J. Vis. Commun. Image R. 24 (2013) 303–317

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

3D shape from focus using LULU operators and discrete pulse transform in the presence of noise

Roushanak Rahmat [a], Aamir Saeed Malik [a,*], Nidal Kamel [a], Humaira Nisar [b]

[a] Department of Electrical and Electronic Engineering, Universiti Teknologi, PETRONAS, 31750 Tronoh, Perak, Malaysia
[b] Department of Electronic Engineering, Faculty of Engineering and Green Technology, Universiti Tunku Abdul Rahman, 31900 Kampar, Perak, Malaysia

A R T I C L E I N F O

Article history:
Received 28 September 2011
Accepted 28 December 2012
Available online 10 January 2013

Keywords:
Shape from focus
LULU operators
Discrete pulse transform

A B S T R A C T

3D shape recovery is an interesting and challenging area of research. Recovering the depth information of an object from a sequence of 2D images with varying focus is known as shape from focus. Focus value of an image carries information about the object and shape from focus is a method which depends on different focused value images. It reconstructs the shape/surface/depth of an object based on the different focused values of the object. These different focused valued images should be captured from the same angle. Calculating the shape of the object from different images with different focused values can be done by applying sharpness detection methods to maximize and detect the focused values. In this paper, we propose new 3D shape recovery techniques based on LULU operators and discrete pulse transform. LULU operators are nonlinear rank selector operators that are efficient with low complexity. They hold consistent separation, total variation and shape preservation properties. Discrete pulse transform is a transform that decomposes image into pulses. Therefore selection of right pulses, give sharpest focus values. The proposed techniques provide better result than traditional techniques in a noisy environment.

1. Introduction

3DTV requires the facility of capturing and analyzing the multiview images and compressing and transmitting massive volume of data through the communication channel. The most primary concern in 3D technology is 3D shape extraction of different surfaces and objects. Additionally, there are a variety of applications that require 3D shape of an object [1], e.g., virtual games, product modeling, facial representation, biomedical imaging, microscopic imaging, vehicle navigation, astronomy, etc.

Image focusing is one of the principal schemes of 3D shape reconstruction. The shape from focus (SFF) is one of the shape recovery methods which reconstruct the 3D shape from a sequence of 2D images taken from the same angle. The SFF images of an object are defined as a number of frames which carry different focused values of object's surface. Each frame carries different focusing information about different parts of the object.

Fig. 1 shows the basic image formation geometry, where f indicates the focal length of the lens, u refers to the distance of the object from the lens and v shows the distance of the image from the lens. Since f and v are known, therefore, u can be calculated which provides the depth map. This figure illustrates that when there is an object at point P, it will be well focused at the point P' and if the object point is not focused in image plane, there will be a blur image around P'' [1]. The relationship between the object distance, focal distance of the lens, and the image distance, is given by the Gaussian lens law:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \tag{1}$$

Reconstructing the 3D shape based on the focused values, requires a sharpness extraction technique which can detect the focused parts in each frame. There are different sharpness measures for detecting the focus in image pixels along all the frames of SFF. Laplacian, modified Laplacian (ML), sum of the modified Laplacian (SML), Tenenbaum, gray level variance (GLV), mean, curvature and M2, are some of the methods which detect the best focus value [2].

Many different focusing techniques are proposed in last few decades. Horn [3] in 1968 proposed a technique based on Fourier transform. Tenenbaum [4] in 1970 built up the gradient magnitude maximization technique which is based on sharpness of edges to optimize focus quality. Buffington [5] in 1974 introduced aperture-plane distortion. Erteza [6] in 1976 obtained an index value for sharpness by considering the intensity distribution of the image. Jarvis [7] in 1976 established a new technique based on the sum-modulus-difference. Pentland [8] in 1985 proposed the assessment of image blur. Grossmann [9] in 1987 suggested the evaluation of depth of edge points by considering the blur of the

* Corresponding author.
E-mail address: aamir_saeed@petronas.com.my (A.S. Malik).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.01.005

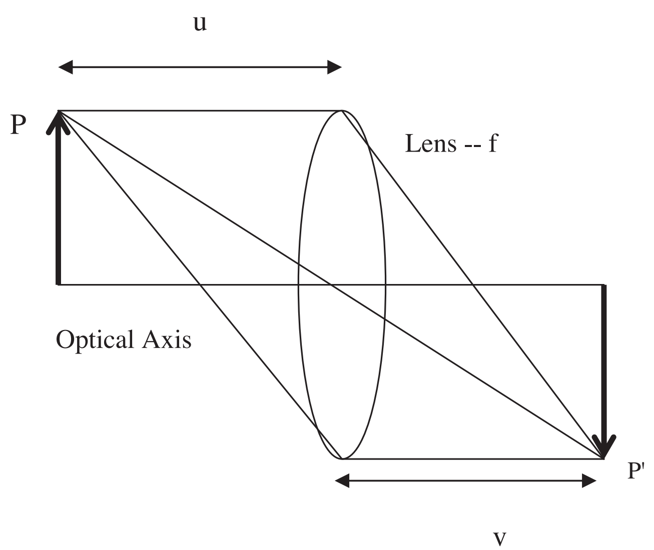

Fig. 1. Formation of focused and defocused images.

edges. Krotkov in 1987 discussed about the distance calculation of the sharply focused point. Darrell and Wohn [10] in 1988 applied Laplacian and Gaussian pyramids for depth estimation. Nayar [11] in 1990 built the first SFF system and introduced Gaussian interpolation in 3D microscope. He also introduced sum of modified Laplacian in shape recovery [12] in 1994. In 1992 Dillion [13] combined shape from focus and stereo to get better result. Asada [14] in 1998 described eliminating windowing method. Zhang [15] in 2000 proposed 2nd/4th order central moment as a sharpness detector. Helmi [16] in 2002 introduced new techniques based on mean, curvature and point focus methods. Malik et. al. [17] proposed a fuzzy-neural approach for estimation of depth map using focus.

Shape from focus (SFF) suffers from various limitations that prevent its usage in practical 3D reconstruction scenarios. One of the major limitation is due to the discrete nature of data, i.e., the number of frames are finite and the focus information between the two frames is lost. Increasing the number of frames is not an option because it increases the computational complexity. Another limitation is the magnification error that is due to change in the focus value. The SFF method requires narrower focus so that only certain pixels are best focused in a frame. Like all computer vision applications, illumination is another factor that affects the SFF technique. The texture of object also contributes to the error in 3D shape recovery using SFF. Various types of noise introduce error in estimation of depth map using SFF. Therefore, there is a need for new focus measures that can address these issues and limitations. In this paper, we propose new focus measures that address the issues of noise and perform well as compared to the existing focus measures. The proposed focus measure addresses noise because it has two inherent properties, i.e., noise reduction and sharpest focus point extraction.

In this work, we propose new focus measures based on LULU filters and discrete pulse transform. The proposed techniques are implemented on simulated and actual SFF data. The proposed methods are also combined with the traditional focus measure methods such as sum of modified Laplacian (SML), Tenenbuam and gray level variance (GLV). Based on the quantitative and qualitative experimental results, the proposed techniques are more accurate in focus value extraction and shape recovery in the presence of noise. The rest of the paper is organized as follows: Section 2 discusses the traditional SFF techniques, Section 3 explains the LULU operators and discrete pulse transform (DPT), Section 4 provides details of the proposed methods and Section 5 discusses results, that is followed by conclusion.

2. Shape from focus techniques

SFF method requires capturing different image frames for the same object from a specific angle. In general, there are two methods of capturing different sequences; one can be by changing the focus value of the lens and keeping the object and camera's position fixed. Other way is keeping the camera's focus value fixed and change the distance between object and camera gently for different shots. The first case is used when the position of both the camera and the object are fixed, and this information is used for recovering the 3D shape of some static object. Therefore, only the camera focus parameter is changed in this case. In the latter case, the position of the object or the camera may not be fixed, e.g., a person walking towards or away from the camera. In this case, the change in focus is due to the change in the position of the object.

Fig. 2 shows a sequence of frames that correspond to different levels of object focus obtained through a single camera. In Fig. 3, the test image shows different focused images of a cone object. This sequence is constructed from 97 different images with different focus values of the cone object.

After collecting the data we determine the exact frame where the depth of the object is in focus or has the maximum sharpness. A sharpness measure or focus measure (FM) for each image in the sequence is computed at each pixel location using a small window around the pixel. The success of any focus measure depends on how accurate the sharpness in image pixels could be detected. By applying different well known mathematical techniques for SFF such as Laplacian [11], modified Laplacian (ML), sum of the modified Laplacian (SML), Tenenbaum (TEN) [16], gray level variance (GLV), mean, curvature and M2, the best depth value for each single point of the object can be obtained. This information shows the highest amount of sharpness or best focusing values from different image captures.

By selecting the pixel with highest focus value among all frames, the 3D shape of the object from a single view can be reconstructed. The 3D shape is reconstructed by calculation of depth map using the best focused point for every pixel, i.e. by calculating the maximum value for each pixel (i,j) in all the frames. Let FM be the focus measure value for each pixel (i,j), k be the image frame number, D_p be a 2D matrix containing the best focused value (sharpness/maximum value) for each of the pixels and D_f be a 2D matrix containing the corresponding frame numbers where the pixels are best focused, then the equation for depth map calculation using SFF method is given as:

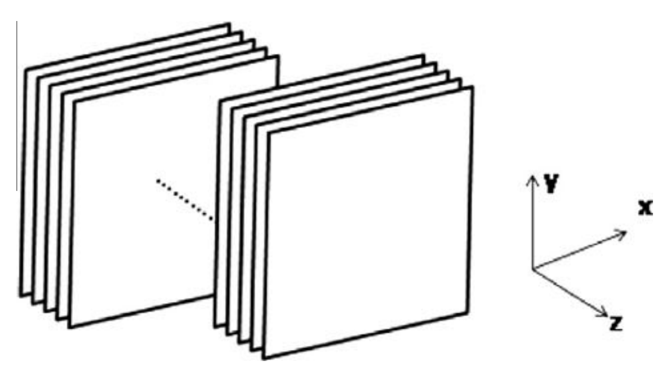

Fig. 2. Sequence of images.

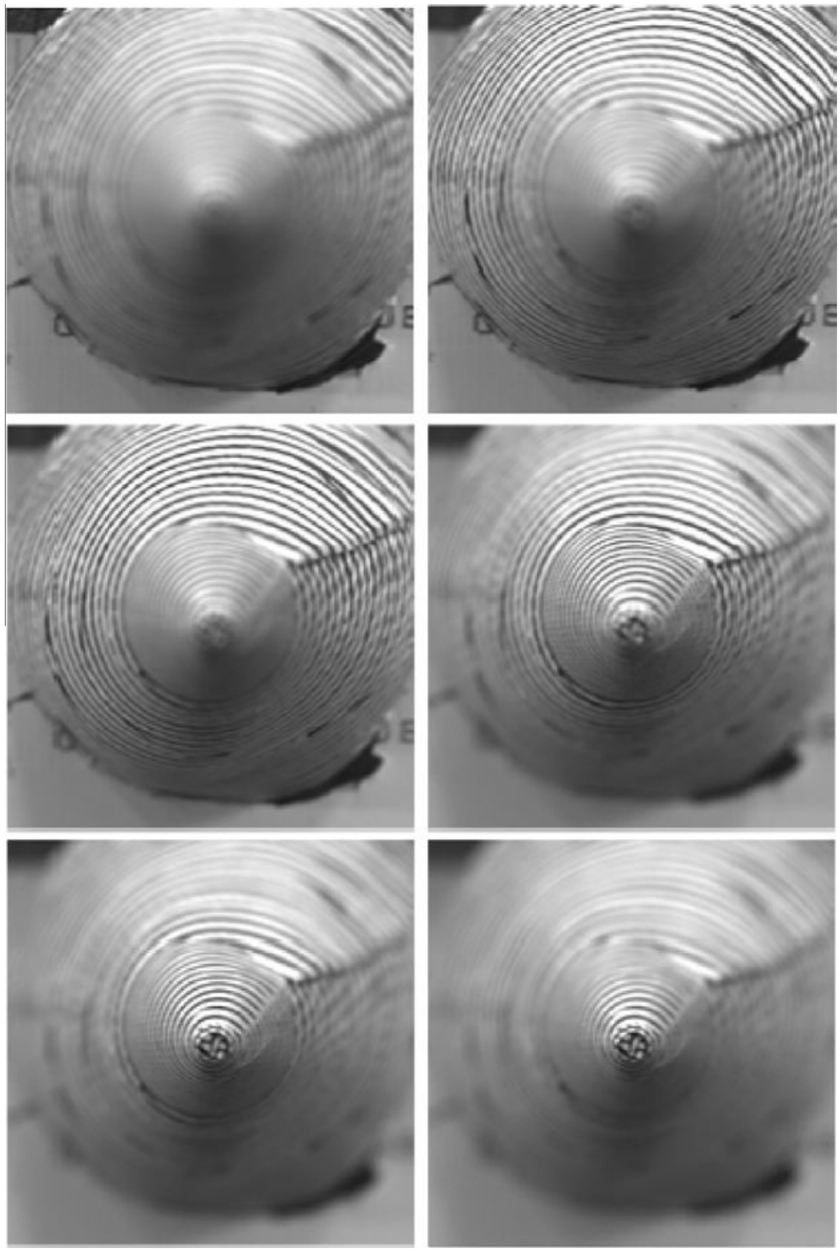

Fig. 3. Test image with different focusing values.

$$[D_p(i,j), D_f(i,j)] = \max[{}^k FM(i,j)] \quad (2)$$

After getting the initial depth map, approximation methods can be used in order to improve the results. Among these methods include the Gaussian interpolation and neural networks [2]. Some of the focus measures which calculate sharpness are used in this work for comparison with the proposed techniques. They include SML, Tenenbaum and GLV. For more information about other focus measure methods, please refer to [2].

2.1. Sum of modified Laplacian method [2]

Sum of modified Laplacian (SML) is a modified version of Laplacian operators. They are powerful sharpness detectors which are widely used in SFF techniques for detecting the focused value for each pixel. Eq. (3) formulates SML techniques when $p(x,y)$ is a pixel in the neighborhood $U(x_0,y_0)$ of pixel $(x_0,y_0))$ [2].

$$SML = \sum_{p(x,y) \in U(x_0,y_0)} \left(\frac{\partial^2 f(x,y)}{\partial^2 x}\right)^2 + \left(\frac{\partial^2 f(x,y)}{\partial^2 y}\right)^2 \quad (3)$$

2.2. Tenenbaum [2]

Tenenbaum (TEN) technique is based on sobel operators. It sums these operators along x axis and y axis. In Eq. (4), $f(x, y)$ is the image function and $p(x,y)$ is a pixel in the neighborhood $U(x_0,y_0)$ of pixel (x_0,y_0) [2].

$$TEN = \sum_{p(x,y) \in U(x_0,y_0)} (f_x(x,y)^2 + f_y(x,y)^2)^2 \quad (4)$$

2.3. Gray level variance [2]

Gray level variance (GLV) method is one of the best methods and has high accuracy in sharpness detection in images. It has been widely used in focus value calculation for SFF data. In Eq. (5), $f(x, y)$ is the image function and $\mu_U(x_0, y_0)$ is the gray values' mean in the neighborhood $U(x_0, y_0)$ of pixel x_0, y_0 [2]).

$$GLV = \frac{1}{N-1} \sum_{p(x,y) \in U(x_0,y_0)} (f(x,y) - \mu_{U(x_0,y_0)})^2 \tag{5}$$

3. LULU operators and discrete pulse transform

3.1. LULU

Rohwer and Toerien in late 1980s introduced a novel, innovative nonlinear smoother, named LULU smoothers, based on extreme order statistics [18]. LULU operators are local and nonlinear, fully trend preserving, which make them an efficient means for multi-resolution analysis of sequences [19]. For detailed discussion of properties as well as their proofs, check Refs. [19,20].

These MinMax (MaxMin) operators consist of the sub-operators L (low) and U (up) with different order for different filters. For a given bi-infinite sequence, $\xi = (\xi i)$, $i \in Z$, the 1D LULU operators can be defined as follows [21]:

$$(L_n\xi)_i = \max\{\min\{\xi_{i-n}, \ldots, \xi_i\}, \min\{\xi_i, \ldots, \xi_{i+n}\}\}, \quad i \in Z \tag{6}$$

$$(U_n\xi)_i = \min\{\max\{\xi_{i-n}, \ldots, \xi_i\}, \max\{\xi_i, \ldots, \xi_{i+n}\}\}, \quad i \in Z \tag{7}$$

Fig. 4 shows the effect of L and U smoothers on a randomly generated signal.

LULU operators can be used on 2D images for different applications like object extraction, noise filtering, etc. In 2D LULU, the neighbors of the pixel $f(i,j)$ is divided into four regions as shown in Eqs. (8)–(11) and is illustrated in Fig. 5;

$$I_1 = [f(i,j-1), f(i,j), f(i+1,j-1), f(i+1,j)], \tag{8}$$

$$I_2 = [f(i-1,j-1), f(i-1,j), f(i,j), f(i,j-1)], \tag{9}$$

$$I_3 = [f(i,j+1), f(i,j), f(i+1,j), f(i+1,j+1)], \tag{10}$$

$$I_4 = [f(i-1,j+1), f(i-1,j), f(i,j), f(i,j+1)]. \tag{11}$$

Based on the defined neighborhood in Eqs. (8)–(11), L and U operators can be defined as follow;

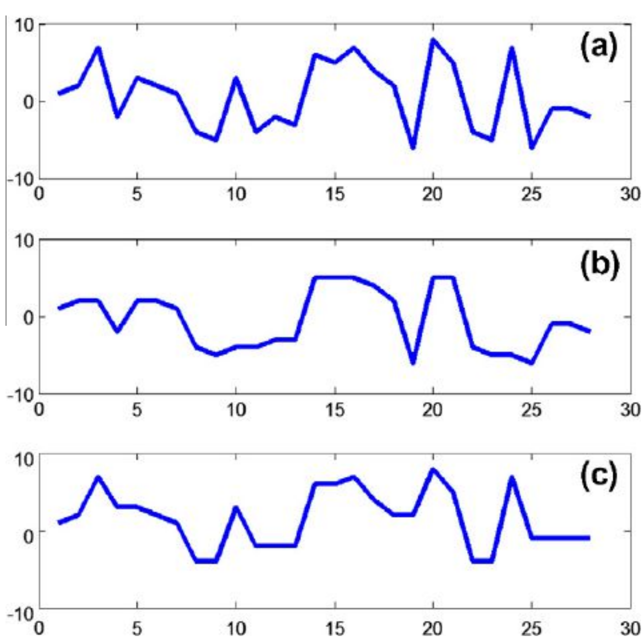

Fig. 4. (a) Original signal and (b) result of L smoother on the signal, and (c) result of U smoother on the signal.

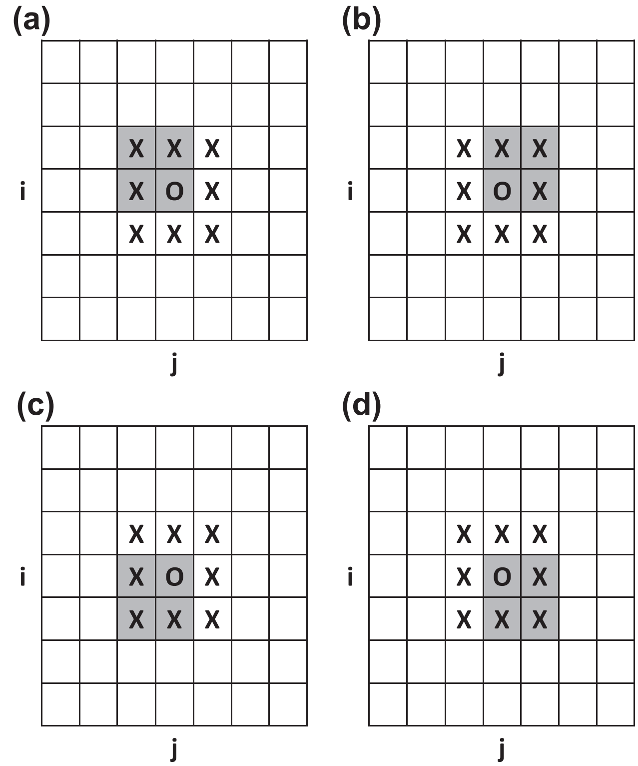

Fig. 5. Illustration of neighbors for Eqs. (8)–(11), (a) I_1, (b) I_2, (c) I_3, and (d).

$$L(i,j) = \max(\min(I_1), \min(I_2), \min(I_3), \min(I_4)), \tag{12}$$

$$U(i,j) = \min(\max(I_1), \max(I_2), \max(I_3), \max(I_4)). \tag{13}$$

3.2. Discrete pulse transform

Discrete pulse transform (DPT) is defined as the composition of pulses; each pulse is a string of zero values which is zero everywhere except for the few consecutive elements. DPT can be applied in images for object extraction by recognizing the corresponding pulses for different objects in the image. DPT in image processing is based on LULU operators on multidimensional arrays. The DPT of a function $f \in A(Z^2)$ is a vector of the form [22];

$$DPT(f) = (D_1(f), D_2(f), \ldots, D_N(f)). \tag{14}$$

In Eq. (14), N refers to the number of total pixels in the image. is summation of pulses and hence is given as $D_N(f) = \sum_{i=1}^{\gamma(n)} \varphi_{ns}$ where φ_{ns} represents the pulses. The pulses are given as: φ_{ns} where $s = 1,2,\ldots,\gamma(n)$, and $\gamma(n)$ is a function of n and it provides information on the total number of pulses required for each pixel. Hence, its value affects the number of pulses that will process each pixel. These functions are discrete pulses with support of size, n, $n = 1,2,\ldots,\gamma(n)$. "Support" indicates that this function will exist for some finite values and will be zero for all other values. In this context, let W be a connected set for which this discrete function exists. The discrete function is zero outside this connected set. The set is called the support of the pulse, i.e. $w = \boldsymbol{supp}(\varphi)$. The value of on is called the value of the pulse, i.e., value of pulse can be positive or negative. If the value of φ is positive then is an up-pulse, if it is negative, is a down-pulse. Using DPT, we represent a function $f \in A(Z^2)$ as a sum of pulses [22].

$$f = \sum_{n=1}^{N} D_n(f) = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \varphi_{ns} \tag{15}$$

The discrete pulse transform for a function $f \in A(Z^2)$ is obtained via iterative application of the operators L_n and with U_n increasing from

1 to N. For a given n, the sequencing of and does not affect the properties of DPT. However, it introduces bias towards up-pulses or down-pulses. Let denote either the composition $L_n \circ U_n$ (for combining L and U, we apply opening operators; In mathematical morphology, Q_n opening is the dilation of the erosion of a set A by a structuring element B: $A \circ B = ((A \ominus B) \oplus B)$ [23] or the composition $U_n \circ L_n$ and let $Q_n = P_n \circ P_{n-1} \circ \cdots \circ P_2 \circ P_1$. In the general theory of mathematical morphology, 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 [24].

However, here we are interested in the portions of the image which are filtered out by the application of, P_n, $n = 1, 2, \ldots, \gamma(n)$. We ultimately obtain $Q_n(f)$, which is a constant function containing no information about the original image except the general level of illumination. The rest of the information carried by f is in the layers peeled off [22], i.e., the number of pulses which are considered. More precisely,

$$f = (id - P_1)(f) + ((id - P_2) \circ Q_1)(f) + ((id - P_3) \circ Q_2)(f) + \cdots ((id - P_{N-1}) \circ Q_{N-2})(f) + ((id - P_N) \circ Q_{N-1})(f) + Q_N(f). \quad (16)$$

For more information, please refer to [22,25,26].

4. Shape from focus using LULU and discrete pulse transform

In this work, we propose a new shape from focus method based on LULU filters and discrete pulse transform to determine the best frame number, i.e. the frame where the pixel is best focused. At this point, the best frame number is chosen according to the best maximum focus value for each pixel along all the frames. This is due to the characteristic of SFF which calculates the depth based on focused values.

4.1. Modified LULU focus measure (MLULU)

This algorithm is based on LULU operators when they are applied in 2D or 3D neighborhood. The effects of the size of neighborhood (window size) have been discussed in detail in [32].

The LULU operations can be performed in 2D or 3D neighborhood. 2D neighborhood implies processing every frame separately with LULU by considering the neighborhood pixels in one frame at a time. 3D neighborhood means that pixels from multiple frames are considered for processing with LULU. The disadvantage of 3D neighborhood is the increase in computational complexity. The following paragraphs provide more detail on the 2D and 3D neighborhood processing.

2D neighborhood means when we apply these operators on each frame separately, regardless of the frames before and after. This concept is illustrated in Fig. 5 and the corresponding Eqs. (8)–(13). Following are the steps of the algorithm when 2D neighborhood is considered:

(1) L and U operators are applied on each frame of the image sequence. Eqs. (10) and (11) explain the fundamental concept for L and U operators.
(2) Initially, four pixels are considered in the neighborhood as shown in first row and corresponding first and second column in Fig. 6. Therefore, the operator L_3 is applied followed by U_3. This operation is referred to as L_3U_3.
(3) Then, the window is expanded to consider nine pixels in the neighborhood as shown in third and fourth column in Fig. 6. This operation is referred to as $L_3U_3L_8U_8$, i.e. L_3U_3 is followed by L_8 and then U_8. It is necessary to mention that higher orders of LULU operators can be performed by increasing the sub-neighbors window size. However, to avoid blurring the image, we applied until $L_3U_3L_8U_8$. This operation is given by the following equation:

$$A' = \begin{bmatrix} f(i-1,j-1), f(i-1,j), f(i,j), f(i,j-1), f(i-2,j), \\ f(i-2,j-1), f(i-2,j-2), f(i-1,j-2), f(i,j-2) \end{bmatrix}. \quad (17)$$

(4) Steps 2 and 3 are repeated in all diagonal directions as shown in row 1–4 in Fig. 6. Eqs. (18)–(20) show the operations corresponding to row 2, 3 and 4 in Fig. 6.

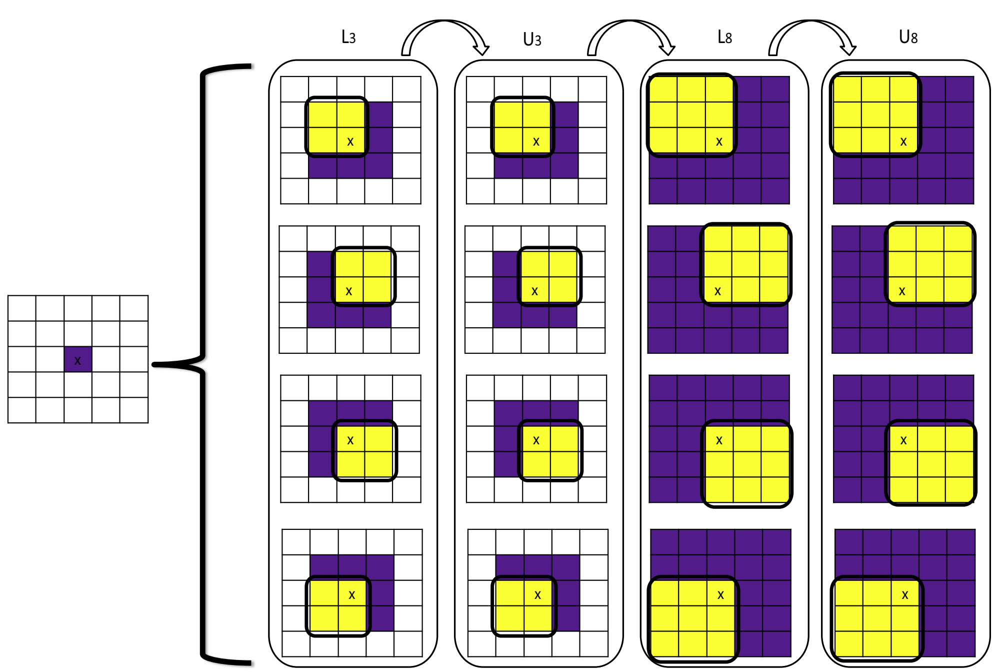

Fig. 6. 2D neighborhood for $L_3U_3L_8U_8$ (pixel x corresponds to $f(i,j)$).

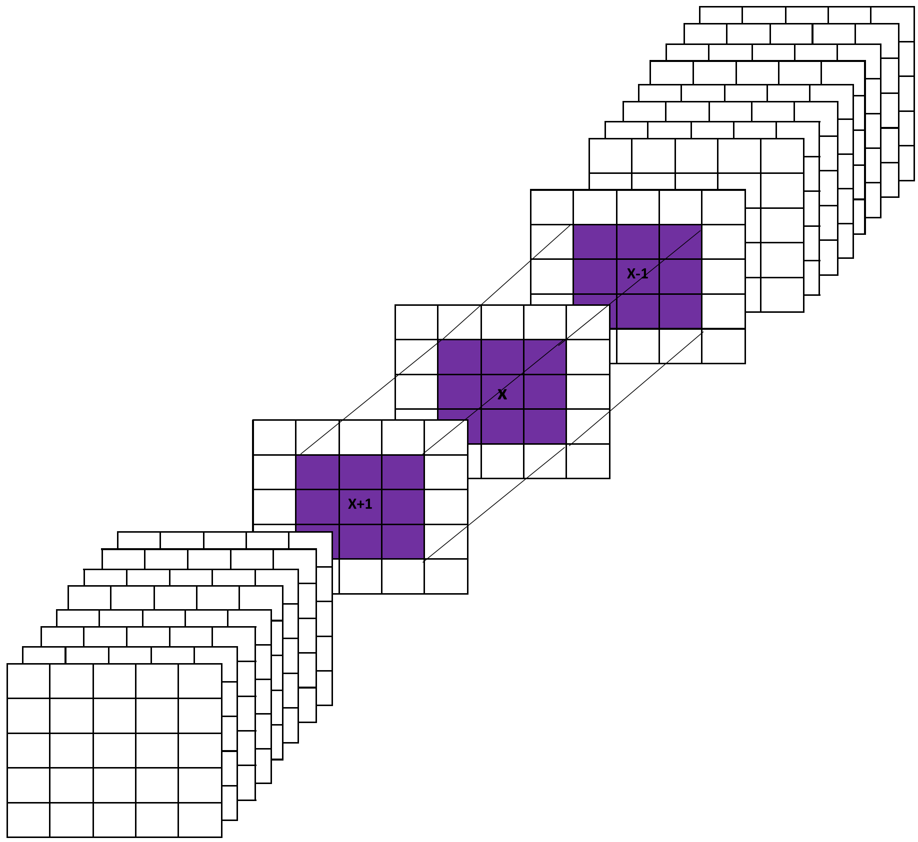

Fig. 7. 3D neighborhood of window size 3 × 3 for pixel "x" (pixel x corresponds to $f(i, j)$).

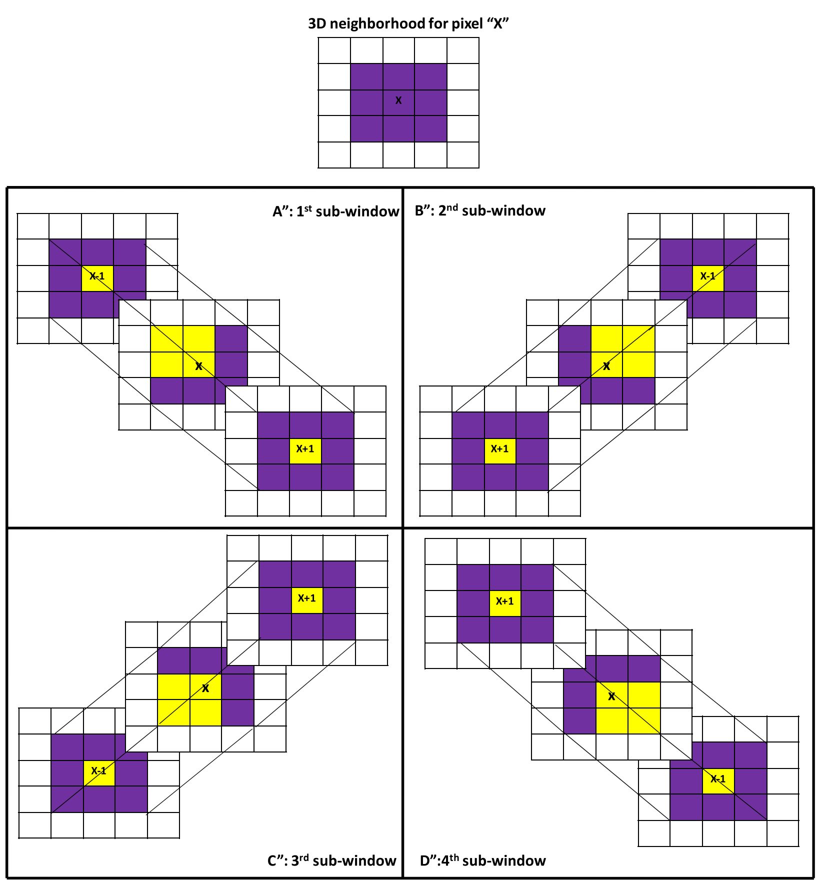

Fig. 8. 3D neighborhood for pixel "x" (pixel x corresponds to $f(i, j)$).

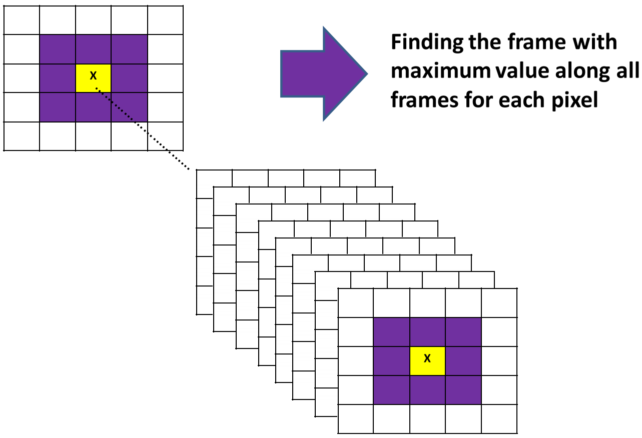

Fig. 9. Choosing the maximum value along all frames.

Table 1
DPT operators for SFF.

LULU operator	DPT Operator
L_3	$f = (id - \mathbf{L_3})(f)$
U_3	$f = (id - \mathbf{U_3})(f)$
L_3U_3	$f = (id - \mathbf{L_3U_3})(f)$
U_3L_3	$f = (id - \mathbf{U_3L_3})(f)$
$L_3U_3L_8$	$f = (id - \mathbf{L_3U_3})(f) + ((id - \mathbf{L_3U_3L_8}) \circ Q_3)(f)$
$U_3L_3U_8$	$f = (id - \mathbf{U_3L_3})(f) + ((id - \mathbf{U_3L_3U_8}) \circ Q_3)(f)$
$L_3U_3L_8U_8$	$f = (id - \mathbf{L_3U_3})(f) + ((id - \mathbf{L_3U_3L_8U_8}) \circ Q_3)(f)$
$U_3L_3U_8L_8$	$f = (id - \mathbf{U_3L_3})(f) + ((id - \mathbf{U_3L_3U_8L_8}) \circ Q_3)(f)$

$$B' = \begin{bmatrix} f(i-1,j+1), f(i-1,j), f(i,j), f(i,j+1), f(i-2,j), \\ f(i-2,j+1), f(i-2,j+2), f(i-1,j+2), f(i,j+2) \end{bmatrix}, \tag{18}$$

$$C' = \begin{bmatrix} f(i,j+1), X(i,j), f(i+1,j), f(i+1,j+1), f(i,j+2), \\ f(i+1,j+2), f(i+2,j+2), f(i+2,j+1), f(i+2,j) \end{bmatrix}, \tag{19}$$

$$D' = \begin{bmatrix} f(i,j-1), f(i,j), f(i+1,j-1), f(i+1,j), f(i+2,j), \\ f(i+2,j-1), f(i+2,j-2), f(i+1,j-2), f(i,j-2) \end{bmatrix}. \tag{20}$$

(5) Subsequently, the L and U formulas are given as:

$$L(i,j) = \max(\min(A'), \min(B'), \min(C'), \min(D')), \tag{21}$$
$$U(i,j) = \min(\max(A'), \max(B'), \max(C'), \max(D')). \tag{22}$$

(6) Finally, the modified LULU (MLULU) focus measure is given as:

$$MLULU = \max[L(i,j), U(i,j)]. \tag{23}$$

For **3D neighborhood**, the neighborhood's window around each pixel is different. It means that LULU value for each pixel not only depends on its neighbors in the same frame but also neighbors of one frame before and one frame after it. Therefore the neighborhood is a 3D one as it is shown in Fig. 7.

With the new 3D neighborhood defined around a pixel "x" (pixel x corresponds to $f(i, j)$) as shown in Fig. 7, the sub-windows are defined as shown in Fig. 8. It can be seen from Fig. 8 that step 1 and 2 remain same as described earlier. However, additionally the two pixels corresponding to $f(i, j)$ in the two consecutive frames (i.e., one frame before and one frame after it) are also considered. Therefore, it's a six pixel neighborhood that includes four pixels from the same frame while one pixel each from the adjoining consecutive frames.

Based on Fig. 8, the sub-window equations for six pixels neighborhood are as follow:

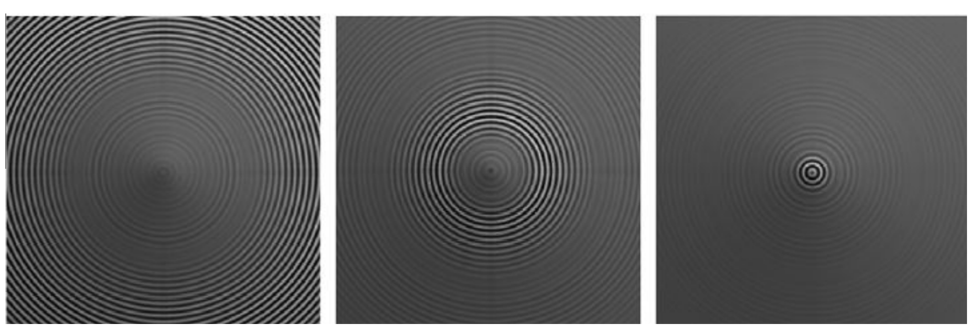

Fig. 10. Test object: simulated cone.

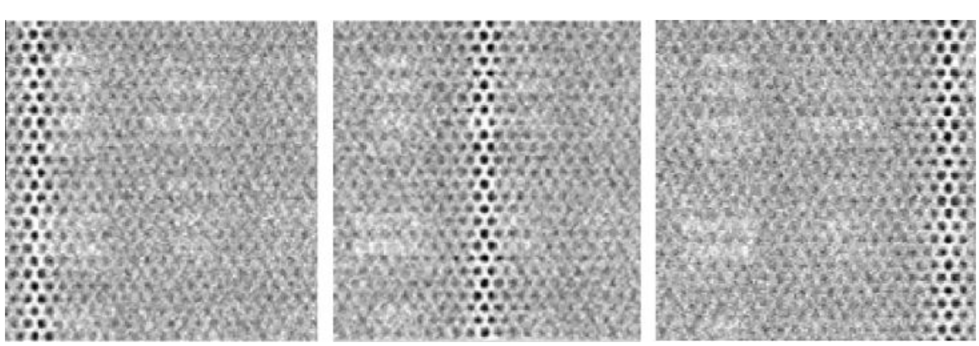

Fig. 11. Test object: simulated slope.

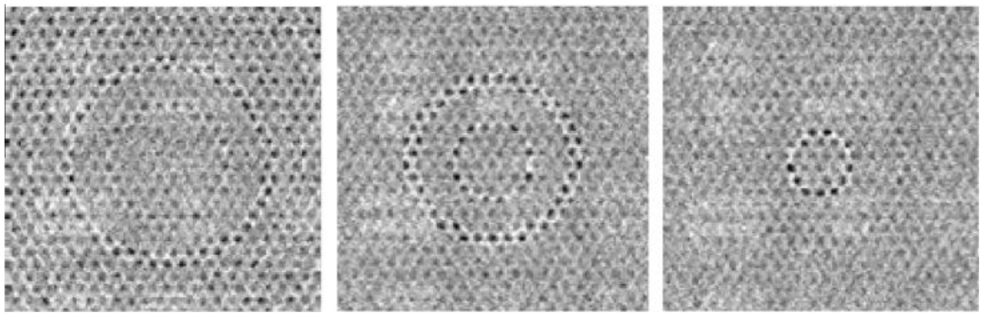

Fig. 12. Test object: simulated cosine.

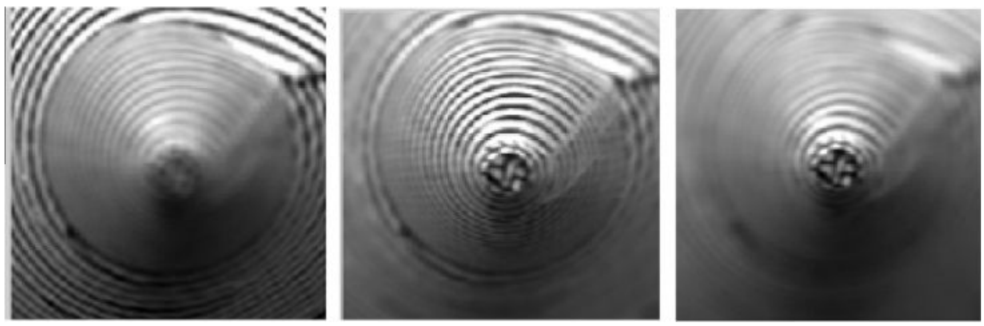

Fig. 13. Test object: real cone.

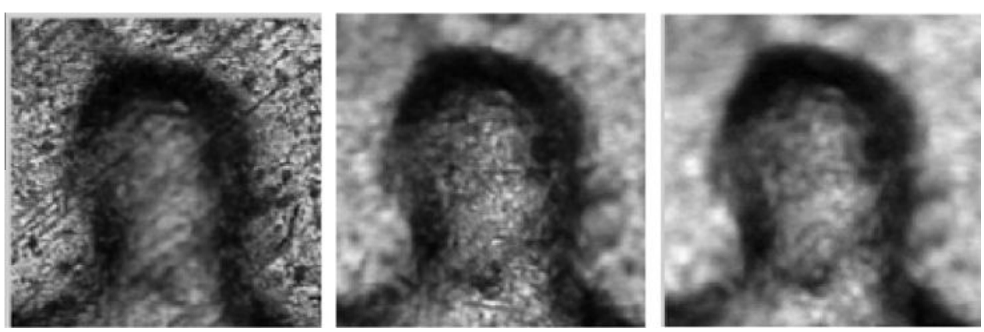

Fig. 14. Test object: real coin.

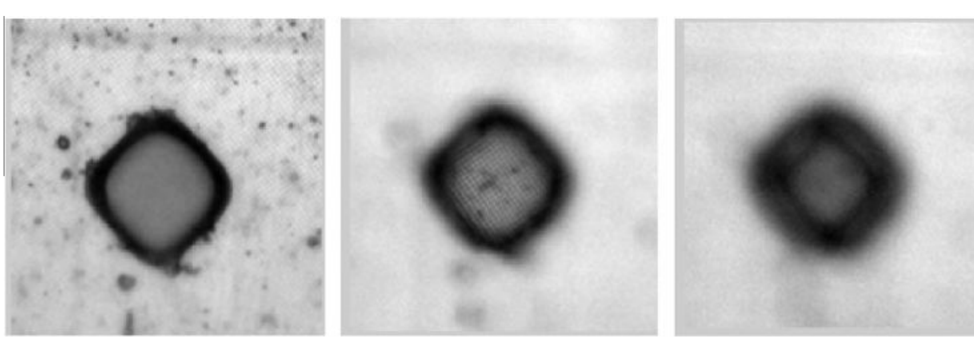

Fig. 15. Test object: real LCD.

$$A'' = [f(i-1,j-1), f(i-1,j), f(i,j), f(i,j-1), f_{-1}(i,j), f_{+1}(i,j)], \quad (24)$$
$$B'' = [f(i-1,j+1), f(i-1,j), f(i,j), f(i,j+1), f_{-1}(i,j), f_{+1}(i,j)], \quad (25)$$
$$C'' = [f(i,j+1), f(i,j), f(i+1,j), f(i+1,j+1), f_{-1}(i,j), f_{+1}(i,j)], \quad (26)$$
$$D'' = [f(i,j-1), f(i,j), f(i+1,j-1), f(i+1,j), f_{-1}(i,j), f_{+1}(i,j)]. \quad (27)$$

The formulas for L and U are the same as Eqs. (21) and (22). Instead of A, B, C and D, we substitute A'', B'', C'' and D''. In Eqs. (8)–(11), (17)–(20) and (24)–(27), it is obvious that the LULU operators detect the peaks and valleys in each sub-window. These operations illustrate that when we apply L or U based on Eqs. (21) and (22), they

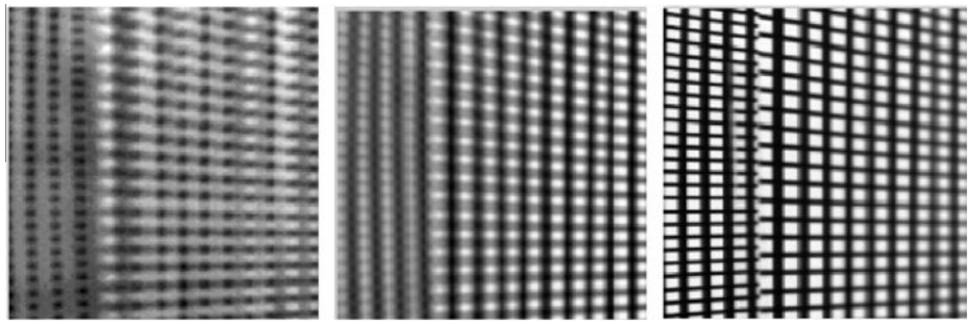

Fig. 16. Test object: real plane.

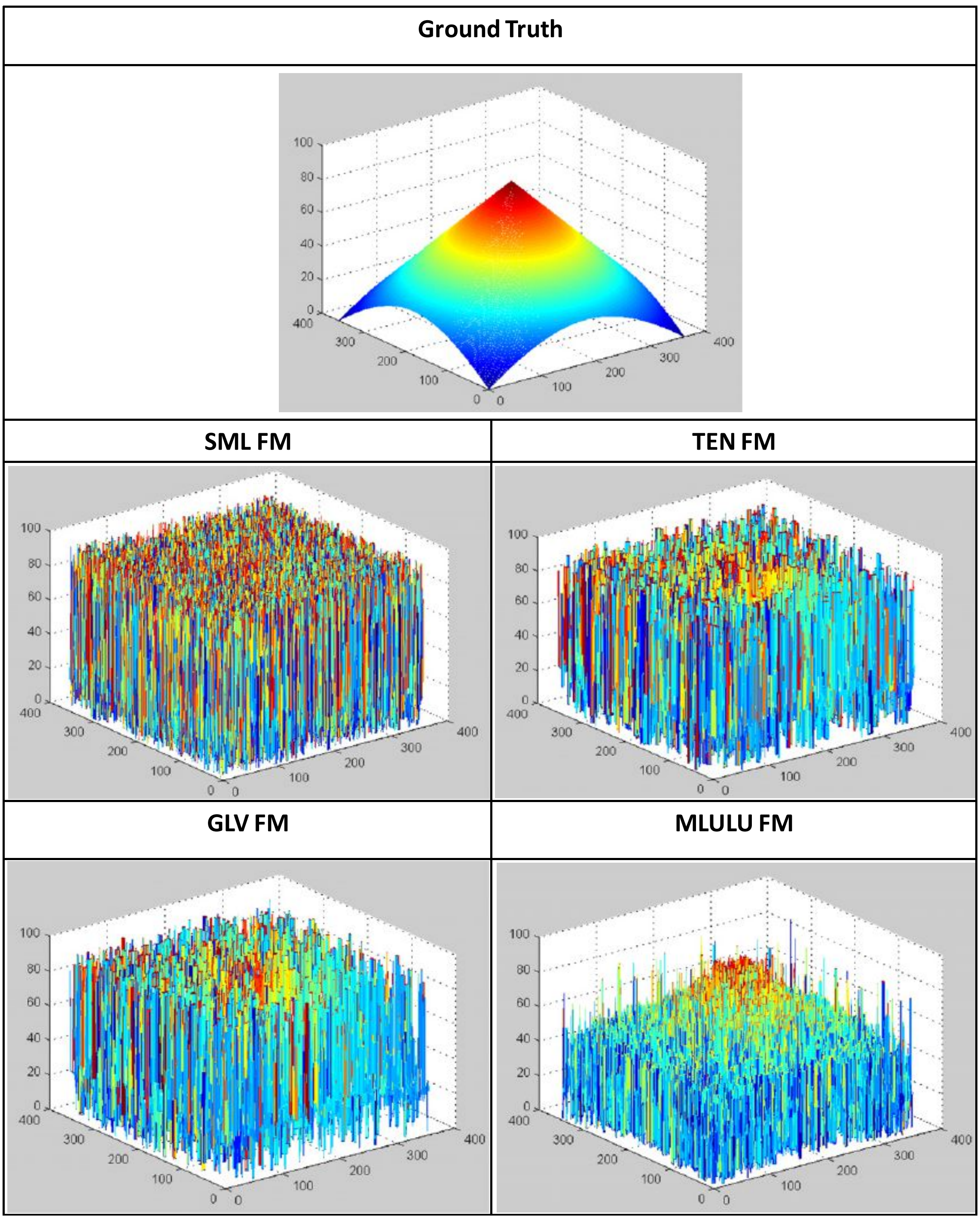

Fig. 17. Simulated cone in the presence of impulse noise (ND = 0.5).

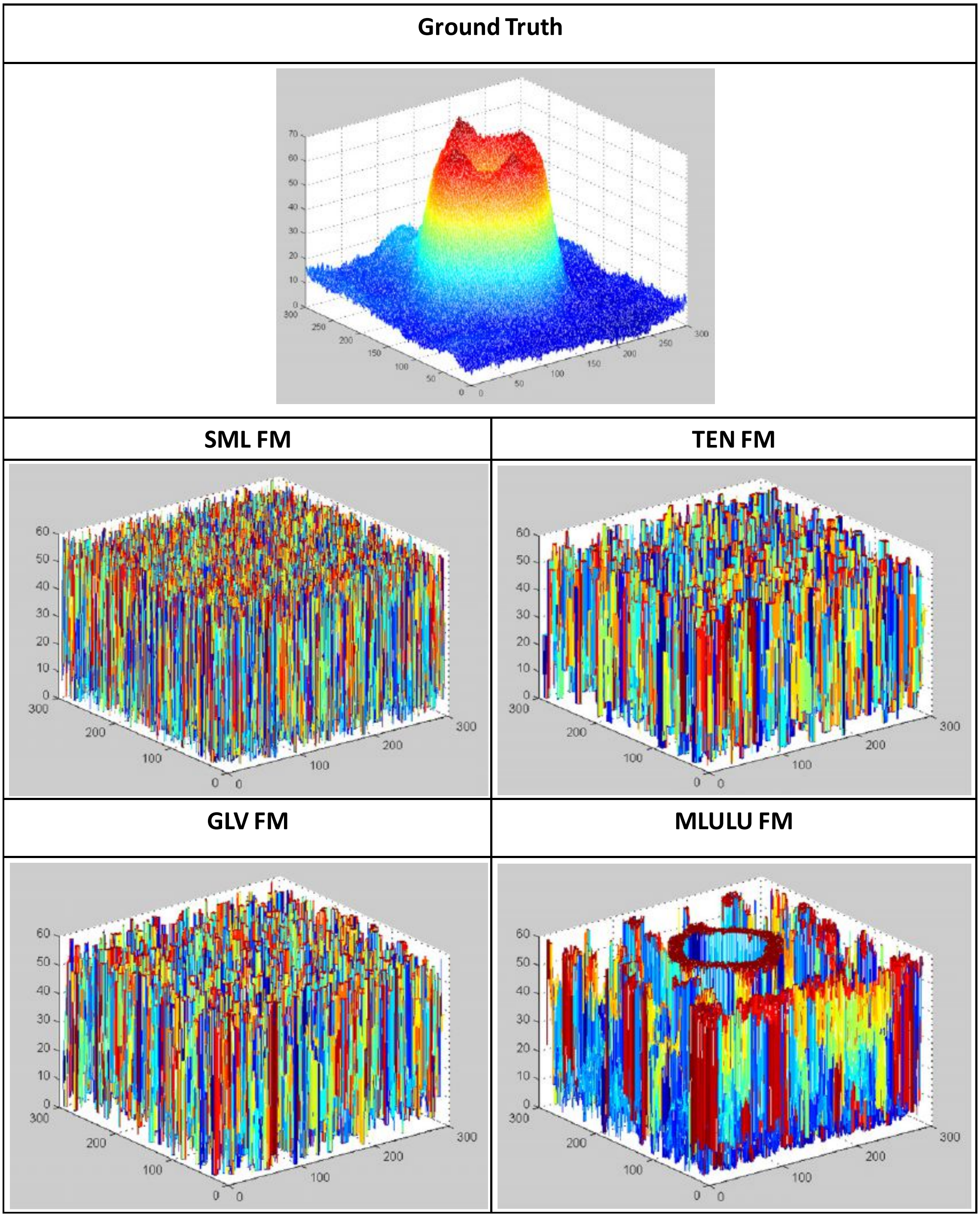

Fig. 18. Real LCD in the presence of impulse noise (ND = 0.05).

minimize noise in each region by eliminating the very high or very low intensities and ensure a smooth focus measure in the presence of noise.

Smoothing characteristic of LULU operators plays a very important role in focus measurement. For images corrupted with noise, the noise value is wrongly interpreted as focused value. However, the focus value should be at least similar to few neighboring pixels, because in each frame, the focusing part is not a point. Rather it refers to a small part, tiny group of pixels, near each other. Based on the concept of focusing, it is obvious that high frequencies may be chosen as the focused values which are in fact the noise values.

After applying LULU operators, we substitute each pixel's intensity by its LULU value. For reconstructing the 3D shape, we select the maximum value for each pixel along all frames as shown in Fig. 9.

Fig. 9, illustrates the sequence of frames for calculating the focused pixel among all the frames for pixel (i,j). As a result, the output of SFF is two 2D matrices; maximum intensity and corresponding frame index. Maximum intensity holds the best focus value for each one of the pixels, and corresponding frame index holds the resultant frame number where the pixel is best focused [2]. Let MLULU be the modified LULU focus measure value for each pixel

Table 2
Proposed methods performance in the presence of impulse noise for simulated cone object.

Noise density	Focus measure (FM)	RMSE	Correlation	PSNR
0.5	**MLULU**	15.69	0.59	24.21
	MDPT	28.74	0.10	18.99
	SML	32.01	0.04	18.02
	GLV	22.37	0.37	21.13
	EN	25.24	0.22	20.09
0.05	MLULU	12.12	0.73	26.46
	MDPT	17.09	0.58	23.47
	SML	29.34	0.19	19.49
	GLV	14.8	0.59	24.72
	TEN	12.13	0.72	26.46
0.005	MLULU	15.01	0.65	24.55
	MDPT	17.32	0.57	23.47
	SML	27.04	0.19	19.49
	GLV	9.71	0.85	28.38
	TEN	8.55	0.92	29.49

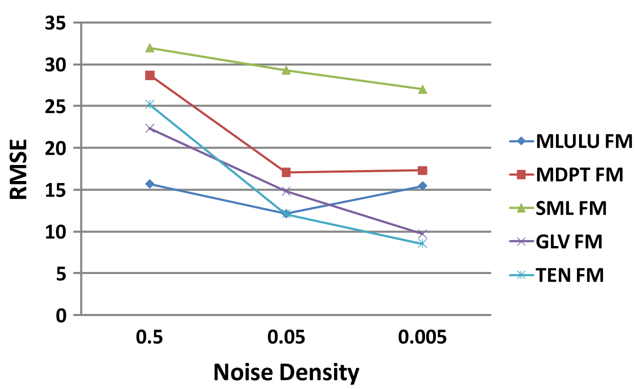

Fig. 19. RMSE comparison between different methods for simulated cone object in the presence of impulse noise.

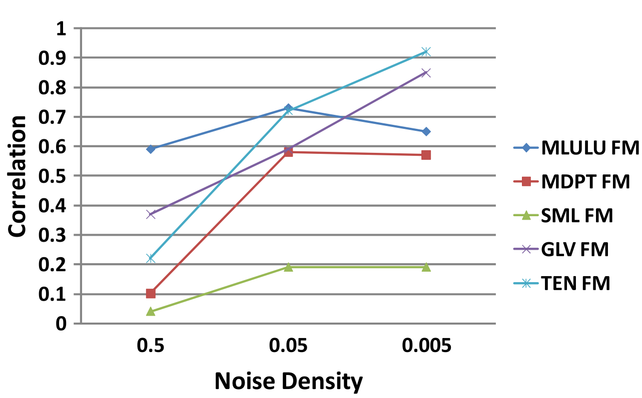

Fig. 20. Correlation comparison between different methods for simulated cone object in the presence of impulse noise.

(i,j), k be the image frame number, D_p be a 2D matrix containing the best focused value (sharpness/maximum value) for each of the pixels and D_f be a 2D matrix containing the corresponding frame numbers where the pixels are best focused, then the equation for depth map calculation using SFF method is given as

$$[D_p(i,j), D_f(i,j)] = \max[{}^{k}MLULU\ (i,j)]. \quad (28)$$

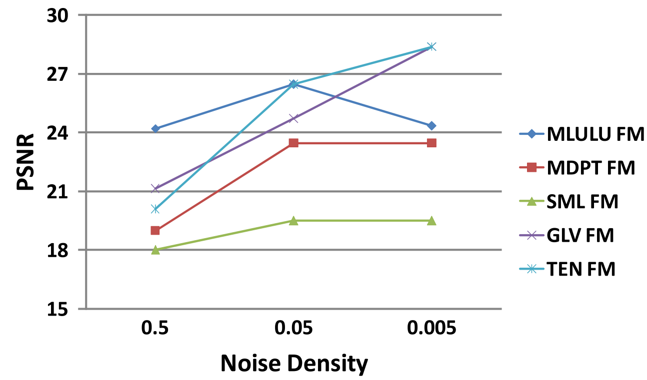

Fig. 21. PSNR comparison between different methods for simulated cone object in the presence of impulse noise.

4.2. Modified discrete pulse transform focus measure (MDPT)

In this proposed focus measure (FM), we apply DPT on each frame based on Eq. (14). Since here we apply LULU and consequently DPT only for L_3, U_3, L_3U_3, U_3L_3, $L_3U_3L_8U_3L_3U_8$, $L_3U_3L_8U_8$, and $U_3L_3U_8L_8$, therefore the DPT operators are also limited as it is shown in Table 1.

where f refers to DPT values, id is the original pixel value and Q_3 refers to the DPT value of the third order operators, which are $(id - L_3)(f)$, $(id - U_3)(f)$, $(id - L_3U_3)(f)$ or $(id - U_3L_3)(f)$ in this work.

The main concept behind this algorithm is to reconstruct the 3D shape based on the very high frequencies. DPT is a very strong technique for object detection. DPT decomposes the image into many pulses and each object in the image can have a specific number of pulses. For detecting the object, we shall find out the range of pulses and eliminate other pulses from the image. We use this concept of pulses in SFF. The focused parts of the images can be selected by choosing the correct pulses.

In this work, we considered the first two pulses in images which are D_3 and D_8 to detect the focused values. We chose the most extreme pulses which may represent the focused values.

5. Results and discussion

The proposed techniques are implemented on simulated and actual SFF data separately and in combination with other focus measure methods including Sum of Modified Laplacian (SML), Tenenbaum and gray level variance (GLV). These methods have been selected because they are the most commonly cited focus measure methods. SML was used by references [2,16,11,27,28], GLV by references [16,29,30] and Tenenbaum by [29,31]. All of these experiments are repeated in the presence of impulse, Gaussian and speckle noises for 3D shape recovery. The reconstructed depth map has been compared with the original data by using three different image quality metrics, which are; peak signal to noise ratio (PSNR), root mean square error (RMSE) and correlation. Based on the quantitative and qualitative experimental results, the proposed techniques are more accurate in focused value extraction and shape recovery in the presence of noise.

The proposed techniques have been tested on seven different objects. The results are shown and discussed later in this section.

5.1. Test Images

Texture of objects play an important role in the identification of best focused pixels. Hence, different objects with different textures have been chosen to be studied in this work. Simulation has been performed using three different quality measures; RMSE, correla-

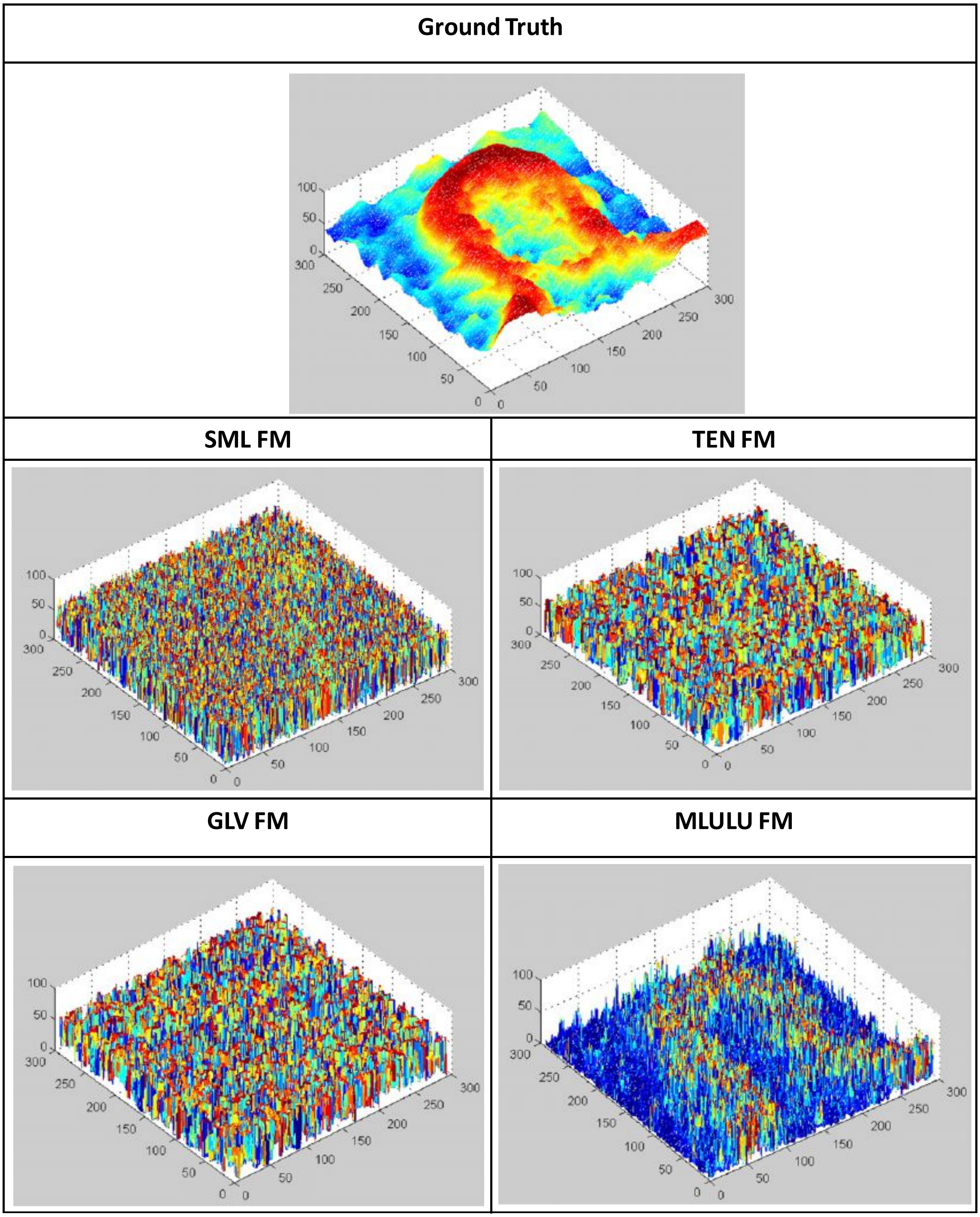

Fig. 22. Coin in the presence of Gaussian noise (variance = 0.5).

tion and PSNR to compare proposed methods with SML, GLV and TEN for different types of noise. For the purpose of comparison seven test sequences are used, including both the simulated and real objects, i.e. simulated cone, simulated slope, simulated cosine, real cone, real coin, real LCD and real plane, as shown in Figs. 10–17. In total, seven objects are evaluated; three simulated objects and four real objects.

The test objects are chosen from different textures with different level of details. Coin, sine and cosine carry good amount of details. These high textured images are good SFF images which help to test the outcomes of focusing. Slope and plane have poor uniformed texture. Cone is a dense textured object and it is considered to have medium level of details, LCD image has low level of details and variance and it is a microscopic image. The resolution of images is $360 \times 360 \times 97$ for simulated and real data cone, $320 \times 320 \times 60$ for slope and cosine objects, $300 \times 300 \times 68$ for Coin, $300 \times 300 \times 60$ for LCD and $200 \times 200 \times 87$ for plane.

Simulated cone: In this case the sequence is constructed from 97 different images with different focus values, with the resolution of 360×360. Fig. 10, illustrates the simulated data of the cone. It has a dense texture [2].

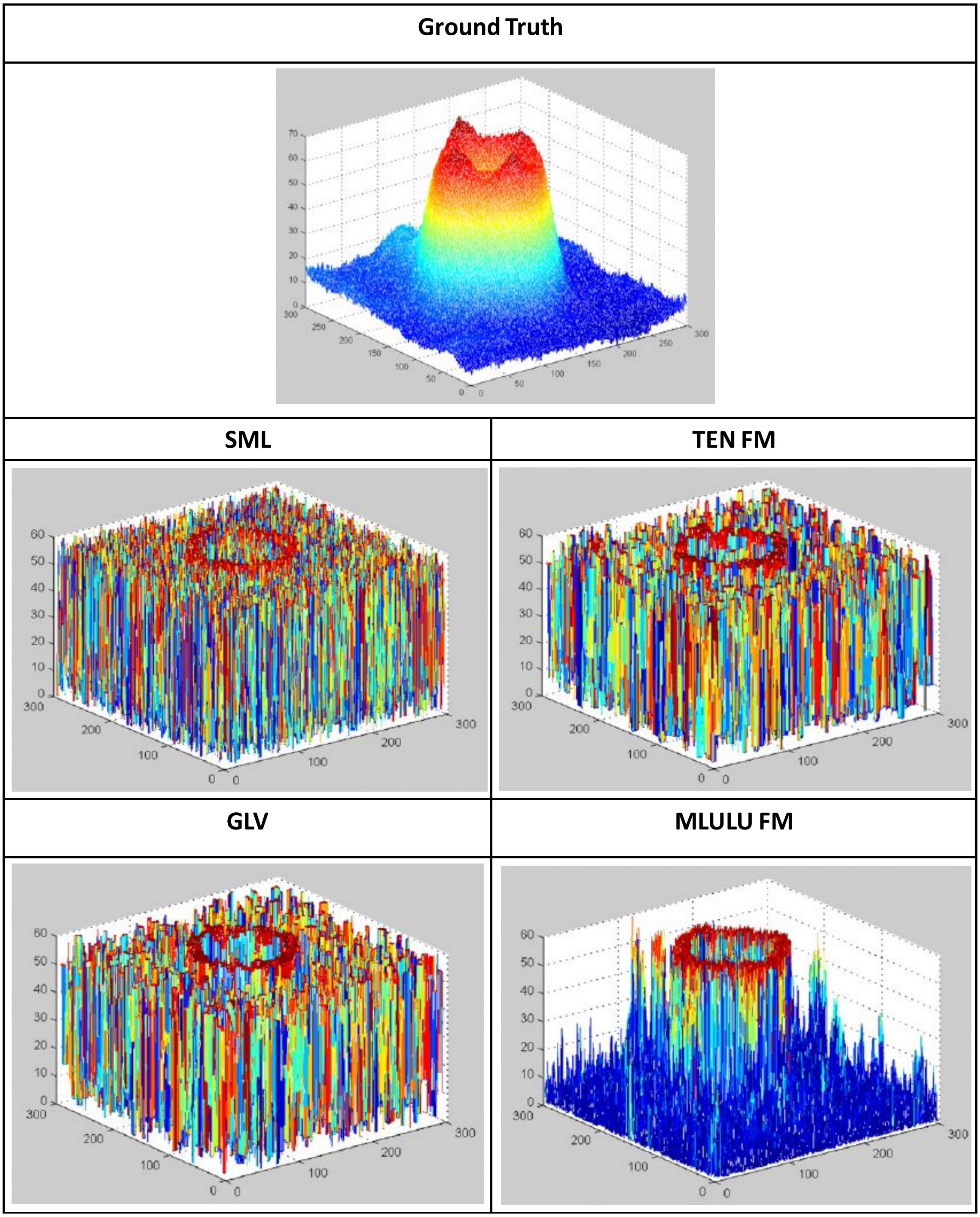

Fig. 23. LCD in the presence of speckle noise (ND = 0.5).

Simulated slope: This data consists of 60 frames with the resolution 320×320. Some of the frames are shown in Fig. 11.

Simulated cosine: Similar to slope and sine simulated data, simulated cosine also consists of 60 frames with the resolution of 320×320. Some of the frames are shown in Fig. 12.

Real cone: Real cone object is the real data of the real cone. The resolution is 360×360 and the number of images in the sequence is 97. Some frames are shown in Fig. 13.

Real coin: This data has been collected from a microscopic object with 68 frames of 300×300 pixels. Fig. 14 shows different focused frames of this object. This object is the head of Lincoln on a one penny coin which is a good sample of a rough texture [2].

Real LCD: LCD is another microscopic object which is a sequence of 60 frames of real data. The resolution of the image is 300×300. Fig. 15 shows three of the frames for TFT-LCD.

Real plane: Real plane is the real data collected from a plane. Its SFF data consists of 87 frames at resolution of 200×200 pixels. This object is a good example of poor texture and some of its different focused frames are illustrated in Fig. 16.

5.2. Experimental results

The seven different test objects are considered in the presence of impulse noise, Gaussian noise and speckle noise. Each noise is evaluated with three different noise densities/variance, which are; 0.005, 0.05 and 0.5. The results of the proposed methods are compared with SML, GLV and TEN techniques. These techniques were explained in Section 2.

In general, all the results obtained for each object are compared qualitatively and quantitatively with SML, the GLV and TEN. This comparison is done via three different image quality metrics which are; PSNR, RMSE and correlation.

5.2.1. Proposed methods

This section shows the result for the proposed methods which are described in Section 4 and comparison is provided both qualitatively and quantitatively with SML, GLV and TEN. The 3D recovered shapes for the seven objects in the presence of various types of noise are shown in this section.

Three noise levels are used for experiments, i.e., high (noise density/variance = 0.5), medium (noise density/variance = 0.05) and low (noise density/variance = 0.005). Fig. 17 illustrates the performance of SML, TEN, GLV, MDPT and MLULU Focus Measures (FM's) for simulated cone object in the presence of impulse noise with the noise density (ND) of 0.5.

In Fig. 18, the performance of MLULU is compared with other three methods in the presence of impulse noise with noise density of 0.05 for real LCD.

In general, proposed focus measure performs well in the presence of impulse noise as is evident from Figs. 17 and 18. MLULU FM removes the locally occurring hills and valleys of signals and images. It is clear that in the presence of impulse noise, the other focus measures (FMs) are not performing well and the 3D shape reconstructed based on them is not clear at all. Their result is a set of noisy data which does not show anything similar to the object. This is true for high noise density (0.5) as well as medium noise density (0.05) levels. But the 3D shape reconstructed based on MLULU FM is clear and shows the shape at all noise levels. The quantitative result for simulated cone is provided in Table 2 and Figs. 19–21.

It is clear in Fig. 19 that MLULU is performing better than SML and MDPT in general and at high and medium impulse noise levels, its performance is better than all focus measures. This is also evident from Figs. 20 and 21.

In Figs. 22 and 23, we show the comparison between MLULU, SML, GLV and TEN for Gaussian and Speckle noises. We conclude that MLULU is not only a good focus measure in the presence of impulse noise, but its performance is comparable in the presence of other types of noise i.e., speckle and Gaussian. In Fig. 22, MLULU method is performing better than other methods in the presence of Gaussian noise with variance (V) = 0.5.

From Fig. 23, it is obvious that MLULU performance is comparable with other focus measures in the presence of Speckle noise.

6. Conclusion

In this paper, new focus measures are proposed and tested for 3D shape recovery. The proposed techniques are implemented on seven simulated and real objects. They are also cascaded with existing focus measure methods, i.e., sum of modified Laplacian (SML), Tenenbaum and gray level variance (GLV). The experiments are repeated in the presence of impulse, Gaussian and speckle noise for 3D shape recovery. The reconstructed depth map has been compared with the ground truth by using three different image quality metrics, which are; RMSE, PSNR and correlation. Based on the quantitative and qualitative experimental result, the proposed techniques are more accurate in focused value extraction and shape recovery especially in the presence of various types of noise. MLULU focus measure performs better than existing methods when the SFF data is noisy. The combined performance of MLULU and existing methods, show a good improvement in shape recovery.

References

[1] Aamir Saeed Malik, Tae-Sun Choi, Humaira Nisar, Depth Map and 3D Imaging Applications: Algorithms and Technologies IGI Global, 2011.
[2] A.S. Malik, T. Sun Choi, A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise, Pattern Recognition 41 (7) (2008) 2200–2225.
[3] B.K.P. Horn, Focussing, M.I.T., Project MAC, AI Memo, 1968.
[4] J.M. Tenebaum, Accommodation in Computer Vision, Ph.D. Thesis, Stanford University, 1970.
[5] R.A. Muller, A. Buffington, Real time correlation of atmospherically degraded telescope images through image sharpening.pdf, Journal of the Optical Society of America 64 (9) (1974) 1200–1210.
[6] A. Erteza, Sharpness index and its application to focus control, Applied Optics 15 (4) (1976) 877–881.
[7] R.A. Jarvis, Focus optimisation criteria for computer image processing, Microscope 24 (2) (1976) 163–180.
[8] A.P. Pentland, R. Ave, M. Park, A New Sense for Depth of Field, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9 (4) (1985) 988–994.
[9] P. Grossmann, Depth from focus, Pattern Recognition Letters 5 (4) (1987) 63–69.
[10] T. Darrell, K. Wohn, Pyramid based depth from focus, in: Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR '88, 1988, pp. 504–509.
[11] S.K. Nayar, Y. Nakagawa, Shape from focus: an effective approach for rough surfaces, in: IEEE International Conference on Robotics and Automation, 1990. Proceedings, 1990, pp. 218–225.
[12] S.K. Nayar, Shape from focus, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (8) (1994) 824–883.
[13] C. Dillon, T. Caelli, Generating complete depth maps in passive vision systems, in: 11th IAPR International Conference on Pattern Recognition, vol. I. Conference A: Computer Vision and Applications, Proceedings, 1992, pp. 562–566.
[14] N. Asada, H. Fujiwara, T. Matsuyama, Edge and depth from focus, International Journal of Computer Vision 26 (2) (1998) 153–163.
[15] Y. Zhang, Y. Zhang, C. Wen, A new focus measure method using moments, Image and Vision Computing 18 (12) (2000) 959–965.
[16] F.S. Helmli, S. Scherer, Adaptive shape from focus with an error estimation in light microscopy, in: Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, ISPA 2001, 2001, pp. 188–193.
[17] Aamir Saeed Malik, Humaira Nisar, Tae-Sun Choi, A fuzzy-neural approach for estimation of depth map using focus, Applied Soft Computing 11 (3) (2011) 1837–1850.
[18] W.J. Conradie, T.D. Wet, M. Jankowitz, Exact and asymptotic distributions of LULU smoothers, Journal of Computational and Applied Mathematics 186 (1) (2006) 253–267.
[19] C.H. Rohwer, D.P. Laurie, The discrete pulse transform, SIAM Journal on Mathematical Analysis 38 (3) (2005) 1–28.
[20] J. Pierre du Toit, The Discrete Pulse Transform and Applications, Masters Thesis, University of Stellenbosch, 2007.
[21] R. Anguelov, I. Fabris-rotelli, LULU operators and discrete pulse transform for multidimensional arrays, IEEE Transaction on Image Processing 19 (11) (2010) 3012–3023.
[22] B. Anguelov, Discrete Pulse Transform of Images: Algorithm and Applications, in: 19th International Conference on Pattern Recognition (ICPR), 2008, pp. 1–4.
[23] A. Morales, R. Acharya, S.-J. Ko, Morphological pyramids with alternating sequential filters, IEEE Transaction on Image Processing 4 (7) (1995) 965–977.
[24] S.-C. Pei, C.-L. Lai, An efficient class of alternating sequential filters in morphology, Graphical Models and Image Processing 59 (2) (1997) 109–116.
[25] I. Fabris-Rotelli, S.J.V.D. Walt, The discrete pulse transform in two dimensions, in: Proceedings of the Twentieth Annual Symposium of the Pattern Recognition Association of South Africa, 2009.
[26] T. de Wet, W. Conradie, Smoothing sequences of data by extreme selectors, in: Proceedings of ICOTS7 2006: The 7th International Conference on Teaching Statistics, Salvador, Bahia, Brazil, 2–7 July, 2006.
[27] J. Yun, T.S. Choi, Accurate 3-D shape recovery using curved window focus measure, in: International Conference on Image Processing (ICIP), 1999, pp. 910–914.
[28] M.B. Ahmad, T. Sun Choi, A heuristic approach for finding best focused shape, IEEE Transactions on Circuits and Systems 15 (4) (2005) 566–574.
[29] M. Subbarao, T. Sun Choi, A. Nikzad, Focusing Techniques, Technical Report 92.09.04, Department of Electrical Engineering, State University of New York at Stony Brook, 1992.

[30] M. Asif, T. Sun Choi, Shape from focus using multilayer feedforward neural networks, IEEE Transactions on Image Processing 10 (11) (2001) 1670–1675.
[31] Y. Xiong, S.A. Shafer, Depth from focusing and defocusing, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings CVPR '93, 1993, pp. 68–73.
[32] Aamir Saeed Malik, Tae-Sun Choi, Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery, Elsevier Pattern Recognition 40 (1) (2007) 154–170.