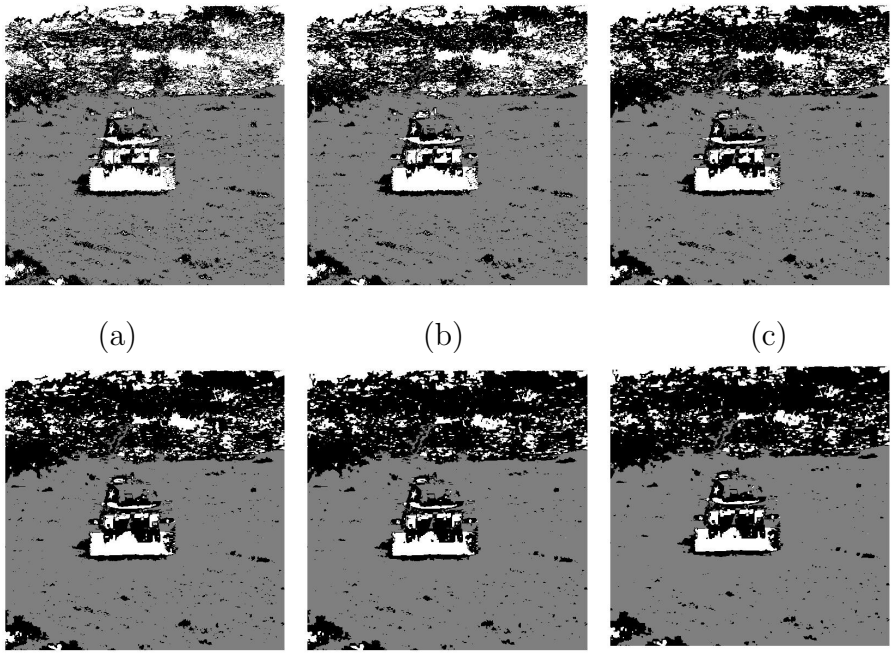
$$\mu_k^{(\alpha)} = \frac{1}{N_k^{(\alpha)}} \sum_{(i,j)\in C_k^{(\alpha)}} f_{ij}.$$

4. Repeat steps 2 and 3 until convergence (no change).

We illustrate the effect of β in Figure 4.21. Notice how the regions in the image are more smoothed with less detail as β increases.

We repeat the ICM segmentation on the image used to illustrate the k-means algorithm in Figures 4.22 to 4.24 . Figure 4.22 shows the ICM algorithm applied to the original image without the use of the DPT. Notice the improvement over the k-means results already in the 3 cluster segmentation. In Figure 4.23 the segmentation is done with the DPT. There doesn't seem to be a huge improvement and in fact the segmentation requires 5 clusters now to pick up the canoeist effectively. However, in Figure 4.24 the segmentation is done again using the DPT but only pulses larger than 100, as was discussed in Chapter 4.8.2 as significant structures are very unlikely to be this small (depending on the total image size of course). The canoeist is picked out in the 3-, 4- and 5-cluster segmentation in this case. It is not surprising the 2-cluster segmentation cannot pick the canoeist out as there are clearly three patterns in the image - the dark water, white water and the canoeist, thus the canoeist will end up being classified with one of the water patterns.

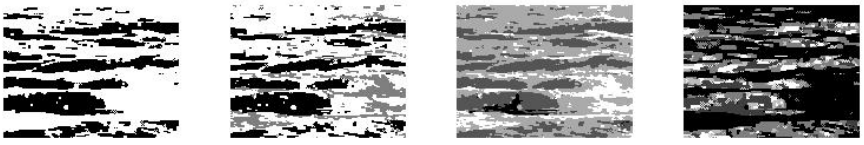
We now look at the Tank image introduced in Figure 4.21. Figure 4.25 presents the ICM segmentation of the Tank image without using the DPT.



(a) (b) (c)

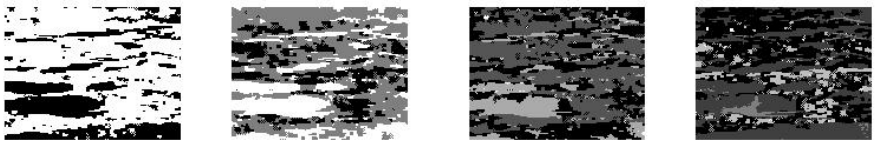
(d) (e) (f)

Figure 4.21: *The effect of parameter β in the ICM algorithm illustrated on the Tank image clustered into 3 clusters (a) $\beta = 0.1$ (b) $\beta = 0.5$ (c) $\beta = 1$ (d) $\beta = 1.5$ (e) $\beta = 2$ (f) $\beta = 2.5$*



(a) (b) (c) (d)

Figure 4.22: *ICM segmentation illustrated on the Canoeist for (a) 2 (b) 3 (c) 4 (d) 5, clusters*



(a) (b) (c) (d)

Figure 4.23: *ICM segmentation of the Canoeist image using the DPT into (a) 2 (b) 3 (c) 4 (d) 5, clusters*

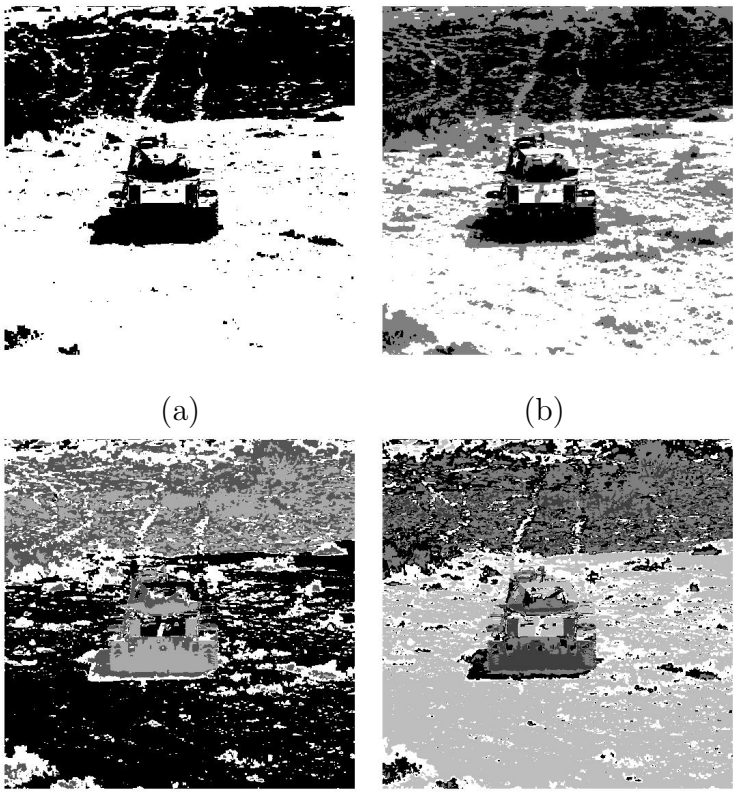
(a) (b) (c) (d)

Figure 4.24: *ICM segmentation of the Canoeist image using the DPT and only pulses 100 and larger into (a) 2 (b) 3 (c) 4 (d) 5, clusters*

Notice has 'messy' the segmentations are - the clusters are not easily discernable. Figure 4.26 shows the improved segmentation using the DPT. The segmentations are more clear.

Since the DPT provides us with all the scale information and the ICM algorithm can be vectorized, further improved segmentation may be obtained by separating the $|\ell_i|$ into bands indicated in (4.9). Figure 4.27 shows this method by separating the number of scales in half, the lower half representing the smaller scales and the upper half the larger scales. Notice that in the vector segmentation on (a) already shows an improvement over Figure 4.26(b). In Figure 4.27(b) the algorithm does not converge as there are not significantly different patterns in the information provided by the lower scales. Figure 4.27(c) also provides better segmentation - the background grass segments more consistently than before. Figure 4.28 shows the segmentation by separating the scales into three bands. By applying the ICM algorithm to the lower and middle band individually doesn't result in convergence and are thus not included.

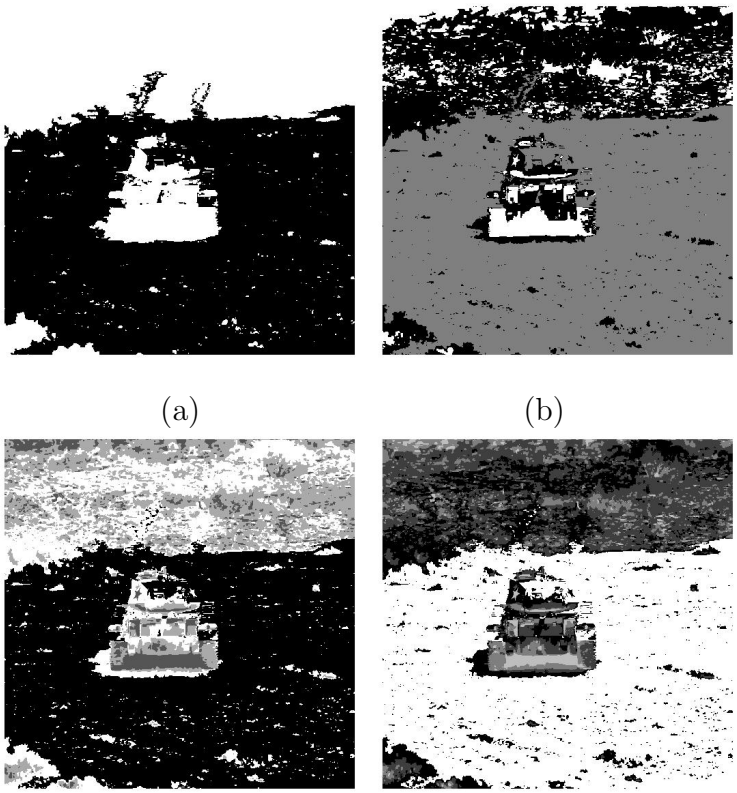
By using the total variation spectrum we can improve the grouping used above. Figure 4.29 shows the variation spectrum [52] for the *Tank* image. There seem to be five distinct bands of total variation, namely, 1 - 30000,



(a) (b)

(c) (d)

Figure 4.25: *ICM segmentation of the Tank image into (a) 2 (b) 3 (c) 4 (d) 5, clusters*



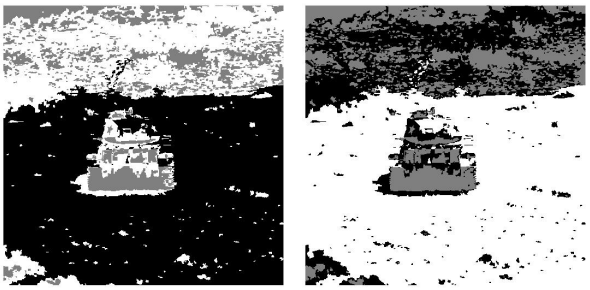
(a) (b)

(c) (d)

Figure 4.26: *ICM segmentation using the DPT of the Tank image into (a) 2 (b) 3 (c) 4 (d) 5, clusters*

(a) (b) (c)

Figure 4.27: *ICM segmentation of the Tank image into three clusters using the DPT separated into two bands (a) both bands clusters (b) lower band clustered (c) upper band clustered*



(a) (b)

Figure 4.28: *ICM segmentation of the Tank image into three clusters using the DPT separated into three bands (a) both bands clustered (b) upper band clustered*

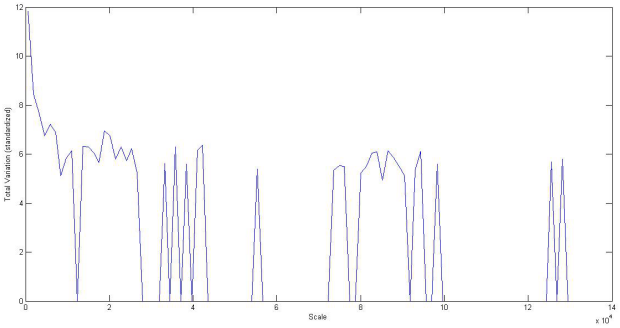


Figure 4.29: *Total Variation Spectrum of the Tank Image*

30001 - 50000, 50001 - 70000, 70001 - 120000, 120001 - 130139. The result of vectorized ICM segmentation using the total variation spectrum is shown in Figure 4.30. The individual segmentations of bands 1 - 30000, 30001 - 50000 and 50001 - 70000 do not converge illustrating the information within each of these bands has low variation. Figure 4.30(a) and (e) present the best segmentations by picking out the two different background grass shades, some significant features in the grass, as well as the tank with its different features.

In Chapter 4.8.2 it was discussed that pulses of size larger than 100 should be used for feature detection as significant structures are very likely to be smaller (depending on the total image size of course). Incorporating this into segmentation gives the results in Figure 4.31. We combine this idea with the total variation spectrum and investigate bands 100 - 30000, 30001 - 50000, 50001 - 70000, 70001 - 120000, 120001 - 130139. The result is shown in Figure 4.31(e) - notice it is very similar to (b).

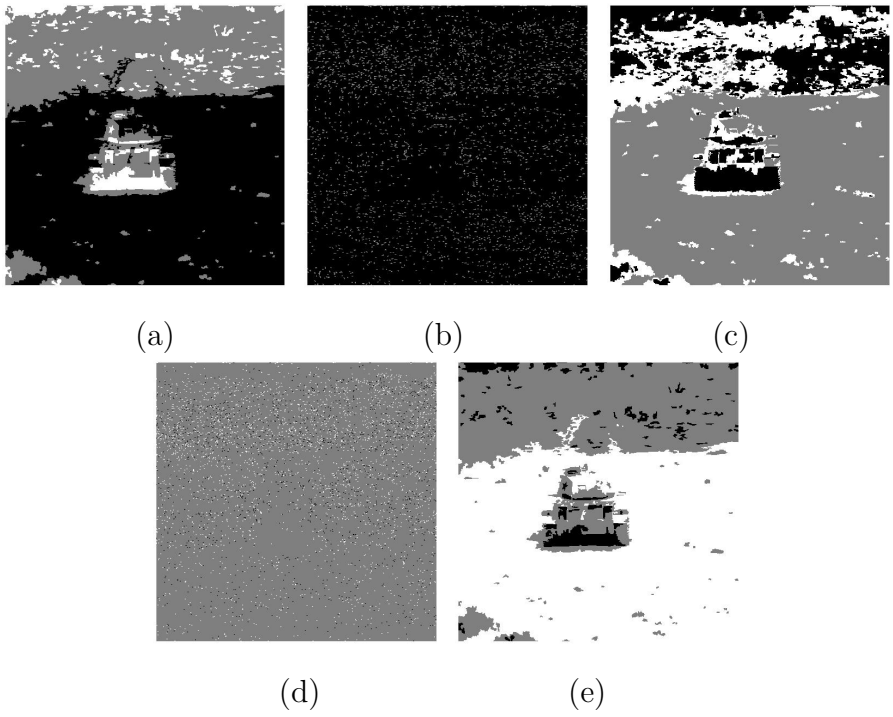


Figure 4.30: *ICM segmentation of the Tank image into three clusters using the DPT and the total variation spectrum shown in Figure 4.29 (a) all 5 TV bands (b) scales 70001 - 120000 (c) scales 120001 - 130139 (d) Scales 1 - 70000 (e) Scales 70001 - 130139*

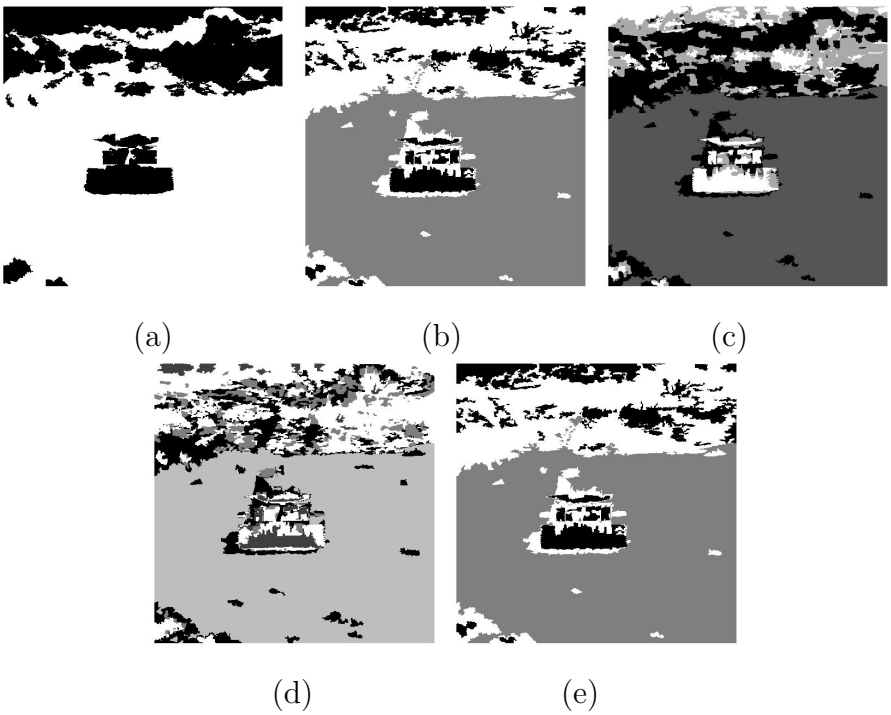


Figure 4.31: *The ICM algorithm illustrated on the Tank image using only pulses of size 100 and larger (a) 2 clusters (b) 3 clusters (c) 4 clusters (d) 5 clusters (e) 3 clusters using the TV bands*

The results presented in this section are by no means the end-all of segmentation with the DPT and have been presented as an indication of the usefulness of the DPT in image segmentation. Future research will involve making use of the LULU scale-space to determine the number of clusters beforehand as Nakamura and Kehtarnavaz [155] do with the Gaussian scale-space; use the scale-space life-times for the segmentation as these may clearly distinguish noise, texture, small detail and large detail; look at alternative connectivity approaches for improved segmentation such as the work done by Soille in [217]; comparisons with state-of-the-art segmentation; and using the shape measures, such as the shape number and shape dispersion matrix, discussed in detail in Section 4.8.2 to obtain further improved segmentation as Urdiales et al [227] do. This last approach ventures into the realm of pattern recognition which will enable the modeling of backgrounds in images and the subsequent removal of them for accurate target detection and tracking.

4.9 Conclusion

In this chapter we have presented an overview of the development of the original Gaussian scale-space of Witkin and Iijima, further works resulting from this, as well the various pre-scale-space notions of introducing scale into analysis of signals and images. We also briefly listed the numerous applications of scale-spaces in image analysis. Most importantly, we provided a formal definition of a scale-space (Section 4.6), which has not been done to our knowledge. The Discrete Pulse Transform results in a scale-space, named the LULU scale-space, according to this definition and we prove this in Section 4.7. The opportunity to investigate the practical use of the LULU scale-space is now available and we delve into this in Sections 4.8.2 and 4.8.3.

Chapter 5

Improving Image Quality

5.1 Introduction

The first step in any analysis of data involves some mechanism of assessing the quality of the data. Any discrepancies or irregularities may result in poor results from the analysis. Image analysis is no different and is simply a special case of data analysis. A low resolution, noisy or blurry image, for example, may result in insignificant, inconclusive or false results from many image techniques such as feature detection, object detection, edge detection and segmentation to name a few. It is thus essential to have the best version of an image available before further analysis is done on it.

In this chapter we look in Section 5.2 at sharpening an image and the effect this has on the resulting DPT of the image. This entails an improvement in the crispness of edges in an image. The edges in an image discriminate its content and should be as clear as possible therefore. We also look at measuring the quality of the approximation of an image via the DPT in Section 5.3. This work was published in [6]. We then investigate the ability of the LULU operators to remove added noise from differently shaped statistical distributions in one dimension as well as two dimensions. The one-dimensional noise removal investigation work was published in [56].

5.2 Sharpening

Sharpening an image entails improving the crispness and clarity of the detail in an image, without, importantly, adding any new detail. This is achieved by enhancing or deblurring the edges. Ironically, sharpening involves first blurring the image via some mechanism, then comparing each pixel with its blurred counterpart. If the pixel luminosity is higher then then its luminosity is lowered accordingly and if it is lower, raised accordingly, resulting in an increase in contrast between pixels and thus a sharpening of the image.

In [204] a morphological approach to sharpening is presented. We shall use this method in this section. The sharpening operator is define as follows,

Definition 39 *The* image sharpening operator class *[204] is*

$$\epsilon_\rho(f)(x) = \begin{cases} F_\rho^{\oplus}(x) & \text{if } F_\rho^{\oplus}(x) - f(x) < f(x) - F_\rho^{\ominus}(x) \\ F_\rho^{\ominus}(x) & \text{if } F_\rho^{\oplus}(x) - f(x) > f(x) - F_\rho^{\ominus}(x) \\ f(x) & \text{otherwise} \end{cases}$$

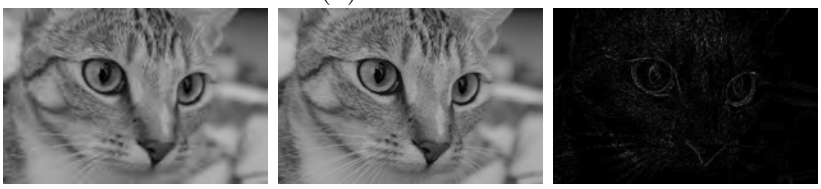
where $F_\rho^{\oplus}$ is a grey-scale dilation of f and $F_\rho^{\ominus}$ a grey-scale erosion of f by a scalable structuring function g^ρ (element c^ρ) where ρ is the size of the structuring function (element).

A scalable structuring function (element) can be obtained from a structuring function g (element c) by umbral scaling, namely, $g^\rho(x) = \rho g\big(x/\rho\big)$ $(c^\rho(x) = \rho c\big(x/\rho\big))$.

We shall look at the effect of sharpening on the DPT of an image f when using a quadratic structuring function $g^\rho(x) = -1/(2\rho)x^T x$. We used $\rho = 2$ in the investigation below. Figures 5.1 and 5.2 show some sample images with their corresponding sharpened counterpart as well as the differences between the two. As can be seen from the last column in the two figures the differences are subtle but mainly occur at edges, except for the images with noise or obvious detail. For example, in Image 8 in Figure 5.2 the far right column of bricks has more detail than the other columns and this detail can be seen more clearly in the sharpened image. Tables 4.1 - 4.3 provide quantitative data on the number of pulses in the DPT for the original and sharpened images. In the tables, column p is the proportion of pulses at scale n in the original image, *psharp* is the proportion of pulses at scale n in the sharpened image, c is the number of pulses at scale n in the original image, and *csharp* is the number of pulses at scale n in the sharpened image.

Notice that the images which include obvious background patterns, namely Images 4, 6, 7 and 9, are those that benefit from the sharpening in that they have a reduced number of total pulses. The sharpening 'cleans' up the patterns in these images and results in fewer pulses associated with the detail of the image. In addition, sharpening results in a significant reduction in the number of pulses of size 1 and 2 in all cases (except Image 7, but this is only a slight increase and only for size 1). This is an important result since pulses of size 1 and 2 make up between 30 and 50 percent of the total number of pulses and thus allows a mechanism for easier implementation of the DPT algorithmically.

(1) 336×381



(2) 300×451

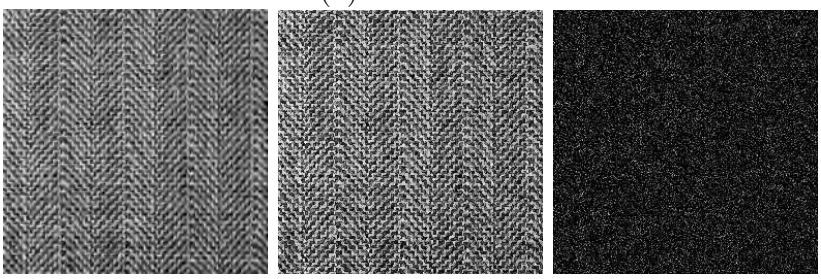
(3) 256×256

(4) 225×300

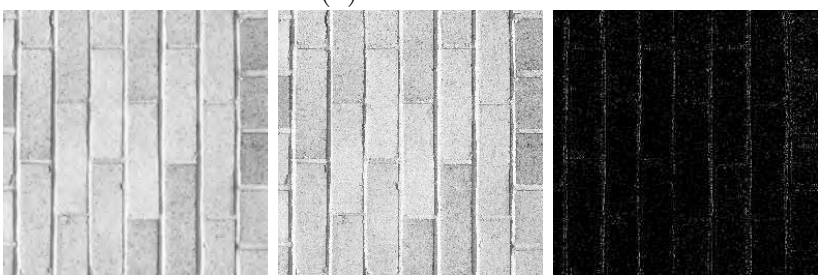
Figure 5.1: *First column: original images, middle column: sharpened images, last column: difference between original and sharpened images*

(5) 261 × 453

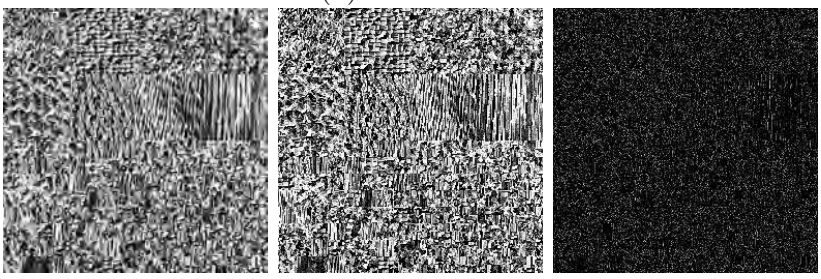
(6) 512 × 512



(7) 256 × 256



(8) 256 × 256



(9) 256 × 256

Figure 5.2: *First column: original images, middle column: sharpened images, last column: difference between original and sharpened images*

Table 5.1: *Quantitative Data for the Original and Sharp DPTs of the images in Figures 5.1 and 5.2*

Image 1				
n	**p**	**psharp**	**c**	**csharp**
1	0.4477916	0.3741685	18685	22443
2	0.1668224	0.1258732	6961	7550
3	0.0817217	0.0934796	3410	5607
4	0.0533947	0.0675714	2228	4053
5	0.0343902	0.0434804	1435	2608
6	0.026266	0.0327937	1096	1967
7	0.019316	0.0244577	806	1467
8	0.0146907	0.0185059	613	1110
9	0.011767	0.0153715	491	922
10	0.0093944	0.0130041	392	780
Totals	**0.8655547**	**0.8087061**	**36117**	**48507**
Image 2				
n	**p**	**psharp**	**c**	**csharp**
1	0.270031	0.2537643	8786	11831
2	0.1523804	0.0733345	4958	3419
3	0.0856871	0.0642615	2788	2996
4	0.0582107	0.0607439	1894	2832
5	0.039094	0.0423191	1272	1973
6	0.0311953	0.0320879	1015	1496
7	0.0252635	0.0277337	822	1293
8	0.0212681	0.0244091	692	1138
9	0.0171497	0.0185535	558	865
10	0.0151827	0.0167946	494	783
Totals	**0.7154624**	**0.614002**	**23279**	**28626**
Image 3				
n	**p**	**psharp**	**c**	**csharp**
1	0.4355907	0.3753071	14601	15278
2	0.1485084	0.1084308	4978	4414
3	0.0833831	0.0887786	2795	3614
4	0.0530131	0.0637467	1777	2595
5	0.0347255	0.0441437	1164	1797
6	0.0273568	0.0335806	917	1367
7	0.0219869	0.0259408	737	1056
8	0.0170644	0.021298	572	867
9	0.01429	0.0160902	479	655
10	0.012679	0.0140267	425	571
Totals	**0.8485979**	**0.7913432**	**28445**	**32214**

Table 5.2: *Quantitative Data for the Original and Sharp DPTs of the images in Figures 5.1 and 5.2*

Image 4

n	p	psharp	c	csharp
1	0.3826489	0.3569501	16540	14550
2	0.1410295	0.0981061	6096	3999
3	0.0792597	0.0874098	3426	3563
4	0.0548294	0.0661155	2370	2695
5	0.0389589	0.0446494	1684	1820
6	0.0296819	0.0330455	1283	1347
7	0.0226258	0.0255385	978	1041
8	0.0197571	0.0231098	854	942
9	0.015616	0.0185222	675	755
10	0.0130249	0.0152102	563	620
Totals	0.797432	0.7686571	34469	31332

Image 5

n	p	psharp	c	csharp
1	0.2952464	0.2591808	13801	13953
2	0.1611116	0.0793164	7531	4270
3	0.0957128	0.0885669	4474	4768
4	0.0664898	0.081434	3108	4384
5	0.048648	0.0545742	2274	2938
6	0.0366036	0.0459738	1711	2475
7	0.0287096	0.0353488	1342	1903
8	0.0240459	0.0303334	1124	1633
9	0.0193608	0.024185	905	1302
10	0.0167722	0.0208043	784	1120
Totals	**0.7927007**	**0.7197177**	**37054**	**38746**

Image 6

n	p	psharp	c	csharp
1	0.4199563	0.3677711	49150	56077
2	0.1394186	0.1012867	16317	15444
3	0.076925	0.0811199	9003	12369
4	0.0500103	0.0607235	5853	9259
5	0.0361171	0.0420061	4227	6405
6	0.0272395	0.0309291	3188	4716
7	0.0209765	0.0247249	2455	3770
8	0.0175929	0.0198389	2059	3025
9	0.0143802	0.016422	1683	2504
10	0.0120988	0.0135823	1416	2071
Totals	**0.8147151**	**0.7584045**	**95351**	**115640**

Table 5.3: *Quantitative Data for the Original and Sharp DPTs of the images in Figures 5.1 and 5.2*

Image 7

n	p	psharp	c	csharp
1	0.2615815	0.2744027	14393	12909
2	0.1005761	0.0455744	5534	2144
3	0.0713702	0.0592424	3927	2787
4	0.0549225	0.0638551	3022	3004
5	0.0439816	0.0474875	2420	2234
6	0.0350581	0.0381558	1929	1795
7	0.0301692	0.0338832	1660	1594
8	0.0265525	0.0292067	1461	1374
9	0.0226996	0.0252955	1249	1190
10	0.0202097	0.02368	1112	1114
Totals	**0.667121**	**0.6407831**	**36707**	**30145**

Image 8

n	p	psharp	c	csharp
1	0.4463266	0.4074106	20157	18879
2	0.1488198	0.1070157	6721	4959
3	0.0842301	0.0851551	3804	3946
4	0.0529649	0.061827	2392	2865
5	0.0371551	0.0405058	1678	1877
6	0.0266817	0.030169	1205	1398
7	0.0201718	0.0236949	911	1098
8	0.0159869	0.0189689	722	879
9	0.0136841	0.0145881	618	676
10	0.0104513	0.0123654	472	573
Totals	**0.8564723**	**0.8017005**	**38680**	**37150**

Image 9

n	p	psharp	c	csharp
1	0.3020252	0.2851626	15107	12775
2	0.1208541	0.0600013	6045	2688
3	0.0775705	0.0692873	3880	3104
4	0.0548592	0.0655595	2744	2937
5	0.0419241	0.0453805	2097	2033
6	0.0340271	0.0359606	1702	1611
7	0.0284692	0.0308043	1424	1380
8	0.0246306	0.0257149	1232	1152
9	0.0208321	0.022188	1042	994
10	0.0180132	0.0190852	901	855
Totals	0.7232052	0.6591442	36174	29529

5.3 Best Approximation

Often filters are defined by requiring proximity in some sense to the original input, e.g. see [223]. In comparison, the LULU operators and, in fact, morphological filters in general, are focused on shape and do not use distance and proximity in their definition. Nevertheless, it turns out that the LULU operators provide in some sense 'near best' approximations by functions of certain kind of local monotonicity. This result which is also the main contribution of this section extends an earlier result in [181] for LULU operators on sequences. The work developed in this section was published in [6].

In the next section we define the LULU operators in the setting of $\mathcal{A}(\mathbb{G})$, for a graph $\mathcal{G}$, and consider their structure preserving properties. The theorems in that section combine results from [183] and [8] and are given here without proofs.

5.3.1 The LULU Operators on a Graph

Let us denote by $\mathcal{C}_n(v)$ the set of connected subgraphs containing the vertex v and n other vertices, that is

$$\mathcal{C}_n(v) = \{C \in \mathcal{G} : v \in C, \text{card}(C) = n + 1\}.$$

Then for any $n \in \mathbb{N}$ the operators $L_n, U_n : \mathcal{A}(\mathbb{G}) \to \mathcal{A}(\mathbb{G})$ are defined as

$$L_n(f)(v) = \max_{C \in \mathcal{C}_n(v)} \min_{w \in C} f(w), \quad U_n(f)(v) = \min_{C \in \mathcal{C}_n(v)} \max_{w \in C} f(w).$$

The smoothing effect of the LULU operators can be described as removing 'peaks' and 'pits' of sufficiently small support. This is made precise through the definitions below.

Definition 40 *Let $C \in \mathcal{G}$. A vertex $v \notin C$ is called* ***adjacent*** *to C if $C \cup \{v\} \in \mathcal{G}$. The set of all vertices adjacent to C is denoted by* $\text{adj}(C)$*, that is,*

$$\text{adj}(C) = \{v \notin C : C \cup \{v\} \in \mathcal{G}\}.$$

Definition 41 *A set $C \in \mathcal{G}$ is called a* **local maximum set** *of $f \in \mathcal{A}(\mathbb{G})$ if*

$$\sup_{w \in \text{adj}(C)} f(w) < \inf_{v \in C} f(v).$$

Similarly C is a ***local minimum set*** *if*

$$\inf_{w\in \mathrm{adj}(C)} f(w) > \sup_{v\in C} f(v).$$

Definition 42 *We say that $f \in \mathcal{A}(\mathcal{G})$ is* ***locally*** *n-**monotone** if every local maximum or local minimum set of f is of size $n+1$ or more. The set of all functions in $\mathcal{A}(\mathbb{G})$ which are n-monotone is denoted by $\mathcal{M}_n$.*

The operator L_n removes local maximum sets (peaks) of size n or less while U_n removes local minimum sets (pits) of size n or less so that we have the following theorem.

Theorem 43 *For any $n \in \mathbb{N}$ and $f \in \mathcal{A}(\mathbb{G})$ we have that $L_nU_n(f) \in \mathcal{M}_n$ and $U_nL_n(f) \in \mathcal{M}_n$. Moreover, $f \in \mathcal{M}_n \iff (L_n(f) = f,\ U_n(f) = f)$.*

An example of the application of the LULU operators is given in Figure 5.3. The figures on the right are the graphs of the luminosity functions of the images on the left. A noisy input is given in Figure 5.3(a). It is well known that impulse noise creates spikes of extreme luminosity with small support. The operator $L_{30}U_{30}$ is applied to remove such random noise and the smoothed image is presented in Figure 5.3(b). The LULU operators can be also used for extracting features of given size. The keys of the calculator are extracted in Figure 5.3(c) by using the composition $(id - L_{3368}U_{3368})L_{624}U_{624}$. Notice that these values are obtained for this specific example and as such an automatic procedure will be developed as future research to determine these values automatically.

5.3.2 Locally monotone approximations

The rationale for locally monotone approximations is given in [179] for one dimensional signals, but it also applies to higher dimensions as well as the general setting of functions on a graph considered here. It can be described shortly as follows. Suppose it is known that the expected signal has particular kind of local monotonicity, e.g. it belongs to $\mathcal{M}_n$ for some $n \in \mathbb{N}$. If the input f in not in $\mathcal{M}_n$ then clearly it is contaminated with noise. Then we take the best approximation of f in $\mathcal{M}_n$ as signal. We should remark that the concepts of signal and noise are relative. Signal generally refers to

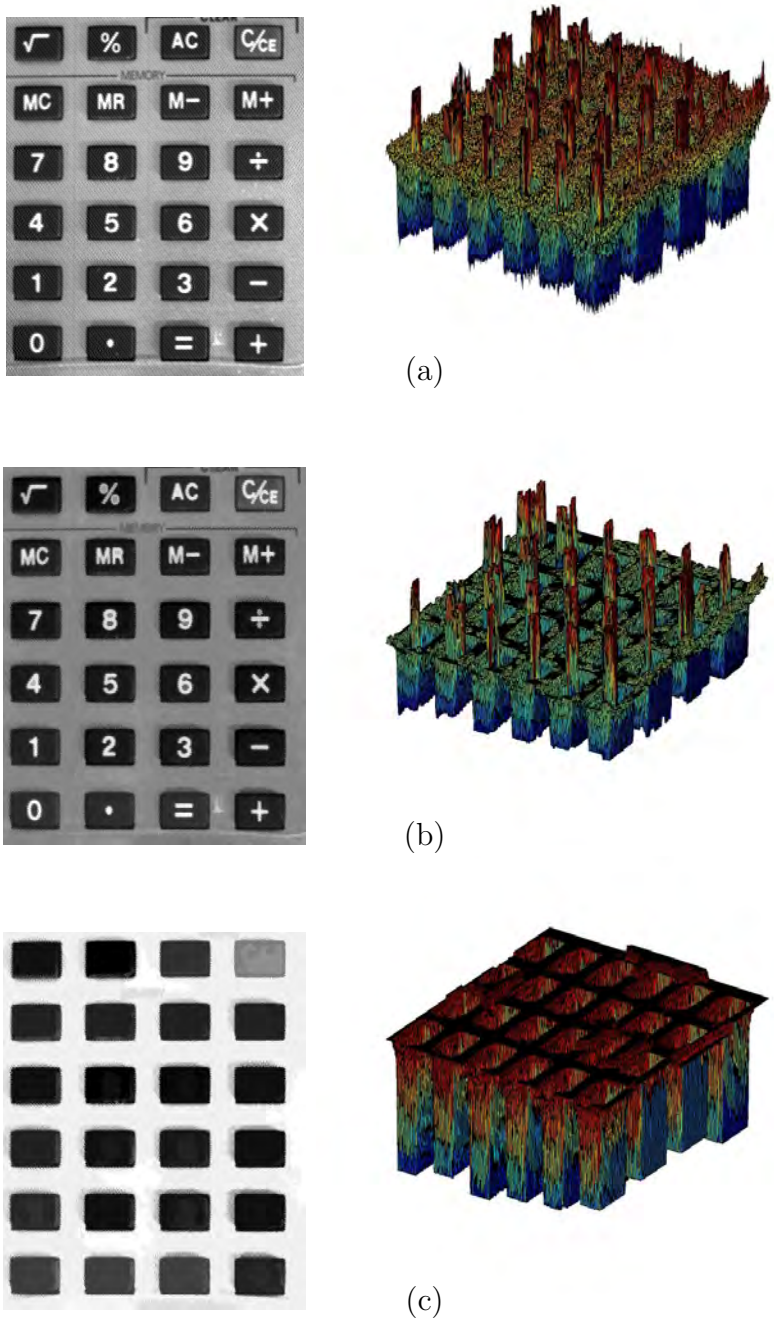


Figure 5.3: *An illustrative example: (a) input; (b) noise removed; (c) features of interest (keys) extracted*

required information or feature that needs to be separated from the input. For example, if from the input on Figure 5.3(a) we require the keys of the calculator as they have been extracted on Figure 5.3, then everything else is considered noise, e.g. including the labels on the keys which are indeed removed.

In the stated formulation the problem of signal extraction is an approximation problem. The issue of proximity can be considered in any of the norms $||\cdot||_p$, $p \in [1, \infty]$. It is easy to see that $\mathcal{M}_n$ is a closed subset of $\mathcal{A}(\mathbb{G})$ in any one of these norms. Therefore, a best approximation exists. Further analysis of this problem is difficult. On the one hand, uniqueness can not be guaranteed since $\mathcal{M}_n$ is not convex. On the other hand, constructive algorithms for the best approximation are not currently available, however, future work aims to resolve this issue. The best approximation takes into account only proximity and does not necessarily preserve any other essential and/or useful properties of the input. Our main result given in Theorem 44 shows that while the LULU operators do not necessarily produce the best approximation, the error of the approximation is bounded by a constant multiple of the error of the best approximation and in this sense it is near best. The involved constant naturally depends on n and on the connectivity of the graph.

We introduce a metric on $\mathbb{G}$ in the usual way. Let $u, v \in \mathbb{G}$. Since $\mathbb{G}$ is connected there exists a path connecting u and v. The shortest path is the one with fewest edges. We denote by $\rho(u, v)$ the number of edges in the shortest path connecting u and v. Then

$$B(v, n) = \{u \in \mathbb{G} : \rho(u, v) \leq n\}$$

can be considered as the ball centered at v with radius n. Let $\mathcal{K}_n = \sup_{v \in \mathbb{G}} \operatorname{card}(B(v, n))$. It is easy to see that $\mathcal{K}_n < \infty$, e.g. we have $\mathcal{K}_n \leq \alpha^n$.

Theorem 44 *Let P be either $L_n \circ U_n$ or $U_n \circ L_n$. For any $f \in \mathcal{A}(\mathbb{G})$ and any $h \in \mathcal{M}_n$ we have*

$$\begin{aligned} \|Pf - f\|_p &\leq \left(1 + \left(\mathcal{K}_n\right)^{1/p}\right)\|h - f\|_p, \ p \in [1, \infty), \\ \|Pf - f\|_\infty &\leq 2\|h - f\|_\infty. \end{aligned}$$

The idea of the proof of the inequalities in Theorem 44 comes from the Lebesgue inequality. For a linear, idempotent and bounded operator P on a

normed space X for every $f \in X$ and $h \in P(X)$ we have

$$\|Pf - f\| \leq (1 + \|P\|)\|f - h\|. \tag{5.1}$$

The LULU operators are not linear so that the inequality (5.1) is not directly applicable. We proceed by establishing the Lipschitz property for these operators.

Theorem 45 *For any $f, g \in \mathcal{A}(\mathbb{G})$ we have*

$$\begin{aligned} \|L_n f - L_n g\|_p &\leq \mathcal{K}_n^{1/p}\|f - g\|_p, \\ \|U_n f - U_n g\|_p &\leq \mathcal{K}_n^{1/p}\|f - g\|_p, \end{aligned}$$

Proof
Let $v \in \mathbb{G}$. Without loss of generality we may assume that $L_n f(v) \geq L_n g(v)$. From the definition of L_n

$$L_n f(v) = \max_{C \in \mathcal{C}_n(v)} \min_{w \in C} f(w) = \min_{w \in C_v} f(w)$$

for some $C_v \in \mathcal{C}_n(v)$. We also have

$$L_n g(v) = \max_{C \in \mathcal{C}_n(v)} \min_{w \in C} g(w) \geq \min_{w \in C_v} g(w) = g(u_v),$$

for some $u_v \in C_v$. Thus

$$\begin{aligned} |L_n f(v) - L_n g(v)| &= L_n f(v) - L_n g(v) \\ &\leq \min_{w \in C_v} f(w) - g(u_v) \\ &\leq f(u_v) - g(u_v). \end{aligned}$$

Using that $u_v \in C_v \in \mathcal{C}_n(v)$ it is easy to see that $\rho(v, u_v) \leq n$. Therefore

$$|L_n f(v) - L_n g(v)|^p \leq |f(u_v) - g(u_v)|^p \leq \sum_{w \in B(v,n)} |f(w) - g(w)|^p. \tag{5.2}$$

Using the inequality (5.2) for every $v \in \mathbb{G}$ we obtain

$$\begin{aligned} \|L_n f - L_n g\|_p^p &= \sum_{v \in \mathcal{G}} |L_n f(v) - L_n g(v)|^p \\ &\leq \sum_{v \in \mathcal{G}} \sum_{w \in B(v,n)} |f(w) - g(w)|^p \\ &\leq \mathcal{K}_n \sum_{w \in \mathcal{G}} |f(w) - g(w)|^p \\ &= \mathcal{K}_n ||f - g||_p, \end{aligned}$$

which proves the Lipschitz property of L_n. The Lipschitz property of U_n is proved similarly. ■

It is easy to obtain from Theorem 45 that the compositions L_nU_n and U_nL_n are also Lipschitz with a constant $\mathcal{K}^{\frac{2}{p}}$ for $p \in [1, \infty)$. However, we actually need a Lipschitz inequality when one of the functions is in $\mathcal{M}_n$. In this case the respective constant is smaller as shown in the next theorem.

Theorem 46 *For all $f \in \mathcal{A}(\mathbb{G})$ and $g \in \mathcal{M}_n$ we have*

$$\|L_nU_nf-g\|_p \leq \mathcal{K}_n^{1/p}\|f-g\|_p \ , \quad \|U_nL_nf-g\|_p \leq \mathcal{K}_n^{1/p}\|f-g\|_p \ .$$

Proof
Let $v \in \mathbb{G}$. If $L_nU_nf(v) < g(v)$ using that $U_n \geq id$ we obtain

$$|L_nU_nf(v) - g(v)| = g(v) - L_nU_nf(v) \leq g(v) - L_nf(v) = |L_nf(v) - L_ng(v)|.$$

Then it follows from inequality (5.2) derived in the proof of Theorem 45 that

$$|L_nU_nf(v) - g(v)|^p \leq |L_nf(v) - L_ng(v)|^p \leq \sum_{w\in B(v,n)} |f(w) - g(w)|^p \quad (5.3)$$

If $L_nU_nf(v) \geq g(v_i)$ then similarly using that $L_n \leq id$ and the inequality for U_n which is analogical to (5.2) we have

$$|L_nU_nf(v) - g(v)|^p \leq |U_nf(v) - U_ng(v)|^p \leq \sum_{w\in B(v,n)} |f(w) - g(w)|^p. \quad (5.4)$$

The combined application of (5.3) and (5.4) for every $v \in \mathbb{G}$ yields

$$\begin{aligned} \|L_nU_nf-g\|_p^p &= \sum_{v\in\mathbb{G}} |L_nU_nf(v)-g(v)|^p \leq \sum_{v\in\mathbb{G}} \sum_{w\in B(v,n)} |f(w) - g(w)|^p \\ &\leq \mathcal{K}_n \sum_{w\in\mathbb{G}} |f(w) - g(w)|^p = \mathcal{K}_n\|f-g\|_p^p \end{aligned}$$

which proves the inequality for L_nU_n. The inequality for U_nL_n is proved in a similar manner. ■

Remark 47 *Letting $p \to \infty$ we obtain from Theorems 45 and 46 that the operators L_n, U_n and their compositions all satisfy the Lipschitz property with a constant 1 with respect to the supremum norm.*

Proof of Theorem 44 Let $p \in [1, \infty)$. Using Theorem 46 we obtain

$$\begin{aligned} \|Pf - f\|_p &\leq \|Pf - h\|_p + \|h - f\|_p \\ &\leq \mathcal{K}_n^{1/p} \|f - h\|_p + \|h - f\|_p \\ &= (1 + \mathcal{K}^{1/p}) \|h - f\|_p. \end{aligned}$$

For $p = \infty$ it follows from Remark 47 that

$$\begin{aligned} \|Pf - f\|_\infty &\leq \|Pf - h\|_\infty + \|h - f\|_\infty \\ &= \|Pf - Ph\|_\infty + \|h - f\|_\infty \\ &\leq 2\|h - f\|_\infty, \end{aligned}$$

which completes the proof. ■

The idea of using monotonicity as a concept of smoothness within approximation theory originates in the works of Sendov and Popov, e.g. [206]. In this section we consider the situation when a signal or a feature with smoothness defined in terms of its local monotonicity needs to be extracted from a given input. We show that the LULU operators typically considered for their structure preserving properties also provide near best locally monotone approximations. The general setting of functions defined on a graph includes as particular cases both sequences as in [183] and multidimensional arrays as in [8]. We have provided a simple example in Figure 5.3 and future research will look into the applicable construction of these near best-approximations.

5.4 Noise Removal

In this section we shall look at the ability of the LULU operators to remove noise, of all types, from a signal. Noise removal and measurement is a widely researched topic as it is inevitable that noise arises in a signal. Consider for example Murtagh and Starck [154], who take a multi-scale approach in this regard, arguing that noise will appear at different scales, and use statistical significance tests to determine which wavelet coefficients (a non-orthogonal form) are due to noise. They model the noise as additive Poisson and/or Gaussian. As an alternative to direct noise removal, Coutinho et al [40] take advantage of the phase, polarization and coherence properties of light to improve feature detection and other image analysis techniques.

The process by which a signal is obtained from a physical phenomenon involves firstly the conversion of the phenomenon into an electrical signal via

a transducer, secondly conditioning of the signal, thirdly conversion of the signal from analog to digital (ADC) which involves sampling it discretely as well as quantizing it, and lastly the conversion into a software compatible form, [117]. Ideally the signal should be accessible on the computer interface as accurately and noiseless as possible, so the measurement hardware used for the extraction should be effective and appropriate for the system requirements.

The most common noise discussed in noise models is white noise. White noise is completely random with an equally distributed frequency distribution. Different colours of noise are also discussed. For example, red noise (Brownian noise) has more energy at lower frequencies, purple/violet noise has more energy at high frequencies, and pink noise, an intermediate between white and red noise, has a frequency distribution inversely proportional to the frequency. The various colours of noise represent the various frequency distribution forms [73]. An interesting relationship between the colours of noise and music is presented by Bulmer [29].

The rest of this section provides an investigation into the ability of the LULU smoothers to remove various distributional types of noise in signals and images.

5.4.1 Noise Removal in One Dimension

The work in this section was published together with a colleague and student as proceedings of the 2010 South African Statistical Association (SASA) conference [56].

Recall that the LULU smoothers for signals (sequences) have been developed over the last three decades by Rohwer and his collaborators, [183]. For a signal $x = (x_i)_{i=-N}^{N}$, the LULU operators L_n and U_n act at position i in the signal and for $n = 1, 2, 3...$ as follows:

$$(L_n(x))_i = \max\{\min\{x_{i-n}, ..., x_i\}, ..., \min\{x_i, ..., x_{i+n}\}\}, \text{ and}$$

$$(U_n(x))_i = \min\{\max\{x_{i-n}, ..., x_i\}, ..., \max\{x_i, ..., x_{i+n}\}\}.$$

We also recall that the LULU operators are nonlinear but have very useful properties to their name, that is, they are separators, are total variation preserving and fully trend preserving as defined in [183]. However, since $L_n(x) \leq x \leq U_n(x)$ the two operators will produce slightly biased results

when used individually, namely, L_n smoothes the signal from above and U_n smoothes from below. We thus use the two together as either $L_n \circ U_n$ or $U_n \circ L_n$. These compositions are also biased, but to a far lesser degree. The Discrete Pulse Transform (DPT) of x, $DPT(x) = (D_1(x), D_2(x), ..., D_N(x))$, is obtained as the iterative application of $L_n \circ U_n$ or $U_n \circ L_n$ for $n = 1, 2, ..., N$. The components D_i are obtained as follows, $D_1(x) = (I - P_1)(x), D_n(x) = (I - P_n) \circ Q_{n-1}(x),\ n = 2, ..., N$, where $P_n = L_n \circ U_n$ or $P_n = U_n \circ L_n$ and $Q_n = P_n \circ ... \circ P_1,\ n \in \mathbb{N}$. The DPT can be seen as the recursive peeling off of pieces of information of width n - we first remove isolated information of width 1, then of width 2, and so on. For some n the remaining signal is considered sufficiently smoothed (denoised). This optimal n is determined by tracking the total variation removed at each step. The total variation of a signal x is defined as, $TV(x) = \sum_{i=-N}^{N} |x_i - x_{i-1}|$. Since our LULU operators are total variation preserving ($TV(x) = TV(Px) + TV((I-P)x)$ where P is either $L_n \circ U_n$ or $U_n \circ L_n$), we can easily track how much variation remains in the smoothed signal, $TV(Px)$, and how much we remove with each iteration or in total, $TV((I - P)x)$, since no variation is lost at any step. Once the optimal n is decided upon, say n_{opt}, the immediate question to ask is how well has the signal been smoothed or equivalently, how well does that which we have removed, $(I - P_{n_{opt}})x$, represents the noise present in the original signal x? It turns out that the DPT is quite effective in removing impulsive noise. One explanation for this is that linear smoothers aren't well suited to removing noise which arises from a long-tailed probability distribution, [233], which is characteristic when there are outliers present, nor noise which is signal dependent, [38], whereas the LULU operators are nonlinear smoothers which is believed to avoid such complications. Here we investigate the ability of the DPT to remove imposed noise and uncover the underlying signal effectively. More specifically, by imposing noise chosen from various distributions, see Table 1, we shall determine if the removed noise $(I - P_{n_{opt}})x$ accurately represents the noise initially imposed.

In order to simulate noise with various distributional shape properties, we use a parameterization of the Generalized Lambda Distribution (GLD) introduced by [232] and defined through its quantile function (QF) by

$$Q(p) = \begin{cases} \alpha + \beta\big((1-\delta)\big(\frac{p^\lambda - 1}{\lambda}\big) - \delta\big(\frac{(1-p)^\lambda - 1}{\lambda}\big)\big) & \text{if } \lambda \neq 0 \\ \alpha + \beta\big((1+\delta)\ln p - \delta \ln(1-p)\big) & \text{if } \lambda = 0 \end{cases}$$

where $0 \leq p \leq 1$, α is a location parameter, $\beta > 0$ is a spread parameter and $0 \leq \delta \leq 1$ and λ are shape parameters. The GLD can be characterized through its first four L-moments, that is, the L-location, L_1 , the L-scale, L_2

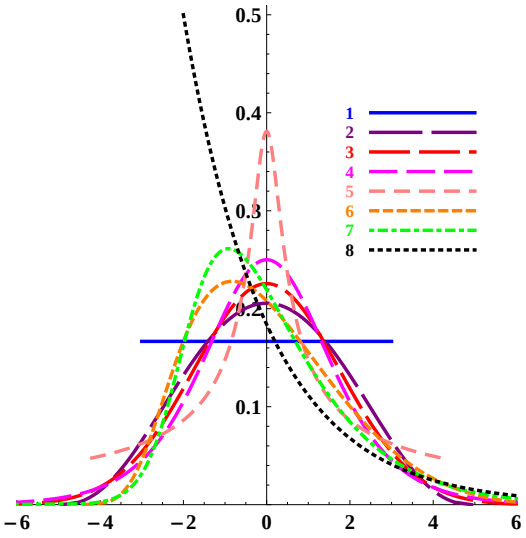


Figure 5.4: *Probability Density Functions of the Noise Distributions in Table 5.5*

and the L-skewness and L-kurtosis ratios, $\tau_3 = L_3/L_2$ and $\tau_4 = L_4/L_2$. As shown in Table 5.5 and Figure 5.4, we selected eight distributions from the GLD with different distributional shapes by choosing appropriate values for τ_3 and τ_4 and calculating the corresponding parameter values. All selected distributions were standardized and/or shifted so that $L_1 = 0$ and $L_2 = 1$.

The fact that we simulate the noise from a family of distributions, the GLD, with a single functional form as defined through its QF, is important for our investigation as it enables a strongly justified comparison amongst the noise types.

For this study the underlying true signal used was $(s_i) = (a\cos(wi) + b\sin(wi))$ where the parameters a and b are chosen in order to obtain a weak, medium and strong signal respectively with respect to the noise. The period was chosen as 100 throughout, and the frequency w was then calculated through the formula $2\pi/100$. The length of the signal was taken to be 100, that is, 100 data points. For this study we thus for simplicity use the subscripts $1, 2, ..., 1000$ instead of $-N, ..., N$. The amplitude of such a signal is $\sqrt{a^2 + b^2}$, [66]. This signal is periodic and thus has an obvious cyclical trend which enables easy detection of the true signal.

Typically the signal-to-noise ratio (SNR), [167], is used to measure the strength of a signal. The three signals were chosen to have SNR 1, 5 and 9 respectively, which correspond to a weak, medium and strong signal according to

Distribution Shape	L**-moments:** $(L_1, L_2, \tau_3, \tau_4)$	**Parameters of the GLD:** $(\alpha, \beta, \delta, \lambda)$
1. Symmetric, Uniform distribution[a]	$(0, 1, 0, 0)$	$(0, 6, 0.5, 1)$
2. Symmetric, short-tailed	$(0, 1, 0, \frac{1}{12})$	$(0, 2.9989, 0.5, 0.3025)$
3. Symmetric, Normal distribution[b]	$\left(0, 1, 0, \frac{30}{\pi}\tan^{-1}(\sqrt{2}) - 9\right)$	$(0, 0.2449, 0.5, 0.1416)$
4. Symmetric, Logistic distribution (heavy-tailed)[1]	$(0, 1, 0, \frac{1}{6})$	$(0, 2, 0.5, 0)$
5. Symmetric, truncated distribution	$\left(0, 1, 0, \frac{1}{6}\right)$	$(0, 42, 0.5, 5)$
6. Skewed, Rayleigh distribution[2]	$\left(0, 1, \frac{3\sqrt{2}+2\sqrt{6}-9}{3(\sqrt{2}-1)}, \frac{20\sqrt{6}-9(4+\sqrt{2})}{6(\sqrt{2}-1)}\right)$	$(-1.0173, 2.6641, 0.7305, 0.2071)$
7. Skewed, Gumbel distribution[2]	$\left(0, 1, \frac{\ln(9/8)}{\ln 2}, \frac{2\ln(256/243)}{\ln 2}\right)$	$(-1.1157, 2.1486, 0.7723, 0.0487)$
8. Skewed, Exponential distribution (J-shaped)[1]	$(0, 1, \frac{1}{3}, \frac{1}{6})$	$(-2, 2, 1, 0)$

Table 5.5: *Distributions Chosen to Simulate Noise ([1]Distribution is special case of the GLD, [2]Distribution approximated by the GLD)*

[a]Distribution is special case of the GLD
[b]Distribution approximated by the GLD

the Rose Criterion [236]. As suggested by its name, the SNR is defined as the signal relative to the noise, (see for instance [47]). To calculate the SNR, it is common practice to use a measure of location for the signal and a measure of spread for the noise. For example, the SNR can be calculated as the mean signal relative to the standard deviation of the noise [167], and is given by $\text{SNR} = \frac{\text{mean signal}}{\text{std. dev(noise)}}$. We used L_2 as measure of spread for the noise. Recall that we set $L_2 = 1$ for all eight GLDs used to simulate the noise. Since our signals are periodic with zero mean levels, we decided to measure each signal by its amplitude. Hence we calculated the SNR with $\text{SNR} = \frac{\text{amplitude of signal}}{L\text{-scale of noise}}$. So, given $L_2 = 1$ and SNR equal to 1, 5 and 9 respectively, it then follows that the signal parameters a and b are given by 0.5, 4.5 and 8.5, and 0.866, 2.179 and 2.958, respectively, for the weak, medium and strong signals. These three signals are shown in Figure 5.5.

The DPT was then applied to $(s_i^j + n_i^k), j = 1, 2, 3,\ k = 1, 2, ..., 8$ where (s_i^j) is the j^{th} underlying signal and (n_i^k) is the k^{th} noise signal. In Figure 5.6, some of these contaminated signals are illustrated. The strength of the signals for the various SNRs can be seen clearly. The DPT was applied in four different ways in order to fully investigate the noise removal and any bias due to the ordering, namely for (1) $L_n \circ U_n \circ L_{n-1} \circ U_{n-1} \circ ... \circ L_1 \circ U_1$, (2) $U_n \circ L_n \circ U_{n-1} \circ L_{n-1} \circ ... \circ U_1 \circ L_1$, (3) $U_n \circ L_n \circ L_{n-1} \circ U_{n-1} \circ ... \circ U_1 \circ L_1$, and (4) $L_n \circ U_n \circ U_{n-1} \circ L_{n-1} \circ ... \circ L_1 \circ U_1$. We shall use the notation LULU, ULUL, LUUL and ULLU for these. The last two options are called the *alternating bias operators* since they alternately swop between the two basic choices $L_n \circ U_n$ and $U_n \circ L_n$. See [100] for other possibilities of reducing the bias.

For the three signals we thus apply the DPT with respect to (1)-(4) for the 8 different noise types. The total variation is tracked throughout the DPT and the cumulative noise removed for (a) n where half of the added total variation has been removed, and for (b) n_0 where all the added total variation has been removed, is investigated. We investigate (a) as it is understood that the most disruptive noise occurs in the first levels of the DPT, and (b) because this is where it is naturally thought that the original signal should be uncovered. The respective true noise distribution for $k = 1, 2, ..., 8$ is fitted to these noise samples using method of L-moment estimation, [232], to investigate if the noise removed up to the two respective points is distributed similarly to the original noise imposed.

Due to the total variation preservation of the DPT, total variation is a good measure to track the smoothing process over n, i.e. from level to level of the DPT. When the total variation removed with each n stabilizes, i.e. doesn't

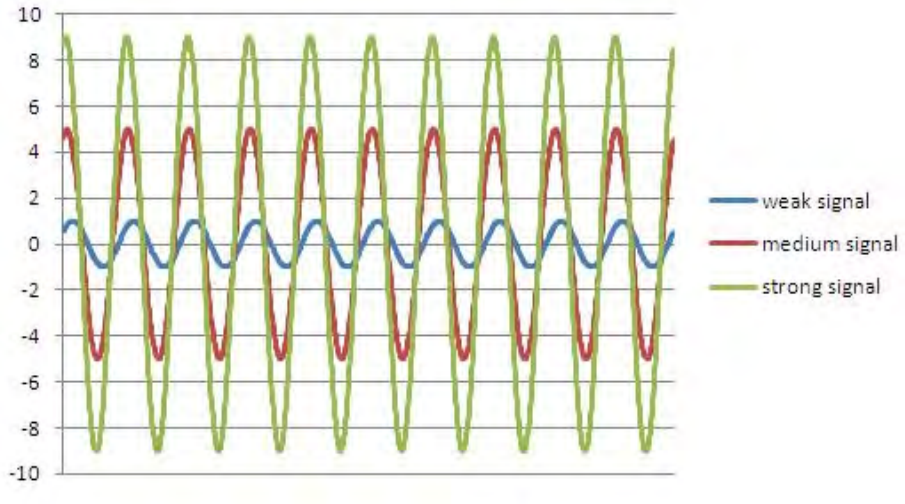


Figure 5.5: *Original Signals with SNR 1 (weak), 5 (medium) and 9 (strong)*

Noise Type	1	2	3	4	5	6	7	8
SNR = 1	2050	2066	2072	2078	2075	2056	2056	2039
SNR = 5	2055	2070	2077	2083	2082	2061	2062	2047
SNR = 9	2070	2087	2094	2101	2101	2078	2079	2071

Table 5.6: *Total Variation (rounded) of the 24 Contaminated Signals*

change significantly from n to $n+1$, the added noise has been removed effectively. From the results of the 24 different contaminated signals, the total variation removed at each level remains similar whichever combination, LULU, ULUL, LUUL or ULLU, is used to obtain the DPT, and for each of the three SNRs. A slight difference can only be seen in the three skewed noise types, namely types 6, 7 and 8. It can be seen in Figure 5.7 how the total variation progresses through the DPT levels. The differences seen between the different SNRs are due to the fact that the weak, medium and strong original signals have a total variation of 40.491, 204.487 and 368.423 respectively, thus the smoothing (decrease in total variation) occurs sharply up until that point and then stabilizes. The contaminated signals have total variation as indicated in Table 5.6. Comparing Table 5.6 with Figure 5.7 it can be seen a huge proportion of the total variation is removed in the first level of the DPT as the remaining total variation drops to around 600 in all cases. The stabilization of the total variation removal varies for the three SNRs investigated. For the weak signal (SNR = 1) half the total variation is removed at around $n = 3$ and all the added total variation (i.e.

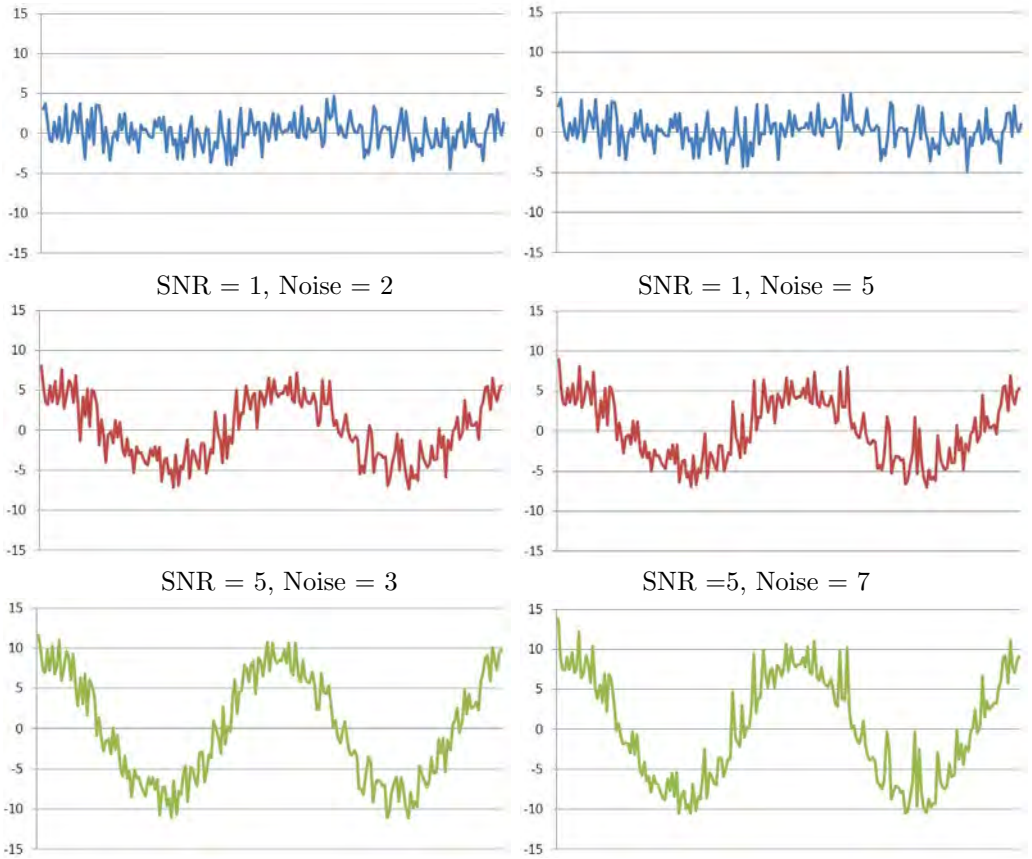
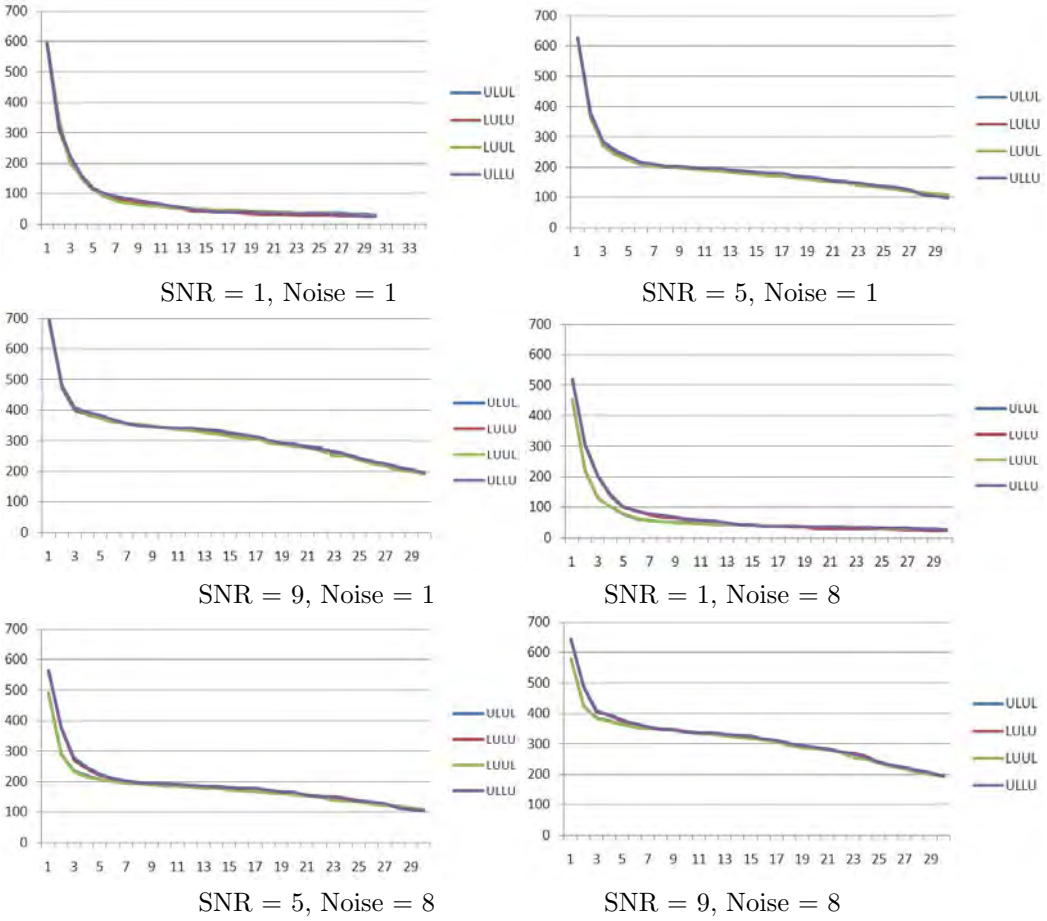


Figure 5.6: *A Sample of the 24 Different Contaminated Signals*



SNR = 1, Noise = 1 SNR = 5, Noise = 1

SNR = 9, Noise = 1 SNR = 1, Noise = 8

SNR = 5, Noise = 8 SNR = 9, Noise = 8

Figure 5.7: *Total Variation Removed at each Level of the DPT for Noise Type 1 (Noise Types 1-5 are similar) and 8 (Noise Types 6-8 are similar)*

at stabilization) is removed by around $n = 14$ for LULU and ULLU and by around $n = 19$ for LUUL and ULUL.

For the medium signal (SNR = 5) half the total variation is removed at around $n = 3$ as well and all the added total variation (i.e. at stabilization) is removed by around $n = 6$. For the strong signal (SNR = 9) half the total variation is removed at around $n = 2$ and all the added total variation (i.e. at stabilization) is removed by around $n = 5$. It would thus seem, as expected, that the stronger the signal (or the weaker the noise) the quicker and more effective the noise removal, and also that the bias between the four L_n and U_n combinations decreases.

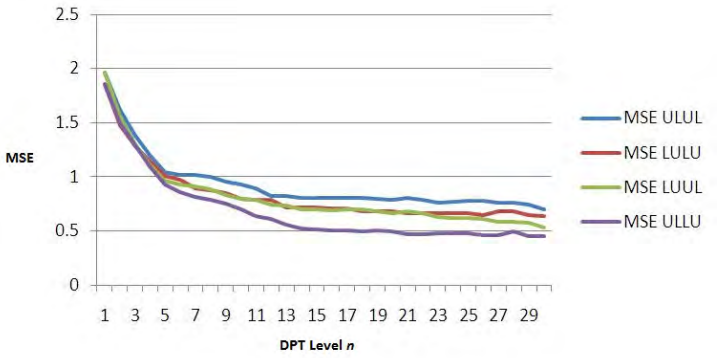
To investigate whether the smoothed signal obtained when the total variation stabilizes does in fact resemble the original uncontaminated signal, the MSE measure was used to calculate the differences between the smoothed signal through the DPT levels and the original signal. The MSE is calculated as

$$MSE(x) = \frac{\sum_{i=1}^{1000}(x_i - \bar{x})^2}{1000}.$$

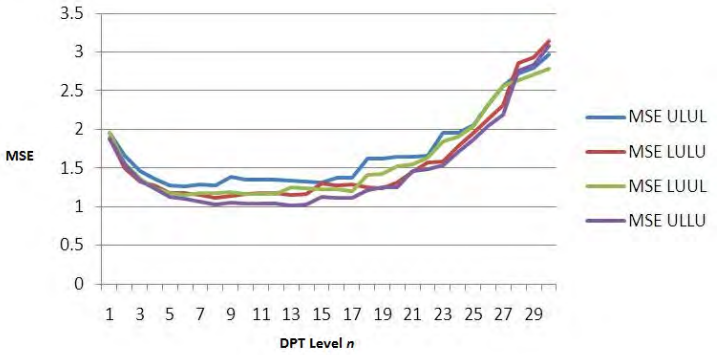
For SNR = 1 the combination ULLU provides the lowest MSE from the beginning of the smoothing process. The combination ULUL gives the highest MSE although the differences between the combinations are not drastic. For SNR = 5 and 9 the same is seen. See Figure 5.8 for the MSE for noise type 1. The medium and strong signal give very interesting results for the MSE measurements. It can be seen in Figure 5.8 that the MSE starts to increase from level 14 of the DPT onwards. This indicates that from this point onwards the smoothing process begins to smooth out the uncovered original signal instead of the noise. As the SNR increases the MSE in the beginning levels of the DPT is more similar for the four L_n and U_n combinations, see Figure 5.8.

In Figure 5.9, the smoothed signals can be visually analysed. The higher the SNR the more effective the noise removal, i.e. the more the smoothed signal resembles the original signal.

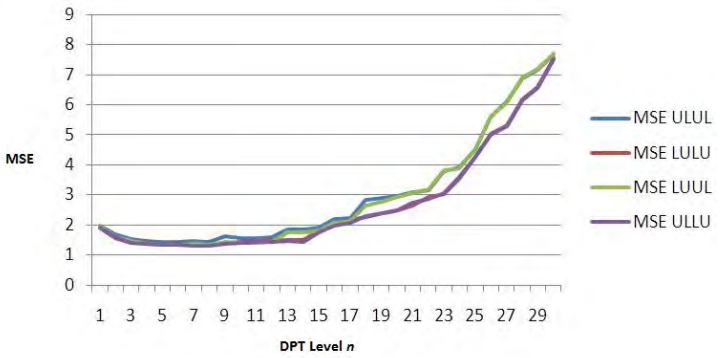
As discussed above, the cumulative noise removed when (a) half of the added total variation has been removed, and (b) when all the added total variation has been removed, is investigated. The original noise distributions were fitted to this cumulative removed noise. The L-moments and four parameters of the GLD were then compared to evaluate the fit of the removed noise. The results for the L-location, L_1, and the L-skewness ratio, τ_3, are given in Tables 5.7 and 5.8. The following observations can be made:



SNR = 1



SNR = 5



SNR = 9

Figure 5.8: *MSE at each Level of the DPT for Noise Type 1 (other noise types are similar)*

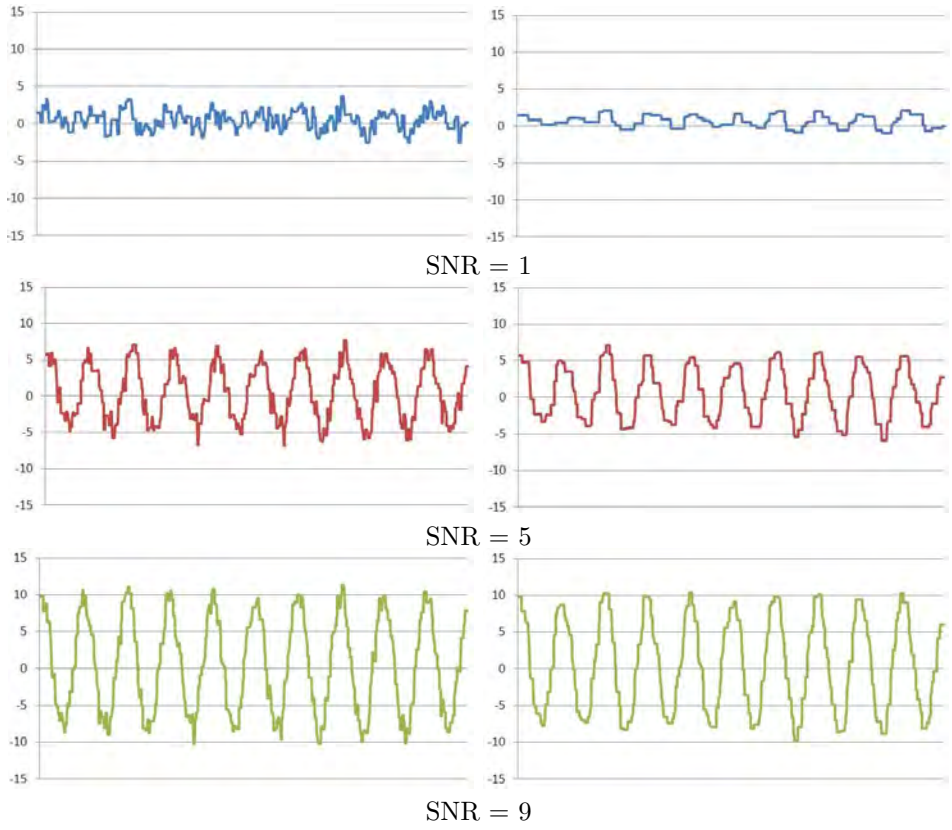


Figure 5.9: *The Smoothed Signals for Noise Type 1 and using LULU (all noise types and L_n, U_n combinations are visually similar): Left column indicates the smoothed signal when half the added TV has been removed, Right column indicates the smoothed signal when all the added TV has been removed*

- From Table 5.7 we see that LULU and ULLU result in a negative shift in location for each SNR, although the shift seems to reduce from noise type 1 through to 8. The shift in location is of course due to the biasedness of the various smoothers, already discussed. For ULUL and LUUL the shift in location is seen to be positive for each SNR, but again generally increases from noise type 1 through to 8. An interesting phenomenon can be seen for noise types 6 and 7 for ULLU. For noise type 7 we see a shift of 0 for SNR $= 1$ and 5, but a negative shift for SNR $= 9$. Noise distributions 6 and 7 are very similar and thus behave similarly. It is thus evident that the smaller SNRs result in poorer removal of noise type 7, i.e. do not as effectively remove the same noise that was imposed.

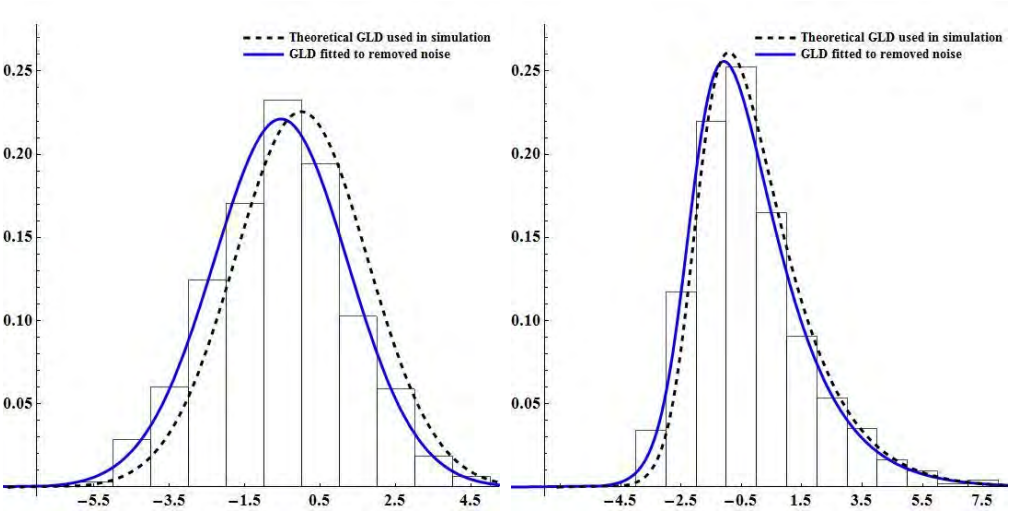
- The shift in L-location must be considered simultaneously with the change in the L-skewness ratio, as seen in Table 5.8, since a change in the level of skewness of a distribution will result in a shift in the location of that distribution. There we see a decrease in the L-skewness ratio for LULU and ULLU. This decrease become less prominent as SNR increases however. See Figure 5.10 for the fitted and original distributions for noise types 3 and 7. The shift in location and change in skewness can be seen. The changes are due to the fact that U_n is applied first in LULU and ULLU. The operator U_n removes negative pulses and thus the removed noise favours slightly the negative direction. For ULUL and LUUL there is a general increase in the L-skewness ratio for noise types 1 to 5, and the trend becomes stronger as SNR increases. The changes are due to the fact that L_n is applied first in ULUL and LUUL. The operator L_n removes positive pulses and thus the removed noise favours slightly the positive direction. For noise types 6 to 8, a decrease in the L-skewness ratio is still observed. This is due to the fact that these noise distributions are already positively skewed and thus contain more negative pulses from the start. Although the above discussed change in the L-skewness ratio is clear, the change is very slight, as can be seen by the values in Table 5.8.

- The fitted distributions fit the original distribution very well when half the added total variation has been removed. Furthermore, in general the fit improves towards the full removal of the added total variation.

- The L-scale does not vary significantly at all for any of the fits and for each SNR investigated. This is an important result as it indicates that the removed noise has very similar spread to the noise which was

	Noise Type	1	2	3	4	5	6	7	8
SNR = 1	LULU	-0.70	-0.60	-0.56	-0.50	-0.43	-0.37	-0.23	0
	LUUL	0.40	0.35	0.30	0.27	0.23	0.53	0.58	0.80
	ULLU	-0.45	-0.38	-0.36	-0.34	-0.29	**-0.16**	**0**	0.27
	ULUL	0.68	0.56	0.51	0.45	0.37	0.73	0.77	1
SNR = 5	LULU	-0.69	-0.62	-0.59	-0.58	-0.54	-0.40	-0.27	0
	LUUL	0.44	0.40	0.38	0.36	0.36	0.60	0.68	0.95
	ULLU	-0.43	-0.41	-0.40	-0.39	-0.37	**-0.20**	**0**	0.22
	ULUL	0.62	0.56	0.52	0.49	0.47	0.74	0.80	1.06
SNR = 9	LULU	-0.76	-0.69	-0.66	-0.63	-0.62	-0.46	-0.33	0
	LUUL	0.49	0.46	0.45	0.44	0.45	0.64	0.72	1.02
	ULLU	-0.57	-0.53	-0.51	-0.49	-0.48	**-0.32**	**-0.20**	0.10
	ULUL	0.67	0.62	0.60	0.57	0.57	0.79	0.85	1.14

Table 5.7: *L-Location Moments of the Fully Removed Noise (Grey: Negative Change, White: Positive Change, In Bold: interesting case)*



SNR = 1, Noise = 3 SNR = 1, Noise = 7

Figure 5.10: *Theoretical and Fitted Noise Distributions*

initially imposed on the signals and almost none of its variation has been left in the smoothed signal.

- The L-kurtosis ratio also does not vary significantly at all for any of the fits and for each SNR investigated, indicating the mass in the tails and centre for each distribution of the imposed noise has been removed intact.

The ability of the LULU smoothers to remove the different noise types for this simple signal is very effective and from the results we see that the noise removed is distributed similarly to the noise originally imposed. The underlying smoothed signal is also effectively uncovered when the total variation removed at each step begins to stabilize. The effect of the different combinations of L_n and U_n produce interesting results as indicated. Future work will look at implementing more effective combinations of L_n and U_n to reduce the bias, such as those in [100]. As expected the fit of the noise removed improves as n increases towards the optimum n_0. A further possibility for this study is to investigate using as a measure of smoothing the number of pulses removed at each level of the DPT, i.e. at each n, and to compare this with using the total variation as a measure for this purpose. In addition, further work into more complicated naturally occurring signals should be investigated.

	Noise Type	1	2	3	4	5	6	7	8
	True Skewness Value	**0**	**0**	**0**	**0**	**0**	**0.114**	**0.17**	**0.333**
SNR = 1	LULU	-0.001	-0.001	-0.003	-0.005	-0.006	0.105	0.161	0.301
	LUUL	0.005	0.003	0.002	-0.002	0.003	0.104	0.154	0.29
	ULLU	-0.003	-0.003	-0.004	-0.005	-0.002	0.102	0.157	0.297
	ULUL	0.009	0.006	0.005	0.002	0.004	0.113	0.168	0.315
SNR = 5	LULU	-0.006	-0.0004	0.002	0.005	0.003	0.093	0.148	0.256
	LUUL	0.001	-0.0001	-0.0001	-0.002	0.003	0.086	0.139	0.271
	ULLU	-0.004	-0.002	0.001	0.003	0.003	0.09	0.145	0.256
	ULUL	0.003	0.0004	-0.0002	-0.002	0.004	0.087	0.14	0.268
SNR = 9	LULU	0.0000	0.002	0.006	0.006	0.007	0.092	0.143	0.24
	LUUL	-0.001	0.004	0.005	0.007	0.009	0.087	0.144	0.255
	ULLU	0.001	0.005	0.008	0.007	0.008	0.086	0.137	0.232
	ULUL	0.002	0.003	0.003	0.004	0.006	0.086	0.144	0.259

Table 5.8: *L-Skewness of the Fully Removed Noise (Grey: decrease in skewness, White: increase in skewness)*

5.4.2 Noise Removal in Two Dimensions

The applicable extension of the noise removal work done in the previous section from signals to images must be carefully considered. Various images should be contaminated with simulated noise and subsequently removed as in the previous section. The nature of the mechanism of adding the noise can vary in a number of ways. Firstly, independent one dimensional noise can be simulated for each pixel and easily removed by retaining only the appropriate pulses of the DPT. The pulses retained are chosen so that the total variation is reduced from the total variation of the contaminated image to that of the original, uncontaminated image. The seven sample images considered can be seen in Figures 5.11 to 5.17, which each show the various levels of contamination. The images were each contaminated with various noise types, namely exponential, gumbel, normal, rayleigh, uniform and weibull, at signal-noise-ratio values 1, 5 and 9, as was similarly done in the previous section. The details of these distributions are given in Table 5.9. The formula used for the signal-noise-ratio was $SNR = \frac{mean(signal)}{std.dev.(noise)}$. All distributions were centered so that the mean is 0. The sample images were chosen to represent a variety of possible image content, that is, textured as well as homogeneous. The increase in pulse numbers as well as total variation in the contaminated images is given in Tables 5.10 and 5.11. Figure 5.18 provides the noise added to the *Chelsea* image at the various signal-noise-ratios and distributions. The distributions look very similar for the other six sample images and were thus not included.

Figures 5.19 to 5.24 show the smooth Chelsea images with the removed noise. Table 5.12 shows the pulse size c in the DPT at which the total variation of the contaminated image has been reduced to the original total variation. The smoothed images are reconstructed from this pulse size upwards. Similarly the noise image is obtained by reconstructing from pulse size 1 up to $c - 1$. By comparing the histograms in Figures 5.19 to 5.24 to those in Figure 5.18 it is clear that the noise removed does not follow the same distribution as the noise added. The removed noise seems to follow a symmetrical distribution in most cases. The remaining sample images produce similar results. It is clear the method of image contamination, noise removal with the DPT as well as fitting of the removed noise to the original distribution must be investigated in more depth.

A more appropriate method of contaminating the images is with two-dimensional noise so that the dependency from pixel to pixel can be realized in the contamination. The fitting of the removed noise should be done with two di-

Table 5.9: Noise distributions used to contaminate the sample images

Distribution	**Parameters**	**Mean**	**Variance**	**Comments**
Exponential	$X \sim exp(\lambda)$	λ	λ^2	Must be shifted
Gumbel	$X \sim gum(\alpha, \beta)$	$\alpha - \gamma\beta$	$\frac{1}{6}\pi^2\beta^2$	γ: Euler-Mascheroni constant
Normal	$X \sim N(\mu, \sigma^2)$	μ	σ^2	
Rayleigh	$X \sim ray(s)$	$s\sqrt{\frac{\pi}{2}}$	$\frac{4-\pi}{2}s^2$	Must be shifted
Uniform	$X \sim U(a, b)$	$\frac{1}{2}(a+b)$	$\frac{1}{12}(b-a)^2$	
Weibull	$X \sim wei(\beta, \alpha)$	$\beta\Gamma(1+\alpha^{-1})$	$\beta^2(1-\pi/4)$	α chosen as 1.5, Must be shifted

Table 5.10: Number of pulses of the DPTs of the various contaminated images

	Image	*Chelsea* 451 × 300	*Tank* 512 × 512	*Fabric* 512 × 512	*Bricks* 512 × 512	*Texture* 512 × 512	*Blocks* 512 × 512	*Regions* 512 × 512
	Original	*32 534*	*113 769*	*184 395*	*159 043*	*194 422*	*68*	*19*
Exponential	Weak	122 336	238 223	231 678	239 696	226 741	222 876	223 612
	Medium	105 754	212 642	204 471	220 011	209 000	214 120	213 940
	Strong	90 428	190 965	194 656	207 163	202 819	200 680	200 992
Gumbel	Weak	123 955	240 590	238 414	219 307	231 490	222 028	224 308
	Medium	107 645	216 600	206 135	222 307	210 156	213 676	214 748
	Strong	92 022	193 145	194 916	211 293	202 581	203 452	202 996
Normal	Weak	123 815	240 392	238 036	228 538	232 110	224 128	217 632
	Medium	109 432	219 236	207 565	224 483	211 382	217 280	205 148
	Strong	93 565	195 530	195 320	212 828	202 819	205 648	225 616
Rayleigh	Weak	123 572	239 656	237 298	232 205	231 770	225 184	225 336
	Medium	109 389	219 610	207 146	225 104	210 588	217 972	218 112
	Strong	93 406	195 551	195 396	212 525	202 906	205 604	205 232
Uniform	Weak	123 180	238 272	237 855	224 829	232 015	222 400	223 864
	Medium	111 737	223 040	208 359	226 863	210 924	219 612	220 180
	Strong	95 015	197 930	195 551	215 489	203 074	208 812	206 928
Weibull	Weak	121 547	235 504	233 531	230 395	229 264	226 346	227 382
	Medium	112 924	224 932	212 703	228 897	214 440	223 992	225 380
	Strong	101 719	207 927	200 092	217 732	206 035	211 848	213 107

Table 5.11: Total Variation of the various contaminated images standardized to a 100 × 100 image

	Image	*Chelsea*	*Tank*	*Fabric*	*Bricks*	*Texture*	*Blocks*	*Regions*
	Original	*104 476*	*146 811*	*534 132*	*258 511*	*773 118*	*29 811*	*28 119*
Exponential	Weak	1 781 355	1 794 930	1 692 171	2 023 529	1 726 580	1 693 810	1 667 785
	Medium	559 331	597 336	751 765	721 073	940 046	569 851	549 092
	Strong	315 361	366 065	623 481	531 754	840 889	341 266	321 171
Gumbel	Weak	1 894 796	1 927 598	1 781 975	2 018 560	1 839 134	1 851 147	1 819 823
	Medium	580 900	621 596	752 650	882 999	950 634	625 283	597 234
	Strong	321 174	370 745	623 131	602 261	842 431	368 494	344 476
Normal	Weak	1 974 865	2 002 748	1 868 928	2 142 431	1 911 105	1 927 362	1 895 941
	Medium	600 613	641 361	766 432	865 754	958 189	642 763	615 413
	Strong	326 102	377 681	625 409	598 959	844 907	375 356	350 786
Rayleigh	Weak	1 988 674	2 015 667	1 878 047	2 197 820	1 913 162	1 931 278	1 896 754
	Medium	598 822	640 380	767 572	832 454	956 990	636 143	608 763
	Strong	326 021	377 710	625 772	585 645	844 758	371 208	347 501
Uniform	Weak	2 134 810	2 158 766	2 002 617	2 276 176	2 029 272	2 075 086	2 043 683
	Medium	611 045	653 898	772 210	894 792	963 614	660 462	631 344
	Strong	330 405	380 886	626 511	613 692	845 476	381 766	356 076
Weibull	Weak	2 138 747	2 160 672	2 042 201	2 347 492	2 070 727	2 103 500	2 071 604
	Medium	745 343	788 130	869 599	943 707	1 036 931	769 550	743 923
	Strong	441 371	487 486	679 910	666 989	886 394	477 309	453 457

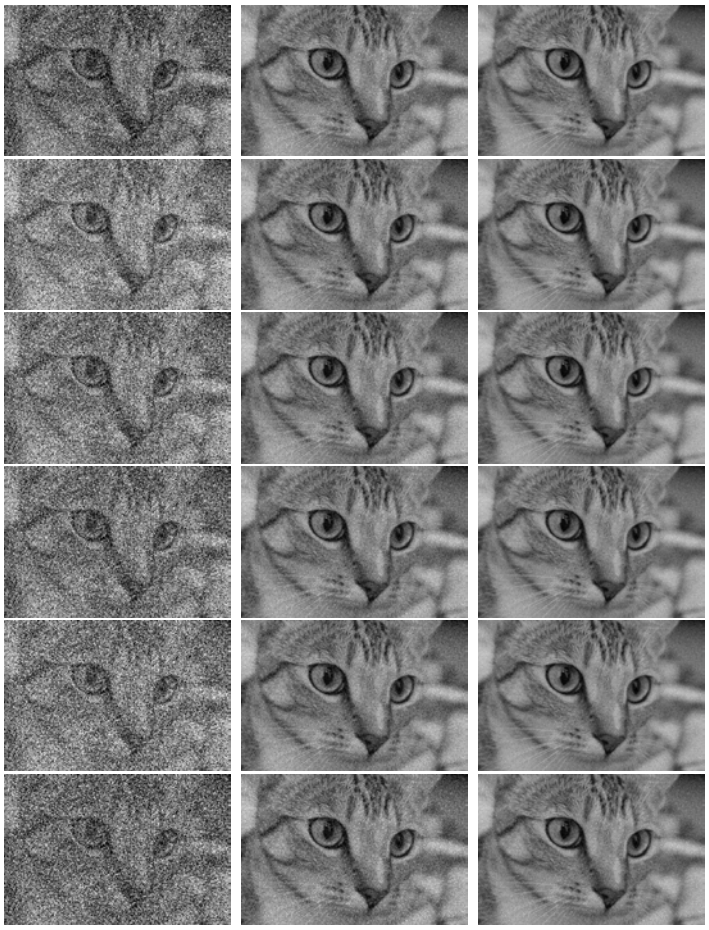


Figure 5.11: *Contaminated Chelsea images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

Table 5.12: Pulse size indicated by the DPT for reduction in total variation to the original total variation

SNR	**1**	**5**	**9**
Exponential	5	3	3
Gumbel	6	3	3
Normal	6	3	3
Rayleigh	6	3	3
Uniform	6	3	3
Weibull	6	3	3

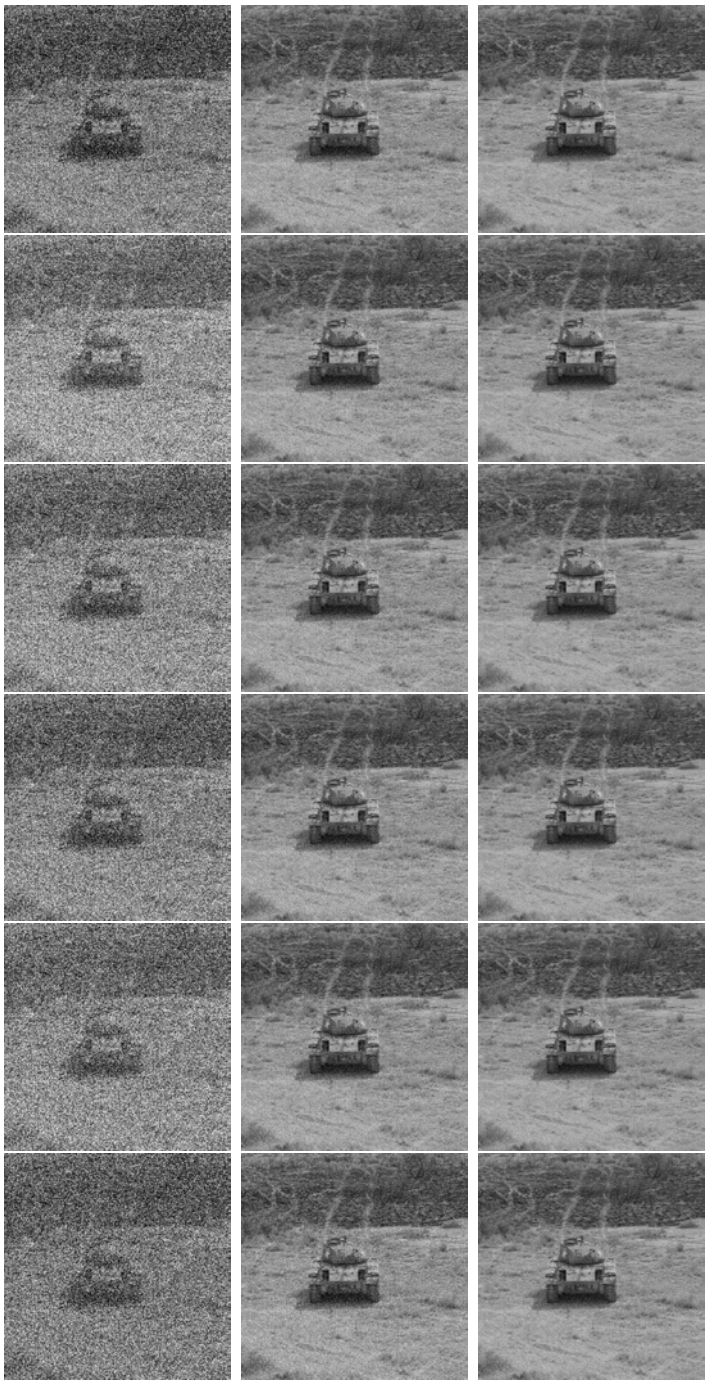


Figure 5.12: *Contaminated Tank images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

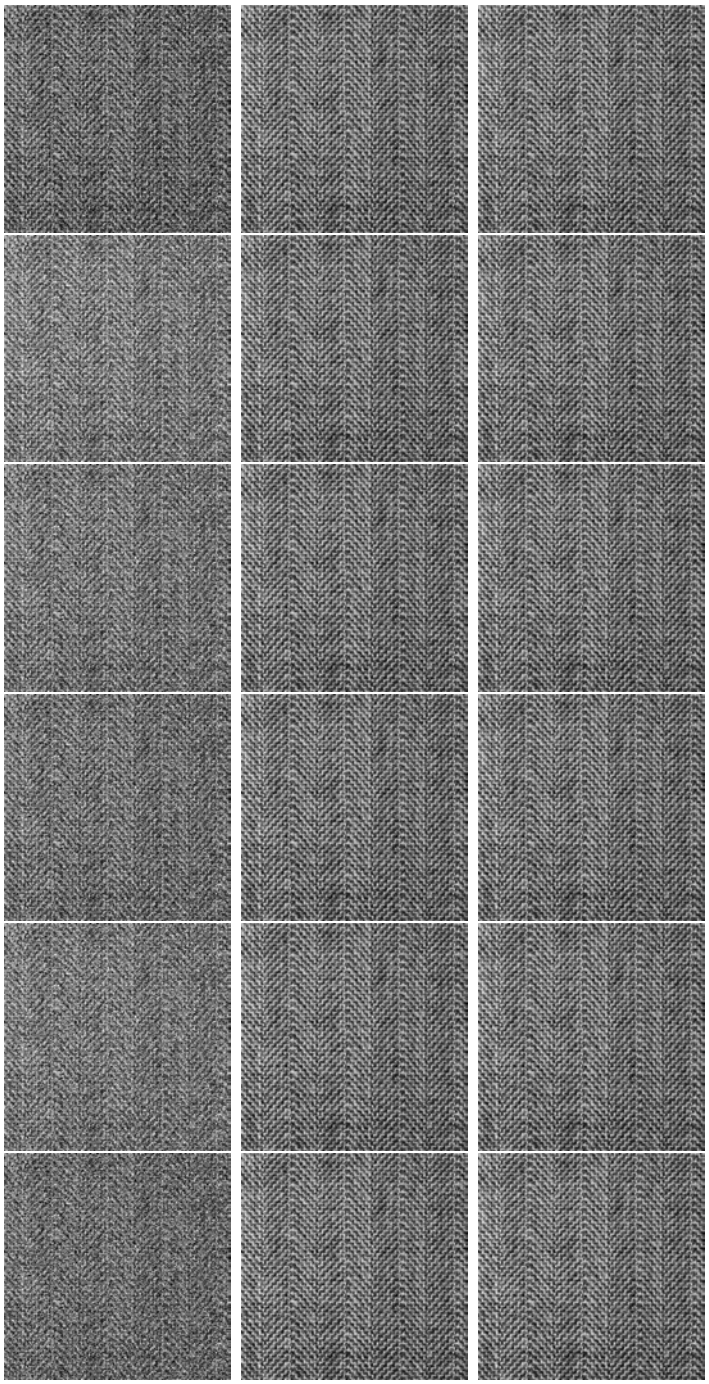


Figure 5.13: *Contaminated Fabric images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

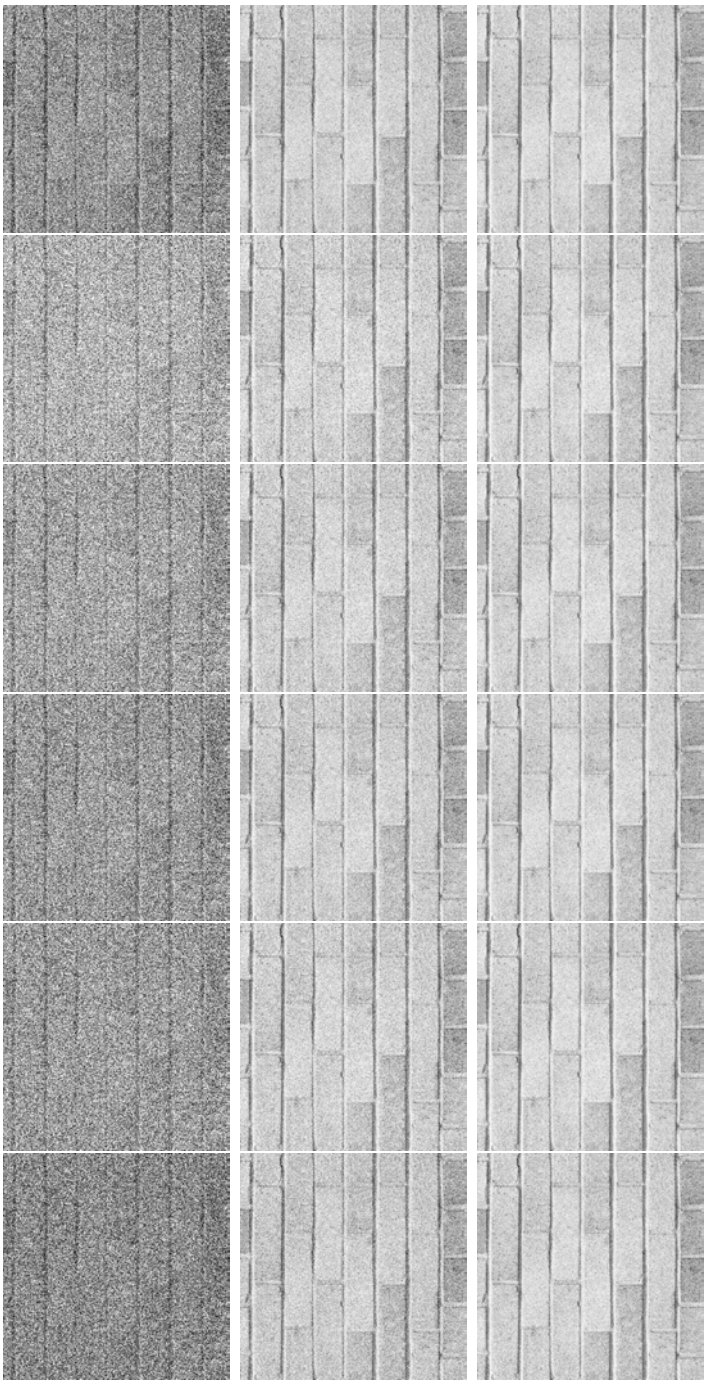


Figure 5.14: *Contaminated Brick images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

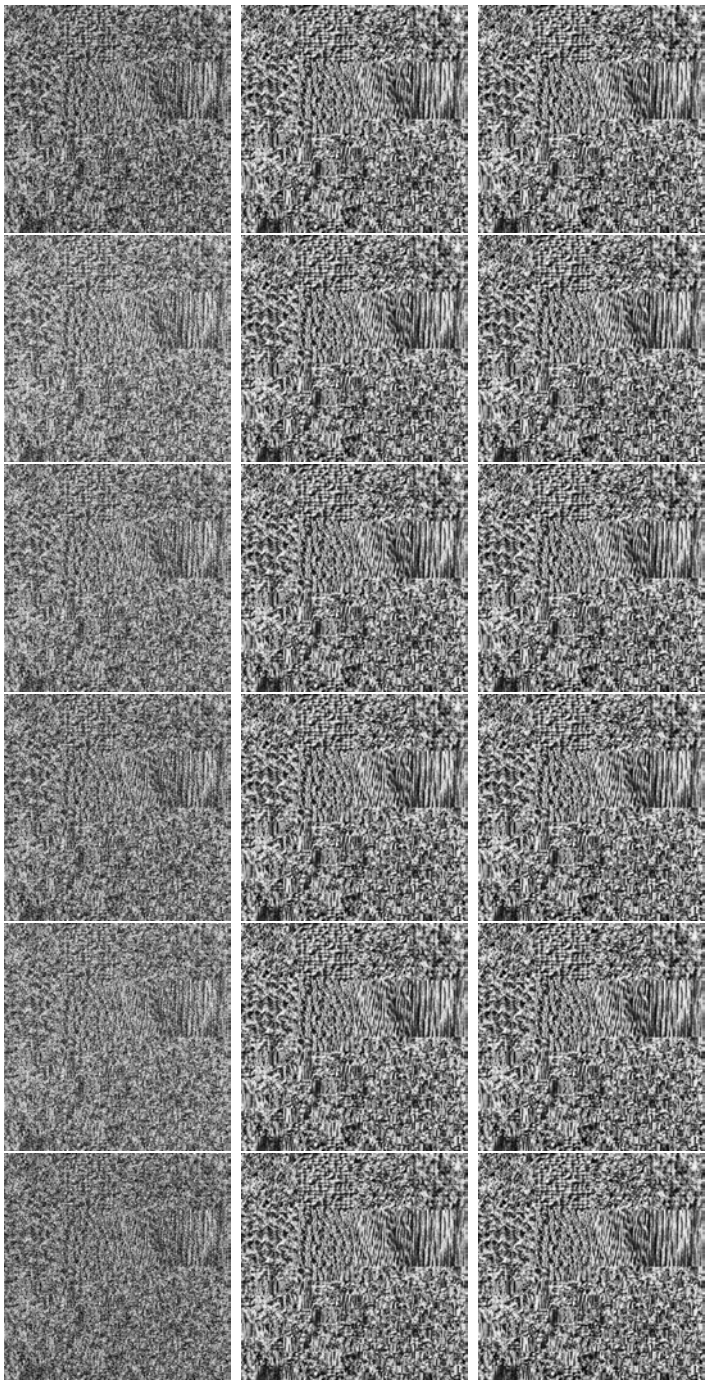


Figure 5.15: *Contaminated Texture images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

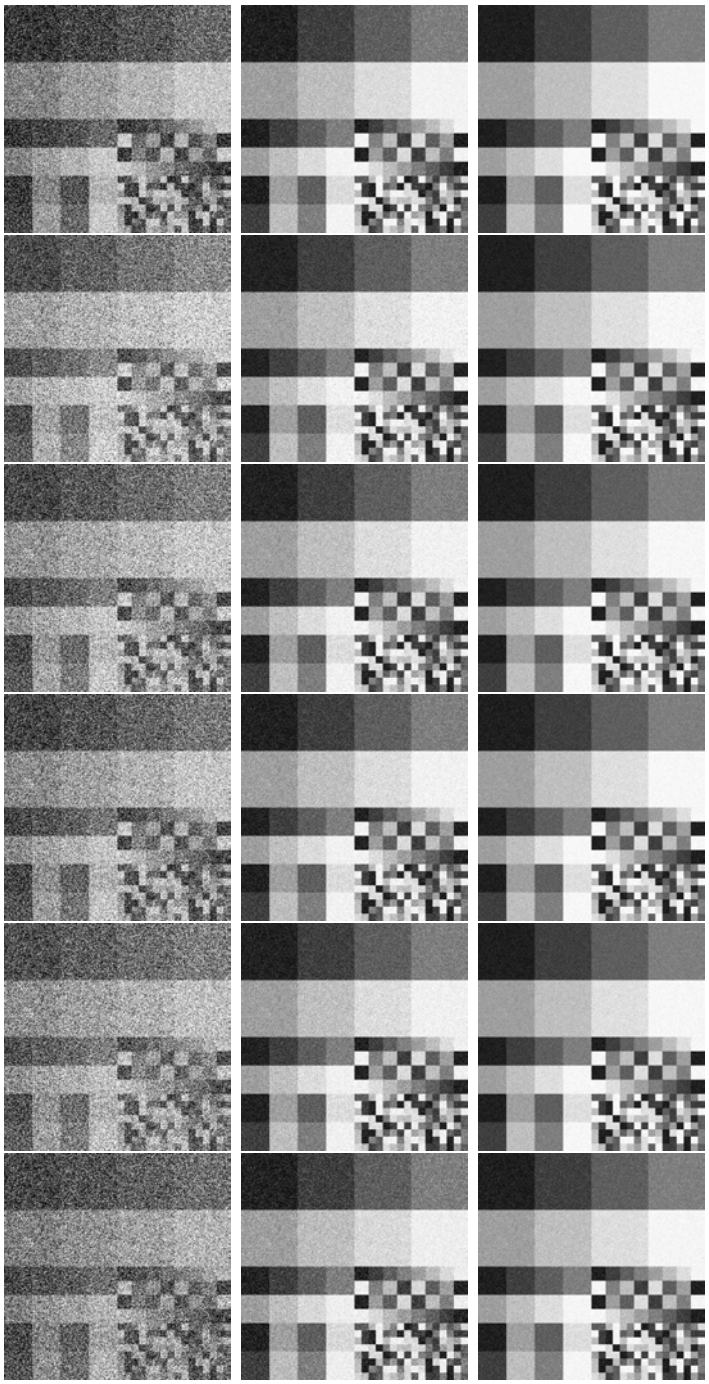


Figure 5.16: *Contaminated Blocks images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

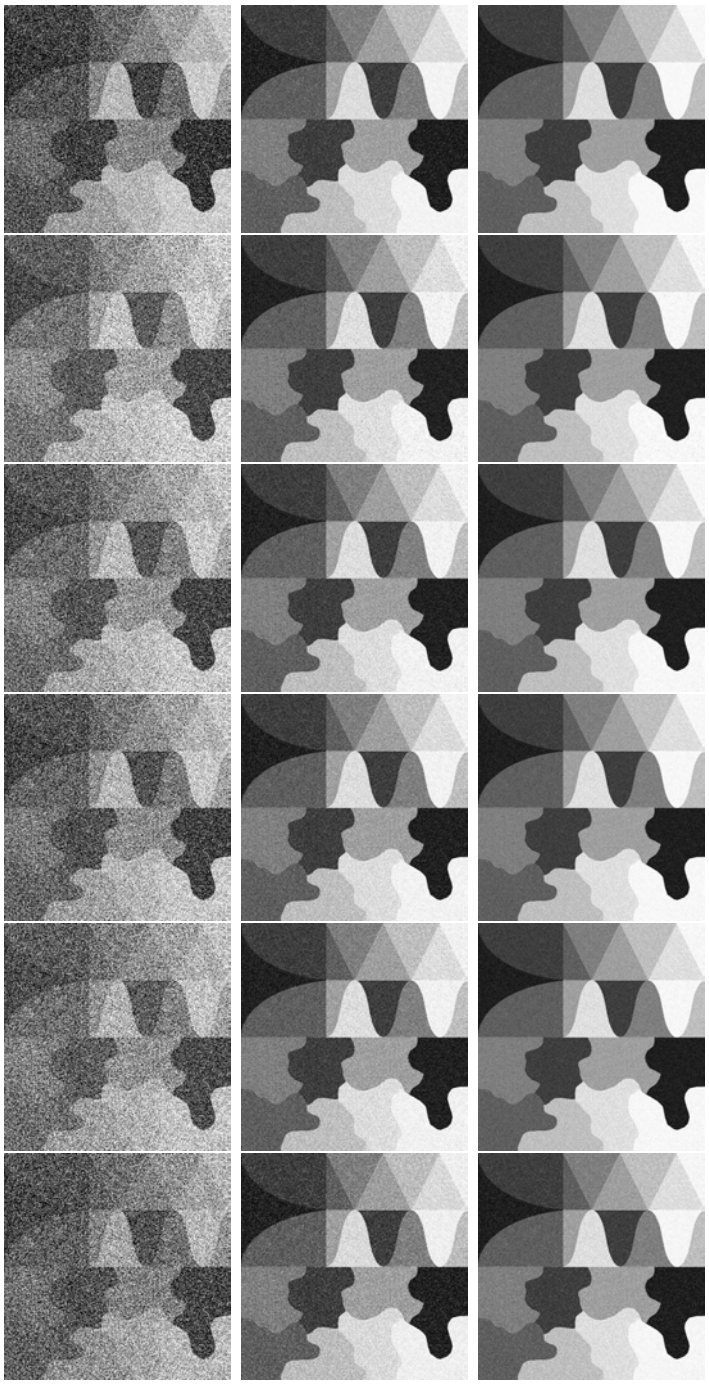


Figure 5.17: *Contaminated Regions images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

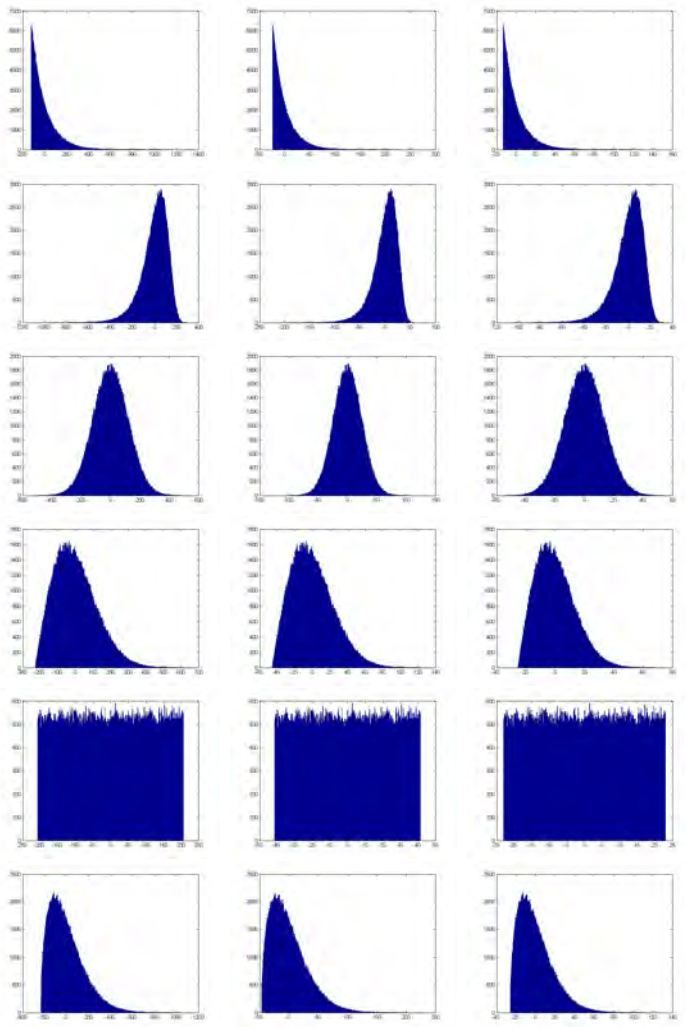


Figure 5.18: *Added noise, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

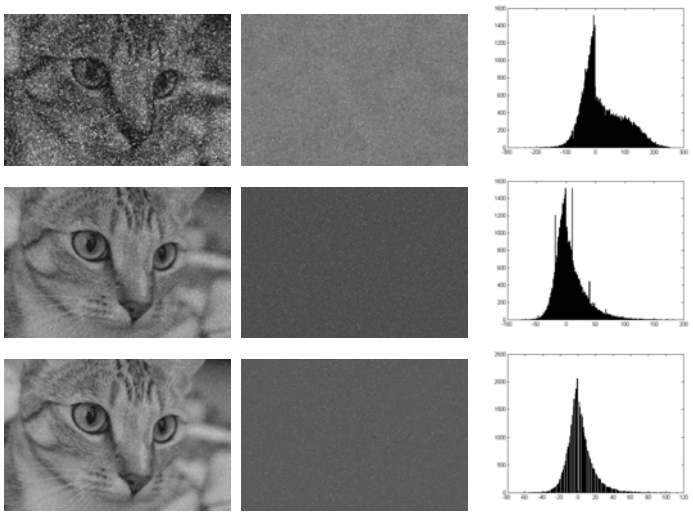


Figure 5.19: *Exponential contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

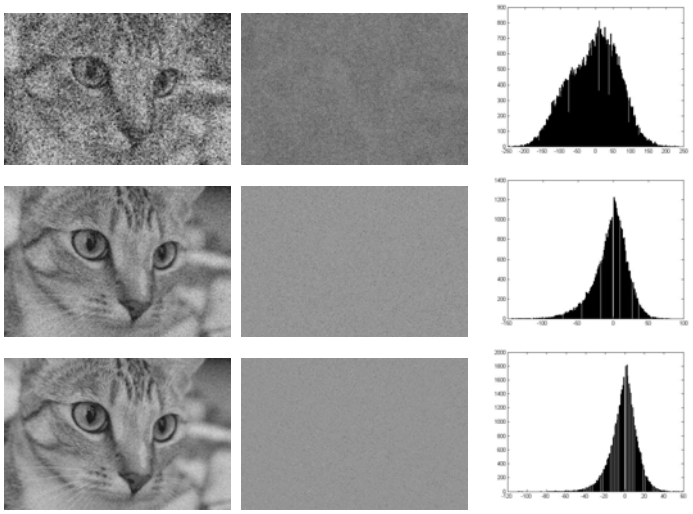


Figure 5.20: *Gumbel contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

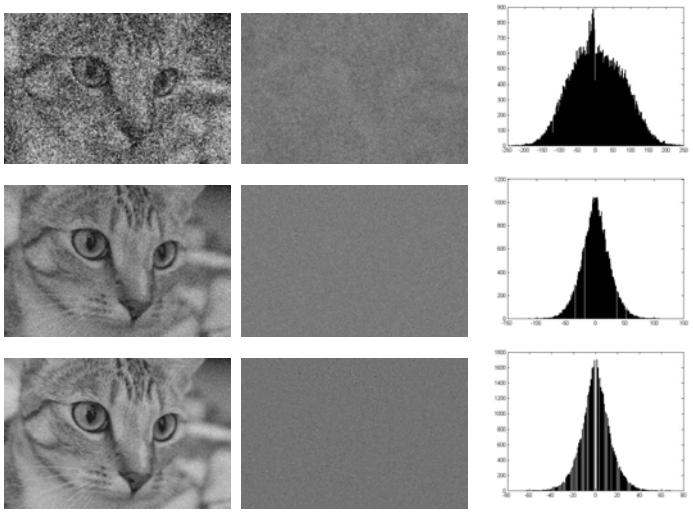


Figure 5.21: *Normal contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

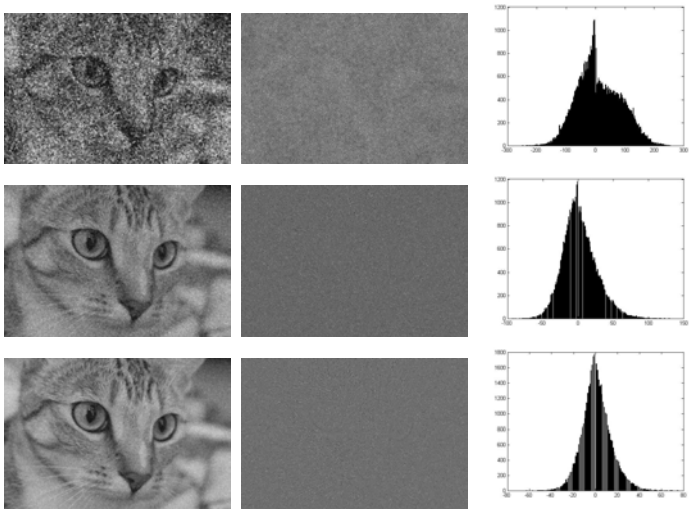


Figure 5.22: *Rayleigh contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

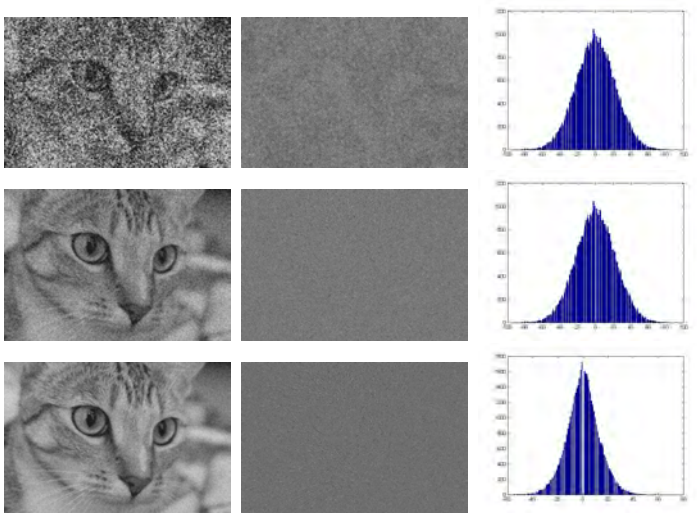


Figure 5.23: *Uniform contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

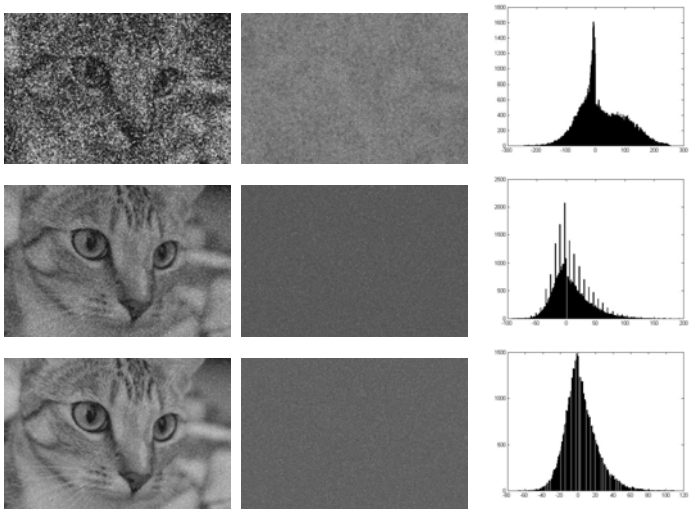


Figure 5.24: *Weibull contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

mensional distribution fitting. This method should prove much more effective since the relationships between neighbouring pixels and pulses will be captured. In addition, an intense look into noise types encountered in real images would improve the investigation into the noise removal ability of the two dimensional DPT. An ideal way to measure the noise removal in images may be by making use of the structural similarity index of Wang et al [235] discussed already in Chapter 3. Current work with junior postgraduate students involves building this research further.

5.5 Conclusion

In this chapter we looked at basic methods to improve the quality of an image before further analysis takes place. Sharpening an image, discussed in Section 5.2, reduces the number of pulses in the DPT thus reducing the computational complexity of the already heavy algorithm. Section 5.3 provides theory for the use of the LULU operators in approximating the true underlying signal from a noisy one. Although, the best approximation is not achieved, the approximation is near best and produces good visible results. Further reconstruction of the approximations can be obtained via image inpainting and various partial differential equation techniques, which essentially add previously unknown non-noise data points into the image, to reduce the 'blockiness' of LULU smoothed images and signals. The noise removal ability of the LULU operators, investigated in Section 5.4, is promising. Further detailed work needs to be done for the case of images but the results are still visibly good for signal-to-noise-ratio values likely to be observed in practice. A study into other types of noise encountered in all areas of practice in image processing would be very helpful in enabling further investigation. There is work done in this direction already.

Chapter 6

Conclusion

We have presented an investigation into the theoretical and practical usefulness of the LULU operators on multidimensional arrays and the resulting Discrete Pulse Transform. An initial summary of the grounding theory of the LULU operators was presented in Chapter 2. In Section 3.4 we provided an in depth look at the theoretical soundness of the nonlinear decomposition obtained via the recursive application of the LULU operators over increasing window or neighbourhood size, that is, the Discrete Pulse Transform. This work is original and new and has been written up as an article. Chapter 4 provides an extensive look at the concept of a scale-space which has become a very prominent phenomenon is image analysis literature. We delve into a history of scale-spaces and specifically the development of the most common scale-space, the Gaussian scale-space. In Section 4.6 we provide new theory on a missing link in the theory of scale-spaces, namely an axiomatic definition of a scale-space operator and the derived scale-space from this operator. In Section 4.7 we define the LULU scale-space related to the DPT and show that it aligns with the axiomatic theory in Section 4.6. Sections 4.8.2 and 4.8.3 as well as Chapter 5 investigate the practical significance of the LULU scale-space in feature detection, image segmentation and improving image quality, respectively, to ensure its usefulness in the fundamental requirements of image analysis. The work in these sections is original and new. Specifically the new results are:

- The consistent decomposition property of a nonlinear decomposition (Section 3.4.3).
- Measuring the ability of the LULU smoothers to smooth an array in

the sense expected (Section 3.4.5).

- Definition of a scale-space operator and scale-space (Section 4.6).
- The derivation of the LULU scale-space (Section 4.7).
- The relationship between sharpening an image and the DPT (Section 5.2).
- Best approximation theory for the LULU operators (Section 5.3).
- Noise removal ability of the LULU operators (Section 5.4).
- Useful feature point detection using the DPT (Section 4.8.2).
- Segmentation with the LULU scale-space (Section 4.8.3).

We have a number of publications from this work namely [8], [56], [53], [6], [7] and [54], and an article under review, [55]. In addition the following research is still open to investigation

1. In [244], it is described that any image can be represented as a linear combination of some basis. If a method can be determined such that most coefficients are zero when adjusted in some way the result is a compression. The bulk of the DPT pulses are in the lowest scales so by compressing these scales in some manner a useful LULU compression may be possible.
2. A faster, hopefully real-time, 1-, 2- and 3D implementation of the DPT is already underway.
3. The connectivity concepts discussed in Section 3.2 should also be investigated for improved image analysis with the DPT, specifically the multi-scale connectivity presented therein. An implementation of the DPT allowing for alternating connections will allow for investigation into ideas such as this.
4. Another idea for improving image quality as discussed in Section 5 is to add the pulses of the DPT back in order of some measure on the pulses instead of simply adding back full scale levels for partial image reconstruction. An optimal value for the level of smoothing n can be found with further research.

5. As mentioned in Chapter 4.8.2 the scale-space life-time and signatures may provide an indication of whether a pixel is noise, texture, small detail, large detail etc. Image segmentation taking this into account may prove fruitful.

6. A sound theoretical study of the distributional properties of the LULU operators in two and higher dimensions is still open for investigation. The method introduced in [33] for deriving the distributions of the output of the LULU operators on sequences will be investigated for an extension to $\mathbb{Z}^d$.

7. Pattern recognition and background modeling using the pulses of the DPT will also prove a fruitful exercise.

8. All items listed above should include comparisons to current state-of-the-art techniques in the various applications to fully ground LULU theory and the DPT within the image processing field.

Bibliography

[1] S. Acton and D. Mukherjee. Scale space classification using area morphology. *IEEE Transactions on Image Processing*, 9(4):623–635, 2000.

[2] C. R. Adams and J. A. Clarkson. Properties of functions $f(x, y)$ of bounded variation. *Transactions of American Mathematical Society*, 36(4), 1934.

[3] A. Almansa and T. Lindeberg. Fingerprint enhancement by shape adpatation of scale-space operators with automatic scale selection. *IEEE Transactions on Image Processing*, 9(12):2027–2042, 2000.

[4] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations in image processing. *Archive for Rational Mechanics and Analysis*, 123:199–257, 1993.

[5] R. Anguelov. *Constructive Theory of Functions*, chapter LULU Operators and Locally δ-Monotone Approximations, pages 22–34. Marin Drinov Academic Publishing House, Sofia, 2006.

[6] R. Anguelov and I. Fabris-Rotelli. Locally monotone approximations of real functions on graphs. In G. Nikolov and R. Uluchev, editors, *Constructive Theory of Functions*, pages 21–29, Sozopol 2010, 2012. Prof. Marin Drinov Academic Publishing House, Sofia. In memory of Borislav Bojanov.

[7] R. Anguelov and I. Fabris-Rotelli. On the axiomatization of scale-space theory. In M. Todorov, editor, *Application of Mathematics in Technical and Natural Sciences, 4th International Conference, AMiTaNS '12*, volume 1487, pages 159–167, St. Constantine and Helena, Bulgaria, 11-16 June 2012 2012. American Institute of Physics.

[8] R. Anguelov and I. N. Fabris-Rotelli. LULU operators and discrete pulse transform for multi-dimensional arrays. *IEEE Transactions on Image Processing*, 19(11):3012–3023, 2010.

[9] R. Anguelov and C. H. Rohwer. LULU operators for functions of continuous argument. *Quaestiones Mathematicae*, 32(2), 2009.

[10] G. Arfken and H. Weber. *Mathematical Methods for Physicists*. Elsivier Academic Press, 6th edition, 2005.

[11] A. Arias-Castro, E. Candés, and A. Durand. Detection of an anomalous cluster in a network. *The Annals of Statistics*, 39(1):278–304, 2011.

[12] J. Babaud, A. Witkin, M. Baudin, and R. Duda. Uniqueness of the gaussian kernel for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):26–33, 1986.

[13] H. Barlow. The efficiency of detecting changes of density in random dot patterns. *Vision Research*, 18:637–650, 1978.

[14] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. SIAM, 2nd edition, 1994.

[15] H. Barrow and Tenenbaum. Recovering intrinsic scene characteristics from images. Techincal Report 157, SRI International, April 1978.

[16] N. Berline, E. Getzler, and M. Vergne. *Heat Kernels and Dirac Operators*. Spinger-Verlag, Berlin, New York, 2004.

[17] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, pages 312–321, San Antonio, Texas, USA, 2002.

[18] R. N. Bracewell. *The Fourier Transform and its Application*. McGraw-Hill, Kogakusha, 1978.

[19] J. Bradley and C. Brislawn. Image compression by vector quantization of multiresolution decompositions. *Physica D*, 60:245–258, 1992.

[20] U. Braga-Neto. Multiscale connected operators. *Journal of Mathematical Imaging and Vision*, 22:199–216, 2005.

[21] U. Braga-Neto. Object-based image analysis using multiscale connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):892–907, 2005.

[22] U. Braga-Neto and J. Goutsias. An axiomatic approach to multiscale connectivity in image analysis. Technical report, The Johns Hopkins University, Baltimore, April 2001.

[23] U. Braga-Neto and J. Goutsias. Connectivity on complete lattices: New results. *Computer Vision and Images Understanding*, 85:22–53, 2002.

[24] U. Braga-Neto and J. Goutsias. A multiscale approach to connectivity. *Computer Vision and Image Understanding*, 89:70–107, 2003.

[25] U. Braga-Neto and J. Goutsias. A theoretical tour of connectivity in image processing and analysis. *Journal of Mathematical Imaging and Vision*, 19:5–31, 2003.

[26] U. Braga-Neto and J. Goutsias. Grayscale level connectivity: Theory and applications. *IEEE Transactions on Image Processing*, 13(12):1567–1580, 2004.

[27] U. Braga-Neto and J. Goutsias. Constructing multiscale connectivities. *Computer Vision and Image Understanding*, 99:126–150, 2005.

[28] R. W. Brockett and P. Maragos. Evolution equations for continuous-scale morphological filtering. *IEEE Transactions on Image Processing*, 42(12):3377–3386, 1994.

[29] M. Bulmer. Music from fractal noise. In *Proceedings of the Mathematics 2000 Festival*, Melbourne, 10-13 January 2000 2000.

[30] R. Burden and J. Faires. *Numerical Analysis*. Brooks/Cole, CA, USA, 2001.

[31] P. Burt. Fast filter transfroms for images processing. *Computer Graphics and Image Processing*, 16(1):20–51, 1981.

[32] P. Burt and T.-H. H. A. Rosenfeld. Segmentation and estimation of image region propertires through cooperative hierarchical computation. *IEEE Transactions*, SMC-II:802–825, 1981.

[33] P. Butler. The transfer of distributions of LULU smoothers. Masters thesis, University of Stellenbosch, 2008.

[34] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76:167–188, 1997.

[35] Y. Chi and M. Leung. A general shape context framework for object identification. *Computer Vision and Image Understanding*, 112:324–336, 2008.

[36] J. A. Clarkson and C. R. Adams. On definitions of bounded variation for functions of two variables. *Transactions of American Mathematical Society*, 35(4), 1933.

[37] M. Cohen and S. Grossberg. Neural dynamics of brightness perception: features, boundaries, diffusion, and resonance. *Attention, Perception, & Psychophysics*, 36(5):428–456, 1984.

[38] W. Conradie, T. de Wet, and M. Jankowitz. An overview of LULU smoothers with application to financial data. *Journal for Studies in Economics and Econometrics*, 29(1):97–121, 2005.

[39] R. Courant and D. Hilbert. *Methoden der mathematischen en physik*. Spinger-Verlag, Berlin, 4th edition, 1993.

[40] R. Coutinho, D. Selviah, and H. Griffiths. Theoretical modeling of a detection system based on optical coherence contrast. In W. Watkins, D. Clement, and W. Reynolds, editors, *Targets and Backgrounds X: Characterization and Representation*, volume 5431 of *Proceedings of SPIE*, pages 24–35, 2004.

[41] J. Crespo and R. Schafer. Locality and adjacency stability constraints for morphological connected operators. *Journal of Mathematical Imaging and Vision*, 7:85–102, 1997.

[42] J. Crespo, J. Serra, and R. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 47:201–225, 1995.

[43] J. Crowley. *A representation for visual information*. Phd thesis, Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennysylvania, 1981.

[44] J. L. Crowley. *A representation for visual information*. Technical report number cmu-ri-tr-82-7, Robotics Institute, Carnegie-Mellon Unievrsity, Pittsburgh, Pa., 1982.

[45] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, XLI:909–996, 1988.

[46] P. Debba, A. Stein, F. van der Merwe, E. Carranza, and A. Lucieer. Field sampling from a segmented image. In *Proceedings of the International Conference on Computational Science and its Applications, Part I*, pages 756–768, Perugia, Italy, 30 June - 3 July 2008. Springer-Verlag.

[47] D. Donoho and I. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[48] T. Duda and M. Canty. Unsupervised classification of satellite imagery: choosing a good algorithm. *International Journal of Remote Sensing*, 23(11):2193–2212, 2002.

[49] I. Duff, R. Grimes, and J. Lewis. Sparse matrix test problems. *ACM Transactions on Mathematical Software*, 15:1–14, 1989.

[50] Editorial. Special issue: Shape representation and similarity for image databases. *Pattern Recognition*, 35:1–2, 2002.

[51] J. Elder and S. Zucker. Scale space localization, blur, and contour-based image coding. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR '96*, pages 27–34, San Fransico, CA, USA, 18-20 June 1996.

[52] I. Fabris-Rotelli. LULU operators on multidimensional arrays and applications. Masters dissertation, University of Pretoria, November 2009.

[53] I. Fabris-Rotelli. The discrete pulse transform for images with entropy-based feature detection. In P. Robinson and A. Nel, editors, *Proceedings of the 22nd Annual Symposium of the Pattern Recognition Association of South Africa*, pages 43–48. PRASA, 22-25 November 2011.

[54] I. Fabris-Rotelli. Characterization of the Discrete Pulse Transform of images. *International Journal on Mathematical Methods and Models in Biosciences*, 1(2), 2012.

[55] I. Fabris-Rotelli and S. van der Walt. Applications and memory efficient implementation of the two-dimensional Discrete Pulse Transform. Under review, December 2010.

[56] I. Fabris-Rotelli, K. van Oldenmark, and P. van Staden. Evaluation of noise removal in signals by LULU operators. In P. Debba, F. Lombard, V. Yadavalli, and L. Fatti, editors, *Peer-reviewed Proceedings of the 52nd Annual Conference of the South African Statistical Association*

for 2010 (SASA 2010), pages 44–51, North-West University: Potchefstroom Campus, 10-12 November 2010. SASA.

[57] P. Falzenszwalb and D. Huttenlocher. Image segmentation using local variation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, 1998.

[58] M. Felsberg. *Low-level image processing with the structures multivector*. Phd thesis, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel Olshausenstr. 40 D 24098 Kiel, March 2002.

[59] M. Ferraro, G. Boccignone, and T. Caelli. On the representation of image structures via scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1–5, 1999.

[60] C. Fiorio and J. Gustedt. Two linear time union-find strategies for image processing. *Theoretical Computer Science*, 154(2):165–181, 1996.

[61] L. Florack. Data, models, and images. volume I, pages 469–472, Lausanne, 16-19 September 1996. 1996 IEEE International Conference on Image Processing, ICIP '96.

[62] L. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12:65–79, 2000.

[63] L. Florack, R. Maas, and W. Niessen. Pseudo-linear scale-space theory. *International Journal of Computer Vision*, 31(2/3):247–259, 1999.

[64] L. Florack, B. Romeny, J. Koenderink, and M. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376 – 388, 1992.

[65] L. Florack, A. Salden, B. ter Haar Romeny, and J. K. andMA Viergever. Nonlinear scale-space. *Image and Vision Computing*, 13(4):279–294, 1995.

[66] P. A. Foerster. *Precalculus with Trigonometry: Concepts and Connections*. Birkhuser, 2002.

[67] J. Fourier. *Théorie analytique de la chaleur*. Translated to English by A Freeman in 1878, reissue by Cambridge University Press in 2009, 1822.

[68] D. Gabor. Theory of communication. *Journal of the Institute of Electrical Engineers Part III*, 93:429–457, 1946.

[69] G. Gagaudakis and P. Rosin. Incorporating shape into histograms for CBIR. *Pattern Recognition*, 35:81–91, 2002.

[70] I. Gath and A. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–781, 1989.

[71] J. Gilles. Image decoposition: Theory, numerical schemes, and performance evaluation. *Advances in Imaging and Electron Physics, Chapter 8*, 158:89–137, 2009.

[72] G. Granlund. In search of a general picture processing operator. *Computer Vision, Graphics, and Image Processing*, 8:155–173, 1978.

[73] M. Greated. The nature of sound and vision in relation to colour. *Optics & Laser Technology*, 2011(43):337–347, 2011.

[74] J. Hadamard. *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, 1923.

[75] R. Hammah and J. Currah. On distance measures for the fuzzy k-means algorithm for joint data. *Rock Mechanics and Rock Engineering*, 32(1):1–27, 1999.

[76] A. R. Hansen and E. M. Riseman. Processing cones: A parallel computational structure for scene analysis. Technical report, University of Massachusetts, Amherst, Massachusetts, 1974.

[77] G. Hay and M. Chesters. A model of visual threshold detection. *Journal of Theoretical Biology*, 67:221–240, 1977.

[78] H. Heijmans and A. Goutsias. Nonlinear multiresolution signal decomposition schemes - part ii: Morphological wavelets. *IEEE Transactions on Image Processing*, 9(11):1897–1913, 2000.

[79] I. Herstein. *Topics in Algebra*. Wiley, second edition, 1975.

[80] D. Hilbert. Grundzüge einer allgemeinen theorie der linearen integralgleichungen. Nachrichten der Göttinger Akademie der Wissenschaften, Mathematisch-Physikalische Klasses, 1904.

[81] R. Hobson. Properties preserved by some smoothing functions. *Journal of the American Statistical Association*, 71(355):763–766, 1976.

[82] A. Hoover, G. Jean-Baptiste, D. Goldgof, and K. Bowyer. A methodology for evaluating range image segmentation techniques. In *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*, pages 264–271, Sarasota, FL, USA, 5-7 December 1994.

[83] C.-T. Hsu and J.-L. Wu. Multiresolution watermarking for digital images. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, 45(8):1097–1101, 1998.

[84] D. H. Hubel. *Eye, Brain, and Vision.* Scientific American Library, 1988.

[85] R. Hummel. *The Scale-Space Formulation of Pyramid Data Structures*, pages 187–223. Parallel Computer Vision. Academic Press, New York, 1987.

[86] T. Iijima. Basic theory on normalization of pattern (in case of typical one-dimensional pattern). *Bulletin of the Electrotechnical Laboratory*, 26:368–388, 1962. in Japanese.

[87] T. Iijima. Basic theory on normalization of two-dimensional visual pattern. *Studies on Information and Control, Pattern Recognition Issue, IECE, Japan,*, 1:15–22, 1963. in Japanese.

[88] T. Iijima. Theory of pattern recognition. *Electronics and Communications in Japan*, pages 123–134, November 1963. in English.

[89] T. Iijima. Basic equation of figure and observational transformation. *Systems, Computers, Controls*, 2(4):70–77, 1971. in English.

[90] T. Iijima. Basic theory on normalization of figures. *Electronics and Communications in Japan*, 54-C(12):106–112, 1971. in English.

[91] T. Iijima. Basic theory on the construction of figure space. *Systems, Computers, Controls*, 2(5):51–57, 1971. in English.

[92] T. Iijima. A suppression kernel of a figure and its mathematical characteristics. *Systems, Computers, Controls*, 2(6):16–23, 1971. in English.

[93] T. Iijima. A system of fundamental functions in an abstract figure space. *Systems, Computers, Controls*, 2(6):96–103, 1971. in English.

[94] T. Iijima. Basic theory of feature extraction for figures. *Systems, Computers, Control*, 3(1):32–39, 1972. in English.

[95] T. Iijima. Basic theory on the structural recognition of figures. *Systems, Computers, Control*, 3(4):30–36, 1972. in English.

[96] T. Iijima. Theoretical studies on the figure identification by pattern matching. *Systems, Computers, Control*, 3(4):37–44, 1972. in English.

[97] T. Iijima. Basic theory of pattern observation. *Papers of Technical Group on Automata and Automatic Control, IECE, Japan*, December 1959. in Japanese.

[98] T. Iijima. Observation theory of two-dimensional visual patterns. *Papers of Technical Group on Automata and Automatic Control, IECE, Japan*, October 1962. in Japanese.

[99] P. T. Jackway. Morphological scale-space. In *Proceedings of 11th IAPR International Conference on Image, Speech and Signal Analysis, Pattern Recognition*, volume III, 1992.

[100] M. Jankowitz. *Some Statistical Aspects of LULU Smoothers*. PhD thesis, Department of Statistics and Actuarial Science, Stellenbosch University, South Africa, 2007.

[101] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, 2002.

[102] R. Jones. Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding*, 75(3):215–228, 1999.

[103] B. Julesz. Textons, the elements of texture perception, and their interaction. *Nature*, 290:91–97, 1981.

[104] B. Julesz. *Dialogues on Perception*. MIT Press, Cambridge, MA, 1995.

[105] A. Kadir and M. Brady. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[106] O. Kao. Modification of the LULU operators for preservation of critical image details. In *Proceedings of CISST'02: International Conference on Imaging Science, Systems and Technology*, Las Vegas, Nevada, USA, 24-27 June 2002 2002.

[107] R. Kessel. Estimating the limitations that image resolution and contrast place on target recognition. In F. Sadjadi, editor, *Automatic Target Recognition XII, Proceedings of SPIE*, volume 4726, pages 316–327, 2002.

[108] A. Klinger. *Pattern and Search Statistics*. Optimizing Methods in Statistics. Academic Press, New York, 1971.

[109] K. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[110] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[111] A. Kuijper, L. Florack, and M. Viergever. Scale space hierarchy. *Journal of Mathematical Imaging and Vision*, 18:169–189, 2003.

[112] D. Laurie. The roadmaker's algorithm for the Discrete Pulse Transform. *IEEE Transactions on Image Processing*, 20(2):361–371, 2011.

[113] D. Laurie and C. Rohwer. The Discrete Pulse Transform. *SIAM Journal of Mathematical Analysis*, 38(3), 2007.

[114] D. P. Laurie and C. H. Rohwer. Fast implementation of the Discrete Pulse Transform. In *Extended Abstracts of ICAAM 2006: International Conference of Numerical Analysis and Applied Mathematics*, pages 484–487, Crete, Greece, 15-19 September 2006.

[115] Y. Leung, J.-S. Zhang, and Z.-B. Xu. Clustering by scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1396–1410, 2000.

[116] Q. Li and J. S. Racine. *Nonparametric Econometrics: Theory and Practice*. Princeton University Press, New Jersey 08540, Oxfordshire OX20 1SY, 2007.

[117] Z. Li and M. J. Crocker. *Handbook of Noise and Vibration Control*, chapter 41: Equipment for Data Aquisition, pages 486–492. John Wiley & Sons, Inc., 2007.

[118] L. Lifshitz and S. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–540, 1990.

[119] C.-L. Lin, T. Chuang, and K.-C. Fan. Palmprint verification using hierarchical decomposition. *Pattern Recognition*, 38:2639–2652, 2005.

[120] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990.

[121] T. Lindeberg. *Discrete Scale Space Theory and the Scale Space Primal Sketch*. Phd thesis, Department of Numerica Analysis and Computing Science, Royal Institute of Technology, Stockholm, May 1991.

[122] T. Lindeberg. On scale selection for differential operators. In *Proceedings of the 8th Scandinavian Conference on Image Analysis*, pages 857–866, Tromsø, Norway, 1993.

[123] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[124] T. Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings of CERN School of Computing*, number 8, pages 27–38, Egmond aan Zee, The Netherlands, 8-21 September 1996.

[125] T. Lindeberg. *Gaussian Scale-Space Theory*, chapter On the Axiomatic Formulations of Linear Scale-Space, pages 75–97. Kluwer, Dordrecht, 1997.

[126] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2), 1998.

[127] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2), 1998.

[128] T. Lindeberg. *Scale-Space*, volume IV of *Encyclopedia of Computer Science*, pages 2495–2504. John Wiley and Sons, Hoboken, New Jersey, 2009.

[129] T. Lindeberg and K. V. Mardia. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1/2):225–271, 1994.

[130] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[131] A. K. Louis and P. Maass. A mollifier method for linear operator equation of the first kind. *Inverse Problems*, 6:427–440, 1990.

[132] S. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2091–2110, 1989.

[133] S. Mallat. A theory for multiresolution signal decoposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[134] S. Mallat and Z. Zhang. Matching persuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[135] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):701–716, 1989.

[136] D. Marr. Early processing of visual information. *Philosophical Transactions of the Royal Society of London, Series B, Biological Sciences*, 275(942):483–519, October 1976.

[137] D. Marr. Artificial intelligence - a personal review. *Artificial Intelligence*, 9:37–48, 1977.

[138] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W H Freeman and Company, 1982.

[139] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B*, 207:187–217, 1980.

[140] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London, Series B, Biological Sciences*, 204(1156):301–328, 1979.

[141] H. Mayer and C. Steger. Scale-spcae events and their link to abstracttion for road extraction. *Journal of Photogrammetry and Remote Sensing*, 53:67–75, 1998.

[142] N. McLachlan. *Bessel Functions for Engineers*. Oxford University Press, London, 1955.

[143] P. Meer, E. S. Baugher, and A. Rosenfeld. Frequency domain analysis and synthesis of image pyramid generating kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:512–522, 1987.

[144] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society, London*, A290:415–446, 1909.

[145] F. Meyer, A. Averbuch, and R. Coifman. Multilayered image representation: Application to image compression. *IEEE TRansaction on Image Processing*, 11(9):1072–1080, 2002.

[146] F. Meyer and P. Maragos. Nonlinear scale-space representation with morphological levelings. *Journal of Visual Communication and Image Representation*, 11:245–265, 2000.

[147] Y. Meyer. *Wavelets and Operators*. Cambridge University Press, 1993.

[148] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Sonar image segmentation using an unsupervised hierarchical mrf model. *IEEE Transactions on Image Processing*, 9(7):1216–1231, 2000.

[149] P. Mishra and B. Jenkins. Hierarchical feature extraction and object recognition based on biologically inspired filters. In *Proceedings of SPIE, IS&T/SPIE Electronic Imaging Conference*, volume 753805-5, California, USA, 17-21 January 2010.

[150] F. Mokhtarian and S. Abbasi. Shape similarity retrieval under affine transforms. *Pattern Recognition*, 35:31–41, 2002.

[151] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proceedings of the British Machine Vision Conference*, pages 53–62, 1996.

[152] F. Mokhtarian, S. Abbasi, and J. Kittler. *Efficient and Robust Retrieval by Shape Content through Curvature Scale Space*, pages 51–58. Series on Siftware Engineering and Knowledge Engineering. World Scientific Publishing, 1997.

[153] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.

[154] F. Murtagh and J.-L. Starck. Image processing through multiscale analysis and measurement noise modeling. *Statistics and Computing*, 10:95–103, 2000.

[155] E. Nakamura and N. Kehtarnavaz. Determining number of clusters and prototype locations via multi-scale clustering. *Pattern Recognition Letters*, 19:1265–1283, 1998.

[156] U. Neisser. Visual search. *Scientific American*, 210(6):94–102, 1964.

[157] M. Nielson, L. Florack, and R. Deriche. Regularization, scale-space, and edge detection filters. In B. Baxton and R. Cipolla, editors, *Computer Vision - ECCV '96*, volume 1065, pages 70–81. Lecture Notes in Computer Science, Kluwer, Berlin, 1996.

[158] L. Nirenberg. A strong maximum principle for parabolic equations. *Communications on Pure and Applied Mathematics*, 6(2):167–177, 1953.

[159] F. Olver, D. Lozier, R. Boisvert, and C. Clark, editors. *NIST Handbook of Mathematical Functions*. Cambridge University Press, New York, first edition, 2010.

[160] S. Oscher, A. Solé, and L. Vese. Image decomposition and restoration using total variation minimization and the h^{-1} norm. *Multiscale Modeling and Simulation*, 1(3):349–370, 2003.

[161] N. Otsu. *Mathematical Studies on Feature Extraction in Pattern Recognition*. Report number 818, phd, Electrotechnical Laboratory, 1-1-4, Umezono, Sakura-mura, Niihari-gun, Ibaraki, Japan, 1981. in Japanese.

[162] G. Ouzounis. *Generalized connected morphological operators for robust shape extraction*. Phd thesis, University of Groningen, 2009.

[163] G. Ouzounis and M. Wilkinson. Countering oversegmentation in partitioning-based connectivities. In *Proceedings of the IEEE International Conference on Image Processing*, volume III, pages 844–847, 2005.

[164] A. B. Owen. *Multidimensional Variation for Quasi-Monte Carlo*, chapter Contemporary Multivariate Analysis and Design of Experiments. 2005.

[165] N. Pal, K. Pal, J. Keller, and J. Bezdek. A possibilistic fuzzy c-means clustering algorithm. *IEEE Transactions on fuzzy systems*, 13(4):517–530, 2005.

[166] A. Papoulis. *The Fourier Integral and its Applications*. McGraw-Hill Book Company, reissue edition, 1987.

[167] T. Parrish, D. Gitelman, K. LaBar, and M.-M. Mesulam. Impact of signal-to-noise on functional MRI. *Magnetic Resonance in Medicine*, 44:925–932, 2000.

[168] E. J. Pauwels, L. van Gool, P. Fiddelaers, and T. Moons. An extended class of scale-invariant and recursive scale-space filters. *IEEE TRansactions on Pattern Analysis and Machine Intelligence*, 17:691–701, 1995.

[169] A. Pentland. Shape information from shading: A theory about human perception. *Spatial Vision*, 4(2-3):165–182, 1989.

[170] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[171] A. Petrovic, O. Escoda, and P. Vandergheynst. Multiresolution segmentation of natural images: from linear to nonlinear scale-space representations. *IEEE Transactions on Image Processing*, 13(8):1104–1114, 2004.

[172] I. Pitas and A. N. Venetsanopoulos. Order statistics in digital image processing. *Proceedings of the IEEE*, 80(12):1893–1921, 1992.

[173] C. Podilchuk and W. Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on Selected Areas in Communications*, 16(4):525–539, 1998.

[174] T. Poggio and V. Torre. Ill-posed problems and regularization analysis in early vision. Technical Report A.I. Memo 773, C.B.I.P. Paper 001, Massachusetts Institute of Technology, AI Laboratory and Center for Biological Information Processing, Whitaker College, 1984.

[175] L. Qi and D. Sun. Smoothing functions and a smoothing newton method for complementarity and variational inequality problems. *Journal of Optimization and Variational Inequality Problems*, 113(1):121–147, 2002.

[176] M. Ramos, S. Hemami, and M. Tamburro. Psychovisually-based multiresolution image segmentation. In *International Conference on Image Processing*, volume 3, pages 60–69, Santa Barbara, CA, USA, 26-29 October 1997.

[177] A. Rattarangsi and R. Chin. Scale-based detection of corners of planar curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(4):430–449, 1992.

[178] F. Relton. *Applied Bessel Functions*. Blackie & Son Limited, London and Glasgow, 1946.

[179] A. Restrepo and A. Bovik. Locally monotone regression. *IEEE Transactions on Signal Processing*, 41(9):2796–2810, 1993.

[180] O. Rioul. A discrete-time multiresolution theory. *IEEE Transactions on Signal Processing*, 41(8):2591–2606, 1993.

[181] C. Rohwer. Fast approximation with locally monotone sequences. *Rend. Circ. Mat. Palermo*, 2 Suppl.(68), 2002.

[182] C. Rohwer. Fully trend preserving operators. *Quaestiones Mathematicae*, 27:217–230, 2004.

[183] C. Rohwer. *Nonlinear Smoothers and Multiresolution Analysis*. Birkhäuser, 2005.

[184] C. Rohwer and M. Wild. LULU theory, idempotent stack filters, and the mathematics of vision of marr. *Advances in Imaging and Electron Physics*, 146:57–162, 2007.

[185] C. H. Rohwer. Variation reduction and LULU-smoothing. 25:163–176, 2002.

[186] C. Ronse. Partial partitions, partial connections and connective segmentation. *Journal of Mathematical Imaging and Vision*, 32:97–125, 2008.

[187] C. Ronse and J. Serra. Geodesy and connectivity in lattices. *Fundamenta Informaticae*, 46:349–395, 2001.

[188] A. Rosenfeld. Fuzzy digital topology. *Information and Control*, 40:76–87, 1979.

[189] A. Rosenfeld. On connectivity properties of grayscale pictures. *Pattern Recognition*, 16:47–50, 1983.

[190] A. Rosenfeld. The fuzzy geometry of image subsets. *Pattern Recognition Letters*, 2:311–317, 1984.

[191] A. Rosenfeld and M. Thurston. Edge and curve detetction for visual scene analysis. *IEEE Transactions on Computers*, C-20(5):562–569, May 1971.

[192] L. Rudin and V. Caselles. Image recovery via multiscale total variation. In *Proceedings of the Second European Conference on Image Processings*, Palma, Spain, 1995.

[193] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[194] T. Sakai and A. Imiya. Gradient structure of image in scale space. *Journal f Mathematical Imaging and Vision*, 28:243–257, 2007.

[195] T. Sakai and A. Imiya. Unsupervised cluster discovery using statistics in scale space. *Engineering Applications of Artificial Intelligence*, 22:92–100, 2009.

[196] E. Salari and Z. Ling. Texture segmentation using hierarchical wavelet decomposition. *Pattern Recognition*, 28(12):1819–1824, 1995.

[197] P. Salembier and A. Oliveras. *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 5 of *Computational Imaging and Vision*, chapter Practical Extensions of Connected Operators. Springer, 1996.

[198] P. Salembier and A. Oliveras. *Practical extensions of connected operators*, pages 97–110. Mathematical Morphology and its Applications to Image and Signal Processing. Kluwer Academic, Dordrecht, 1996.

[199] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4):555–570, 1998.

[200] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filers by reconstruction. *IEEE Transactions on Image Processing*, 4(8):1153–1160, August 1995.

[201] P. Salembier and M. H. F. Wilkinson. Connected operators: A review of region-based morphological image processing techniques. *IEEE Signal Processing Magazine*, 136:136–157, November 2009.

[202] H. Samet. The quadtree and related hierarchical data structures. *Computing Surveys*, 16(2):187–260, 1984.

[203] I. Santillán, A. M. Herrera-Navarro, J. D. Mendiola-Sahtibá nez, and I. R. Terol-Villalobos. Morphological connected filtering on viscous lattices. *Journal of Mathematical Imaging and Vision*, 36:254 – 269, 2010.

[204] J. Schavemaker, M. Reinders, J. Gerbrands, and E. Backer. Image sharpening by morphological filtering. *Pattern Recognition*, 33:997–1012, 2000.

[205] B. Schölkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT Press, Cambridge, Massachusetts, London, England, 2002.

[206] B. Sendov and V. Popov. On some properties of the hausdorff metric. *Mathematica*, 7(1):145–154, 1965.

[207] J. Serra. *Image Analysis and Mathematical Morphology, Volume II: Theoretical Advances*, chapter Mathematical Morphology for Boolean Lattices, pages 37–58. Academic Press, London, 1988.

[208] J. Serra. Connectivity on complete lattices. *Journal of Mathematical Imaging and Vision*, 9(3):231–251, 1998.

[209] J. Serra. Set connections and discrete filters. *Lecture Notes in Computer Science: Discrete Geometry and Computer Imagery*, 1568:191–206, 1999.

[210] J. Serra. Connections for sets and functions. *Fundamenta Informaticae*, 41:147–186, 2000.

[211] J. Serra. Connection, image segmentation and filtering. In *Proceedings of the XI International Computing Conference*, Mexico, November 2002.

[212] J. Serra. Viscous lattices. *Journal of Mathematical Imaging and Vision*, 22:269–282, 2005.

[213] J. Serra. A lattice approach to image sementation. *Journal of Mathematical Imaging and Vision*, 24:83–130, 2006.

[214] J. Serra and G. Matheron. *Image Analysis and Mathematical Morphology, Volume II: Theoretical Advances*, chapter Strong Filters and Connectivity, pages 141–157. Academic Press, London, 1988.

[215] J. Serra and P. Salembier. Connected operators and pyramids. In *Proceedings of SPIE: Nonlinear Algebra and Morphological Image Analysis*, volume 2030, pages 65–76, 1993.

[216] J. Shah. Segmentation by nonlinear diffusion. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 202–207, 1991.

[217] P. Soille. Constrained connectivity for hierarchical image partitioning and simplification. *IEEE TRansactions on Pattern Analysis and Machine Intelligence*, 30(7):1132–1145, 2008.

[218] X. Song and G. Fan. A study of supervised, semi-supervised and unsupervised multiscale bayesian image segmentation. In *The 2002 45th Midwest Symposium on Circuits and Systems*, volume 2, 2002.

[219] J. Sporring. The entropy of scale-space. In *Proceedings of the 13th International Conference on Pattern Recognition*, volume 1, pages 900–904, Vienna, Austria, 25-29 August 1996.

[220] J. L. Stansfield. Conclusions from the commodity expert project. Technical Report A.I. memo 671, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA, 1980.

[221] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, Massachusetts, 1986.

[222] E. Tadmor, S. Nezzar, and L. Vese. A multiscale image representation using hierarchical (bv, l^2) decompositions. *Multiscale Modeling and Simulation*, 2(4):554–579, 2004.

[223] E. Tadmor, S. Nezzar, and L. Vese. Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. *Communications in Mathematical Sciences*, 6(2):281–307, 2008.

[224] S. Tanimoto and T. Pavlidis. A hierarchical structure for image processing. *Computer Vision, Graphics, and Image Processing*, 4:104–119, 1975.

[225] A. Toet. Hierarchical image fusion. *Machine Vision and Applications*, 3:1–11, 1990.

[226] L. Uhr. Layered 'recognition cone' networks the preprocess, classify, and describe. *IEEE Transactions on Computers*, 21(7):758–768, 1972.

[227] C. Urdiales, A. Bandera, and F. Sandoval. Non-parametric planar shape representation based on adpative curvature functions. *Pattern Recognition*, 35:43–53, 2002.

[228] C. Vachier. Morphological scale-space analysis and feature extraction. In *Proceedings of the 2001 International Conference on Image Processing*, volume 3, pages 676–679, Thessaloniki, Greece, 7-10 October 2001.

[229] C. Valens. A really friendly guide to wavelets. Technical report, 1999.

[230] R. van den Boomgaard and A. Smeulders. The morphological structure of images: The differential equations of morphological scale-space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1101–1113, 1994.

[231] S. J. van der Walt. *Super-Resolution Imaging*. Phd thesis, Stellenbosch University, December 2010.

[232] P. van Staden and M. Loots. Method of L-moment estimation for the generalized lambda distribution. In *Proceedings of the Third Annual ASEARC Conference*, Newcastle, Australia, 7-8 December 2009.

[233] P. Velleman. Robust nonlinear data smoothers: Definitions and recommendations. *Proceedings of the Natural Academy of Sciences*, 74(2):434–436, 1977.

[234] P. F. Velleman. Definition and comparison of robust nonlinear data smoothing algorithms. *Journal of the American Statistical Association*, 75(371):609–615, 1980.

[235] Z. Wang, A. Bovik, H. Sheik, and E. Somincelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[236] M. Watanabe, D. Williams, and Y. Tomokiyo. Comparison of detection limits for elemental mapping by EF-TEM and STEMXEDS. *Microscopy and Microanalysis*, 8:1588–1589, 2002.

[237] G. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, London, 2nd edition, 1944.

[238] J. Weickert, S. Ishikawa, and A. Imiya. On the history of gaussian scale-space axiomatics. *Computational Imaging and Vision*, 8:45–49, 1994.

[239] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in japan. *Journal of Mathematical Imaging and Vision*, 10:237–252, 1999.

[240] E. Weisstein. Convolution. From MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/Convolution.html.

[241] R. Whitaker and S. Pizer. A multi-scale approach to nonuniform diffusion. *Computer Vision, Graphics, and Image Processing*, 57(1), 1993.

[242] D. Widder. *The Heat Equation*. Acdemic Press, New York, 1975.

[243] M. Wild. Idempotent and co-idempotent stack filters and min-max operators. *Theoretical Computer Science*, 299:603–631, 2003.

[244] M. H. F. Wilkinson and G. K. Ouzounis. *Advances in Imaging and Electron Physics*, volume 161, chapter 5: Advances in Connectivity and Connected Attribute Filters, pages 211–275. 2010.

[245] A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'84*, pages 150–153, March 1984.

[246] A. P. Witkin. Scale-space filtering. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence IJCAI '83*, pages 1019–1022, Karlsruhe, 8-12 August 1983.

[247] A. Wong and A. Mishra. Generalized probabilistic scale space for image restoration. *IEEE Transactions on Image Processing*, 19(10):2774–2780, 2010.

[248] K. Wu, E. Otoo, and A. Shoshani. Optimizing connected component labeling algorithms. http://repositories.cdlib.org/lbnl/LBNL-56864, 2005.

[249] X. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[250] A. Yuille and T. Poggio. Scaling theorems for zero-crossings. A.i. memo 722, Massachusetts Institute of Technology Artificial Intelligence Laboratory, June 1983.

[251] C. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20(1):68–86, 1971.

[252] S. Zeki. *A Vision of the Brain*. Blackwell Scientific Publications, 1993.

[253] Y. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[254] L. Zusne. *Contemporary Theory of Visual Form Perception: III. Global Properties*, chapter Chapter 4, pages 108–174. Academic Press, 1970.