

Article

# Sampling Based on Kalman Filter for Shape from Focus in the Presence of Noise

Hoon-Seok Jang <sup>1,†</sup> , Mannan Saeed Muhammad <sup>2,†</sup> , Guhnoo Yun <sup>3</sup>   
and Dong Hwan Kim <sup>3,\*</sup> <sup>1</sup> Center for Imaging Media Research, Korea Institute of Science and Technology, Seoul 02792, Korea<sup>2</sup> College of Information and Communication, Natural Sciences Campus, Sungkyunkwan University, Suwon 16419, Korea<sup>3</sup> Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology, Seoul 02792, Korea

\* Correspondence: gregorykim@kist.re.kr; Tel.: +82-2-958-6719

† These authors contributed equally as first authors to this work.

Received: 10 July 2019; Accepted: 7 August 2019; Published: 9 August 2019



**Abstract:** Recovering three-dimensional (3D) shape of an object from two-dimensional (2D) information is one of the major domains of computer vision applications. Shape from Focus (SFF) is a passive optical technique that reconstructs 3D shape of an object using 2D images with different focus settings. When a 2D image sequence is obtained with constant step size in SFF, mechanical vibrations, referred as jitter noise, occur in each step. Since the jitter noise changes the focus values of 2D images, it causes erroneous recovery of 3D shape. In this paper, a new filtering method for estimating optimal image positions is proposed. First, jitter noise is modeled as Gaussian or speckle function, secondly, the focus curves acquired by one of the focus measure operators are modeled as a quadratic function for application of the filter. Finally, Kalman filter as the proposed method is designed and applied for removing jitter noise. The proposed method is experimented by using image sequences of synthetic and real objects. The performance is evaluated through various metrics to show the effectiveness of the proposed method in terms of reconstruction accuracy and computational complexity. Root Mean Square Error (RMSE), correlation, Peak Signal-to-Noise Ratio (PSNR), and computational time of the proposed method are improved on average by about 48%, 11%, 15%, and 5691%, respectively, compared with conventional filtering methods.

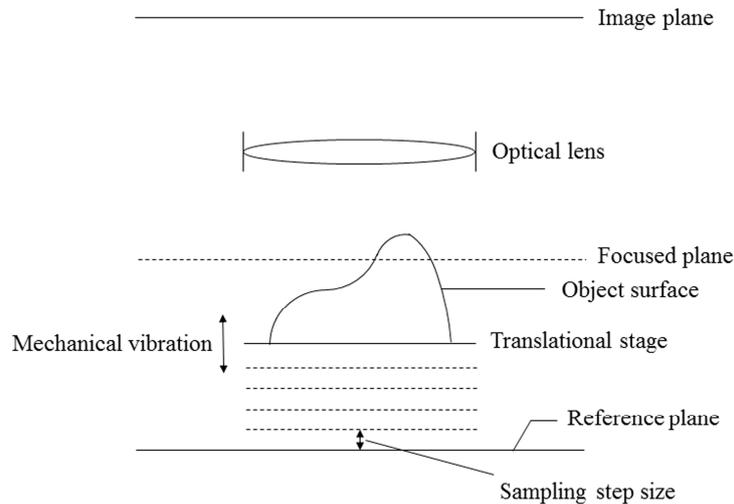
**Keywords:** shape from focus (SFF); jitter noise; focus curve; Kalman filter

## 1. Introduction

Inferring three-dimensional (3D) shape of an object from two-dimensional (2D) images is a fundamental problem in computer vision applications. Many 3D shape recovery techniques have been proposed in literature [1–5]. The methods can be categorized into two categories based on the optical reflective model. The first one includes active techniques which use projected light rays. The second category consists of passive techniques which utilize reflected light rays without projection. The passive methods can further be classified into Shape from X, where X denotes the cue used to reconstruct the 3D shape as *Stereo* [6], *Texture* [7], *Motion* [8], *Defocus* [9], and *Focus* [10]. Shape from Focus (SFF) is a passive optical method that utilizes a series of 2D images with different focus levels for estimating 3D information of an object [11]. For SFF, a focus measure is applied to each pixel of the image sequence, to evaluate the focus quantity at every point. The best focused position is acquired by maximizing the focus measure values along the optical axis.

Many focus measures have been reported in literature [12–16]. Initial depth map, obtained through any of the focus measure operators, has the problem of information loss between consecutive frames

due to the discreteness of predetermined sampling step size. To solve this problem, refined depth map is acquired using approximation techniques, as reported in literature [17–22]. As an important issue of SFF, when images are obtained by translating the object plane with constant step size, mechanical vibration, referred to as jitter noise, occurs in each step, as shown in Figure 1 [21].



**Figure 1.** Image acquisition for Shape from Focus.

Since this noise changes the focus values of images by oscillating along the optical axis, accuracy of 3D shape recovery is considerably degraded. Unlike any image noise [23,24], this noise is not detectable by simply observing images.

Many filtering methods for removing the jitter noise have been reported [25–27]. In [25,26], Kalman and Bayes filtering methods for removing Gaussian jitter noise have been proposed, respectively. In [27], a modified Kalman filtering method for removing Lévy noise has been presented.

In this paper, a new filtering method for removing the jitter noise is proposed as an extended version of [25]. First, jitter noise is modeled as Gaussian or speckle function to reflect more types of noise that can occur in SFF. At the second stage, the focus curves acquired by one of the focus measure operators are modeled as Gaussian function for application of the filter and a clearer performance comparison of various filters. Finally, Kalman filter as the proposed method is designed and applied. Kalman filter is a recursive filter that tracks the state of a linear dynamic system containing noise, and is used in many fields such as computer vision, robotics, radar, etc. [28–33]. In many cases, this algorithm is based on measurements made over time. More precise results can be expected than from using only measurements at that moment. As the filter recursively processes input data, including noise, optimal statistical prediction for the current state can be performed. The proposed method is experimented by using image sequences of synthetic and real objects. The performance of the proposed method is analyzed through various metrics to show its effectiveness in terms of reconstruction accuracy and computational complexity. Root Mean Square Error (RMSE), correlation, Peak Signal-to-Noise Ratio (PSNR), and computational time of the proposed method are improved by an average of about 48%, 11%, 15%, and 5691%, respectively, compared with conventional filtering methods. In the remainder of this paper, Section 2 presents the concept of SFF and a summary of previously proposed focus measures as background. Sections 3 and 4 provide the modeling of jitter noise and focus curves, respectively. Section 5 explains the Kalman filter as the proposed method in detail. Experimental results and discussion are presented in Section 6. Finally, Section 7 concludes this paper.

## 2. Related Work

### 2.1. Shape from Focus

In SFF methods, images with different focus levels (such that some parts are well focused and the rest of the parts are defocused with some blur) are obtained by translating the object plane at a predetermined step size along the optical axis [11]. By applying a focus measure, the best focused frame for each object point is acquired to find the depth of the object with an unknown surface. The distance of the corresponding object point is computed by using the camera parameters for the frame, and utilizing the lens formula as follows:

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

where,  $f$  is the focal length,  $u$  and  $v$  are the distances of object and image from the lens, respectively. Figure 2 shows the image formation in the optical lens. The object point at the distance  $u$  is focused to the image point at the distance  $v$ .

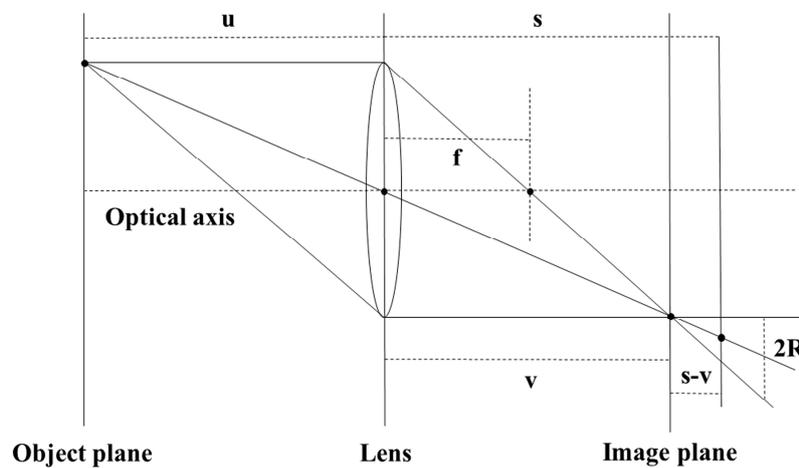


Figure 2. Image formation in optical lens.

### 2.2. Focus Measures

A focus measure operator calculates the focus quality of each pixel in the image sequence, and is evaluated locally. As the image sharpness increases, the value of the focus measure increases. When the image sharpness is maximum, the best focused image is attained. Some of the popular gradient-based, statistical-based, and Laplacian-based operators are briefly given in [12].

First, there are Modified Laplacian (ML) and Sum of Modified Laplacian (SML) as Laplacian-based operators. When Laplacian is used in textured images,  $x$  and  $y$  components of the Laplacian operator may cancel out and provide no response. ML is calculated by adding the squared second derivatives for each pixel of the image  $I$  as:

$$F_{ML}(x, y) = \left( \frac{\partial^2 I(x, y)}{\partial x^2} \right)^2 + \left( \frac{\partial^2 I(x, y)}{\partial y^2} \right)^2 \tag{2}$$

If the image has rich textures with high variability at each pixel, focus measure can be evaluated for each pixel. In order to improve robustness for weak-textured images, SML is computed by adding the ML values in a  $W \times W$  window as:

$$F_{SML}(i, j) = \sum_{x \in W} \sum_{y \in W} \left\{ \left( \frac{\partial^2 I(x, y)}{\partial x^2} \right)^2 + \left( \frac{\partial^2 I(x, y)}{\partial y^2} \right)^2 \right\} \tag{3}$$

where,  $i$  and  $j$  are the  $x$  and  $y$  coordinates of center pixel in a  $W \times W$  window, respectively.

Next, there is Tenenbaum (TEN) as a gradient-based operator. TEN is calculated by adding the squared responses of horizontal and vertical Sobel operators. For robustness, it is also computed by adding the TEN values in a  $W \times W$  window as:

$$F_{TEN}(i, j) = \sum_{x \in W} \sum_{y \in W} \left\{ (G_x(x, y))^2 + (G_y(x, y))^2 \right\} \tag{4}$$

where,  $G_x(x, y)$  and  $G_y(x, y)$  are images acquired through convolution with the horizontal and vertical Sobel operators, respectively.

Finally, there is Gray-Level Variance (GLV) as a statistics-based operator. It has been proposed on the basis of the idea that the variance of gray level in a sharp image is higher than in a blurred image. GLV for a central pixel in a  $W \times W$  window is calculated as:

$$F_{GLV}(i, j) = \frac{1}{N^2} \sum_{x \in W} \sum_{y \in W} \left\{ (I(x, y) - \mu)^2 \right\} \tag{5}$$

where,  $\mu$  is the mean of the gray values in a  $W \times W$  window.

### 3. Noise Modeling

When a sequence of 2D images is obtained by translating the object at a constant step size along the optical axis, mechanical vibrations, referred as jitter noise, occur in each step. In this manuscript, two probability density functions are used for modeling the jitter noise. At first, the jitter noise is modeled as Gaussian function with mean  $\mu_n$  and standard deviation  $\sigma_n$ , as shown in Figure 3.

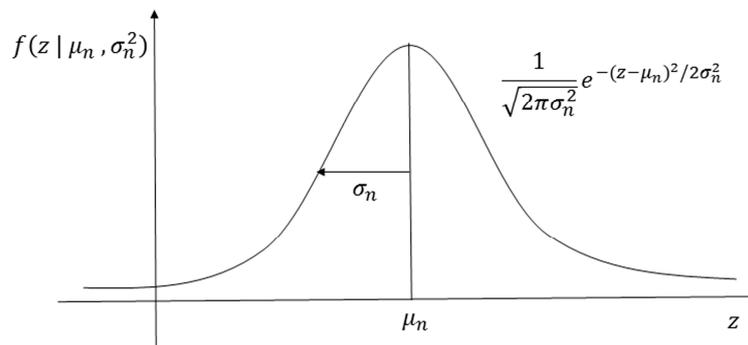


Figure 3. Noise modeling through Gaussian function.

$\mu_n$  represents the position of each image frame without the jitter noise, and  $\sigma_n$  represents the amount of jitter noise occurred in each image frame.  $\sigma_n$  is determined by checking depth of field and corresponding image position. The depth of field is affected by magnification and different factors.  $\sigma_n$  is selected as  $\sigma_n \leq 10 \mu\text{m}$  through repeated experiments with real objects used in this manuscript. Second, the jitter noise is modeled as speckle function as follows [34,35]:

$$f(\zeta) = \frac{1}{2\sigma_n^2} \times e^{\frac{-\zeta}{2\sigma_n^2}} \tag{6}$$

where,  $\zeta$  is the amount of jitter noise before or after filtering.

### 4. Focus Curve Modeling

In order to filter out jitter noise, the focus curve obtained by one of the focus measure operators is modeled by Gaussian approximation with mean  $z_f$  and standard deviation  $\sigma_f$  [11]. This focus curve modeling is shown in Figure 4.

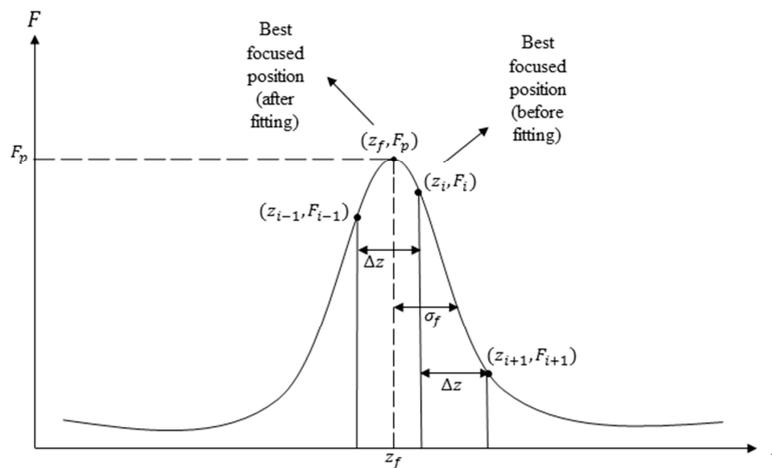


Figure 4. Gaussian fitting of the focus curve.

Related equation about this method is given as:

$$F(z) = F_p \times e^{\left(-\frac{1}{2} \left(\frac{z-z_f}{\sigma_f}\right)^2\right)} \tag{7}$$

where,  $z$  is the position of each image frame,  $F(z)$  is the focus value at  $z$ ,  $z_f$  is the best-focused position of the object point,  $\sigma_f$  is standard deviation of the approximated focus curve (by Gaussian function), and  $F_p$  is amplitude of the focus curve. Using the natural logarithm to (7), (8) is obtained:

$$\ln(F(z)) = \ln(F_p) - \frac{1}{2} \left(\frac{z-z_f}{\sigma_f}\right)^2 \tag{8}$$

Using (8) and initial best-focused position obtained through one of the focus measure operators  $z_i$  and the positions below and above initial best-focused position  $z_{i-1}$  and  $z_{i+1}$  and their corresponding focus values  $F_i$ ,  $F_{i-1}$  and  $F_{i+1}$ , (9) and (10) are obtained:

$$\ln(F_i) - \ln(F_{i-1}) = -\frac{1}{2} \frac{((z_i - z_f)^2 - (z_{i-1} - z_f)^2)}{\sigma_f^2} \tag{9}$$

$$\ln(F_i) - \ln(F_{i+1}) = -\frac{1}{2} \frac{((z_i - z_f)^2 - (z_{i+1} - z_f)^2)}{\sigma_f^2} \tag{10}$$

Using (10), (11) is acquired as follows:

$$\frac{1}{\sigma_f^2} = \frac{\ln(F_i) - \ln(F_{i+1})}{-\frac{1}{2}((z_i - z_f)^2 - (z_{i+1} - z_f)^2)} \tag{11}$$

Applying (11) to (9), (12) is obtained:

$$\ln(F_i) - \ln(F_{i-1}) = \frac{((z_i - z_f)^2 - (z_{i-1} - z_f)^2) \times (\ln(F_i) - \ln(F_{i+1}))}{(z_i - z_f)^2 - (z_{i+1} - z_f)^2} \tag{12}$$

Assuming  $\Delta z = z_{i+1} - z_i = z_i - z_{i-1} = 1$  and utilizing (12), (13) is acquired as:

$$z_f = \frac{(\ln(F_i) - \ln(F_{i+1}))(z_i^2 - z_{i-1}^2) - (\ln(F_i) - \ln(F_{i-1}))(z_i^2 - z_{i+1}^2)}{2((\ln(F_i) - \ln(F_{i-1})) + (\ln(F_i) - \ln(F_{i+1})))} \tag{13}$$

Using (11) and (13), (14) is obtained as:

$$\sigma_f^2 = -\frac{(z_i^2 - z_{i-1}^2) + (z_i^2 - z_{i+1}^2)}{2((\ln(F_i) - \ln(F_{i-1})) + (\ln(F_i) - \ln(F_{i+1})))} \tag{14}$$

Utilizing (7), (13), and (14),  $F_p$  is acquired by following:

$$F_p = \frac{F_i}{e^{(-\frac{1}{2}(\frac{z_i - z_f}{\sigma_f})^2)}} \tag{15}$$

Substituting (13), (14), and (15) into (7), final focus curve obtained by Gaussian approximation is acquired. Since jitter noise is considered in this paper, Equation (7) is modified as follows:

$$F_n(z) = F_p \times e^{(-\frac{1}{2}(\frac{(z+\zeta) - z_f}{\sigma_f})^2)} \tag{16}$$

where,  $\zeta$  is previously modeled (jitter) noise, approximated by Gaussian or speckle function. Using the proposed filter (in the next section), this noise is filtered to obtain a noise-free focus curve.

### 5. Proposed Method

Various filters can be used for removing the jitter noise. In this manuscript, Kalman filter is used as an optimal estimator and is designed accordingly. It is a recursive filter, which tracks the state of a linear dynamic system that contains noise, and is based on measurements made over time. More accurate estimation results can be obtained than by using only measurements at that moment. The Kalman filter, which recursively processes input data including noise, can predict optimal current state statistically [36–39]. The application of the Kalman filter to the SFF system is shown in Figure 5.

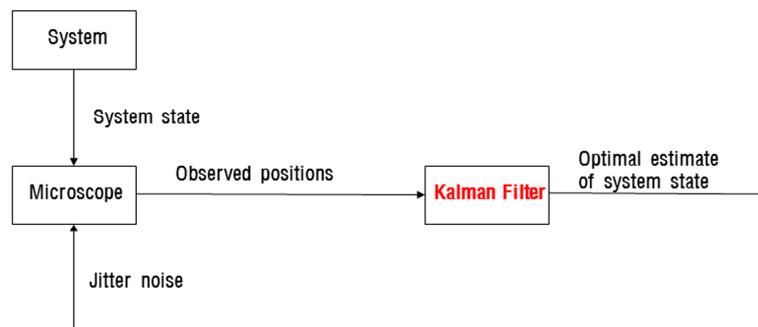


Figure 5. Application of Kalman filter to SFF system.

The system is defined by the position of each image frame in a 2D image sequence. The system state is changed by the jitter noise, which is the measurement noise, in the microscope. The optimal estimate of the system state is obtained by removing the jitter noise through the Kalman filter.

The entire Kalman filter algorithm can be divided into two parts: prediction and update. The prediction refers to the prediction of the current state, and update means that a more accurate

prediction can be made through the values from the present state to the observed measurement. The prediction of the state and its variance is represented as follows:

$$S = T \times S + C \times U \tag{17}$$

$$V = T \times V \times T' + N_p \tag{18}$$

where,  $S$  is “estimate of the system state”,  $T$  is “transition coefficient of the state”,  $U$  is “input”;  $C$  is “control coefficient of the input”,  $V$  is “variance of the state estimate”, and  $N_p$  is “variance of the process noise”. In the SFF system,  $S$  is represented as the position of each image frame in the 2D image sequence estimated by the Kalman filter, and  $C$ ,  $U$ , and  $N_p$  are all set as 0, since there is no control input in the SFF system, and only jitter noise, as the measurement noise, is considered in this manuscript. Next, the computation of the Kalman gain is given for updating the predicted state as follows:

$$G = V \times A' \times inv(A \times V \times A' + N_m) \tag{19}$$

where,  $G$  is “Kalman gain”,  $A$  is “observation coefficient”, and  $N_m$  is “variance of the measurement noise”. In an SFF system,  $N_m$  is defined as the variance of the previously modeled jitter noise. Finally, the update of the predicted state on the basis of the observed measurement is provided by:

$$S = S + G \times (O - A \times S) \tag{20}$$

$$V = V - G \times A \times V \tag{21}$$

where,  $O$  is “observed measurement”. In an SFF system,  $O$  is represented as the position of each image frame in the image sequence before filtering. Parameters that are not set to values,  $T$  and  $A$ , are all set to 1 for simplicity. For the start of the algorithm,  $S$  and  $V$  are initialized as:

$$S = inv(A) \times O \tag{22}$$

$$V = inv(A) \times N_m \times inv(A') \tag{23}$$

Through the Kalman filter algorithm, the optimal position  $S$  of each image frame in the image sequence is estimated. The pseudo code for the Kalman filter algorithm is shown in Algorithm 1.

---

**Algorithm 1** Computing optimal position of each image frame and remaining jitter noise

---

- 1: **procedure** Optimal position  $S$  & remaining jitter noise  $\zeta$
  - 2:  $S \leftarrow O$  ▷ Set initial position of image frame with observed position
  - 3:  $V \leftarrow N_m$  ▷ Initialize variance of position of image frame to variance of jitter noise
  - 4: **for**  $i = 1 \rightarrow N$  **do** ▷ Total number of iterations of Kalman filter
  - 5:  $G \leftarrow V \cdot (V + N_m)^{-1}$  ▷ Compute Kalman gain
  - 6:  $V \leftarrow V - G \cdot V$  ▷ Correct variance of position of image frame
  - 7:  $S \leftarrow S + G \cdot (O - S)$  ▷ Update position of image frame
  - 8:  $\zeta \leftarrow |S - \mu_n|$  ▷ Compute remaining jitter noise
  - 9: **end for**
  - 10: **end procedure**
- 

The difference between the true position  $\mu_n$ , which is the position of each image frame without the jitter noise, and optimal position  $S$ , is put to  $\zeta$ . This algorithm is repeated for all image frames in the image sequence. After acquiring a filtered image sequence, a depth map is obtained by maximizing the focus measure obtained by using the previously modeled focus curve, for each pixel in the image sequence. A list of frequently used symbols and notations is shown in Table 1.

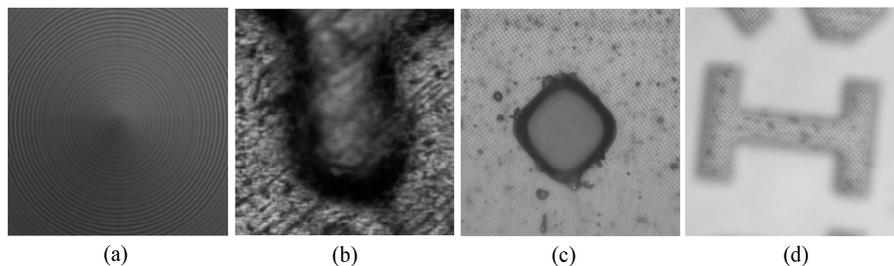
**Table 1.** List of frequently used symbols and notation.

Notation	Description
$\mu_n$	Position of each image frame without jitter noise
$\sigma_n$	Standard deviation of jitter noise
$z_f$	Best focused position through Gaussian approximation in each object point
$\sigma_f$	Standard deviation of Gaussian focus curve
$\zeta$	The amount of jitter noise before or after filtering
$S$	Position of each image frame after Kalman filtering
$O$	Position of each image frame before Kalman filtering
$N$	Total number of iterations of filters

## 6. Results and Discussion

### 6.1. Image Acquisition and Parameter Setting

For experiments, four objects were used, as shown in Figure 6, consisting of one simulated and three real objects.



**Figure 6.** 10th frame of experimented objects: (a) Simulated cone, (b) Coin, (c) Liquid Crystal Display-Thin Film Transistor (LCD-TFT) filter, (d) Letter-I.

First, a simulated cone image sequence consisting of 97 images, with dimensions of  $360 \times 360$  pixels, was acquired. These images were generated using camera simulation software [40].

The real objects used for experiments were: coin, Liquid Crystal Display-Thin Film Transistor (LCD-TFT) filter, and letter-I. The coin images were magnified images of Lincoln's head from the back of the US penny. The coin sequence consisted of 80 images, with dimensions of  $300 \times 300$  pixels. The LCD-TFT filter images consisted of microscopic images of an LCD color filter. The image sequence of LCD-TFT filter had 60 images, with the dimensions of  $300 \times 300$  pixels each. The third image sequence consisted of letter-I, engraved on the metallic surface. It consisted of 60 images, with dimensions of  $300 \times 300$  pixels each. The real objects were acquired through a microscopic control system (MCS) [18]. The system consists of a personal computer integrated with a frame grabber board (Matrox Meteor-II) and a CCD camera (SAMSUNG CAMERA SCC-341) mounted on a microscope (NIKON OPTIPHOT-100S). Computer software obtains images by translating the object plane through a stepper motor driver (MAC 5000), possessing a 2.5 nm minimum step size. The coin and letter-I images were obtained under  $10\times$  magnification, while the LCD-TFT filter images were acquired under  $50\times$  magnification.

In parameter setting, the standard deviation of the jitter noise for each object was assumed to be ten times the sampling step size of each image sequence, i.e., 254 mm, 6.191  $\mu\text{m}$ , 1.059 m, and 1.529  $\mu\text{m}$  for simulated cone, coin, LCD-TFT filter, and letter-I, respectively. For comparison of 3D shape recovery results, a local window  $7 \times 7$  for focus measure operators was used. The total number of iterations  $N$  of the Kalman filter was set as 100.

For performance comparison, Bayes filter and particle filter were employed [41–46]. The depth estimation through the Bayes filter is presented in Figure 7.

In Figure 7,  $z_0$  is defined as a total number of 2D images obtained for SFF, and  $p_j(i)$  is presented as follows:

$$p_j(i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(z(j)-r(i))^2}{2\sigma_n^2}}, \quad 1 \leq i \leq M, \quad 1 \leq j \leq N \quad (24)$$

where,  $p_j(i)$  is Gaussian probability density function,  $z(j)$  is the position of each image frame changed by the jitter noise,  $r(i)$  is the possible positions of each image frame in the presence of the jitter noise,  $\sigma_n$  is standard deviation of previously modeled jitter noise,  $M$  is the total length of  $r(i)$  with intervals of 0.01, and  $N$  is the total number of iterations of Bayes filter. The reason why  $3\sigma_n$  is set in the range of  $r$ , is because  $3\sigma_n$  makes  $z(j)$  be in the range of  $r$  with the probability of 99.7% due to the Gaussian probability density function. The recursive Bayesian estimation was applied to all 2D image frames obtained for SFF. After the filtered image sequence was acquired, an optimal depth map was obtained using the previously modeled focus curve.

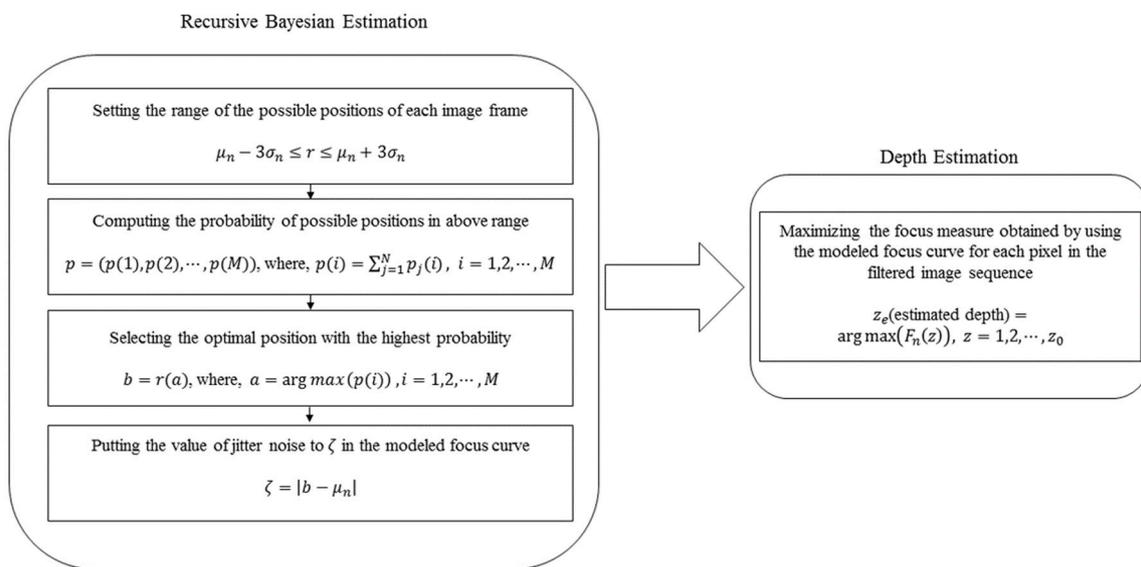


Figure 7. Depth estimation through Bayes filter.

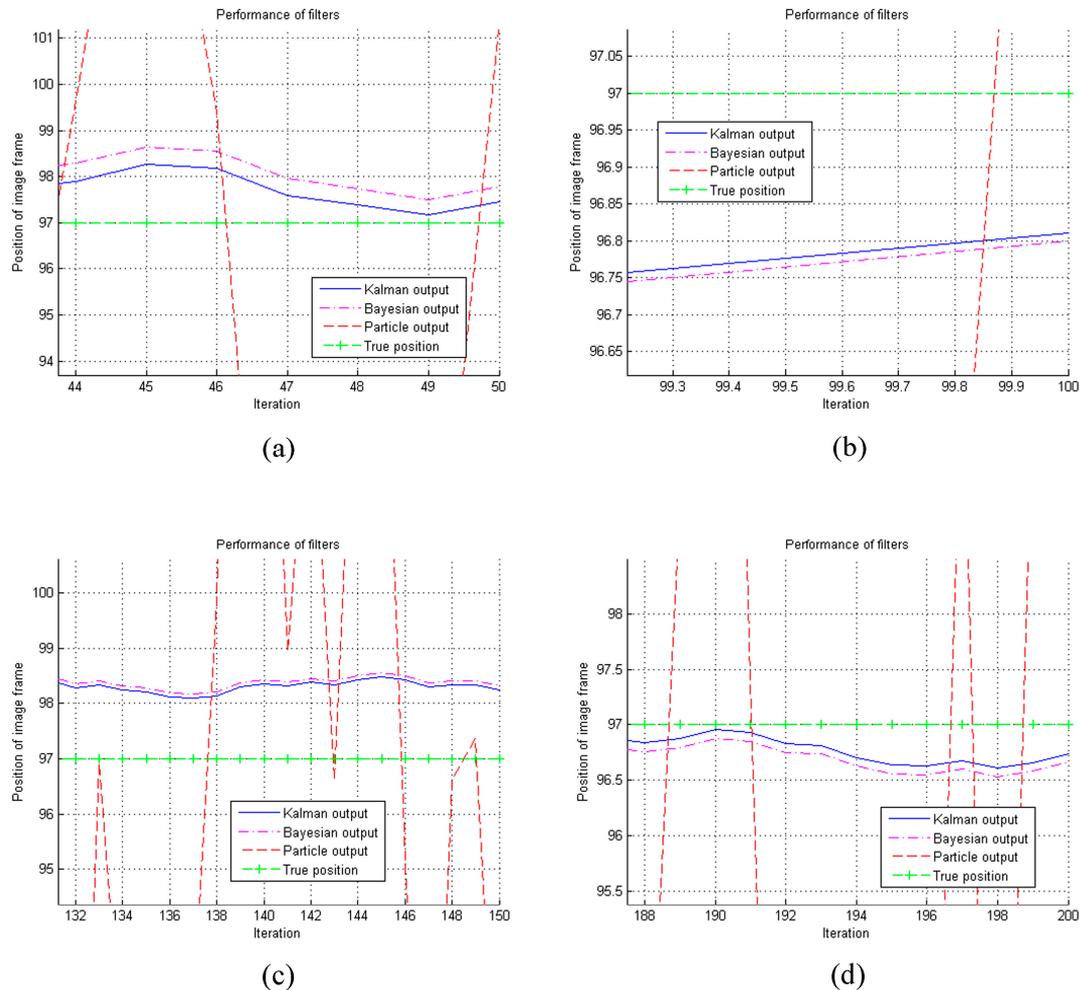
Particle filter algorithm is mainly divided into two steps: Generating the weights for each of particles and resampling for acquiring new estimated particles. In the first step, the weights are based on the probability of the given observation for each of the particles as:

$$p_w(i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(z-z_p(i))^2}{2\sigma_n^2}}, \quad 1 \leq i \leq P \quad (25)$$

where,  $p_w(i)$  is Gaussian probability density function,  $z$  is the observed position of each image frame changed by the jitter noise,  $z_p(i)$  is vector of particles, and  $P$  is the number of particles the SFF system generates. In this manuscript,  $z_p(i)$  is initialized by randomly selecting the values on the x-axis from the previously modeled jitter noise, and  $P$  is set as 1000. After the weights are normalized, resampling, as the second step, is needed for acquiring new estimated particles. The new estimated particles are obtained by sampling the cumulative distribution of the normalized  $p_w(i)$ , randomly and uniformly. Through this sampling, the particles with the higher weights are selected. This particle filter algorithm is repeated  $N$  times, as the total number of iterations of the particle filter. The optimal position of each image frame is the mean of the final estimated particles  $z_p(i)$  obtained through resampling in iteration  $N$ . After the filtered image sequence is acquired through application of the particle filter to all 2D image frames, optimal depth map is obtained in the same way as the depth estimation in Figure 7.

6.2. Experimental Results

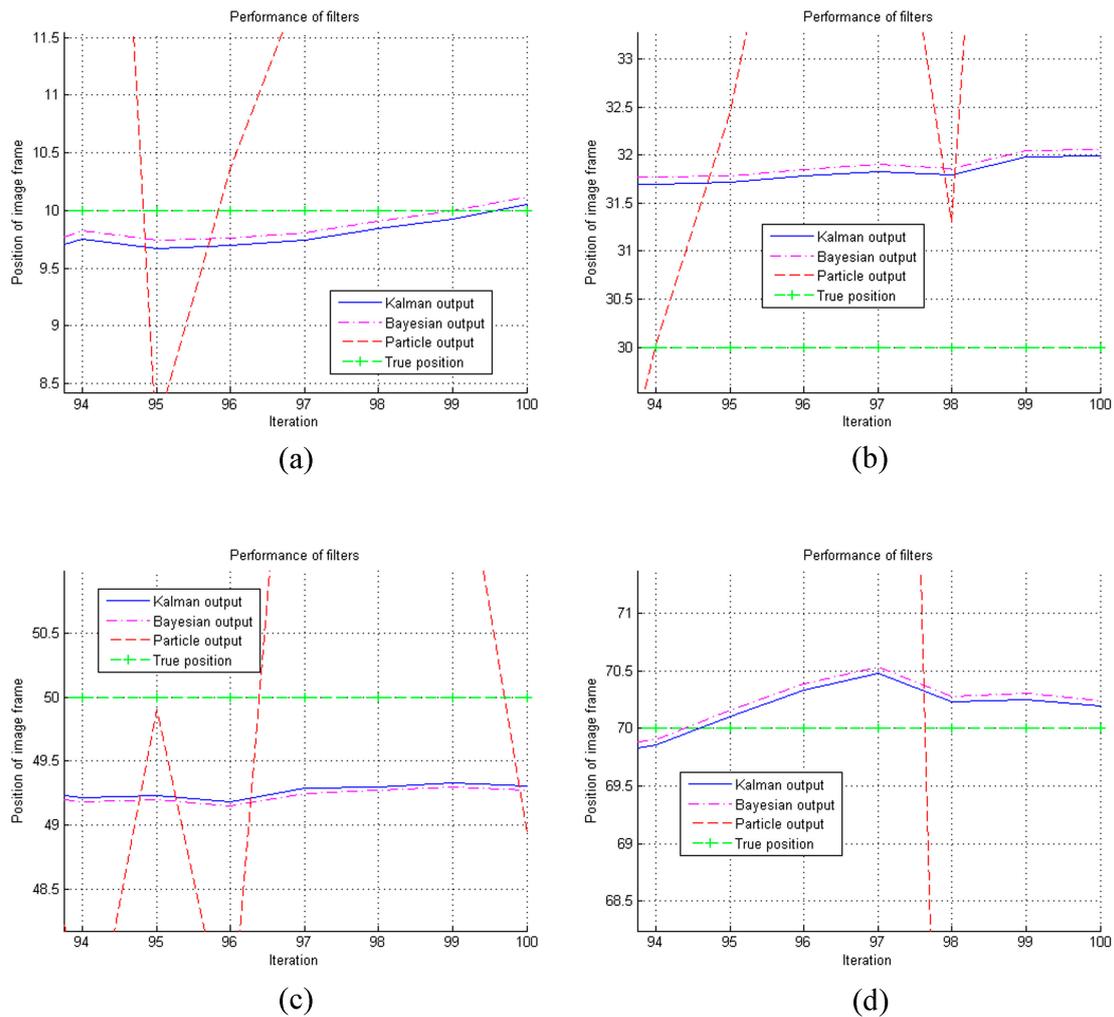
Figure 8 presents the performance comparison of the filters in the 97th frame of the simulated cone using various iterations in the presence of Gaussian jitter noise.



**Figure 8.** Performance of the filters: (a) Iteration—50, (b) Iteration—100, (c) Iteration—150, (d) Iteration—200.

Figure 9 provides performance comparison of the filters in the 100th iteration using various frames of the simulated cone in the presence of Gaussian jitter noise.

These Figures are intensively enlarged versions of the last iteration. “Kalman output” is the estimated position through Kalman filter, “Bayesian output” is the estimated position through Bayes filter, “Particle output” is the estimated position through particle filter, and “True position” is the position without the jitter noise. It is clear from these Figures that Kalman output converged better to True position than Bayesian output and Particle output. It means that Kalman filter outperformed the other filters compared for experiments.

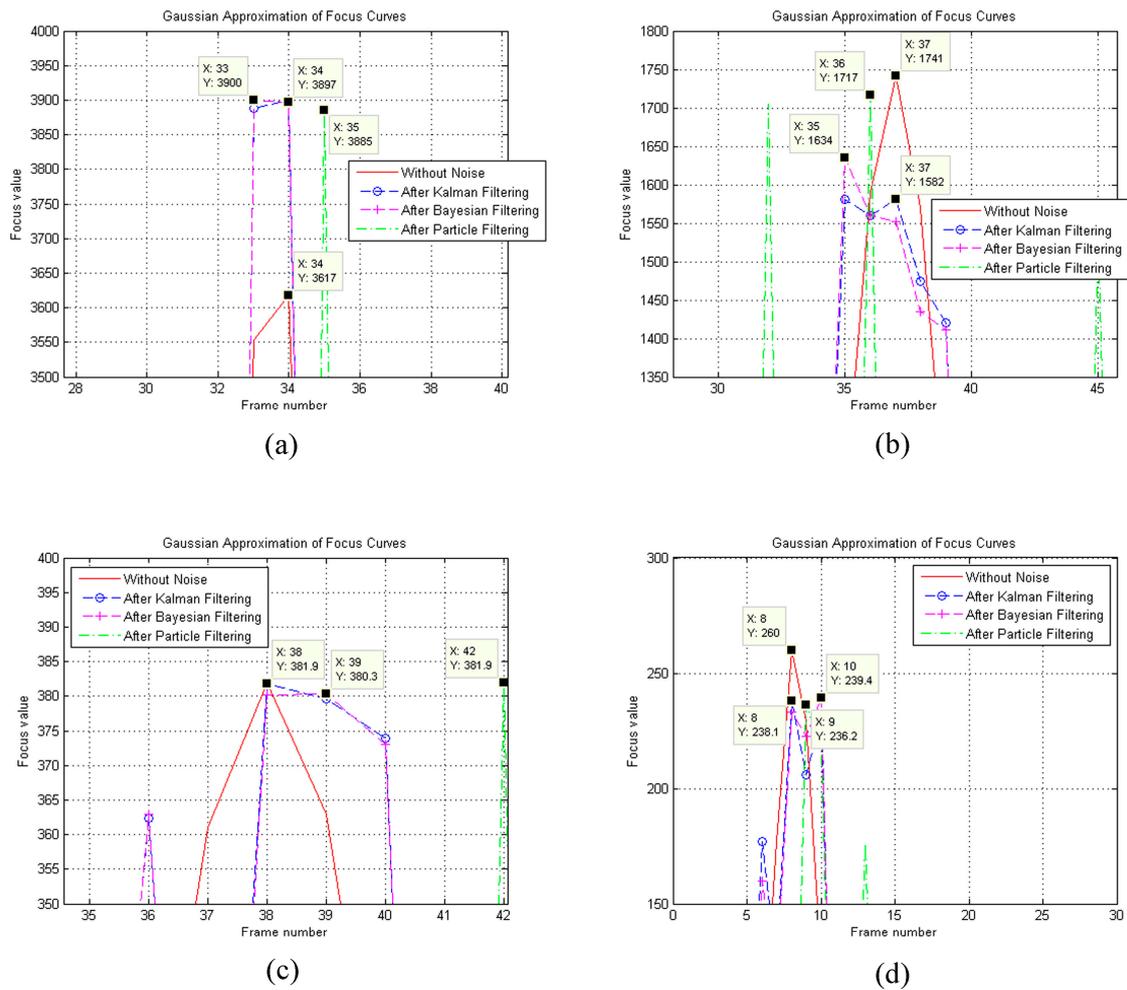


**Figure 9.** Performance of the filters: (a) Frame number—10, (b) Frame number—30, (c) Frame number—50, (d) Frame number—70.

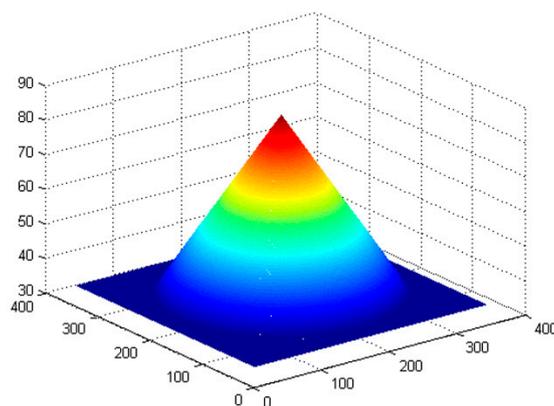
Figure 10 shows the Gaussian approximation of the focus curves using experimented objects in the presence of Gaussian jitter noise.

“Without Noise” is the Gaussian approximation of the focus curve without the jitter noise, “After Kalman Filtering” is the Gaussian approximation of the focus curve after Kalman filtering, “After Bayesian Filtering” is the Gaussian approximation of the focus curve after Bayes filtering, and “After Particle Filtering” is the Gaussian approximation of the focus curve after particle filtering. It is clear from Figure 10 that the optimal position with the highest focus value in After Kalman Filtering is closer to the optimal position in Without Noise than the optimal positions in the focus curves obtained after using other filtering techniques.

For performance evaluation of 3D shape recovery, three metrics were used in case of simulated cone, since the synthetic object had an actual depth map, as in Figure 11 [47].



**Figure 10.** Gaussian approximation of focus curves: (a) Simulated cone (60, 60), (b) Coin (120, 120), (c) LCD-TFT filter (180, 180), (d) Letter-I (240, 240).



**Figure 11.** Actual depth map of simulated cone.

The first one is Root Mean Square Error (RMSE), which is a commonly used measure when dealing with the difference between estimated and actual value, as follows:

$$RMSE = \sqrt{\frac{1}{XY} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (d(x, y) - \hat{d}(x, y))^2} \quad (26)$$

where,  $d(x, y)$  and  $\hat{d}(x, y)$  are the actual and estimated depth map, respectively, and  $X$  and  $Y$  are width and height of 2D images, which are used for SFF, respectively.

The second one is correlation, which shows the linear relationship and strength between two variables as:

$$\text{Correlation} = \frac{\sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (d(x, y) - \bar{d}(x, y))(\hat{d}(x, y) - \overline{\hat{d}(x, y)})}{\sqrt{(\sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (d(x, y) - \bar{d}(x, y))^2)(\sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (\hat{d}(x, y) - \overline{\hat{d}(x, y)})^2)}} \tag{27}$$

where,  $\bar{d}(x, y)$  and  $\overline{\hat{d}(x, y)}$  are the means of the actual and estimated depth map, respectively.

The third one is Peak Signal-to-Noise Ratio (PSNR), which is the power of noise over the maximum power a signal can have. It is usually represented in terms of the logarithmic decibel scale as:

$$\text{PSNR} = 10 \log_{10} \left( \frac{d_{max}^2}{MSE} \right) \tag{28}$$

where,  $d_{max}$  is maximum depth value in the depth map and  $MSE$  is the Mean Square Error, which is the square of the RMSE. The lower the RMSE, the higher the correlation, and the higher the PSNR, the higher the accuracy of 3D shape reconstruction.

Tables 2–4 provide the quantitative performance of 3D shape recovery of the simulated cone using three focus measures, SML, GLV, and TEN, before and after filtering in the presence of Gaussian jitter noise.

**Table 2.** Comparison of focus measure operators with proposed method for simulated cone in the presence of Gaussian noise by using RMSE (Root Mean Square Error). SML: Sum of Modified Laplacian; GLV: Gray-Level Variance; TEN: Tenenbaum.

Focus Measure Operators	SML	GLV	TEN
Before Filtering	9.2629	12.4038	15.2304
After Particle Filtering	9.1993	10.9459	11.2293
After Bayesian Filtering	7.3260	8.2659	8.4961
After Kalman Filtering	7.3169	8.1400	8.3652

**Table 3.** Comparison of focus measure operators with proposed method for simulated cone in the presence of Gaussian noise by using correlation.

Focus Measure Operators	SML	GLV	TEN
Before Filtering	0.7831	0.7430	0.7121
After Particle Filtering	0.7925	0.8157	0.7914
After Bayesian Filtering	0.9536	0.9427	0.9200
After Kalman Filtering	0.9541	0.9438	0.9206

**Table 4.** Comparison of focus measure operators with proposed method for simulated cone in the presence of Gaussian noise by using PSNR.

Focus Measure Operators	SML	GLV	TEN
Before Filtering	20.1276	17.8644	16.0812
After Particle Filtering	20.4604	18.9504	18.7284
After Bayesian Filtering	22.3481	21.3897	21.1510
After Kalman Filtering	22.3588	21.5229	21.2859

The order of the general performance of the focus measures is that SML is the best, then the GLV, and finally the TEN. In Before Filtering, it is difficult to distinguish the performance of the focus

measures due to the jitter noise. However, in After Bayesian Filtering and After Kalman Filtering, it is shown in Tables 2–4 that the performance order of the focus measures is almost correct, as described above. The particle filter suitable for nonlinear systems does not remove jitter noise well in a linear SFF system. It is seen in Tables 2–4 that the performance order of the focus measures in After Particle Filtering is slightly different from the one presented above. Tables 5–7 provide the quantitative performance of 3D shape recovery of the simulated cone using three focus measures, SML, GLV, and TEN, before and after filtering in the presence of speckle noise. The performance order of the focus measures for each filtering technique is almost the same as that of focus measures when Gaussian jitter noise is present. However, in the presence of speckle noise, After Kalman Filtering and After Bayesian Filtering have poor performance in terms of RMSE and PSNR. This is because these two filters estimate the position of each 2D image after assuming the jitter noise to be Gaussian function. It is evident from Tables 2–7 that the best overall performance is that of the Kalman filter as the proposed method, which provides the optimal estimation results in a linear system. The Bayes filter comes second, and finally the particle filter, which estimates the optimal value in a nonlinear system.

**Table 5.** Comparison of focus measure operators with proposed method for simulated cone in the presence of speckle noise by using RMSE.

Focus Measure Operators	SML	GLV	TEN
Before Filtering	21.6257	19.5639	19.1356
After Particle Filtering	18.2074	18.1522	18.3674
After Bayesian Filtering	16.9866	17.4630	17.9747
After Kalman Filtering	16.9458	17.4064	17.9344

**Table 6.** Comparison of focus measure operators with proposed method for simulated cone in the presence of speckle noise by using correlation.

Focus Measure Operators	SML	GLV	TEN
Before Filtering	0.8133	0.8661	0.8570
After Particle Filtering	0.8941	0.9083	0.8873
After Bayesian Filtering	0.9518	0.9496	0.9316
After Kalman Filtering	0.9527	0.9504	0.9325

**Table 7.** Comparison of focus measure operators with proposed method for simulated cone in the presence of speckle noise by using PSNR (Peak Signal-to-Noise Ratio).

Focus Measure Operators	SML	GLV	TEN
Before Filtering	12.2884	13.3517	13.3074
After Particle Filtering	13.2806	13.8092	13.5967
After Bayesian Filtering	14.0878	13.8476	13.6162
After Kalman Filtering	14.1087	13.8758	13.6389

Tables 8 and 9 present the time taken to estimate the position of one image frame by using the filters for the experimented objects in the presence of Gaussian and speckle noise, respectively.

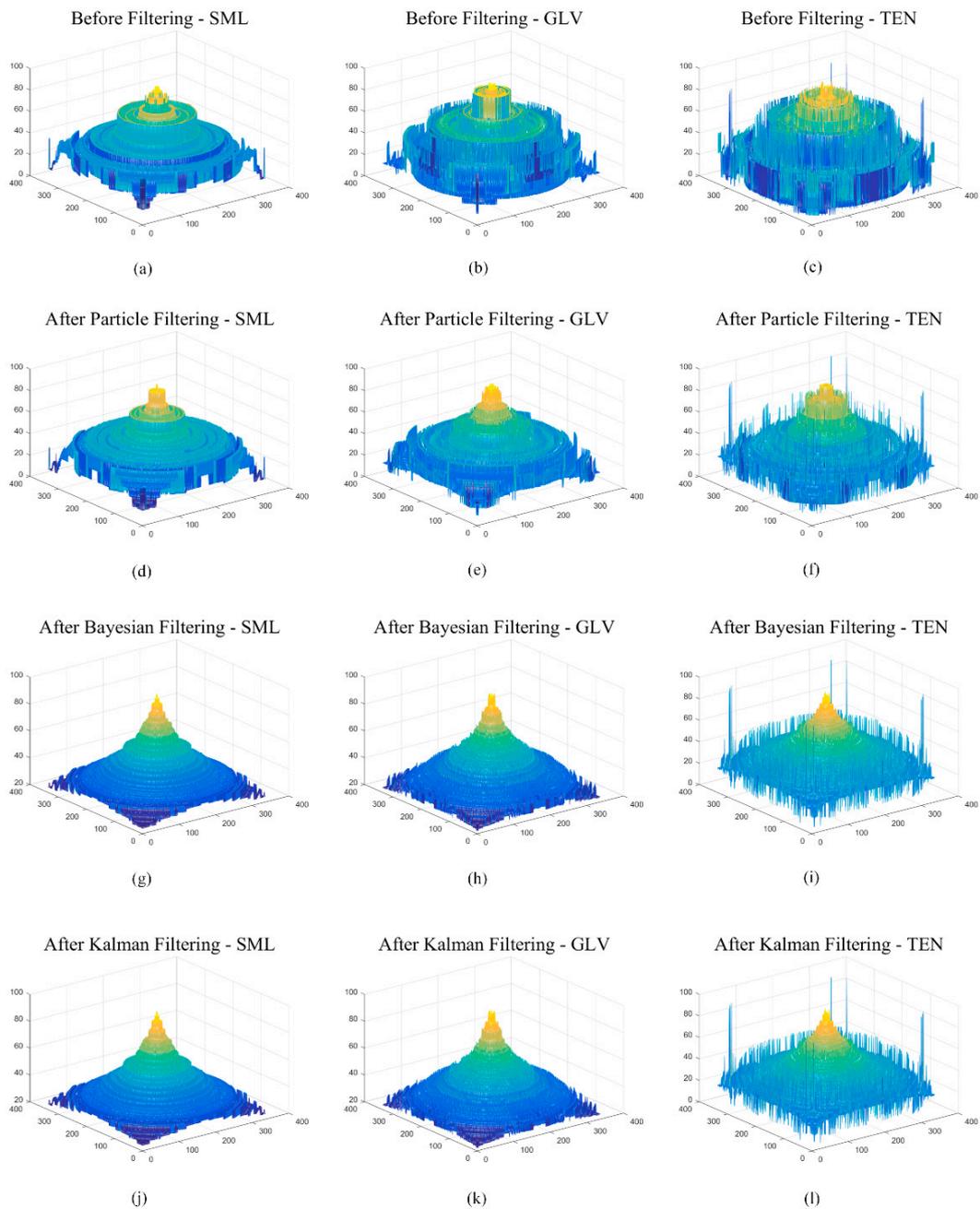
**Table 8.** Computation time of filters for the experimented objects in the presence of Gaussian noise.

Experimented Objects	Particle Filter	Bayes Filter	Kalman Filter
Simulated cone	0.651821	0.112929	0.006780
Coin	0.699162	0.125152	0.009283
LCD-TFT filter	0.624884	0.112957	0.007769
Letter-I	0.688404	0.112758	0.007259

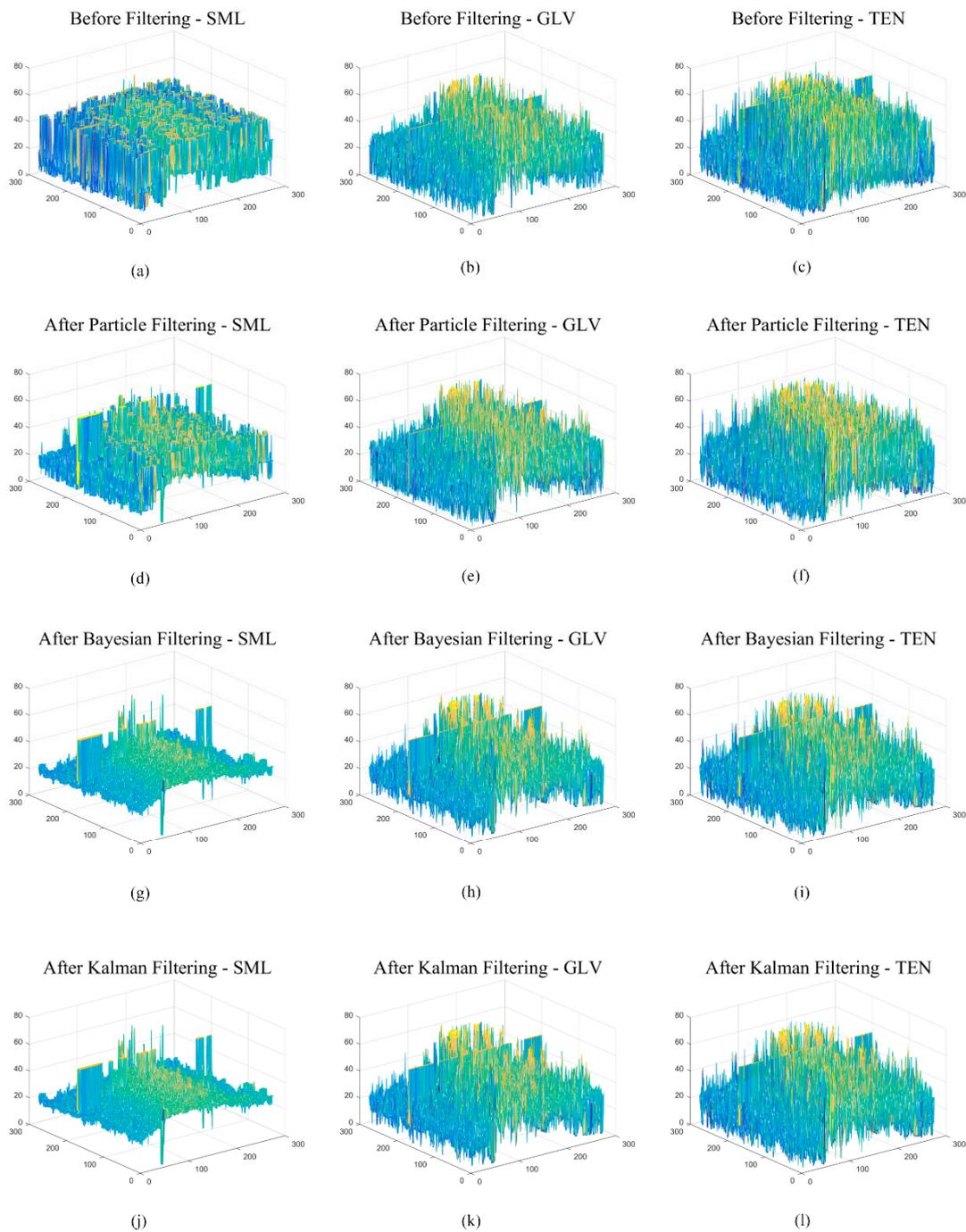
**Table 9.** Computation time of filters for the experimented objects in the presence of speckle noise.

Experimented Objects	Particle Filter	Bayes Filter	Kalman Filter
Simulated cone	0.637766	0.113022	0.008112
Coin	0.677874	0.124471	0.009683
LCD-TFT filter	0.625544	0.116263	0.007738
Letter-I	0.636709	0.112466	0.009179

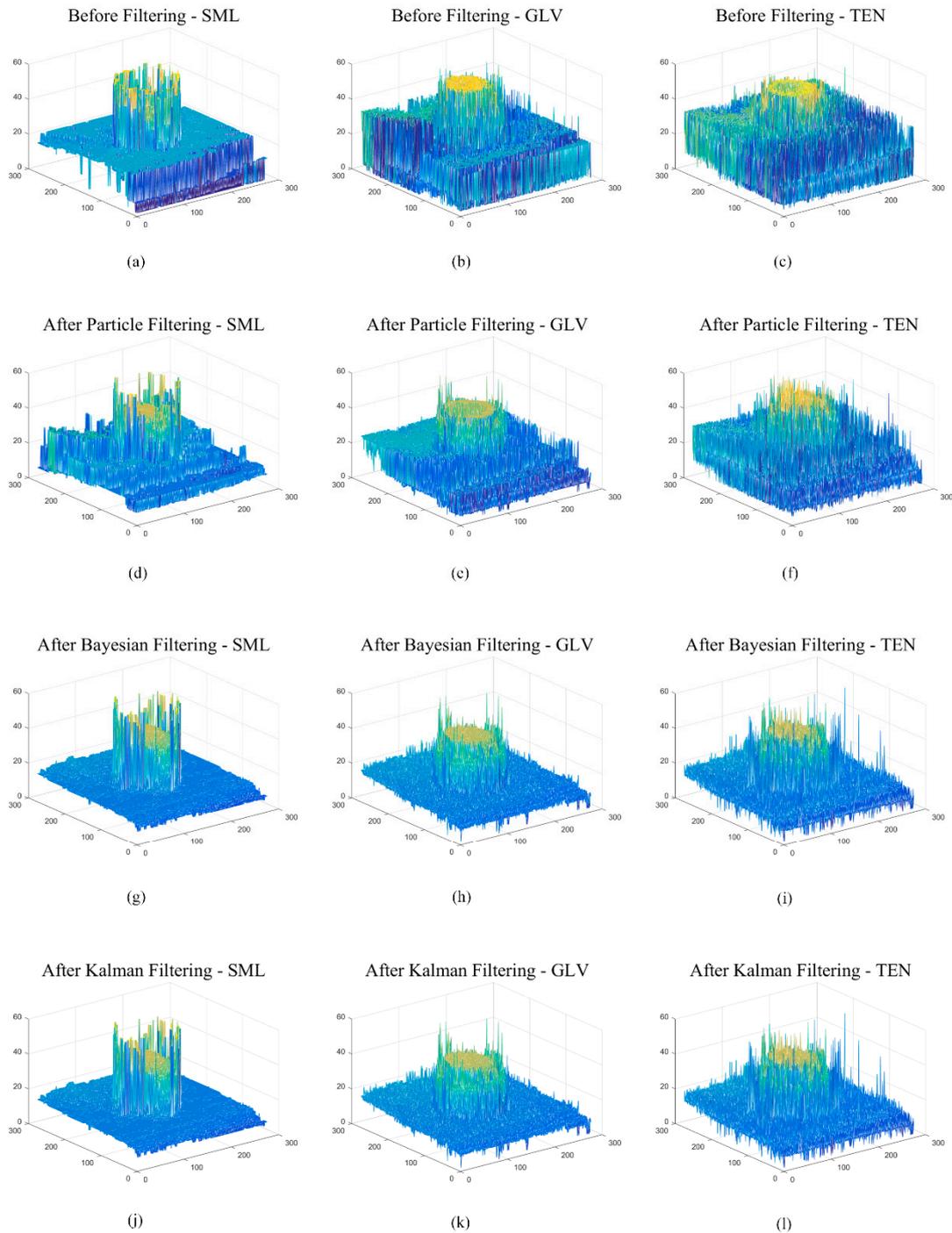
The computation time in Tables 8 and 9 is expressed in seconds. It is evident that the computation time of the Kalman filter was about 14 times better than the Bayes filter and about 80 times better than the particle filter. Figures 12–14 show the qualitative performance of 3D shape reconstruction of the experimented objects using three focus measures, SML, GLV, and TEN, before and after filtering in the presence of Gaussian noise. Figures 15 and 16 provide the qualitative performance of 3D shape reconstruction of the experimented objects using three focus measures, SML, GLV, and TEN, before and after filtering in the presence of speckle noise. In Before Filtering and After Particle Filtering, it can be seen that the performance of 3D shape reconstruction was very poor due to unremoved or poorly removed jitter noise. However, in After Bayesian Filtering and After Kalman Filtering, it is evident that the performance of the 3D shape recovery was greatly improved due to the elimination of most of the jitter noise. It is proved from these experimental results that filtering the jitter noise using the Kalman filter improves the 3D shape reconstruction faster and more accurately.



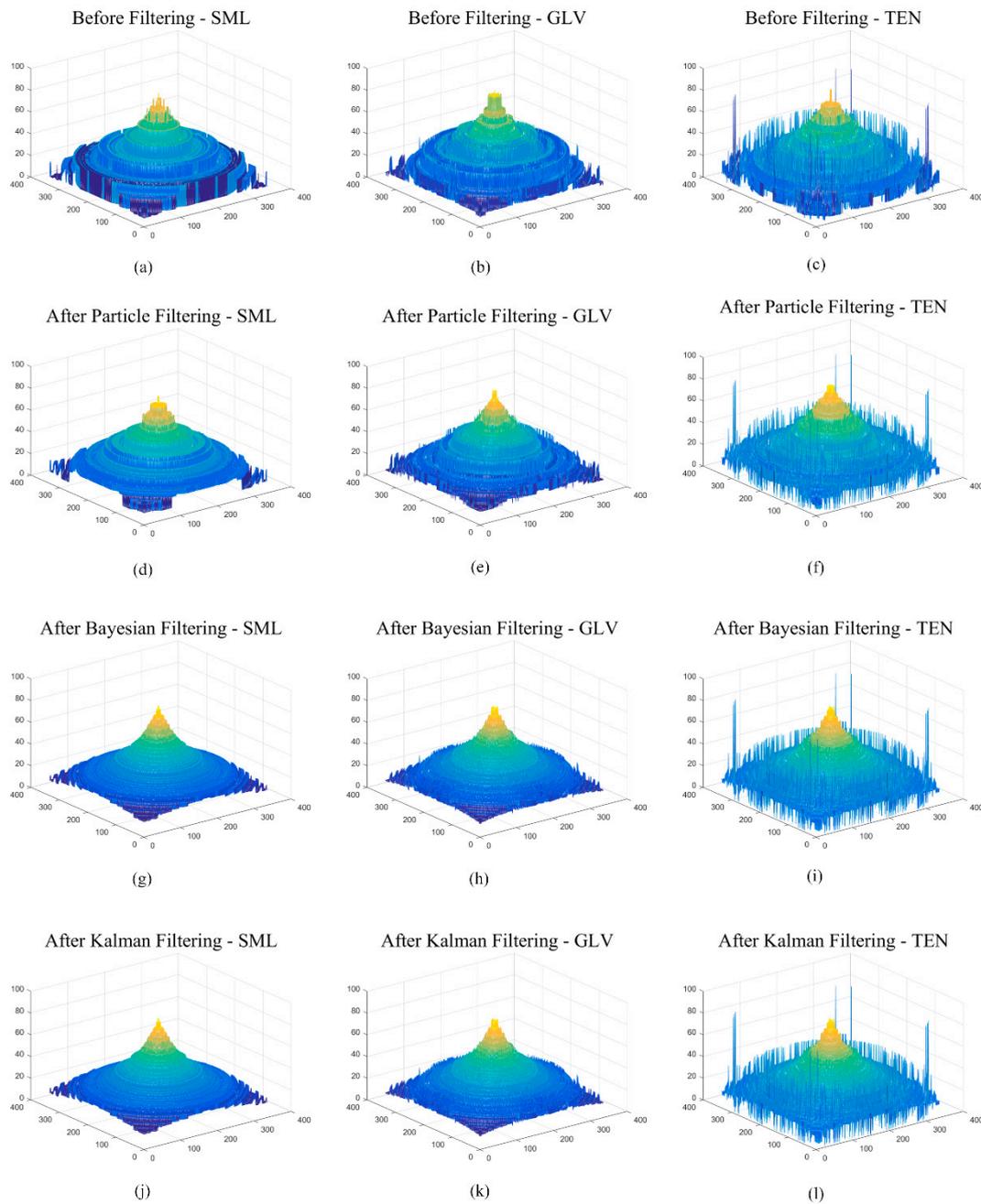
**Figure 12.** 3D shape recovery of simulated cone, before and after filtering, using SML, GLV, and TEN in the presence of Gaussian noise. (a) Before filtering for SML; (b) Before filtering for GLV; (c) Before filtering for TEN; (d) After particle filtering for SML; (e) After particle filtering for GLV; (f) After particle filtering for TEN; (g) After Bayesian filtering for SML; (h) After Bayesian filtering for GLV; (i) After Bayesian filtering for TEN; (j) After Kalman filtering for SML; (k) After Kalman filtering for GLV; (l) After Kalman filtering for TEN.



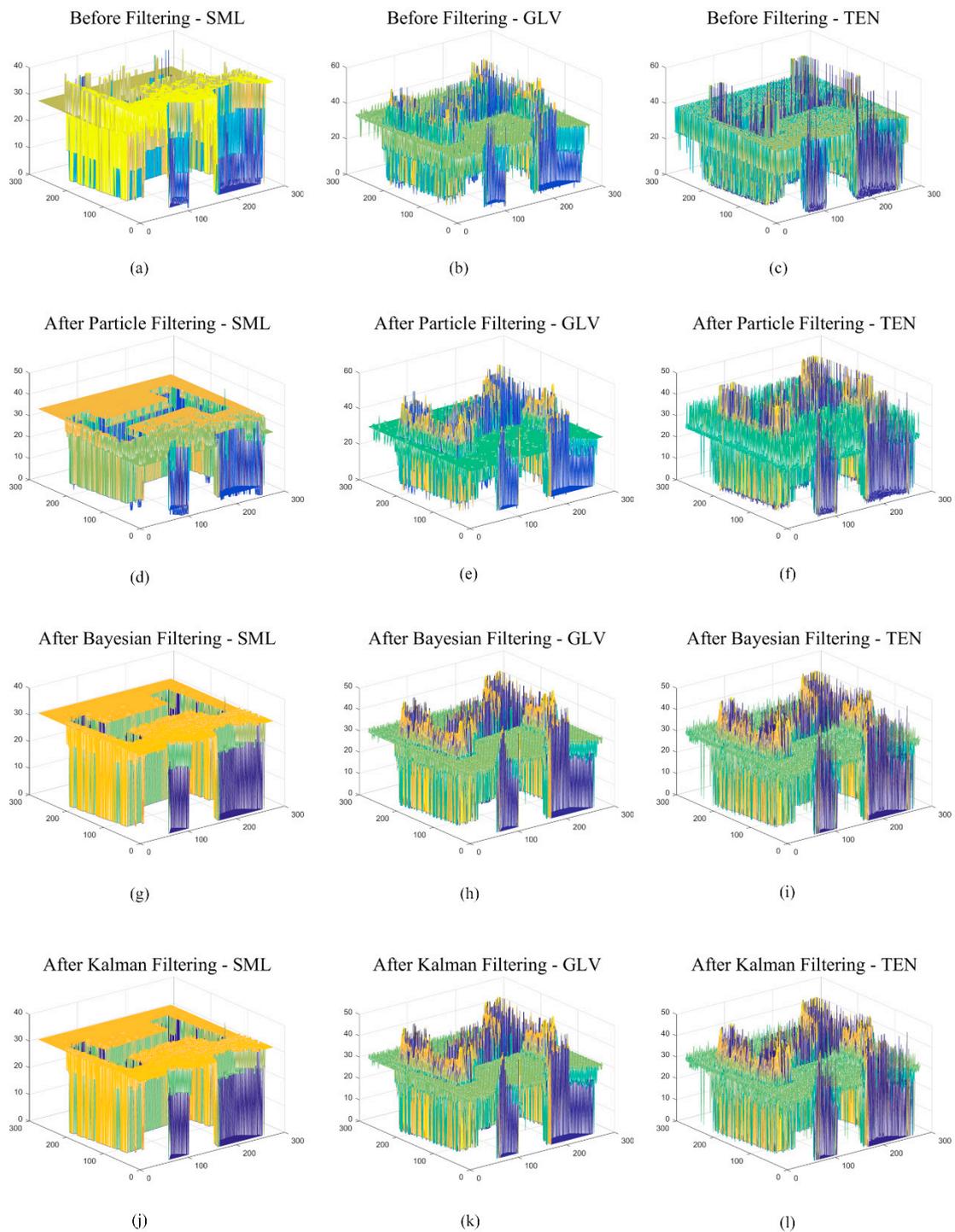
**Figure 13.** 3D shape recovery of coin, before and after filtering, using SML, GLV, and TEN in the presence of Gaussian noise. (a) Before filtering for SML; (b) Before filtering for GLV; (c) Before filtering for TEN; (d) After particle filtering for SML; (e) After particle filtering for GLV; (f) After particle filtering for TEN; (g) After Bayesian filtering for SML; (h) After Bayesian filtering for GLV; (i) After Bayesian filtering for TEN; (j) After Kalman filtering for SML; (k) After Kalman filtering for GLV; (l) After Kalman filtering for TEN.



**Figure 14.** 3D shape recovery of LCD-TFT filter, before and after filtering, using SML, GLV, and TEN in the presence of Gaussian noise. (a) Before filtering for SML; (b) Before filtering for GLV; (c) Before filtering for TEN; (d) After particle filtering for SML; (e) After particle filtering for GLV; (f) After particle filtering for TEN; (g) After Bayesian filtering for SML; (h) After Bayesian filtering for GLV; (i) After Bayesian filtering for TEN; (j) After Kalman filtering for SML; (k) After Kalman filtering for GLV; (l) After Kalman filtering for TEN.



**Figure 15.** 3D shape recovery of simulated cone, before and after filtering, using SML, GLV, and TEN in the presence of speckle noise. (a) Before filtering for SML; (b) Before filtering for GLV; (c) Before filtering for TEN; (d) After particle filtering for SML; (e) After particle filtering for GLV; (f) After particle filtering for TEN; (g) After Bayesian filtering for SML; (h) After Bayesian filtering for GLV; (i) After Bayesian filtering for TEN; (j) After Kalman filtering for SML; (k) After Kalman filtering for GLV; (l) After Kalman filtering for TEN.



**Figure 16.** 3D shape recovery of letter-I, before and after filtering, using SML, GLV, and TEN in the presence of speckle noise. (a) Before filtering for SML; (b) Before filtering for GLV; (c) Before filtering for TEN; (d) After particle filtering for SML; (e) After particle filtering for GLV; (f) After particle filtering for TEN; (g) After Bayesian filtering for SML; (h) After Bayesian filtering for GLV; (i) After Bayesian filtering for TEN; (j) After Kalman filtering for SML; (k) After Kalman filtering for GLV; (l) After Kalman filtering for TEN.

### 7. Conclusions

For SFF, an object is translated at a constant step size along the optical axis. When an image of the object is captured in each step, mechanical vibrations occur, which are referred as jitter noise.

In this manuscript, jitter noise is modeled as Gaussian function with mean  $z_n$  and standard deviation  $\sigma_n$  for simplicity. Then, the focus curves obtained by one of the focus measure operators are also modeled as Gaussian function, with mean  $z_f$  and standard deviation  $\sigma_f$ , for the application of the proposed method. Finally, a new filter is proposed to provide optimal estimation results in a linear SFF system with the jitter noise, utilizing Kalman filter to eliminate jitter noise in the modeled focus curves. Through experimental results, it was found that the Kalman filter provided significantly improved 3D reconstruction of the experimented objects compared with before filtering, and that the 3D shapes of the experimented objects were recovered with more accurate and faster performance than with other existing filters, such as the Bayes filter and the particle filter.

**Author Contributions:** Conceptualization, H.-S.J. and M.S.M.; Methodology, H.-S.J. and M.S.M.; Software, H.-S.J. and G.Y.; Validation, H.-S.J.; Writing—original draft preparation, H.-S.J.; Writing—review and editing, M.S.M.; Supervision, D.H.K.; Funding acquisition, D.H.K.

**Funding:** This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00677, Development of Robot Hand Manipulation Intelligence to Learn Methods and Procedures for Handling Various Objects with Tactile Robot Hands).

**Acknowledgments:** We thank Tae-Sun Choi for his assistance with useful discussion.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lin, J.; Ji, X.; Xu, W.; Dai, Q. Absolute Depth Estimation from a Single Defocused Image. *IEEE Trans. Image Process.* **2013**, *22*, 4545–4550. [[PubMed](#)]
2. Humayun, J.; Malik, A.S. Real-time processing for shape-from-focus techniques. *J. Real Time Image Process.* **2016**, *11*, 49–62. [[CrossRef](#)]
3. Lafarge, F.; Keriven, R.; Brédif, M.; Vu, H.H. A Hybrid Multiview Stereo Algorithm for Modeling Urban Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 5–17. [[CrossRef](#)] [[PubMed](#)]
4. Ciaccio, E.J.; Tennyson, C.A.; Bhagat, G.; Lewis, S.K.; Green, P.H. Use of shape-from-shading to estimate three-dimensional architecture in the small intestinal lumen of celiac and control patients. *Comput. Methods Programs Biomed.* **2013**, *111*, 676–684. [[CrossRef](#)] [[PubMed](#)]
5. Barron, J.T.; Malik, J. Shape, Illumination, and Reflectance from Shading. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1670–1687. [[CrossRef](#)]
6. De Vries, S.C.; Kappers, A.M.L.; Koenderink, J.J.; Vries, S.C. Shape from stereo: A systematic approach using quadratic surfaces. *Percept. Psychophys.* **1993**, *53*, 71–80. [[CrossRef](#)] [[PubMed](#)]
7. Super, B.J.; Bovik, A.C. Shape from Texture Using Local Spectral Moments. *IEEE Trans. Pattern Anal. Mach. Intell.* **1995**, *17*, 333–343. [[CrossRef](#)]
8. Parashar, S.; Pizarro, D.; Bartoli, A. Isometric Non-Rigid Shape-From-Motion in Linear Time. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4679–4687.
9. Favaro, P.; Soatto, S.; Burger, M.; Osher, S.J. Shape from Defocus via Diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 518–531. [[CrossRef](#)] [[PubMed](#)]
10. Tiantian, F.; Hongbin, Y. A novel shape from focus method based on 3D steerable filters for improved performance on treating textureless region. *Opt. Commun.* **2018**, *410*, 254–261.
11. Nayar, S.K.; Nakagawa, Y. Shape from Focus. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 824–831. [[CrossRef](#)]
12. Pertuz, S.; Puig, D.; García, M.A. Analysis of focus measure operators for shape-from-focus. *Pattern Recognit.* **2013**, *46*, 1415–1432. [[CrossRef](#)]
13. Rusiñol, M.; Chazalon, J.; Ogier, J.M. Combining Focus Measure Operators to Predict OCR Accuracy in Mobile-Captured Document Images. In Proceedings of the 2014 11th IAPR International Workshop on Document Analysis Systems, Tours, France, 7–10 April 2014; pp. 181–185.
14. Xie, H.; Rong, W.; Sun, L. Construction and evaluation of a wavelet-based focus measure for microscopy imaging. *Microsc. Res. Tech.* **2007**, *70*, 987–995. [[CrossRef](#)] [[PubMed](#)]
15. Choi, T.S.; Malik, A.S. *Vision and Shape—3D Recovery Using Focus*; Sejong Publishing: Busan, Korea, 2008.

16. Lee, I.-H.; Mahmood, M.T.; Choi, T.-S. Robust Focus Measure Operator Using Adaptive Log-Polar Mapping for Three-Dimensional Shape Recovery. *Microsc. Microanal.* **2015**, *21*, 442–458. [[CrossRef](#)] [[PubMed](#)]
17. Asif, M.; Choi, T.-S. Shape from focus using multilayer feedforward neural networks. *IEEE Trans. Image Process.* **2001**, *10*, 1670–1675. [[CrossRef](#)] [[PubMed](#)]
18. Mahmood, M.T.; Choi, T.-S.; Choi, W.-J.; Choi, W. PCA-based method for 3D shape recovery of microscopic objects from image focus using discrete cosine transform. *Microsc. Res. Tech.* **2008**, *71*, 897–907.
19. Ahmad, M.; Choi, T.-S. A heuristic approach for finding best focused shape. *IEEE Trans. Circuits Syst. Video Technol.* **2005**, *15*, 566–574. [[CrossRef](#)]
20. Muhammad, M.S.; Choi, T.S. A Novel Method for Shape from Focus in Microscopy using Bezier Surface Approximation. *Microsc. Res. Tech.* **2010**, *73*, 140–151. [[CrossRef](#)] [[PubMed](#)]
21. Muhammad, M.; Choi, T.-S. Sampling for Shape from Focus in Optical Microscopy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 564–573. [[CrossRef](#)]
22. Lee, I.H.; Mahmood, M.T.; Shim, S.O.; Choi, T.S. Optimizing image focus for 3D shape recovery through genetic algorithm. *Multimed. Tools Appl.* **2014**, *71*, 247–262. [[CrossRef](#)]
23. Malik, A.S.; Choi, T.-S. Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery. *Pattern Recognit.* **2007**, *40*, 154–170. [[CrossRef](#)]
24. Malik, A.S.; Choi, T.-S. A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise. *Pattern Recognit.* **2008**, *41*, 2200–2225. [[CrossRef](#)]
25. Jang, H.-S.; Muhammad, M.S.; Choi, T.-S. Removal of jitter noise in 3D shape recovery from image focus by using Kalman filter. *Microsc. Res. Tech.* **2017**, *81*, 207–213.
26. Jang, H.-S.; Muhammad, M.S.; Choi, T.-S. Bayes Filter based Jitter Noise Removal in Shape Recovery from Image Focus. *J. Imaging Sci. Technol.* **2019**, *63*, 1–12. [[CrossRef](#)]
27. Jang, H.-S.; Muhammad, M.S.; Choi, T.-S. Optimal depth estimation using modified Kalman filter in the presence of non-Gaussian jitter noise. *Microsc. Res. Tech.* **2018**, *82*, 224–231. [[PubMed](#)]
28. Vasebi, A.; Bathaee, S.; Partovibakhsh, M. Predicting state of charge of lead-acid batteries for hybrid electric vehicles by extended Kalman filter. *Energy Convers. Manag.* **2008**, *49*, 75–82. [[CrossRef](#)]
29. Frühwirth, R. Application of Kalman filtering to track and vertex fitting. *Nucl. Instrum. Methods Phys. Res. A* **1987**, *262*, 444–450. [[CrossRef](#)]
30. Harvey, A.C. *Applications of the Kalman Filter in Econometrics*; Cambridge University Press: Cambridge, UK, 1987.
31. Boulfelfel, D.; Rangayyan, R.; Hahn, L.; Kloiber, R.; Kuduvalli, G. Two-dimensional restoration of single photon emission computed tomography images using the Kalman filter. *IEEE Trans. Med. Imaging* **1994**, *13*, 102–109. [[CrossRef](#)] [[PubMed](#)]
32. Bock, Y.; Crowell, B.W.; Webb, F.H.; Kedar, S.; Clayton, R.; Miyahara, B. Fusion of High-Rate GPS and Seismic Data: Applications to Early Warning Systems for Mitigation of Geological Hazards. In Proceedings of the American Geophysical Union Fall Meeting, San Francisco, CA, USA, 15–19 December 2008.
33. Miall, R.C.; Wolpert, D. Forward Models for Physiological Motor Control. *Neural Netw.* **1996**, *9*, 1265–1279. [[CrossRef](#)]
34. Frieden, B.R. *Probability, Statistical Optics, and Data Testing*; Springer: Berlin, Germany, 2001.
35. Goodman, J.W. *Speckle Phenomena in Optics*; Roberts and Company Publishers: Englewood, CO, USA, 2007.
36. Akselsen, B. *Kalman Filter Recent Advances and Applications*; Scitus Academics: New York, NY, USA, 2016.
37. Pan, J.; Yang, X.; Cai, H.; Mu, B. Image noise smoothing using a modified Kalman filter. *Neurocomputing* **2016**, *173*, 1625–1629. [[CrossRef](#)]
38. Ribeiro, M.I. *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*; Institute for Systems and Robotics: Lisbon, Portugal, 2004.
39. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
40. Subbarao, M.; Lu, M.-C. Image sensing model and computer simulation for CCD camera systems. *Mach. Vis. Appl.* **1994**, *7*, 277–289.
41. Tagade, P.; Hariharan, K.S.; Gambhire, P.; Kolake, S.M.; Song, T.; Oh, D.; Yeo, T.; Doo, S. Recursive Bayesian filtering framework for lithium-ion cell state estimation. *J. Power Sources* **2016**, *306*, 274–288. [[CrossRef](#)]
42. Hajimolahoseini, H.; Amirfattahi, R.; Gazor, S.; Soltanian-Zadeh, H. Robust Estimation and Tracking of Pitch Period Using an Efficient Bayesian Filter. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 1. [[CrossRef](#)]

43. Li, T.; Corchado, J.M.; Bajo, J.; Sun, S.; De Paz, J.F. Effectiveness of Bayesian filters: An information fusion perspective. *Inf. Sci.* **2016**, *329*, 670–689. [[CrossRef](#)]
44. Yang, T.; Laugesen, R.S.; Mehta, P.G.; Meyn, S.P. Multivariable feedback particle filter. *Automatica* **2016**, *71*, 10–23. [[CrossRef](#)]
45. Zhou, H.; Deng, Z.; Xia, Y.; Fu, M. A new sampling method in particle filter based on Pearson correlation coefficient. *Neurocomputing* **2016**, *216*, 208–215. [[CrossRef](#)]
46. Wang, D.; Yang, F.; Tsui, K.-L.; Zhou, Q.; Bae, S.J. Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Spherical Cubature Particle Filter. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1282–1291. [[CrossRef](#)]
47. Mahmood, M.T.; Majid, A.; Choi, T.-S. Optimal depth estimation by combining focus measures using genetic programming. *Inf. Sci.* **2011**, *181*, 1249–1263. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).