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*

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

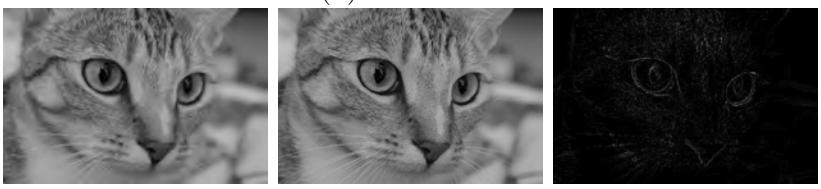
where $F_\rho^{\oplus}$ is a grey-scale dilation of f and $F_\rho^{\ominus}$ a grey-scale erosion of f by a scalable structuring function g^ρ (element c^ρ) where ρ is the size of the structuring function (element).

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

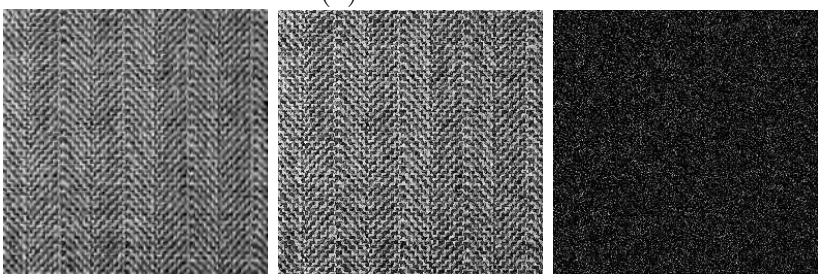
(3) 256×256

(4) 225×300

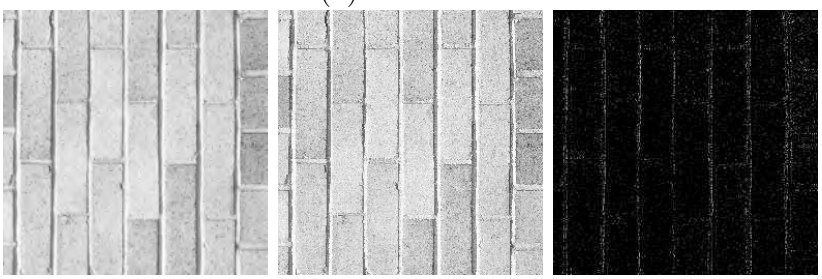
Figure 5.1: *First column: original images, middle column: sharpened images, last column: difference between original and sharpened images*

(5) 261 × 453

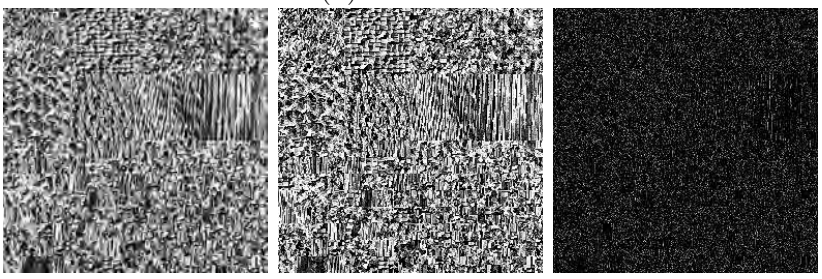
(6) 512 × 512



(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

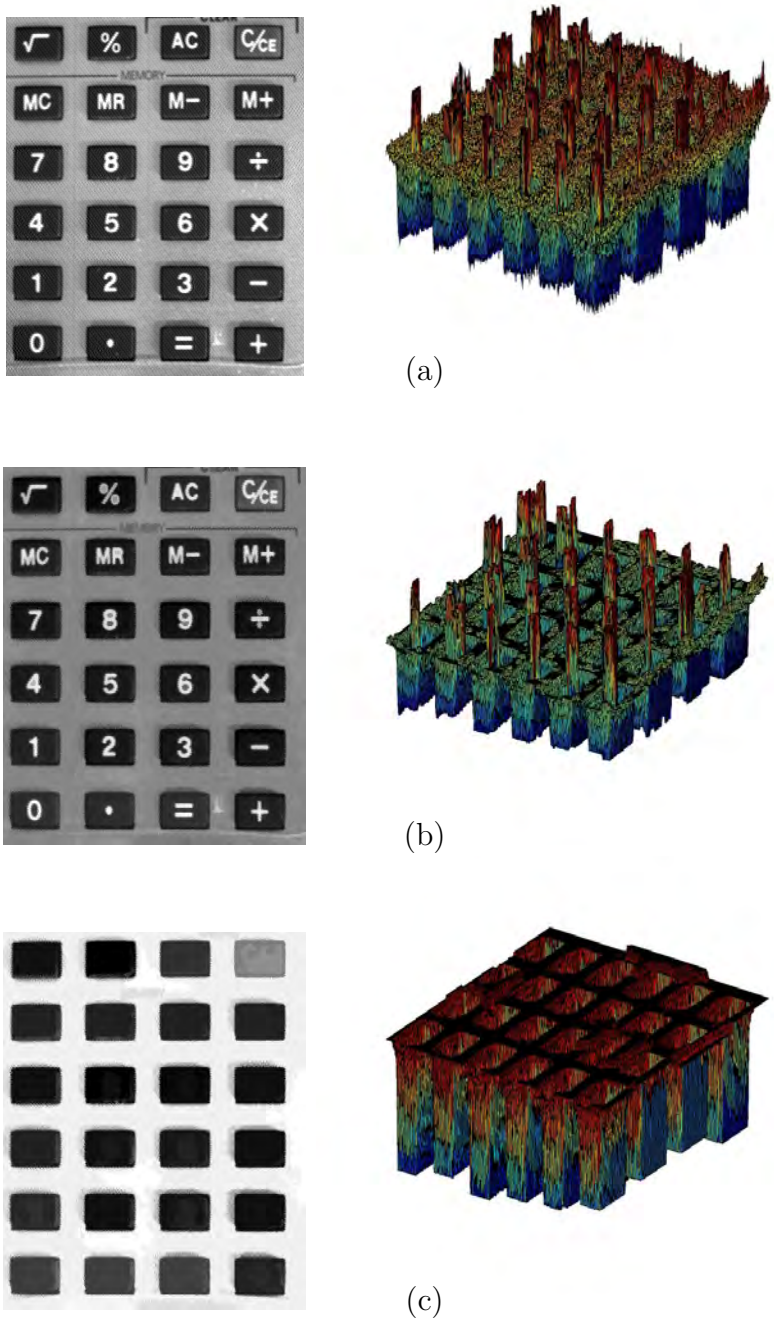


Figure 5.3: *An illustrative example: (a) input; (b) noise removed; (c) features of interest (keys) extracted*

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

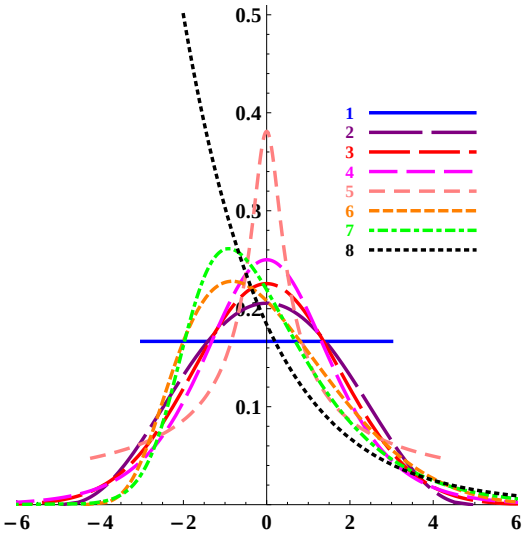


Figure 5.4: *Probability Density Functions of the Noise Distributions in Table 5.5*

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

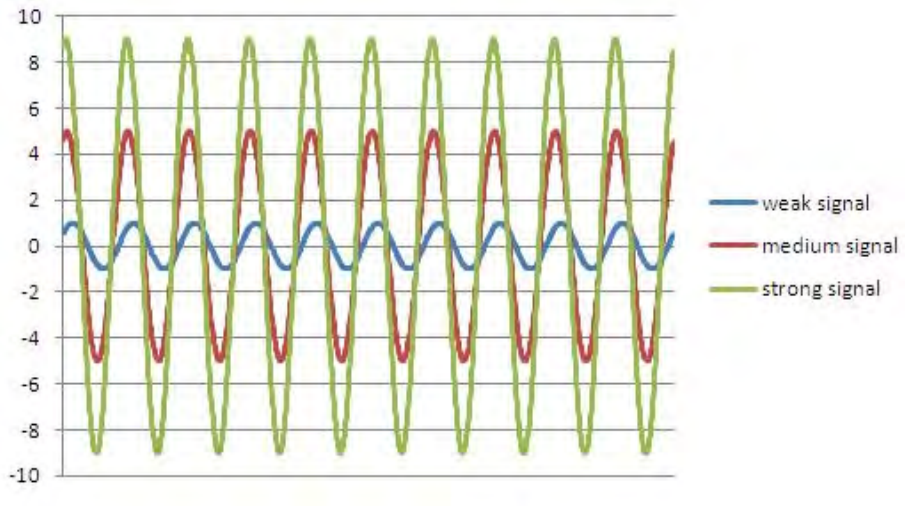


Figure 5.5: *Original Signals with SNR 1 (weak), 5 (medium) and 9 (strong)*

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.

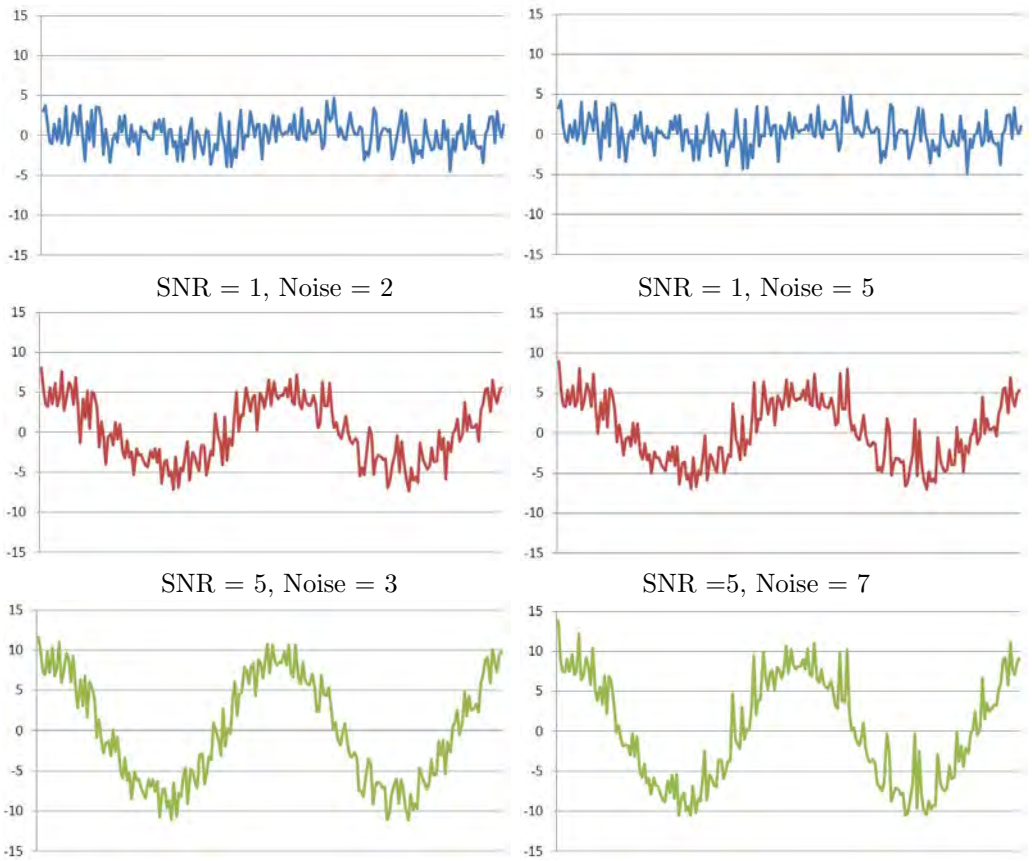
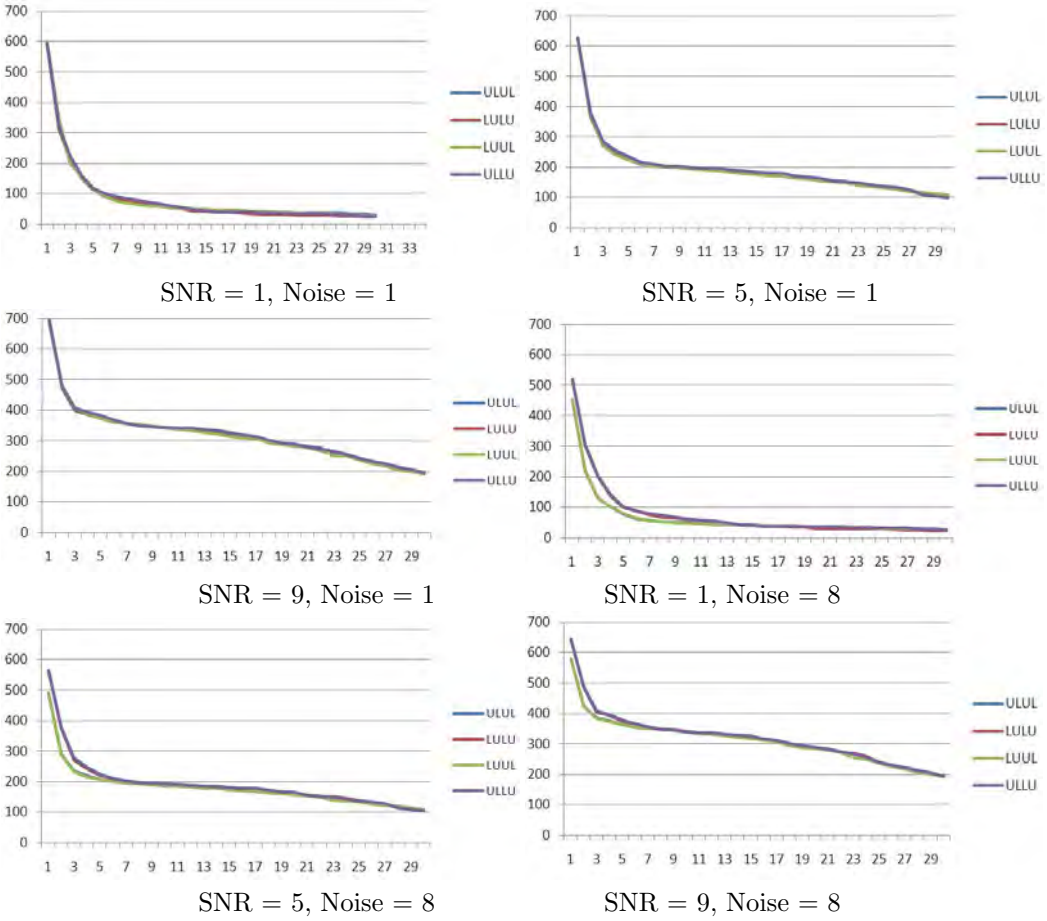


Figure 5.6: *A Sample of the 24 Different Contaminated Signals*



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.

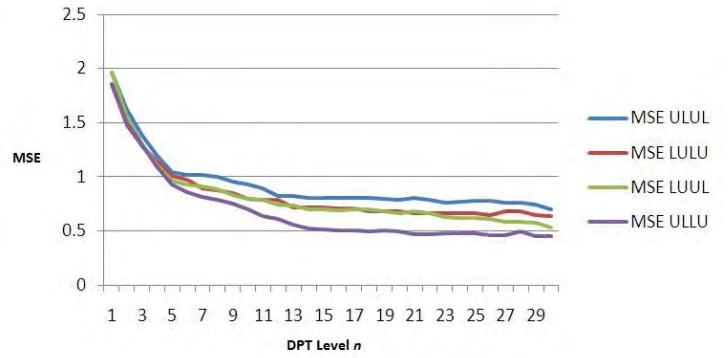
To investigate whether the smoothed signal obtained when the total variation stabilizes does in fact resemble the original uncontaminated signal, the MSE measure was used to calculate the differences between the smoothed signal through the DPT levels and the original signal. The MSE is calculated as

$$MSE(x) = \frac{\sum_{i=1}^{1000}(x_i - \bar{x})^2}{1000}.$$

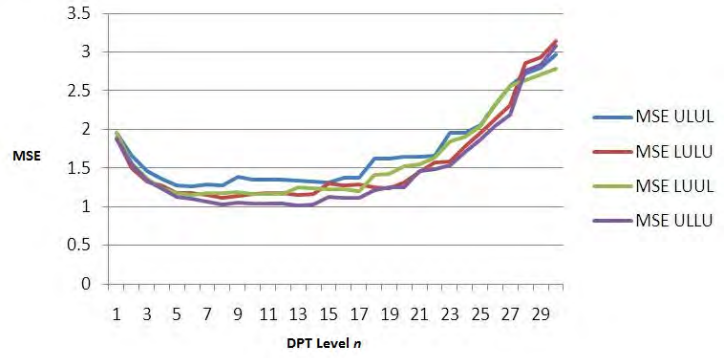
For SNR = 1 the combination ULLU provides the lowest MSE from the beginning of the smoothing process. The combination ULUL gives the highest MSE although the differences between the combinations are not drastic. For SNR = 5 and 9 the same is seen. See Figure 5.8 for the MSE for noise type 1. The medium 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.

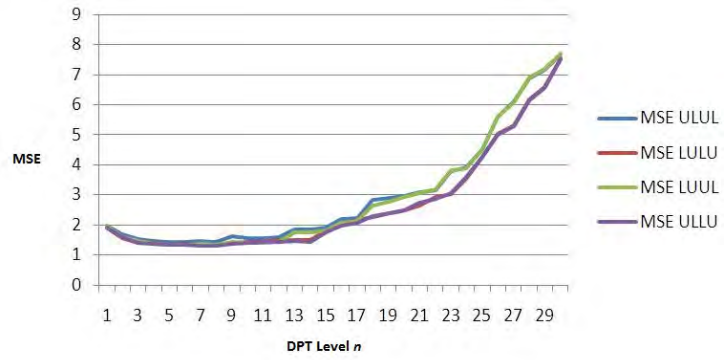
As discussed above, the cumulative noise removed when (a) half of the added total variation has been removed, and (b) when all the added total variation has been removed, is investigated. The original noise distributions were fitted to this cumulative removed noise. The L-moments and four parameters 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)*

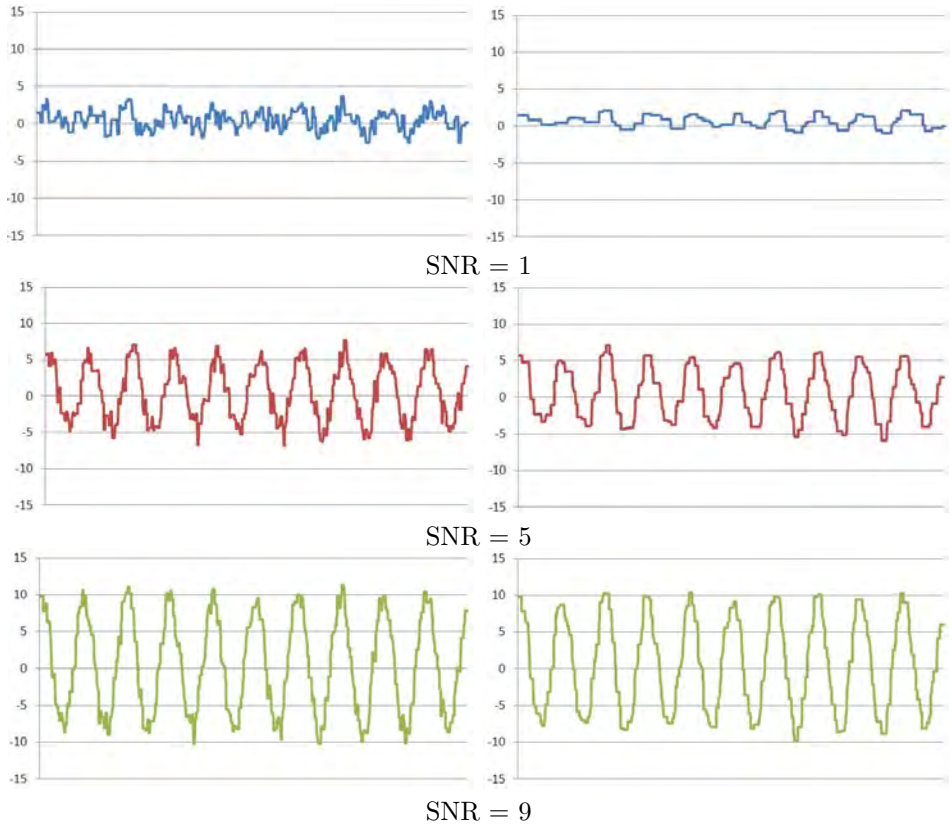


Figure 5.9: *The Smoothed Signals for Noise Type 1 and using LULU (all noise types and L_n, U_n combinations are visually similar): Left column indicates the smoothed signal when half the added TV has been removed, Right column indicates the smoothed signal when all the added TV has been removed*

- 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.

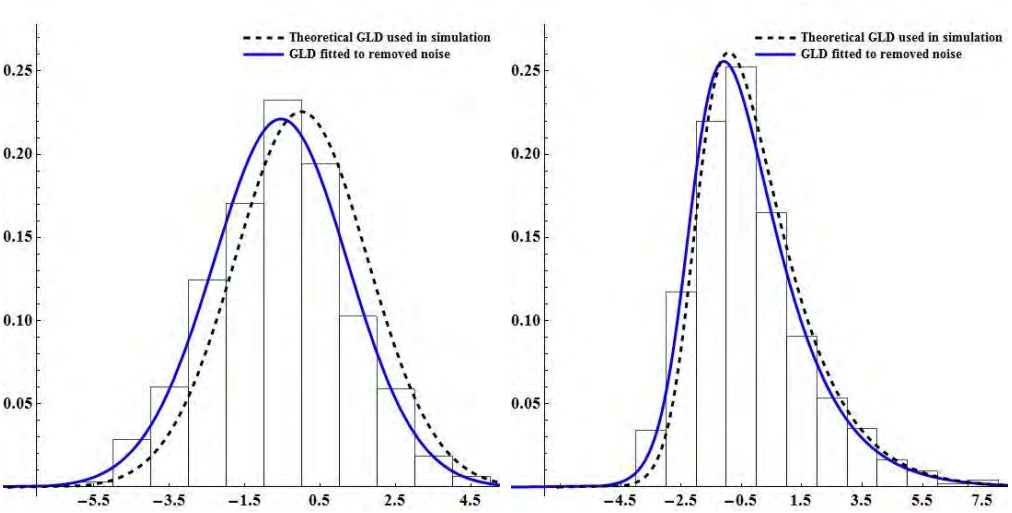
- The shift in L-location must be considered simultaneously with the change in the L-skewness ratio, as seen in Table 5.8, since a change in the level of skewness of a distribution will result in a shift in the location of that distribution. There we see a decrease in the L-skewness ratio for LULU and 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

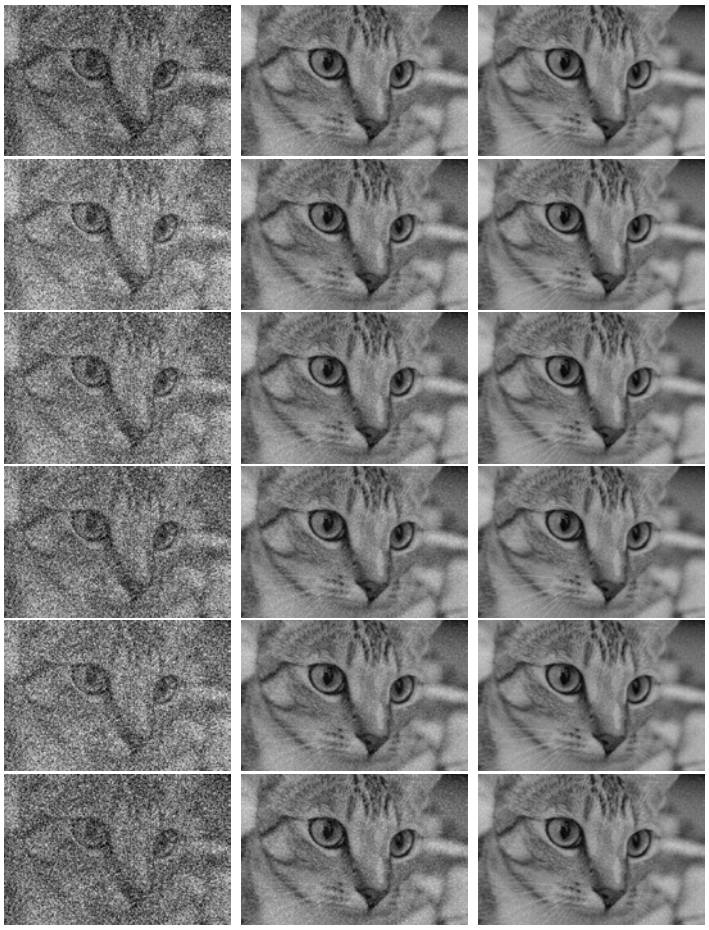


Figure 5.11: *Contaminated Chelsea images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

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

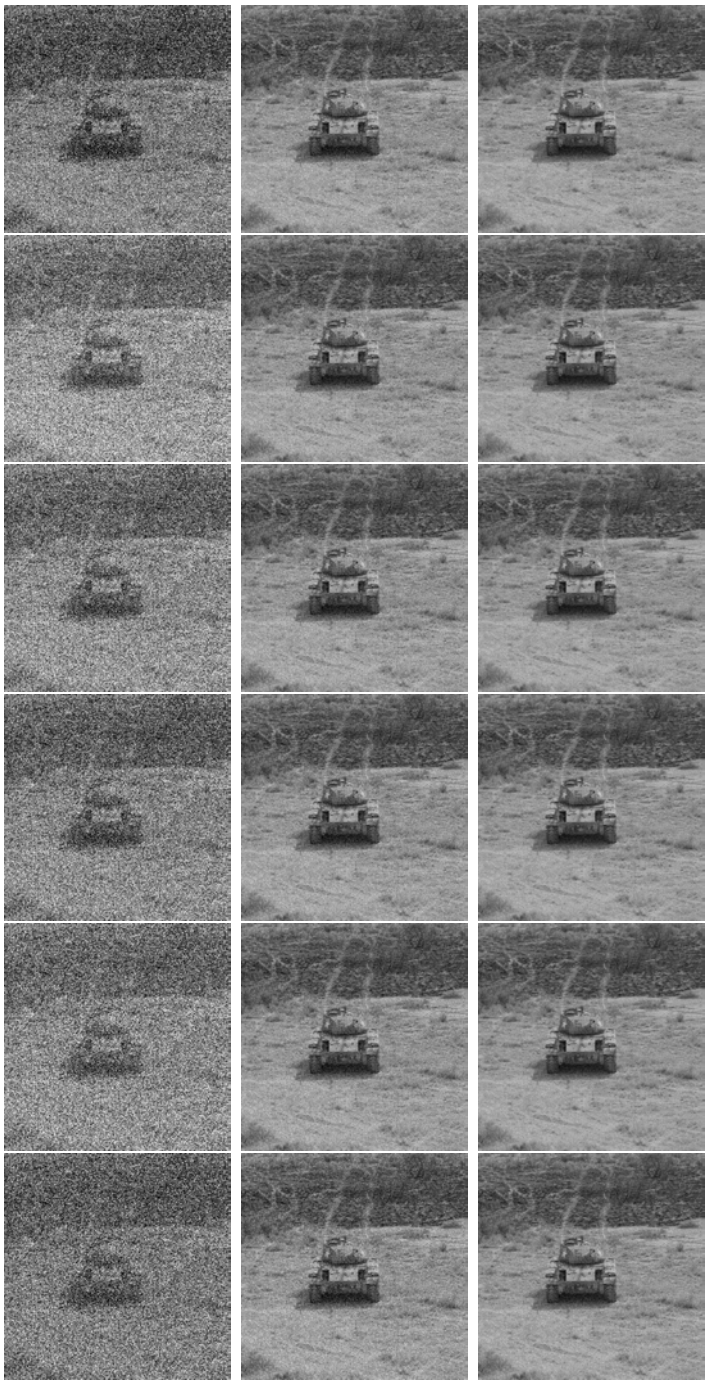


Figure 5.12: *Contaminated Tank images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

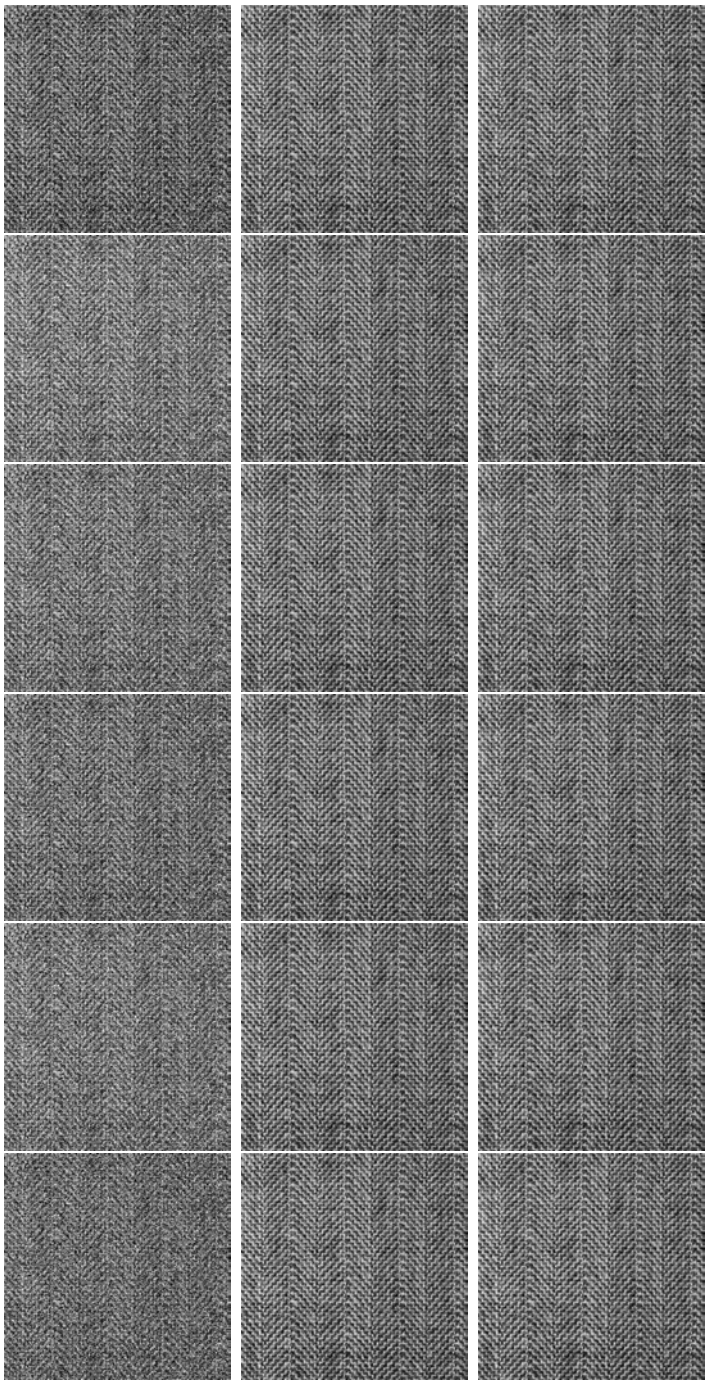


Figure 5.13: *Contaminated Fabric images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

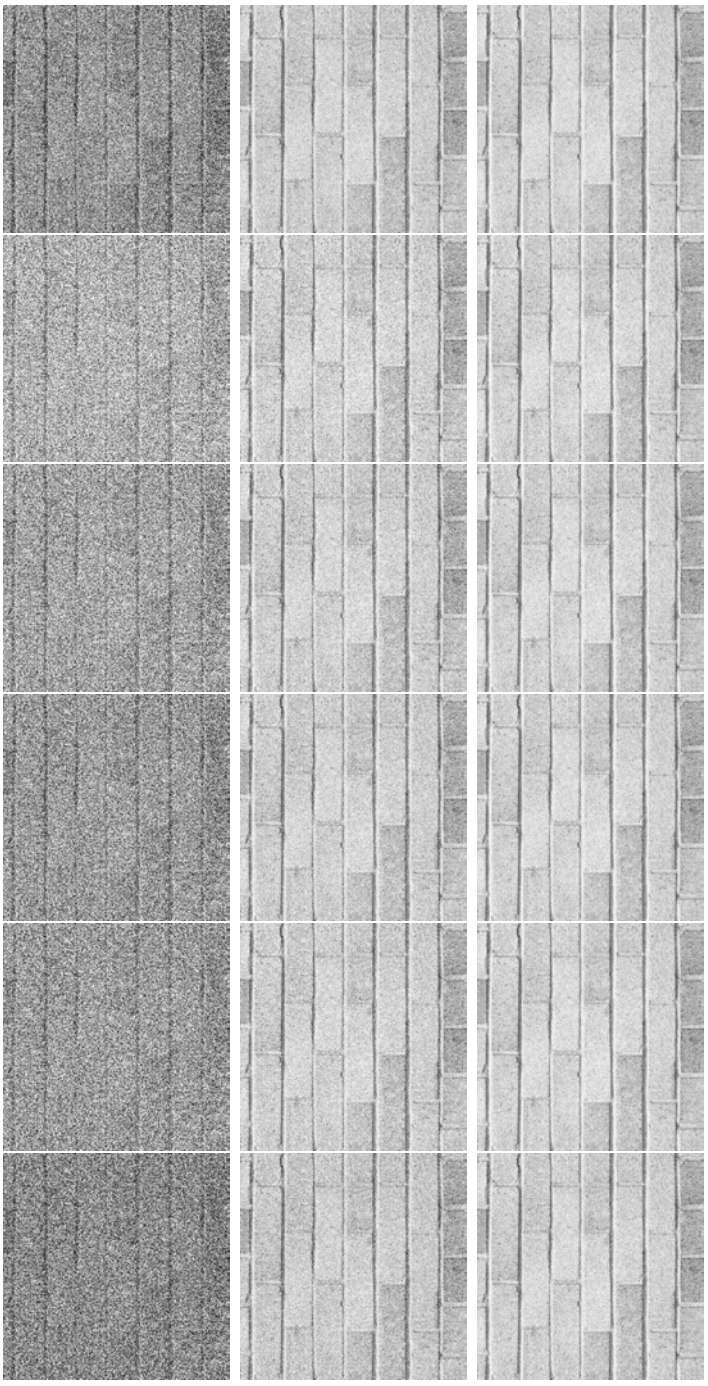


Figure 5.14: *Contaminated Brick images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

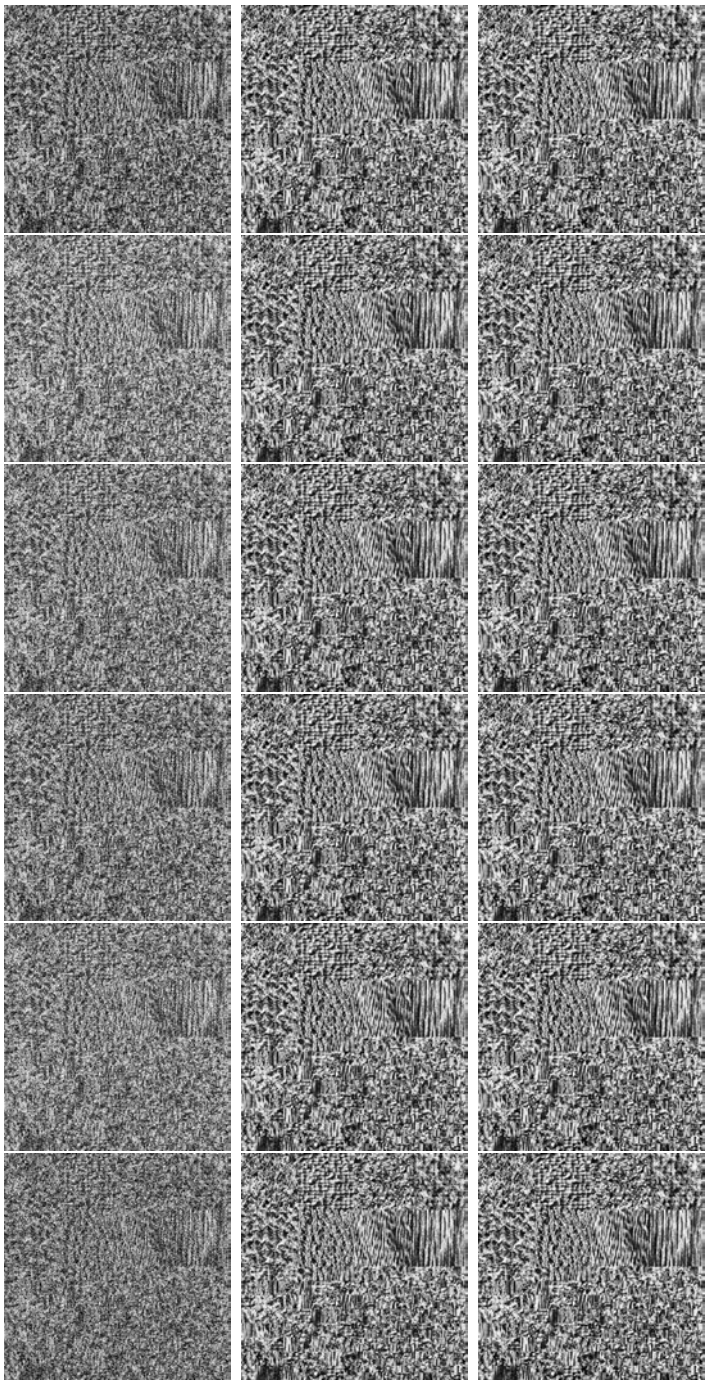


Figure 5.15: *Contaminated Texture images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

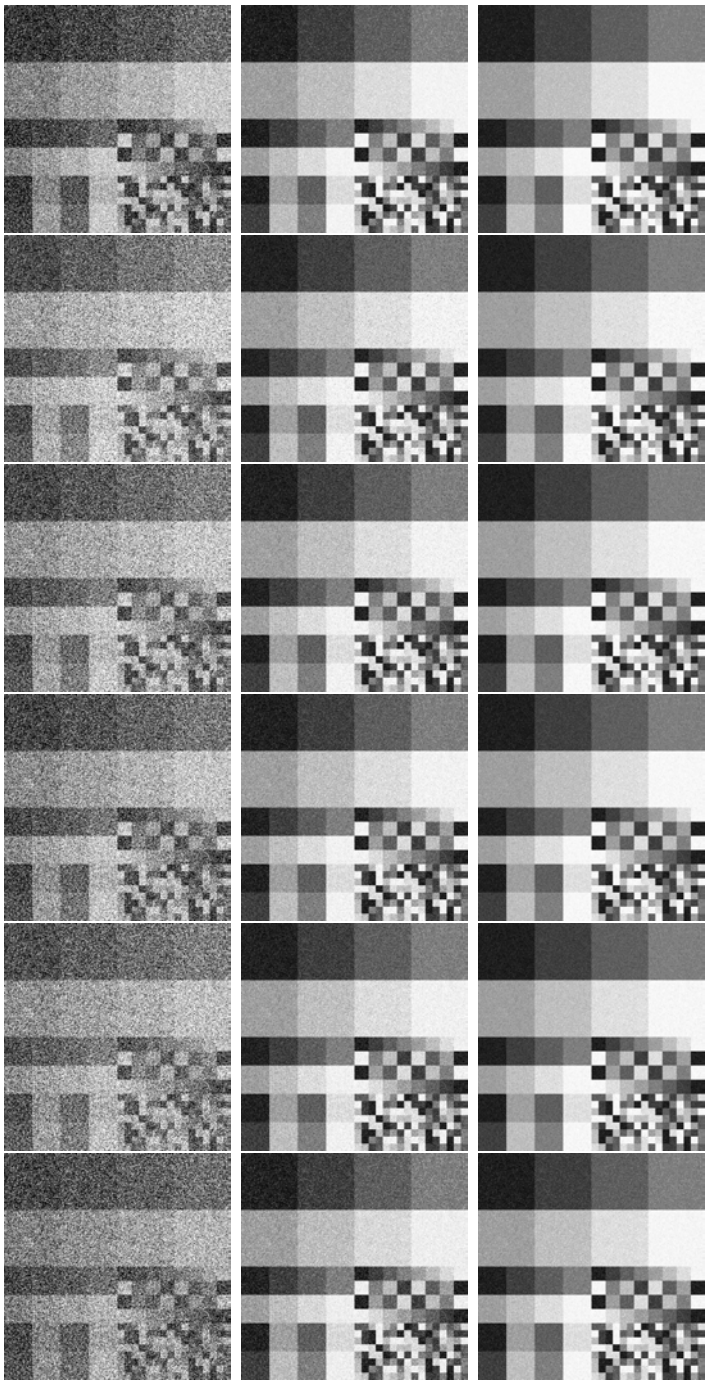


Figure 5.16: *Contaminated Blocks images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

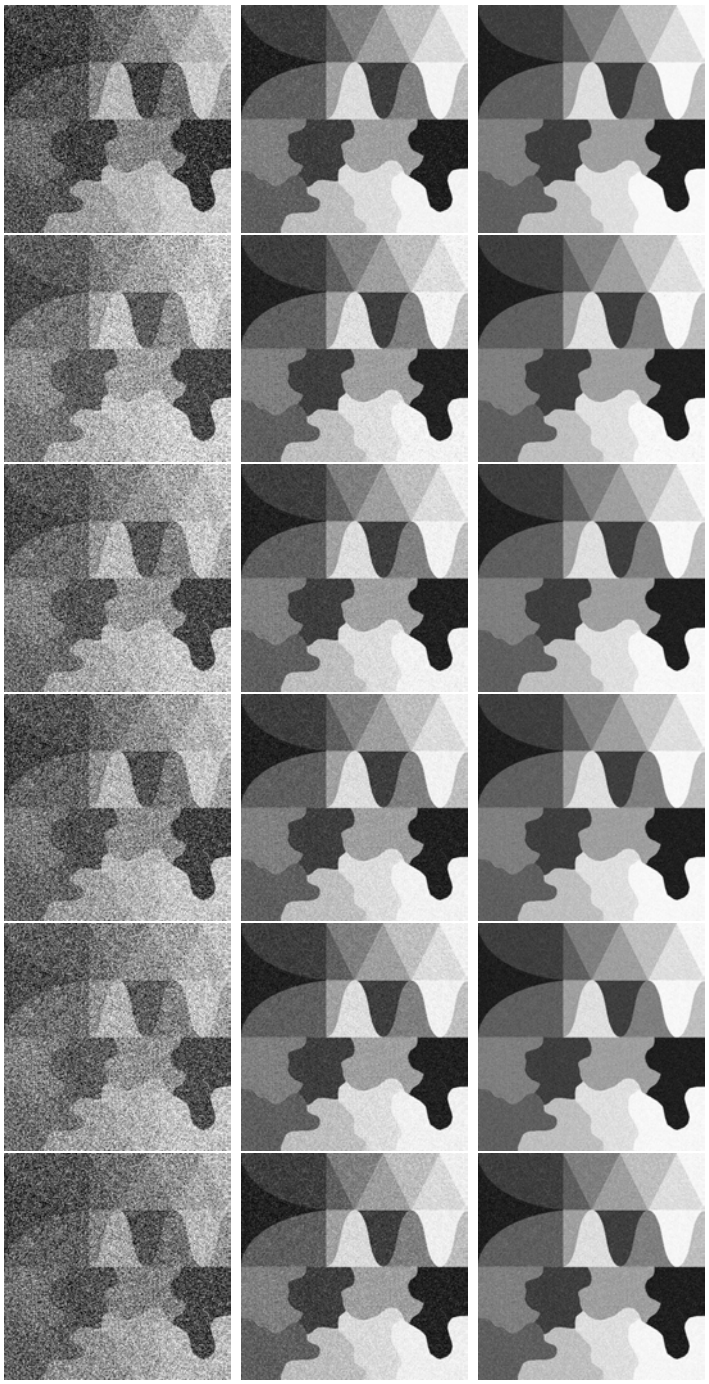


Figure 5.17: *Contaminated Regions images, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

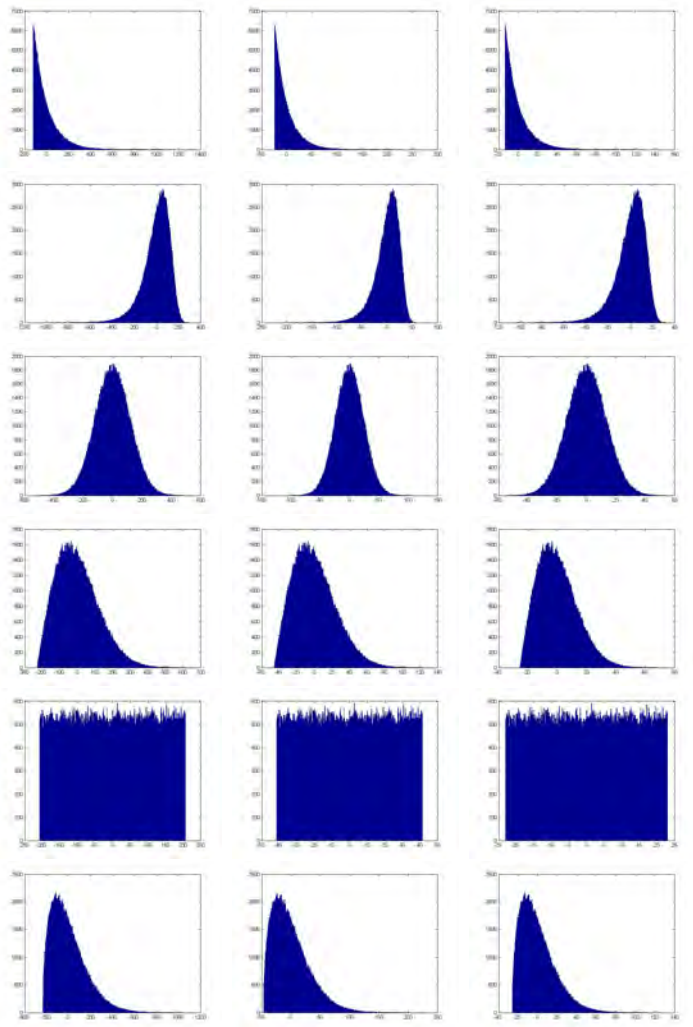


Figure 5.18: *Added noise, rows (top to bottom): Exponential, Gumbel, Normal, Rayleigh, Uniform and Weibull noise, columns (left to right): SNR = 1, 5 and 9*

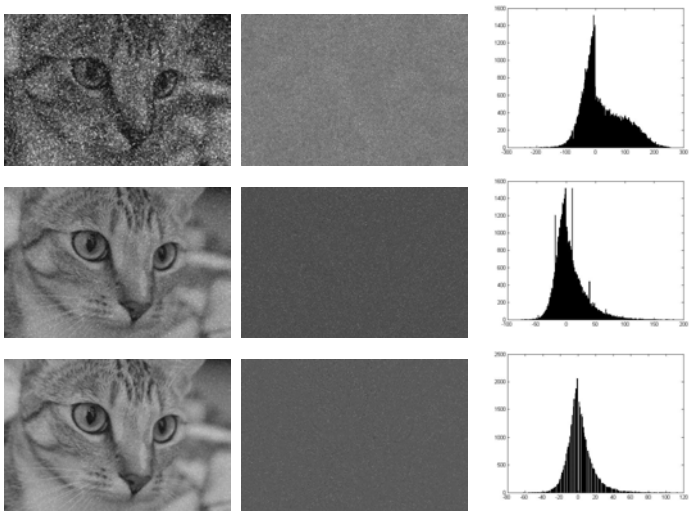


Figure 5.19: *Exponential contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

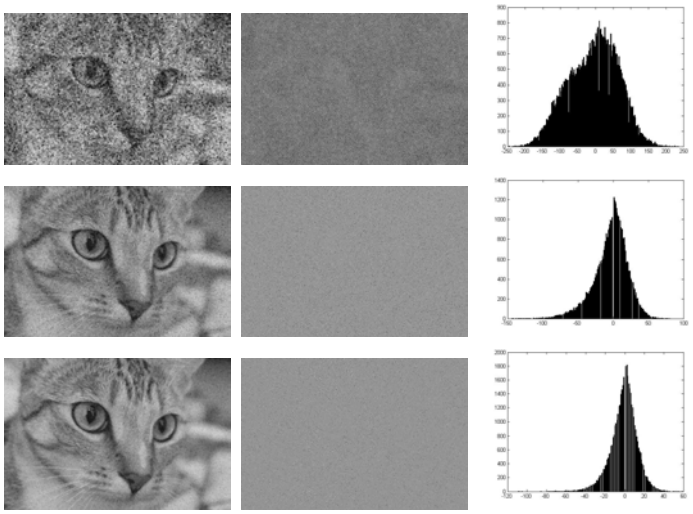


Figure 5.20: *Gumbel contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

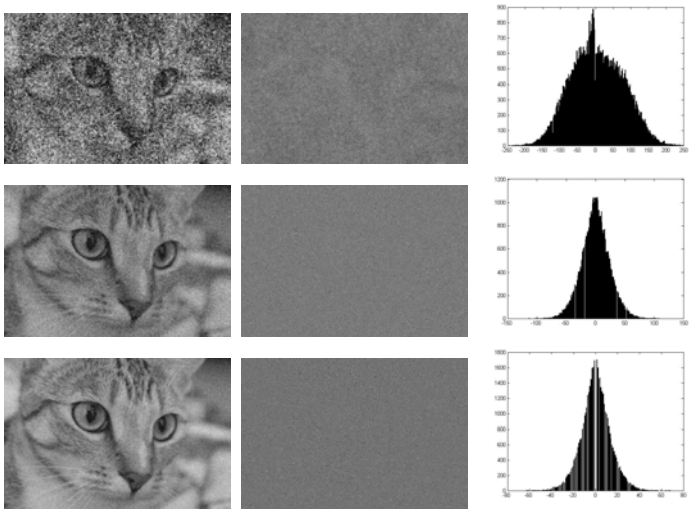


Figure 5.21: *Normal contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

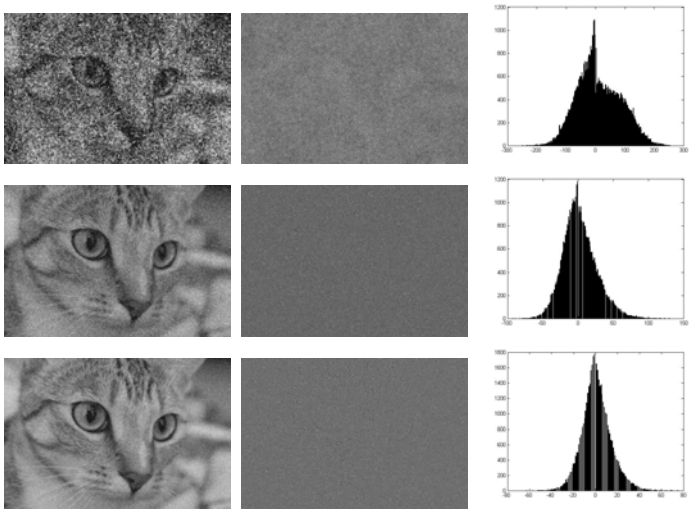


Figure 5.22: *Rayleigh contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

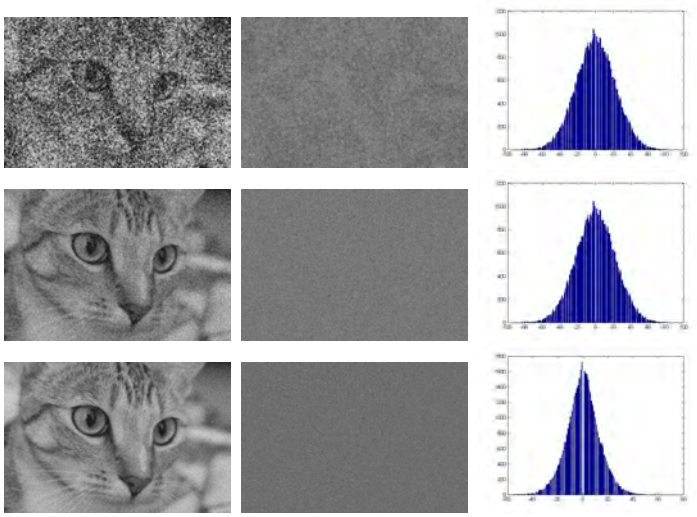


Figure 5.23: *Uniform contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

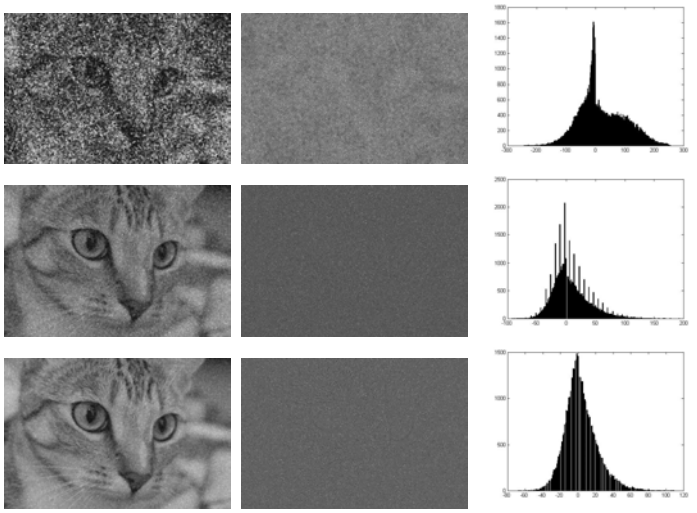


Figure 5.24: *Weibull contamination, columns (left to right): smoothed Chelsea image, noise removed, histogram of removed noise, rows: SNR = 1, 5 and 9*

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.