An overview of LULU operators and discrete pulse transform for image analysis

R Rahmat[a], **A S Malik***[a], **I Faye**[b], **N Kamel**[a] and **H Nisar**[c]
[a]Center for Intelligent Signal and Imaging Research, Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Perak, Malaysia

[b]Department of Fundamental and Applied Science, Universiti Teknologi PETRONAS, Perak, Malaysia

[c]Department of Electronic Engineering, Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, Kampar, Malaysia

Abstract: This paper presents an overview of LULU operators and discrete pulse transform (DPT). Data extraction from signals and images is a popular area of research. Different methods are being used for data extraction such as different types of linear and nonlinear operators. LULU operator is one of the most important rank selector nonlinear filters used for data analysis which is widely being used for signal analysis, especially in impulse noise filtering. It is computationally more efficient and the behaviour of the operator is simple to describe. Based on the composition of different orders of LULU operators, DPT on multi-resolution is defined, which describes the sequences into pulses with different magnitudes. DPT allows a multi-resolution measure of roughness of images and sequences. It is a powerful technique for image analysis and can also be used for the estimation of standard deviation of a random distribution.

Keywords: discrete pulse transform, impulse noise, object extraction

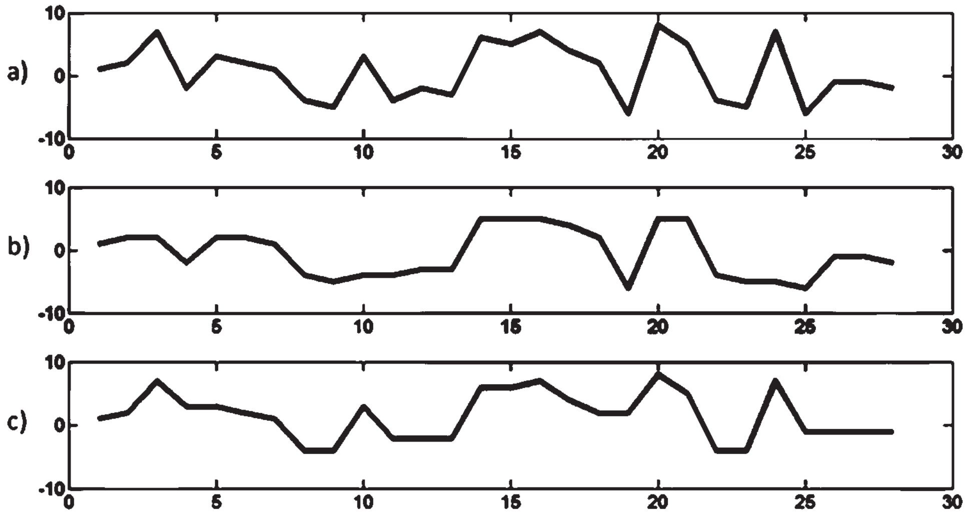
1 INTRODUCTION

In signal and image processing techniques, we come across many challenges analysing the data and recovering the original data from the noise corrupted ones. Sometimes, we look for a specific property or part of data within an image or any signal sequence. This analysis is performed using many different methods, including both, linear and nonlinear methods. Linear filters are good smoothers that respond well to data with Gaussian noise. However, their performance degrades for data with impulse noise. Meanwhile, nonlinear methods deal with discontinuities or large impulses, which relatively provide better results. In this paper, we focus on LULU which is one of the nonlinear methods.

Rohwer and Toerien in the late 1980s introduced a novel, innovative nonlinear smoother, named LULU smoothers, based on extreme order statistics.[1] LULU operators remove impulse noise before a signal is extracted from a sequence. They are computationally convenient and conceptually simpler compared to the median smoothers which are usually considered to be basic smoothers. LULU operators have particular properties, e.g. they are fully trend preserving, preserve the total variation, etc., which make them an essential tool for multi-resolution analysis of sequences. Furthermore, it was demonstrated during the last decade or so that these operators were specific cases of morphological filter. They have a critical role in the analysis and comparison of nonlinear smoothers

The MS was accepted for publication on 9 June 2011.
* *Corresponding author: Aamir Saeed Malik, Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Perak 31750, Malaysia; email: aamir_saeed@petronas.com.my*

 IMAG 113 © RPS 2013 DOI: 10.1179/1743131X11Y.0000000029



1 (a) Original signal; (b) result of L smoother on the signal; (c) result of U smoother on the signal

[an operator A is a smoother if $AE=EA$ where $Ex_i=x_{i+1}$; $A(x+b)=Ax+b$ for all constant sequences b; and $A(cx)=c(Ax)$ for all scalars for $c\geqslant 0$].[2]

The other application of LULU smoothers is in applying discrete pulse transform (DPT) to images. DPT is a new and strong method for the analysis of signals and can be extended to images by using LULU operators. While DPT decomposes the image into different pulses, it can be used to extract the specific objects in the image by selecting the appropriate pulses. Furthermore, DPT is being used in the estimation of standard deviation of a random distribution.[3]

A multi-resolution analysis of a space consists of a sequence of nested subspaces that satisfies certain self-similarity relations in time/space and scale/frequency, as well as completeness and regularity relations. DPT and wavelet are two of the most important multi-resolution analysis methods. The properties of multi-resolution analysis are described in more details in Refs. 4 and 5.

In this paper, first, we explain LULU operators and then discuss their properties. Next, we will discuss the main concept of DPT and then, different applications of signals and images will be discussed which have the potential to be addressed by LULU and DPT.

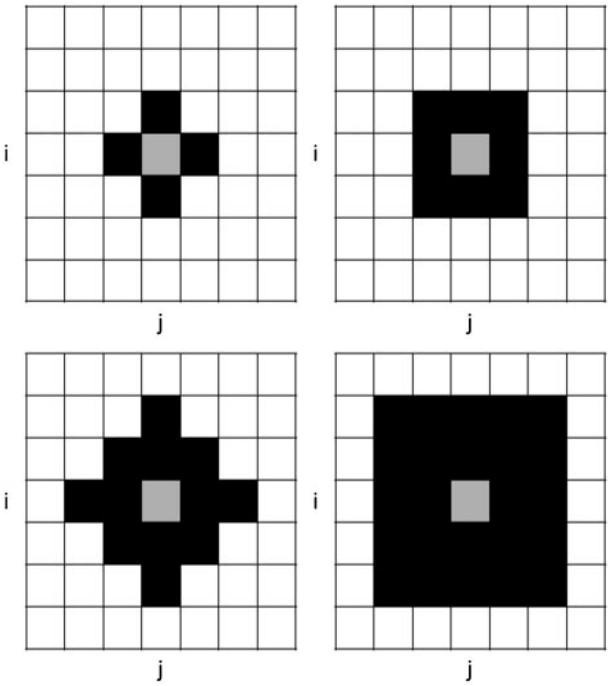
2 LULU OPERATORS

LULU operators are called MaxMin and MinMax filters due to their characters. They are local and nonlinear operators used for the removal of impulse noise. LULU operators consist of the sub-operators L (low) and U (upper) with different orders for different filters.

For one-dimensional (1D) analysis of the sequences of signal, noise removal can be done via LU or UL operators. These operators remove the positive and negative peaks which have small width similar to impulse noise. The result of LU and UL operators is a local and monotone (the sequence ξ is n-monotone if either $\xi \leqslant \xi_{i+1} \leqslant \ldots \leqslant \xi_{i+n} \leqslant \xi_{i+n+1}$ or $\xi_i \geqslant \xi_{i+1} \geqslant \ldots \geqslant \xi_{i+n} \geqslant \xi_{i+n+1}$, for all values of i, such ξ_i and ξ_{i+n+1} are both members of the sequence[2]) sequence without any detectable noise, and the 1D LULU operators fulfil the idempotent condition (A is idempotent if $A^2=A$ and co-idempotent if $I-A$ is idempotent[2]). LULU operators are also used in two-dimensional (2D) analysis, i.e. image analysis, for smoothing or filtering the image and also for object detection and extraction (by using DPT) with composition of different L and U operators.

2.1 1D LULU

When LULU is being used for signals, it is similar to a simple comparison method to remove the peaks. Figure 1 illustrates the power of L and U operators in filtering/smoothing the signal. In this figure, the top one is the original signal, while the middle and the bottom ones show the smoothed signals after applying L and U operators, respectively. For a given bi-infinite sequence, $\xi=(\xi_i)$, $i\in Z$, the 1D LULU operators are defined by equations (1) and (2), as follows[6]



2 Four different neighbouring regions of pixel (i,j)

$$(L_n\xi)_i =$$

$$\max\{\min\{\xi_{i-n}, \ldots, \xi_i\}, \min\{\xi_i, \ldots, \xi_{i+n}\}\}, i \in Z \quad (1)$$

$$(U_n\xi)_i =$$

$$\min\{\max\{\xi_{i-n}, \ldots, \xi_i\}, \max\{\xi_i, \ldots, \xi_{i+n}\}\}, i \in Z \quad (2)$$

2.2 2D LULU

When LULU smoothers are applied on a 2D array, it shall compare any single element with all the neighbours around it. The neighbourhood of a pixel is definable in different ways such as the ones shown in the Fig. 2.

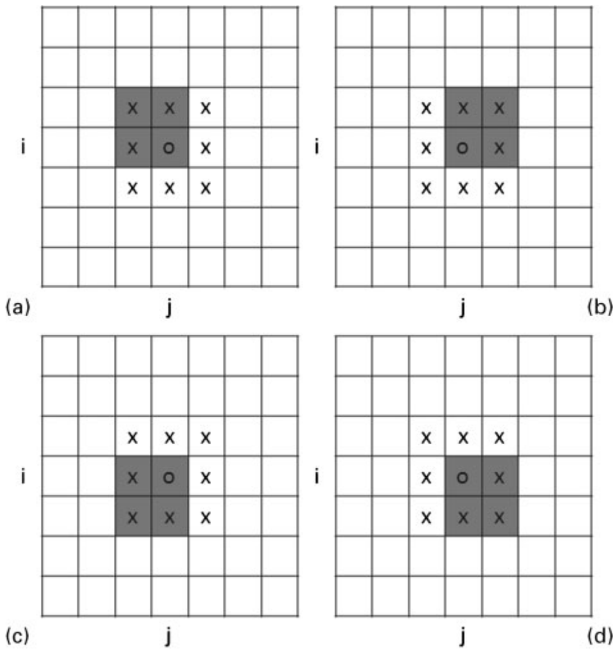
To further elaborate the concept of 2D processing using LULU operators, an example is provided. This example illustrates one of the many different possible neighbourhood and sub-neighbourhoods for a pixel. LULU in 2D, similar to 1D LULU, can be extended to neighbourhoods by considering more pixels surrounding each pixel.

In this example, the neighbours of the pixel $I(i,j)$ is divided to four different regions as shown in equations (3)–(6). Others possible neighbours are not considered here

$$I_1 = [I(i,j-1), I(i,j), I(i+1,j-1), I(i+1,j)] \quad (3)$$

$$I_2 = [I(i-1,j-1), I(i-1,j), I(i,j), I(i,j-1)] \quad (4)$$

$$I_3 = [I(i,j+1), I(i,j), I(i+1,\mathrm{j}), I(i+1,j+1)] \quad (5)$$



3 Illustration of neighbours for equations (3)–(6): (a) I_1; (b) I_2; (c) I_3; (d) I_4

$$I_4 = [I(i-1,j+1), I(i-1,j), I(i,j), I(i,j+1)] \quad (6)$$

Figure 3 illustrates equations (3)–(6).

Then the L and U operators were applied as follows

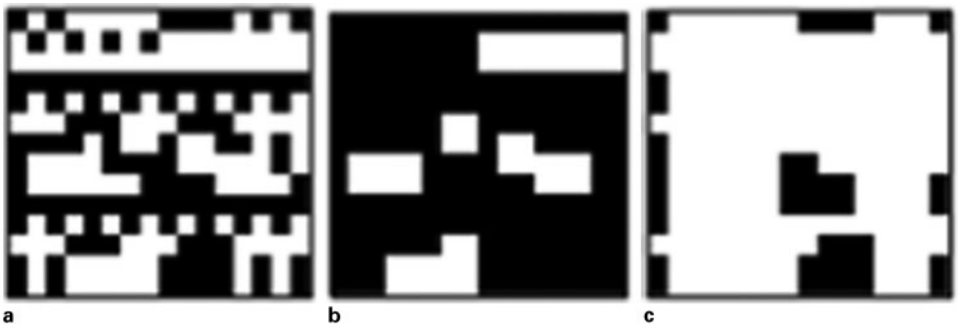
$$L(i,j) = \max(\min(I_1), \min(I_2), \min(I_3), \min(I_4)) \quad (7)$$

$$U(i,j) = \min(\max(I_1), \max(I_2), \max(I_3), \max(I_4)) \quad (8)$$

Figure 4a shows a randomly generated binary image and Fig. 4b and c shows the smoothed images after applying L and U smoothers. The L and U in equations (7) and (8) are actually L_3 and U_3 because of considering a neighbourhood of 4 pixels in each region. In this example, the binary image has balanced number of black and white parts. After applying L smoother on the image, the black parts increased. That can be described according to equation (7). L operator maximises the minima of the neighbourhood which actually removes the lower peaks (this is the reason that Fig. 4b has more black spots than the original image). U smoother is opposite to L smoothers. Therefore, Fig. 4c is whiter compared to the original image.

3 PROPERTIES OF LULU OPERATORS

Some of the properties of LULU operators, as introduced by Rohwer and Laurie, are listed in Table 1. For detailed discussion of properties as well



4 (a) Original binary image; (b) result of L smoother on the image; (c) result of U smoother on the image

as their proofs, check Refs. 2 and 7. In Table 1, M denotes the median, I denotes the identity operator, and C and F are LULU operators which are called ceiling (biased towards lower limits) and floor (biased towards upper limits) operators, respectively.

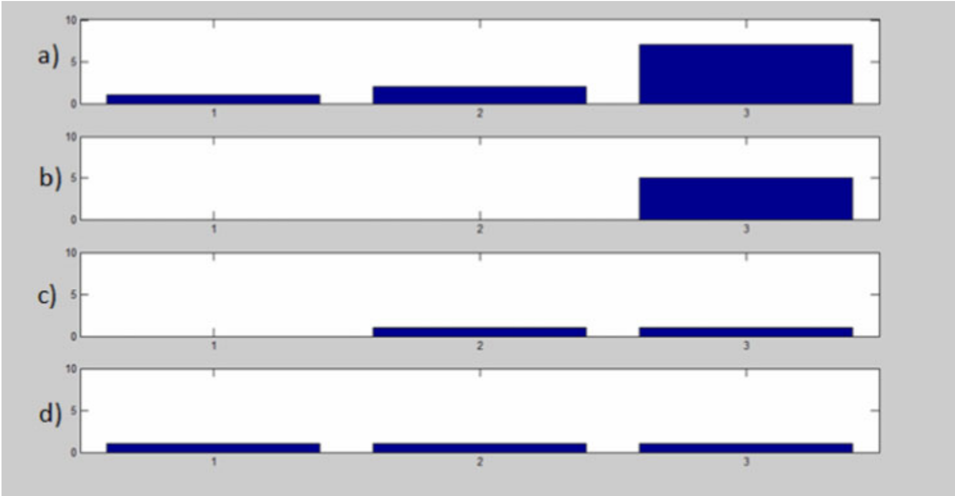
4 THE DPT

DPT is a composition of different pulses, where a pulse is a string of zero values which is non-zero for few consecutive elements. DPT is naturally discrete unlike discrete Fourier and wavelet transforms and its results are better than the median transform especially in terms of computational complexity. DPT is very similar to DFT, except that DPT separates the signal to positive and negative parts (pulses) but DFT divides the signal to even and odd parts.

In image processing, DPT is being used to separate the objects in the image by identifying the pulses corresponding to different objects in the image. For

Table 1 Some of the properties of LULU operators

Property	*Comment*
$L \leqslant I \leqslant U$	I represents identity operator
$L^2=L$, $U^2=U$	Repetition of same operator would not affect the result
$L \leqslant M \leqslant U$	M denote the median operator
$(LUL)^2=LUL$, $(ULU)^2=ULU^2$[2]	Repetition of same operator would not affect the result
$LUL \leqslant ULU^2$[2]	LUL due to applying L operator twice in different orders make the result smaller than ULU which applies the U operator more
$(LU)^2=LU$, $(UL)^2=UL^2$[2]	Repetition of same operator would not affect the result
U_n (x)$=L_n(x)=x$	'where x is a constant sequence and $x \in M_n$'
$L_n \leqslant U_nL_n \leqslant C_n \leqslant F_n \leqslant L_nU_n \leqslant U_n$[2]	'The C_n and F_n operators (ceiling and floor) are given by: $C_0=L_0U_0=I=U_0L_0=F_0$ $C_{n+1}=L_{n+1}U_{n+1}C_n$; $F_{n+1}=U_{n+1}L_{n+1}F_n$'[2]
$U_nU_k=U_m$ *and* $L_nL_k=L_m$[2]	where m$=\max\{n, k\}$[2]
L_nU_n (and U_nL_n) are idempotent and co-idempotent[2]	'A is idempotent if $A^2=A$ and co-idempotent if I - A is idempotent, therefore they are separators'[2]
$U_nL_n \leqslant M_n \leqslant L_nU_n$[2]	M_n denote the median operator of order n[2] $(M_n\ x)_i=\text{median}\{\ x_{i-n},\ldots,\ x_i,\ldots,\ x_{i+n}\ \}$
L_nU_n (and U_nL_n) are syntone operators[2]	An operator S is syntone if $x>y \rightarrow S_x>S_y$[2]
L_nU_n (and U_nL_n) are ntp operators[2]	'An operator A is neighbour trend preserving (ntp) if for each sequence x, $x_i \geqslant x_{i+1} \rightarrow (Ax)_i \geqslant (Ax)_{i+1}$ $x_i \leqslant x_{i+1} \rightarrow (Ax)_i \leqslant (Ax)_{i+1}$'[2]
L_nU_n (and U_nL_n) are ftp operators[2]	'An operator A is fully trend preserving (ftp) if A is ntp and, $\lvert (Ax)_i - (Ax)_{i+1} \rvert \leqslant \lvert x_i - x_{i+1} \rvert$'[2]
U_n and L_n are variation preserving	A parameter expression that preserves orthonormality under variation up to n order
The operators L_n and U_n are duals in that $U_n(-x)=-L_n(x)$[2]	Negation property[2]
$U_n(x+c)=U_nx+c$ (and $L_n(x+c)=L_nx+c$) for any constant sequence c[2]	Constant Shift property[2]
$U_n(\alpha x)=\alpha U_n(x)$ (and $L_n(\alpha x)=\alpha L_nx$) for any $\alpha>0$[2]	Constant Multiple property[2]
F_n and C_n are separators[2]	'A smoother A is a separator if it is both idempotent and co-idempotent'[2]



5 DPT decomposition for 1D sequence: (a) original signal; (b) D_1; (c) D_2; (d) D_3

processing images with DPT, we need to use the LULU operators on multidimensional arrays. Sub-images are constructed based on the disparity of neighbouring pixels and DPT is based on capturing the contrast in the original image on the boundary of their supports. Detailed comparison of DFT and DPT is provided by Rohwer and Laurie.[2] The summary of comparison between DPT and all DFT, wavelet transform and median transform is given in Table 2.

4.1 1D DPT

In general, DPT can map the bi-infinite sequences such as $\xi=(\ldots, \xi_{-1}, \xi_0, \xi_1, \xi_2, \ldots)$ onto an infinite vector

$$\mathrm{DPT}(\xi)=(D_1(\xi), D_2(\xi), \ldots) \tag{9}$$

where $D_n(\xi)$ is a sequence consisting of a well-separated discrete block pulses with the support n (the set of non-zero values of a function is called the function's support).[8]

As shown in equation (9), DPT of a sequence is a composition of DPT for different orders (pulses), and we shall calculate D_1, D_2, …, D_n, one by one to be able to reconstruct the signal. D_n is a sequence consisting of block pulses with the support n, for instance, it only compares the values of any position with n before and n after, and removes the pulses with width size n.

Table 2 Comparison of DPT with other transforms

Property	*DFT*	*Wavelet Transform*	*Median Transform*	*DPT*
Multi-resolution	√	√	√	√
Predictability	√	√		√
Efficiency	√	√		√
Locality		√	√	√
Incisiveness			√	√

As an example for 1D DPT, consider a sequence of $\xi=\{1, 2, 7\}$, which is shown in Fig. 5a. Following is provided step by step explanation for processing this sequence with 1D DPT. In this example, the calculation for all DPT decomposition for this sequence which is D_1, D_2 and D_3 has been shown. This signal has only three elements; therefore, its DPT can be calculated only up to three decompositions.

Step 1. First, we have to filter the signal with L_1U_1 operator. For calculating L_1, we shall filter the signal with L_1 and remove all the signal's peaks with width of size one; Then we apply U_1 on the result to remove all the valleys with width of size one. L_1U_1 smoothes the signal by removing all the local maximum and minimum pulses of width 1.

Step 1. Please note that, for processing boundary elements and also maintaining the size of the signal, we add zeros to the sequence. For example, for calculating L_1, because it considers the neighbourhood with only one element before and one after, we add one zero to the beginning of ξ and one at the end.

$$L_1=\{1, 2, 2\}$$

The first element of $L_1$is 1 which is obtained by using equation (1), i.e. $\min\{\xi_{i-1}, \xi_i\}=0$ (since $\xi_{i-1}=0$ and $\xi_i=1$), and $\min\{\xi_i, \xi_{i+1}\}=1$ (since $\xi_i=1$ and $\xi_{i+1}=2$), and then $\max\{\min\{\xi_{i-1}, \xi_i\},\min\{\xi_i, \xi_{i+1}\}\}_i=1$. This process is repeated for all elements of ξ. Next, U_1 operator is applied on the result of L_1 and the following result is obtained

$L_1U_1=\{1, 2, 2\}$

The first element of L_1U_1 is 1 which is obtained by using equation (2) on the result of L_1, (here we represent the elements of L_1U_1 by 'x'), i.e., $\max\{x_{i-1}, x_i\}=1$, also $\max\{x_i,x_{i+1}\}=2$ and then $\min\{\max\{x_{i-1},x_i\}, \max\{x_i, x_{i+1}\}\}_i=1$ where $x_i=1$ in sequence L_1.

Step 2. For calculating D_1, we subtract the smoothed signal L_1U_1 from the original signal to get all peaks and valleys of size one as shown in Fig. 5b.

$$D_1=\xi-L_1U_1=\{0, 0, 5\}$$

Step 3. For calculating D_2, we shall find the pulses of width two, and for this reason, we need to apply L_2U_2 operator on the result of step one. It means that we applied $L_1U_1L_2U_2$ operator on the signal according to their orders and we remove all the peaks and valleys of the signal with widths one and two. Please note that, we shall increase the previous sequence's size by adding two zeros at the beginning and two at the end of it to consider the L_2 and U_2 neighbourhood of size two.

$L_1U_1L_2=\{1, 0, 1\}$

The first element of $L_1U_1L_2$ is 1 which is obtained by using equation (1) on the result of L_1U_1 (here we represent the elements of $L_1U_1L_2$ by 'z'), i.e. $\min\{z_{i-2}, z_{i-1}, z_i\}=0$, $\min\{z_i, z_{i+1}, z_{i+2}\}=1$ and then $\max\{\min\{z_{i-2}, z_{i-1}, z_i\}, \min\{z_i, z_{i+1}, z_{i+2}\}\}_i=1$ where $\xi_i=1$ in sequence L_1U_1. In the same way, we calculate $L_1U_1L_2U_2$.

$$L_1U_1L_2U_2=\{1, 1, 1\}$$

Step 4. Here we need to subtract the result of Step 3 from original signal, to get all the pulses with widths one and two.

$$\xi-L_1U_1L_2U_2=\{0, 1, 6\}$$

Step 5. The result of Step 4 gives us the peaks and the valleys with widths one and two, but for calculating D_2, our concern is only to find pulses with width two. Therefore, we shall remove the width one pulses from the result of last step by applying L_1U_1 filter on that.

$$(\xi-L_1U_1L_2U_2)L_1=\{0, 1, 1\}$$

$$D_2=(\xi-L_1U_1L_2U_2)L_1U_1=\{0, 1, 1\}$$

The result of D_2 is shown in Fig. 5c.

Step 6. For calculating D_3, we need to apply L_3U_3 and at the end, keep the pulses with width three. Please note that, at this time for calculating L_3U_3, we shall increase the previous sequence's size by adding three zeros at the beginning and three at the end of it to consider the L_3 and U_3 neighbourhood of size three.

$$L_1U_1L_2U_2L_3=\{0, 0, 0\}$$

The first element of $L_1U_1L_2U_2L_3$ is 0 which is obtained by using equation (1) on the result of $L_1U_1L_2U_2$ (here we represent the elements of $L_1U_1L_2$ by 'w'), i.e. $\min\{w_{i-3}, w_{i-2}, w_{i-1}, w_i\}=0$, also $\min\{w_i, w_{i+1}, w_{i+2}, w_{i+3}\}=0$ and then $\max\{\min\{w_{i-3}, w_{i-2}, w_{i-1}, w_i\}, \min\{w_i, w_{i+1}, w_{i+2}, w_{i+3}\}\}_i=0$, where $w_i=1$ in sequence $L_1U_1L_2U_2$. In the same way, we calculate $L_1U_1L_2U_2L_3U_3$.

$$L_1U_1L_2U_2L_3U_3=\{0, 0, 0\}$$

Step 7. By reducing the filtered sequence with $L_1U_1L_2U_2L_3U_3$ from the original one, we can sift all the peaks and valleys remaining from the $L_1U_1L_2U_2L_3U_3$ filter.

$$\xi-L_1U_1L_2U_2L_3U_3=\{1, 2, 7\}$$

Step 8. This step is similar to Step 5. The difference is that we need to filter with $L_2U_2\,L_1U_1$ to take all the pulses with width less than three out.

$$(\xi-L_1U_1L_2U_2L_3U_3)L_2=\{1, 0, 1\}$$

$$(\xi-L_1U_1L_2U_2L_3U_3)L_2U_2=\{1, 1, 1\}$$

$$(\xi-L_1U_1L_2U_2L_3U_3)L_2U_2L_1=\{1, 1, 1\}$$

1	0	1	1	1
1	0	1	1	0
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1

6 DPT decomposition for 2D

$$D_3 = (\xi - L_1 U_1 L_2 U_2 L_3 U_3) L_2 U_2 L_1 U_1$$
$$= \{1, 1, 1\}$$

The result of D_3 is shown in Fig. 5d.

Step 9. In this step, we want to show that after summing all the DPT decompositions for different pulses, we can get the sequence ξ again.

$$D_1 + D_2 + D_3 = \{1, 2, 7\}$$

We can extend the work from 1D sequences to the multi-dimensional arrays, namely, functions on Z^d ($d>1$). The notation Z^d refers to an n-dimensional space with integer coordinates; for example, a value of Z^3 consists of three integer numbers and specifies a location in three-dimensional (3D) space.[9]

4.2 2D DPT

$A(Z^2)$ denotes the set of all functions with limited support defined on Z^2. A greyscale image is a function $f \in A(Z^2)$ such that the support of f is a finite rectangular subset Ω of Z^2. The DPT of a function $f \in A(Z_2)$ is a vector of the form[10]

7 2D DPT for four-connectivity

D_1

1	0	1	1	1
1	0	1	1	0
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1

local maximum sets

D_1

1	0	1	1	1
1	0	1	1	0
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1

local minimum sets

D_2

1	0	1	1	1
1	0	1	1	1
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1

local maximum sets

D_2

0	0	1	1	1
0	0	1	1	1
0	0	1	1	1
0	1	1	1	0
1	0	1	0	1

local minimum sets - none

D3 until D7

0	0	1	1	1
0	0	1	1	1
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1

local maximum sets - none

D3 until D7

0	0	1	1	1
0	0	1	1	1
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1

local minimum sets - none

D_8

0	0	1	1	1
0	0	1	1	1
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1

local maximum sets - none

D_8

0	0	1	1	1
0	0	1	1	1
0	0	1	1	1
0	1	1	1	1
1	0	1	1	1

local minimum sets - none

Decomposition complete (constant image reached)

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

8 2D DPT for eight-connectivity

$$\mathrm{DPT}(f) = [D_1(f), D_2(f), \ldots, D_N(f)] \tag{10}$$

which is finite due to the finite support of f. In the equation above, N represents pixels in the image. $D_N(f)$ is given as

$$D_n(f) = \sum_{s=1}^{r(n)} \varphi_{ns}$$

where φ_{ns} represents the pulses. The functions $\varphi_{ns}=1, 2, \ldots, \gamma(n)$, where $\gamma(n)$ is a function of n, affecting the number of pulses of each pixel. These functions are discrete pulses with support of size n, $n=1, 2, \ldots, \gamma(n)$. In this context, a discrete pulse is a function $\varphi \in A(Z_2)$ which is constant on a connected set W and zero elsewhere. The set W is called the support of the pulse φ, $W=\mathrm{supp}(\varphi)$. The value of φ on W is called the value of the pulse. If the value of φ is positive, then φ is an up-pulse; if it is negative, φ is a down-pulse. Using DPT, we represent the function $f \in A(Z^2)$ as a sum of pulses.[10]

$$f = \sum_{n=1}^{N} D_n(f) = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \varphi_{ns} \tag{11}$$

Furthermore, similar to the DPT of sequences, the decomposition above preserves the total variation (TV), a parameter expression that preserves orthonormality under variation up to order n, of f as

$$\mathrm{TV}(f) = \sum_{n=1}^{N} \mathrm{TV}[D_n(f)] = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \mathrm{TV}(\varphi_{ns}) \tag{12}$$

The TV is an important characteristic of an image. It has been successfully used in noise removal procedures.[11,12] The equalities show that the decomposition in equation (11) does not generate noise. The pulses in equation (11) also have a more direct meaning. The contrast in the original image at the boundary of the support of any pulse is at least as much as the value of that pulse.[10]

The DPT for a function $f \in A(Z^2)$ is obtained via iterative application of the operators L_n and U_n with n increasing from 1 to N. For a given n, the sequencing of L_n and U_n does not affect the properties stated earlier. However, it introduces bias towards up-pulses or down-pulses. Let P_n denote either the composition $L_n \circ U_n$ (for combining L and U, we apply opening operators; in mathematical

9 Impulse noise removal of the Lena's image with different LULU operators

morphology, opening is the dilation of the erosion of a set A by a structuring element B:$A \circ B$=$(A \odot B) \oplus B)$[13] or the composition $U_n \circ L_n$ and let $Q_n = P_n \circ P_{n-1} \circ \ldots \circ P_2 \circ P_1$. In the general theory of mathematical morphology, Q_n is known as an alternating sequential filter (an alternating sequential filter is an iterative application of openings and closings with structuring elements of different sizes[14]).

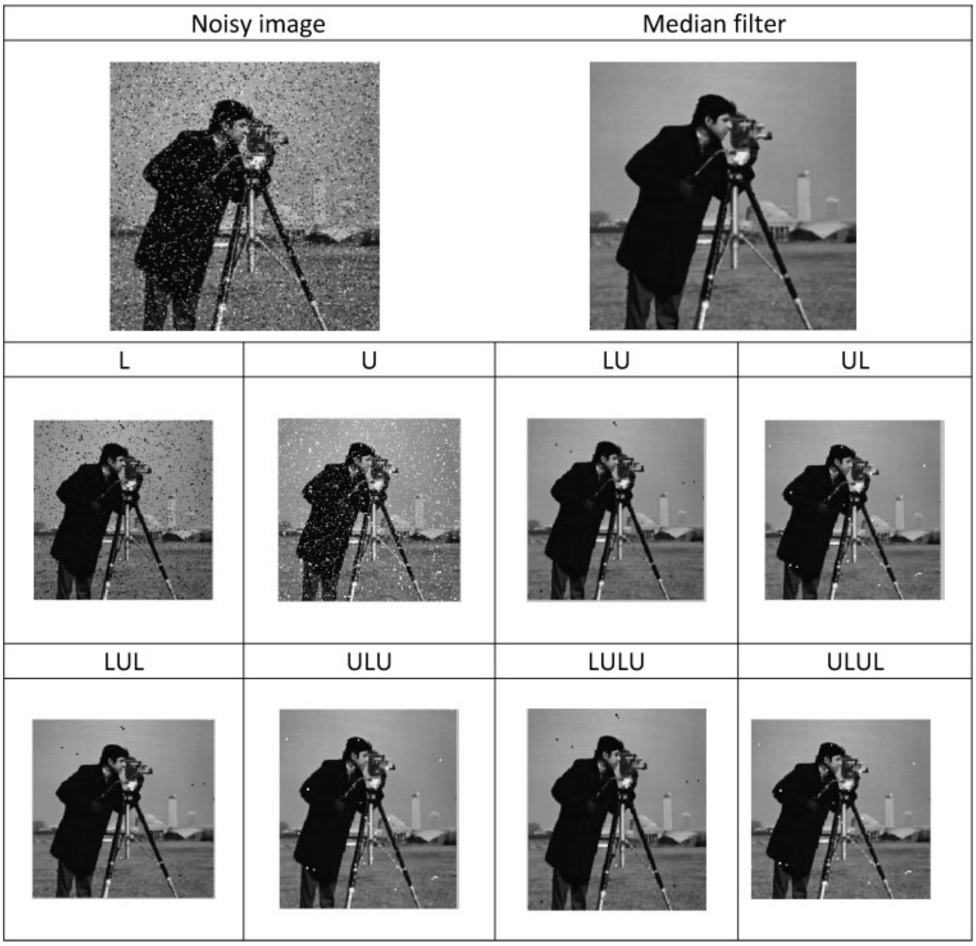
However, here we are interested in the portions of the image which are filtered out by the application of P_n, n=1, 2, …, N. We ultimately obtain $Q_N(f)$, which is a constant function containing no information about the original image, except possibly the general level of illumination. The rest of the information carried by f is in the layers peeled off,[10] i.e. the number of pulses which are considered. More precisely

$$f = (id - P_1)(f) + [(id - P_2) \circ Q_1](f) + [(id - P3) \circ Q_2](f) + \ldots + [(id - P_{N-1}) \circ Q_{N-2}](f) + [(id - P_N) \circ Q_{N-1}](f) + Q_N(f) \tag{13}$$

where *id* denotes the identity operator. Let us note that a similar iterative application of area opening and area closing operators is used by Acton and Mukherjee[15] for image classification. Filtering is done for selected values of n and instead of the layers of peeled off portions, the authors keep a record of filtered images at every scale. This would be $Q_n(f)$ in the notation adopted here.[16] For more information, please refer to Refs. 10, 16 and 17.

DPT for 2D considers a wider neighbourhood for each pixel compared to 1D. Besides, the size of support can vary up to the matrix's size. An example for 2D DPT is provided here, which shows the effects of different pulses on the image. The following steps show the DPT decomposition for the image with the pixel values illustrated in Fig. 6.

Please note that we can consider a different neighbourhood, but here we just illustrated the result of the four-connectivity and eight-connectivity neighbourhoods as shown in Figs. 7 and 8, respectively. 2D DPT is concerned about connectivity for calculating



10 Impulse noise removal of the Cameraman's image with different LULU operators

different decompositions; we shall follow the steps below. The properties of connectivity and segmentation are described in more detail in Refs. 18 and 19.

Step 1. The first step is finding the local maximum sets. For this, we shall find the connected sets. For example, to calculate D_1, we can consider all the pixels one by one because each one makes a set of size one. For any D_n, any n pixels with the same value that are connected can be considered as one set of size n. Any set which has a higher value than its neighbours will be highlighted and its value will be changed to its neighbour values.

Step 2. This step is the same as the first step; the difference is that we are looking for local minimum sets on the result of the previous step. After finding the local minimum sets, we convert the whole set's value to its neighbour values and continue with the next decomposition (D_{n+1}), which shall repeat Step 1 followed by Step 2 for the result of D_n.

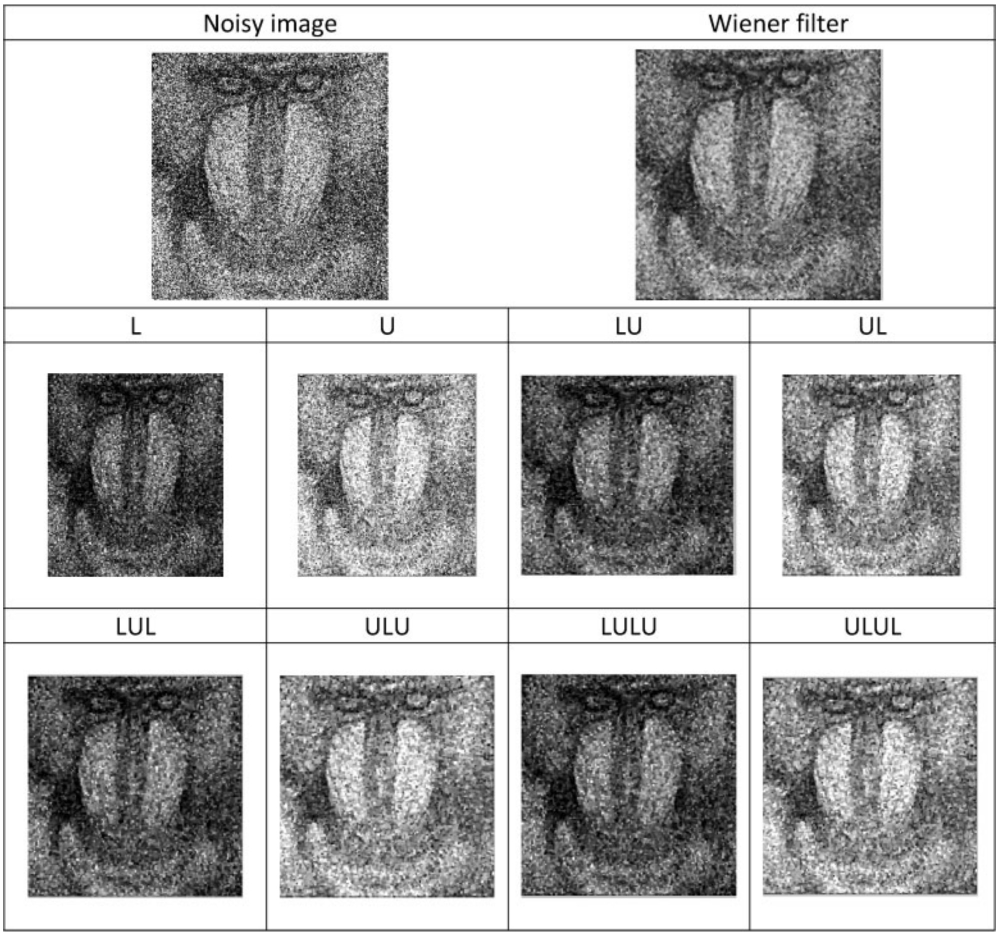
Step 3. The last step is when all values of the image become the same. This will be the point when we shall stop.

5 APPLICATIONS

As mentioned earlier, LULU is being used in 1D array as well as in 2D arrays. The following provides some of the applications of LULU operators and DPT.

5.1 Filtering

For 1D sequence, LULU operators are being used as smoothers or filters, which help to remove the peaks and pulses that have a small width. Depending on the application, various operators can be used and the result of each operator is different from the others. Usually, all these LULU operators are being used for discrete data. Anguelov[20] extended it to continuous



11 Gaussian noise removal of the Baboon's image with different LULU operators

time data. In Refs. 6 and 7, LULU operators are compared and discussed for 1D filtering.

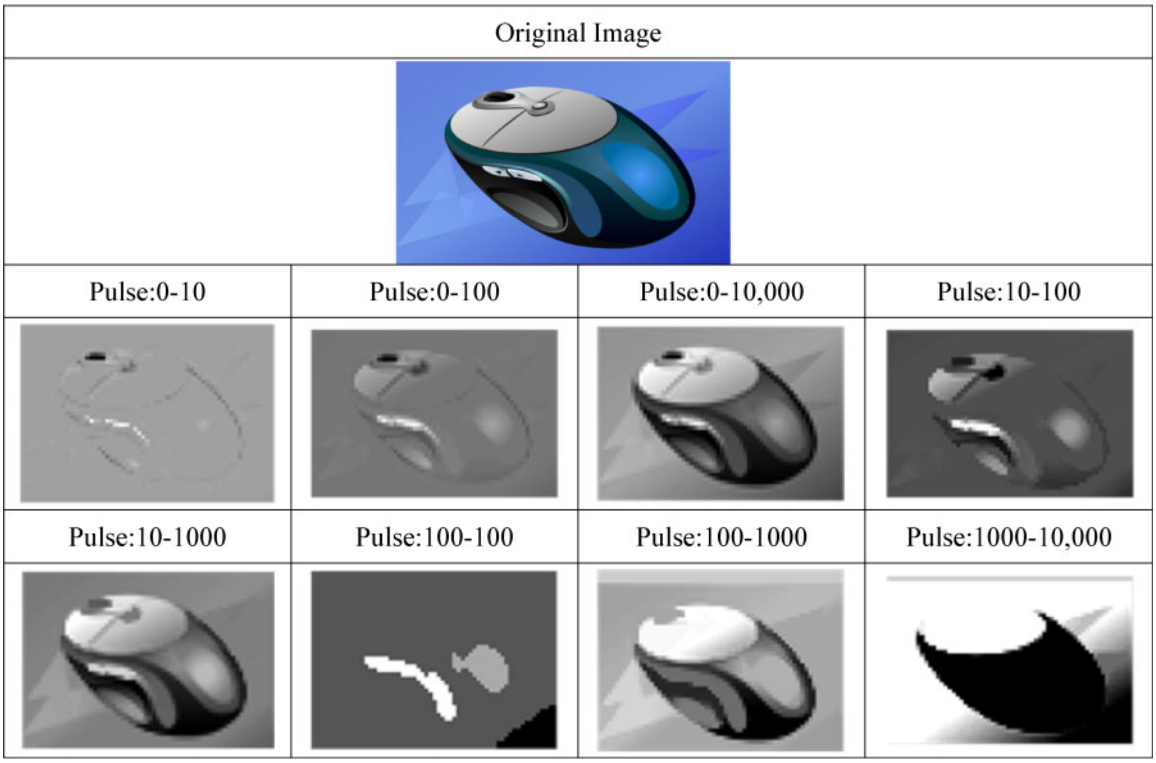
5.2 Statistics

LULU smoothers can be used in the analysis of financial data. They have also been used in econometrical and statistical literature. The usefulness of these operators can be understood better in the case of distributions with high kurtosis, distributions that are very common in the financial risk environment as well as in distributions with fat tails on one side like the gamma distribution.[21] LULU smoothers are still not very well known in this field. Simulations have been performed for financial data analysis.

5.3 Noise reduction in images

Reducing noise is one of the most important requirements in image processing because of the loss of information that results when an image is corrupted by noise. There are many different types of noise, but usually images are affected by impulse noise whose major source is atmosphere. Usually, median filters are used to remove impulse noise due to simplicity but Kao[22] used the LULU operators for preserving the image details. He offered parallel smoothing with two sub-operators, L and U. Any of these two properties can be applied first and then followed by the other, which uses morphological opening function. As more operators are applied on the noisy image, the image becomes smoother and less noisy. Filtering can be stopped according to some metric measure, for example, comparing the value of the smoothed pixel with its original value using mean square error. If the image gets too smooth, some detailed information may be removed from the image which may not be noise.

The result of applying different LULU operators on a noisy image with impulse noise is shown in Fig. 9. In Fig. 9, the first image shows the noisy image, the second is the image after being filtered



12 Results of different ranges of pulses in DPT of an image

with Median filter and the other images are showing the result of L, U, LU, UL, LUL, ULU, LULU and ULUL, respectively. The figure shows that by applying more LULU operators, there is significant reduction in impulse noise. At first when only L or U operators are applied, L removes more white points of impulse noise and U removes more black points. LU smoother applies L first and then U and it helps to remove more noise than a single operator; this procedure can be continued till we get our desired result. Figure 10, shows the same result as Fig. 9 on different image with different noise densities.

Figure 11 shows the comparison of LULU filters and Wiener filter for Gaussian noise removal. The Wiener filter is performing better than LULU filters for Gaussian noise removal due to its characteristic which matches this type of noise behaviour, but for LULU filters, since they are locally operators, their performance is not as good as Wiener filter.

5.4 Object detection

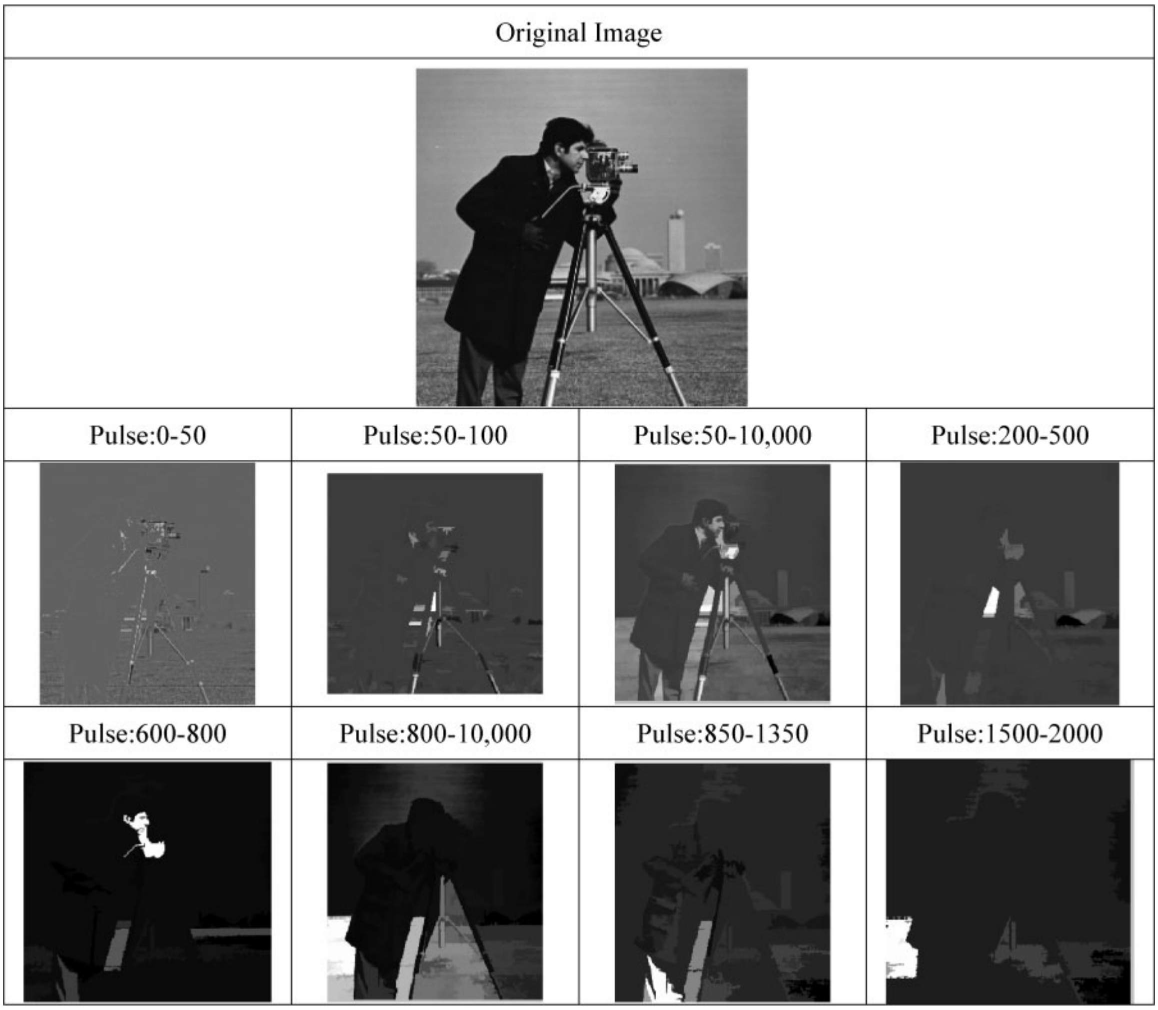
A very useful application of the discrete pulse decomposition is via partial reconstructions of images. This is a new approach in object detection. LULU operators enable DPT decomposition to be applied to images and remove some undesirable parts of the image or extract desirable object from image. As mentioned earlier in DPT properties, each image will be reconstructed from pulses with different sizes. Playing with the number of these pulses can remove the unnecessary parts of the image or extract object of interest.

In Refs. 8, 10 and 23, the authors worked on object detection and object removal in an image. Figure 12 shows the application of DPT on 'Mouse' image. As shown in the figure, when the number of pulses is larger, more parts of the image are visible and when it is restricted to a small interval, then we can only view few parts. By applying different ranges of pulses, we can extract the desired information from the image that is of interest. Figure 13 illustrated the concept of DPT with different pulses for an image of a cameraman of people.

Table 3 shows the computational complexity with various pulses. As shown in the table, the computational times for different pulses are independent of the number of pulses. This can be explained according to the concept of support.

6 RECOMMENDATION AND CONCLUSION

In this paper, LULU operators and the concept of DPT based on LULU operators were discussed.



13 Results of different range of pulses in DPT of an image

These two methods are implemented for 1D sequences as well as 2D arrays (images) for different applications. They appear to perform better compared to other methods. Table 2 illustrates the comparison of DPT with other methods. LULU is already being used widely in filtering and smoothing operations, especially in econometrical and statistical literature. Now, many researchers employ LULU and DPT for image analysis too. DPT is a very efficient operator for multi-dimensional arrays unlike median operator. It is one of the best filtering methods for removing impulse noise from images as well as 1D sequences. Currently, it is being used for edge deduction and contour tracing for object detection and object extraction applications. The next step is to use DPT for 3D applications, i.e. depth map estimation, 3D shape extraction, etc.

Table 3 Computational complexity with various pulses of DPT for original image in Fig. 10

Serial Number	Number of Pulses	Computational Time (Seconds)
1	0–10	1.882516
2	0–100	1.943584
3	0–10,000	2.149764
4	10–100	1.921685
5	10–1000	2.02025
6	100–100	1.774137
7	100–1000	1.979073
8	1000–10,000	1.957978

ACKNOWLEDGEMENTS

This work is supported by E-Science grant funded by the Ministry of Science, Technology and Innovation, Government of Malaysia (no. 01-02-02-SF0064). We would like to thank Professor Roumen Anguelov and

Ms Inger Fabris-Rotelli for their support for providing us with information on 2D DPT.

REFERENCES

1 Conradie, W. J., Wet, T. D. and Jankowitz, M. Exact and asymptotic distributions of LULU smoothers. *J. Comput. Appl. Math.*, 2006, **186**, 253–267.

2 Rohwer, C. H. and Laurie, D. P. The discrete pulse transform. *SIAM J. Math. Anal.*, 2007, **38**, 1–28.

3 Jankowitz, M. D. 'Some statistical aspects of LULU smoothers', Master thesis, University of Stellenbosch, Stellenbosch, South Africa, 2007.

4 Anguelov, R. and Fabris-Rotelli, I. Properties of the discrete pulse transform for multi-dimensional arrays. *IEEE Trans. Image Process.*, 2010, **19**, 3012–3023.

5 Heller, P. N. and Wells, R. O. The Spectral Theory of Multiresolution Operators and Applications, 1994, pp. 13–31 (Academic Press, San Diego, CA).

6 Anguelov, R. and Fabris-Rotelli, I. LULU operators and discrete pulse transform for multi-dimensional arrays. *IEEE Trans. Image Process.*, 2010, **19**, 1–13.

7 Du Toit, J. P. 'The discrete pulse transform and applications', University of Stellenbosch, Stellenbosch, South Africa, 2007.

8 Anguelov, R. and Fabris-Rotelli, I. N. Discrete pulse transform of images. *Lect. Notes Comput. Sci.*, 2008, **5099/2008**, 1–9.

9 Marogos, P. and Schafer, R. W. Morphological systems for multidimensional signal processing. *Proc. IEEE*, 1990, **78**, 690–710.

10 Anguelov, B. Discrete pulse transform of images: algorithm and applications, Proc. 19th Int. Conf. on *Pattern recognition: ICPR 2008*, Tampa, FL, USA, December 2008, IEEE, pp. 1–4.

11 Chambolle, A. and Lions, P. Image recovery via total variation minimization and related problems. *Numer. Math.*, 1997, **76**, 167–188.

12 Leonid, I., Osher, S. and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D*, 1992, **60**, 259–268.

13 Morales, A., Acharya, R. and Ko, S. Morphological pyramids with alternating sequential filters. *IEEE Trans. Image Process.*, 1995, **4**, 965–977.

14 Pei, S., Lai, C. and Shih, F. An efficient class of alternating sequential filters in morphology. *Graph. Models Image Process.*, 1997, **59**, 109–116.

15 Acton, S. T. and Mukherjee, D. P. Scale space classification using area morphology. *IEEE Trans. Image Process.*, 2000, **9**, 623–635.

16 Fabris-Rotelli, I. and Walt, S. J. The discrete pulse transform in two dimensions, Proc. 20th Annual Symp. of the Pattern Recognition Association of South Africa, Stellenbosch, South Africa, December 2009, Pattern Recognition Association of South Africa.

17 de Wet, T. and Conradie, W. Smoothing sequences of data by extreme selectors, Proc. 7th Int. Conf. on *Teaching statistics: ICOTS-7*, Salvador, Bahia, Brazil, July 2006, IASE, pp. 1–5.

18 Serra, J. In Mathematical Morphology and Its Applications to Image and Signal Processing, 1996, pp. 81–96 (Kluwer Academic Publishers, Boston, MA).

19 Serra, J. A lattice approach to image segmentation. *J. Math. Imag. Vision*, 2006, **24**, 83–130.

20 Anguelov, R. In Constructive Theory of Functions, 2006, pp. 22–34 (Marin Drinov Academic Publishing House, Sofia).

21 Conradie, W. J. and de Wet, T. 'LULU smoothers and their statistical properties', Technical report 03/01, University of Stellenbosch, Stellenbosch, South Africa, 2005, pp. 1–37.

22 Kao, O. Modification of the LULU operators for preservation of critical image details, Proc. Int. Conf. on *Imaging science, systems and technology: CISST 2001*, Las Vegas, NV, USA, June 2001, CSREA Press.

23 Anguelov, R. and Plaskitt, I. A class of LULU operators on multi-dimensional arrays', Technical report UPWT2007/22, University of Pretoria, Pretoria, South Africa, 2008.