Pixel-wise parallel calculation for depth from focus with adaptive focus measure

Yoichi Matsubara¹ · Keiichiro Shirai² · Yuya Ito² · Kiyoshi Tanaka²

Received: 9 March 2021 / Revised: 7 August 2021 / Accepted: 15 August 2021 /
Published online: 28 August 2021
© The Author(s) 2021

Abstract

Depth-from-focus methods can estimate the depth from a set of images taken with different focus settings. We recently proposed a method that uses the relationship of the ratio between the luminance value of a target pixel and the mean value of the neighboring pixels. This relationship has a Poisson distribution. Despite its good performance, the method requires a large amount of memory and computation time because it needs to store focus measurement values for each depth and each window radius on a pixel-wise basis, and filtering to compute the mean value, which is performed twice, makes the relationship among neighboring pixels too strong to parallelize the pixel-wise processing. In this paper, we propose an approximate calculation method that can give almost the same results with a single time filtering operation and enables pixel-wise parallelization. This pixel-wise processing does not require the aforementioned focus measure values to be stored, which reduces the amount of memory. Additionally, utilizing the pixel-wise processing, we propose a method of determining the process window size that can improve noise tolerance and in depth estimation in texture-less regions. Through experiments, we show that our new method can better estimate depth values in a much shorter time.

Keywords Depth from focus · Focus measure · Depth estimation

Abbreviations

DfF	Depth-from-focus
RM	Ratio against mean

Yoichi Matsubara and Keiichiro Shirai have contributed equally to the work.

✉ Keiichiro Shirai
keiichi@shinshu-u.ac.jp

Yoichi Matsubara
matsubara-yoichi-r@pref.nagano.lg.jp

Kiyoshi Tanaka
ktanaka@shinshu-u.ac.jp

¹ Nagano Prefecture Nanshin Institute of Technology, Minami-Minowa, Japan

² Academic Assembly School of Science and Technology Institute of Engineering, Shinshu University, Nagano, Japan

AHO	Adaptive focus measure and high order derivatives
CSTD	Curve standard deviation
FWHM	Full width at half maximum
SAT	Summed-area table
ML	Modified Laplacian
VW	Variable-windowed
GT	Ground truth
RMSE	Root mean square error
VDFF	Variational depth-from-focus

1 Introduction

In the inspection and quality control processes of industrial products, there is a need to easily measure the dimensions of an object when observing it with a light microscope or digital microscope. A depth measurement method which does not require any special optical system is the *depth-from-focus* (DfF) method (Nayar and Nakagawa 1990) (also called *shape-from-focus* methods). DfF uses a set of multiple images taken with different focus settings to estimate the depth from the sharpness of the image texture. The depth of a pixel in the object image can be obtained using the focal position giving the clearest image. Therefore, evaluating the sharpness of images is important (hereafter, called the “focus measure”).

As for focus measures of DfF methods, the best focused position is generally determined from the sharpness of the image texture on an object surface, and the amount of luminance variation is used as the numerical value. For example, (Krotkov 1988; Subbarao et al. 1993; Ahmad and Choi 2007) use the first-order differential of luminance, (Nayar and Nakagawa 1990; Thelen et al. 2009; An et al. 2009) use the second-order differential (such as Laplacian), (Kautsky et al. 2002; Yang and Nelson 2003; Xie et al. 2006) use the wavelet transformation, (Subbarao et al. 1993; Śliwiński et al. 2013; Malik and Choi 2007) use the variance of luminance values. In a survey on focus measure methods including the above (Pertuz et al. 2013), the method using the Laplacian differential gives good results. These methods, however, have the following two major problems:

- (i) Focus measuring fails around edges with large luminance variations. When the amount of variation changes due to spread and blurred edges and exceeds that of in-focus textures, the blurred region is erroneously determined to be in-focus.
- (ii) The accuracy of the depth estimation degrades over flat and texture-less regions, and the variation is small and gradual, so the distinction between blurred and in-focus images is not clear perceptually.

To address these problems, we focus on the relationship between a target pixel value and the mean value of the surrounding pixels, and have proposed a method using the fact that the relationship has a Poisson distribution (Matsubara et al. 2017). Specifically, the method uses the ratio of the luminance value of a target pixel to the mean value of the neighboring pixels, i.e., ratio against mean: RM, as the focus measure (hereafter, we refer to the method (Matsubara et al. 2017) as RM). Additionally, multiple window sizes are used when considering the neighbors, and the evaluation value is integrated by using Frommer’s processing flow (Frommer et al. 2015) (hereafter referred to as AHO: adaptive focus measure and high order derivatives, which is the abbreviation used in the paper). However, RM has the following problem.

- (iii) The calculation of a focus measure is necessary at each depth level, pixel, and window radius, requiring a lot of memory capacity and computation time.

In this paper, to address (iii), we transform the formula of RM and approximately replace the terms with calculations to obtain alternative image features. This approximation also enables pixel-wise parallel processing, and does not require a lot of memory to store the focus measure as volume data. We further utilize the advantage of the pixel-wise processing, and propose an adaptive method of determining a processing window size at each pixel. These proposals can improve performance in noise tolerance and in depth estimation in texture-less regions, making it possible to more accurately estimate depth values in a much shorter time.

The rest of this paper is organized as follows. Section 2 briefly describes the processing flow used in RM and the AHO methods, which form the basis of our method. Section 3 introduces a proposed parallel RM method, and also describes a method using variable processing window sizes as a VW-parallel RM method. Section 4 discusses the effectiveness of our method through experimental results. Finally, Sect. 5 concludes this paper.

2 RM method and definitions

We first describe the processing flow of RM (Matsubara et al. 2017) as the base of the proposed method, and define some numerical notations through the description of RM in this section. Incidentally, RM uses the processing flow of the aforementioned AHO (Frommer et al. 2015), but uses a different focus measurement method. The processing flow of RM consists of the following four steps. Steps 1 and 2 are modified in the proposed method.

1. *Calculation of focus measure* At each pixel, by calculating the degree of sharpness at each image taken with different focus settings, a transition curve of sharpness, i.e., focus measure curve for evaluating the focus measure, is obtained. Multiple curves are calculated using multiple window radii. This calculation method of focus measure values differs from that used in AHO.
2. *Integration of focus measure curves* At each pixel, focus measure curves with clearer, single modal shapes are selected (weighted) and integrated to generate a highly reliable curve. This step is the same as in AHO.
3. *Aggregation* The integrated curve at a pixel is compared with those of its neighbors, and outlier values are removed.
4. *Interpolation of depth values* Each depth value is estimated from the peak position of each focus measure curve. The depth values are interpolated with floating point precision so as to be continuous.

We describe the details of each step in the following sections especially about steps (1) and (2) that are related to the proposed method, and summarize the tasks in Sect. 2.6.

2.1 Preliminary

DfF methods use multiple images taken with different focus settings. The image index is denoted by $z \in \{1, \dots, Z\}$ and also serves as the integer depth value. At the z -th image, values for a pixel (x, y) , such as a pixel value and a focus measure value, are denoted by $V(x, y, z)$ (mainly as a scalar value $V(x, y, z) \in \mathbb{R}^1$) and treated as the volume data of a multi dimensional array. Also, we introduce the notation $V_{x,y}(z)$ when considering the transition of values in the z direction as a curve, and use the notation $V_z(x, y)$ when performing an

averaging/smoothing process using neighboring values on the same xy plane:

$$V(x, y, z) = \begin{cases} V_{x,y}(z) & \text{as the curve at } (x, y), \\ V_z(x, y) & \text{as the } z\text{-th image.} \end{cases} \quad (1)$$

As for the neighboring pixels used in the calculation at a target pixel, let (x, y) and (x', y') be the position (coordinates) of the target pixel and a neighbor. Then let $\mathcal{N}_r(x, y)$ be the set of neighbors contained in a square window of radius $r \in \{1, \dots, R\}$ centered at the target pixel, and $|\mathcal{N}_r| := (2r + 1)^2$ be the number of neighbors.

2.2 Focus measure

In RM, as the focus measure representing the sharpness of a local image region, the deviation between the target pixel luminance value $I_z(x, y)$ (≥ 0) and the mean value of its neighboring pixels $M_z(x, y)$ (≥ 0) is used:

$$M_z^r(x, y) := \frac{1}{|\mathcal{N}_r|} \sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} I_z(x', y'), \quad (2)$$

where the superscript r of $M_z^r(x, y)$ indicates mean values calculated with different window radii r are used. Note that, unlike general calculations such as standard deviation and mean deviation, a ratio is used instead of the difference from the mean value, i.e., ratio against mean (RM):

$$\text{RM}_z^r(x, y) := \left| \frac{I_z(x, y)}{M_z^r(x, y) + \varepsilon} - 1 \right|, \quad (3)$$

where $|\cdot|$ in (3) denotes the absolute value, ε is a small value to avoid division by zero. Using the RM value, a focus measure curve at the target pixel is calculated as

$$\Phi_{x,y}^r(z) := \frac{1}{|\mathcal{N}_r|} \sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} \text{RM}_z^r(x', y'), \quad (4)$$

and multiple curves are calculated with different window radii r . The derivation of the above formula is described in “Appendix A”.

2.3 Integration of focus measure curves

The variation toward the z -direction of focus measure values obtained in (4) typically forms a curve having a local maximum value around the indices of sharp images, and the curve ideally has the shape of Lorentzian functions (Cauchy distributions). We call this curve a “focus measure curve”, and use curves after normalizing the amplitude to 1 by dividing by the maximum value among the images:

$$\widehat{\Phi}_{x,y}^r(z) := \frac{\Phi_{x,y}^r(z)}{\max_z \Phi_{x,y}^r(z)}. \quad (5)$$

The shape of a focus measure curve is different at each window radius r . In texture-less regions, as the window radius increases, the focus measure curve sharpens. In contrast, in texture-full regions having edges with large luminance variations, the curve shape becomes multimodal when a large radius is used. Using this feature, we choose curves having a single sharp modal shape and integrated them to obtain a single curve $\Psi_{x,y}(z)$ as

$$\Psi_{x,y}(z) := \sum_{r \in \{1, \dots, R\}} \alpha_{x,y}^r \hat{\Phi}_{x,y}^r(z), \quad (6)$$

where $\alpha_{x,y}^r \in [0, 1]$ is a weight for integration, which shows the degree of similarity to the ideal unimodal shape, and the weight is represented using the aforementioned Lorentzian function. To calculate it, two parameters of the distribution, i.e., the expected value μ and the *curve standard deviation* (CSTD) $\sigma_{x,y}^r$, are estimated as

$$\begin{aligned} \mu &:= \frac{\sum_{z \in \mathcal{Z}_1} z \cdot \hat{\Phi}_{x,y}^r(z)}{\sum_{z \in \mathcal{Z}_1} \hat{\Phi}_{x,y}^r(z)}, \\ \sigma_{x,y}^r &:= \sqrt{\frac{1}{C_1} \sum_{z \in \mathcal{Z}_1} (z - \mu)^2 \hat{\Phi}_{x,y}^r(z)}, \end{aligned} \quad (7)$$

where the summation range $\mathcal{Z}_1$ is the *full width at half maximum* (FWHM), i.e., the range where the function exceeds half of the maximum value, and $C_1 := \sum_{z \in \mathcal{Z}_1} \hat{\Phi}_{x,y}^r(z)$ is a normalization factor. The weight is represented by using them and a Lorentzian function so that the observed curve has a small CSTD value and a sharp shape:

$$\alpha_{x,y}^r := \left[1 + (\sigma_{x,y}^r / \rho)^2 \right]^{-1}, \quad (8)$$

where the parameter ρ is the damping constant and set to $\rho = 6$ in the experiment.

2.4 Aggregation

Smoothing is applied to the voxel data $\Psi(x, y, z) = \Psi_{x,y}(z)$ in (6) so that outliers are removed and also the focus measure value in a flat region is interpolated from the neighboring values, and the smoothed volume data is denoted by $\bar{\Psi}(x, y, z)$. This smoothing is applied iteratively to each image $\Psi_z(x, y)$:

$$\bar{\Psi}_z^{(t+1)}(x, y) := \sum_{\{x', y'\} \in \mathcal{N}_{r'}(x, y)} \hat{\omega}(x', y') \bar{\Psi}_z^{(t)}(x', y'), \quad (9)$$

where the initial value is set to $\bar{\Psi}^{(0)} = \Psi$. As for the weight $\hat{\omega}$, similar to (6) to (8), it is represented by a Lorentzian function and the CSTD $\sigma_{x,y}$ of $\Psi_{x,y}(z)$, and is normalized by dividing by the maximum value of all pixels as follows:

$$\begin{aligned} \omega(x, y) &:= \left[1 + ((\sigma_{x,y} - \bar{\sigma}) / \rho)^2 \right]^{-1}, \\ \hat{\omega}(x, y) &:= \frac{\omega(x, y)}{\max_{x,y} \omega(x, y)}, \end{aligned} \quad (10)$$

where the damping constant ρ is the same value in (8), and $\bar{\sigma}$ is the mode of CSTD, e.g., the median value of all the pixels.

2.5 Interpolation of depth values

By estimating the depth value at each pixel, a depth map D is obtained. Although the peak position of a focus measure curve $\bar{\Psi}_{x,y}(z)$ can be determined as a depth, in order to obtain the position between two images z and $z + 1$, interpolation using values near the peak position is performed.

Nearest neighbor uses the depth z giving the maximum focus measure value (actually no-interpolation):

$$D(x, y) := \arg \max_z \bar{\Psi}_{x,y}(z). \quad (11)$$

Expected value treats the curve as a distribution function and uses the expected value of the peak position:

$$D(x, y) := \frac{1}{C_2} \sum_{z \in \mathcal{Z}_2} z \bar{\Psi}_{x,y}(z), \quad (12)$$

where $C_2 := \sum_{z \in \mathcal{Z}_2} \bar{\Psi}_{x,y}(z)$ is a normalization coefficient, and $\mathcal{Z}_2$ is the range satisfying the threshold condition $\bar{\Psi}_{x,y}(z) \geq \tau_\Psi$.

2.6 Summary of the processing flow and tasks

Figure 1a shows the processing flow of RM. As for the steps of the focus measuring and the integration of focus measure curves, the order of iterations with respect to pixel, window size, and image are different. This requires that all calculation results are held as volume data and makes it difficult to parallelize the calculation with respect to a pixel.

The tasks that need to be addressed are the following two.

Memory usage The aforementioned steps are processed image-wisely, and the obtained information per pixel has correlation among neighbors and needs to be held as volume data such as $\{\{\hat{\Phi}_z^r\}_{z=1}^Z\}_{r=1}^R$ and Ψ , requiring a large amount of memory. When the number of pixels is N , the amount of data becomes NZ . For example, with $R = 10$, $Z = 20$, and using 4-byte single-precision floating point numbers, $800N$ bytes of memory is required.

Processing time The calculations in (3) and (4) are performed RZ times per pixel, requiring much computation time. To accelerate it, basically, pixel-wise parallel calculation is needed, but it is difficult with these calculations because after the focus measure value is obtained by averaging the neighboring pixels in (3), the neighbors are further averaged in (4). The schematic diagram of this process is shown in Fig. 2a. In (a), summation/averaging calculations are shown by red and blue arrows, and the averaging after the “Nonlinear calculation” (shown by the red arrow, and corresponding to the RM calculation in (3)) requires neighboring results, so it must wait for all the pixels to be processed. This makes the relationship between a target pixel and its neighbors too strong to perform pixel-wise processing.

Incidentally, summation/averaging calculations in Fig. 2a are efficiently calculated by using a constant time calculation method, which is independent of the window radii, called the *summed-area table* (SAT) (Lewis 1995). That is, if a SAT for a summation is generated in advance, summation with any window radius can be calculated quickly. However, when two summations are placed sequentially, the second SAT must be recalculated whenever the output of the Nonlinear calculation changes. This is also a time consuming process.

3 Pixel-wise parallelization of the RM method

This section describes the proposed method and how to address the tasks of RM summarized in Sect. 2.6. As described in the previous sections, the key point for introducing pixel-wise processing is the modification of the calculation in (3) and (4). Therefore, we transform the formula and approximately replace the terms with calculations to obtain alternative image

Fig. 1 Processing flow charts of RM (a) and Parallel RM (b). In the steps of focus measuring, weight calculation, and integration, the order of iterations are different and colored with different colors

features. Hereafter we call the method a “Parallel RM” method. The processing flow is shown in Fig. 1b. In the proposed method, the order of iterations becomes the same and this results in some advantages. The details are described in the following sections.

Note that aggregation and interpolation for depth values are the same as in RM. In our experiments, we do not use aggregation and interpolation for simplicity and also to compare only the changed parts.

3.1 Modification of the focus measure in RM

To modify Eqs. (2)–(4), we introduce the following two assumptions. First, the pixel value $I_z(x, y)$ is represented by the sum of the mean value $M_z^r(x, y)$ and the difference, i.e., Laplacian as

$$I_z(x, y) := M_z^r(x, y) + L_z^r(x, y). \quad (13)$$

Second, the mean value in a local range $\{x', y'\} \in \mathcal{N}_r(x, y)$ is almost constant, i.e., $M_z^r(x'_1, y'_1) \approx M_z^r(x'_{|\mathcal{N}|}, y'_{|\mathcal{N}|})$ and they can be approximated as

$$M_z^r(x'_i, y'_i) \approx M_z^r(x, y). \quad (14)$$

Fig. 2 Schematic diagram of focus measure calculation when the window radius $r = 1$

Substituting (13) into (3), we can rewrite the equation as

$$\text{RM}_z^r(x, y) = \left| \frac{M_z^r(x, y) + L_z^r(x, y)}{M_z^r(x, y)} - 1 \right| = \left| \frac{L_z^r(x, y)}{M_z^r(x, y)} \right|, \quad (15)$$

where ε in (3) to avoid division by zero is omitted. Substituting (14) and (15) into (4), we transform and approximate (4) as

$$\begin{aligned} \Phi_z^r(x, y) &= \frac{1}{|\mathcal{N}_r|} \sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} \left| \frac{L_z^r(x', y')}{M_z^r(x', y')} \right| \\ &\approx \frac{1}{|\mathcal{N}_r|} \frac{\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} |L_z^r(x', y')|}{M_z^r(x, y)} \\ &= \frac{1}{|\mathcal{N}_r|} \frac{\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} |L_z^r(x', y')|}{\frac{1}{|\mathcal{N}_r|} \sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} I_z(x', y')} \\ &= \frac{\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} |L_z^r(x', y')|}{\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} I_z(x', y')}, \end{aligned} \quad (16)$$

where the summations included in the numerator and denominator reduce to one. Actually, the Laplacian $L_z^r(x, y)$ in the numerator implicitly includes summation (filtering is used to calculate it), but we later show that we can calculate the Laplacian efficiently.

On the basis of the deformation, We define Parallel RM as

$$\tilde{\Phi}_z^r(x, y) := \frac{\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} |L_z^r(x', y')|}{\left(\sum_{\{x', y'\} \in \mathcal{N}_r(x, y)} I_z(x', y') \right) + \varepsilon}, \quad (17)$$

where $\tilde{\Phi}_z^r(x, y)$ is the approximation of $\Phi_z^r(x, y)$. ε is a small value to avoid division by zero. We use $\varepsilon = 1.0 \times 10^{-7}$ in (15) and (17) in the experiment. The schematic diagram corresponding to (17) is shown in Fig. 2b. In (b), the Laplacian is calculated in advance of the summation, and then the two summations shown with the blue arrows are performed before the Nonlinear calculation, i.e., division. These summations are efficiently calculated by using summed-area tables (SATs) (Lewis 1995) as aforementioned in Fig. 2a. Once two SATs are generated in advance, summation with different window radii can be calculated quickly, and we can independently combine the focus measure curves in (6) to (8) at each pixel.

3.2 Representation of the Laplacian

The important part in the approximation (15) is the representation of $L_z^r(x', y')$, i.e., the Laplacian covering a wide pixel range. We experimentally find the modified Laplacian (ML) (Nayar and Nakagawa 1990), which with a little modification becomes a good approximation. Although ML has been used in DfF methods and is known to have good performance, combining it with our method gives better performance.

Originally, ML uses second-order differentials with a d pixel interval in the horizontal and vertical directions

$$\begin{aligned}\Delta_x &:= |-I(x-d, y) + 2I(x, y) - I(x+d, y)|, \\ \Delta_y &:= |-I(x, y-d) + 2I(x, y) - I(x, y+d)|,\end{aligned}\quad (18)$$

and is represented by their sum as

$$\text{ML}(x, y) := \Delta_x(x, y) + \Delta_y(x, y), \quad (19)$$

but this original representation is too sensitive to noise, so we represent it by the maximum of the two differential values:

$$L_z^r(x, y) := \max(\Delta_x(x, y), \Delta_y(x, y)). \quad (20)$$

In our experiment, the pixel interval is set to $d = 3$. Note that d is robust to the parameters r and z . Therefore, we can calculate ML in advance of the summation as shown in Fig. 2b. The summation of ML can be calculated by using the summed-area table as well as another summation. Additionally, although ML considers the Laplacian over a wide range of pixels, the calculation is very simple and requires only 5 pixels per target pixel.

3.3 Variable process window size

In Parallel RM, the focus measure curves of each pixel can be calculated independently. We utilize this feature to decide the range of window radii, and propose a pixel-wise variable window size method, referred as to variable-windowed parallel RM (VW-Parallel RM).

Experimentally, in texture-full regions, small window sizes are sufficient for generating clear focus measure curves, whereas texture-less regions require large window sizes. Therefore, changing the window sizes variably, in texture-full regions, can reduce the influence of edges having large luminance changes, whereas in texture-less regions, can improve the depth estimation accuracy.

The algorithm flow of VW-Parallel RM is shown in Algorithm 1 and consists of the following four steps. We use a larger R (the maximum window radius) than that of RM, and truncate the ascending order search for a window size, dependent on the transition of focus measure values.

1. *Features of a focus measure curve* We calculate the maximum value $\Phi_{\max}$, the minimum value $\Phi_{\min}$, the mean value Φ_{mean} , and position to determine the peak $z_{\text{peak}} := \arg \max_z \Phi(z)$. Also, we count the number of peaks n_{peak} , i.e., the number of times a curve exceeds the mean value, and calculate the width of the main peak w_{peak} (this value is only used when the number of peaks is one).
2. *Judgment of starting the truncate mode* When a focus measure curve has a clear single modal peak ($n_{\text{peak}} = 1$), the truncate mode is turned ON. We calculate the contrast of

Algorithm 1 Calculate Ψ at a pixel in VW-Parallel RM.

```

 $r^* \leftarrow -1$  ▷ initialize the truncate mode to OFF
for  $r = 2$  to  $R$  do
  /* Step 1 */
  calculate  $\{\Phi^r(z)\}_z$  using (16)
  calculate CSTD  $\sigma^r$  and  $\alpha^r$  from  $\Phi^r$  using (7), (8)
  calculate  $\Phi_{\max}$ ,  $\Phi_{\min}$ ,  $\Phi_{\text{mean}}$ ,  $z_{\text{peak}}$ ,  $n_{\text{peak}}$ ,  $w_{\text{peak}}$ 

  if  $r^* = -1$  then /* Step 2 */
    if  $n_{\text{peak}} = 1$  and  $\frac{\Phi_{\max} - \Phi_{\min}}{\Phi_{\max}} > 0.5$  then
       $r^* \leftarrow r$  ▷ turn the truncate mode ON
       $\Phi^* \leftarrow \Phi^r$  ▷ set the best curve
       $\alpha^* \leftarrow \alpha^r$ 
       $z^* \leftarrow z_{\text{peak}}$ 
    end if
  else /* Step 3 */
    if  $\alpha^r < \alpha^*$  or  $\frac{|z^* - z_{\text{peak}}|}{w_{\text{peak}}} > 0.5$  or
       $r - r^* > 5$  then
       $\Psi_{x,y} \leftarrow \Phi^*$  return
    else
       $\alpha^* \leftarrow \alpha^r$  ▷ update  $\alpha^*$ 
    end if
  end if
end for

```

the shape and check if it is sufficiently large using

$$\frac{\Phi_{\max} - \Phi_{\min}}{\Phi_{\max}} > 0.5. \quad (21)$$

3. *Truncation decision* When the truncate mode is ON, the ascending order window radius search is truncated if any of the following conditions is filled:

- The weight α^r in (8) becomes smaller than the maximum value α^* .
- The change in peak position z_{peak} is large compared with the peak width. We check if it is sufficiently large using

$$\frac{|z^* - z_{\text{peak}}|}{w_{\text{peak}}} > 0.5. \quad (22)$$

- The above two conditions are not satisfied five consecutive times.

4. *Integration of focus measure curves* In this method, we do not integrate multiple focus measure curves with weights, but use a single focus measure curve, Ψ , at the truncated window radius. If the ascending order window radius search is not truncated, the focusing measure curve at $r = R$ is used.

4 Results and discussion

We show the comparison of the conventional methods and the proposed methods. The conventional methods used are Frommer et al. (2015) and Matsubara et al. (2017). The proposed methods used are Parallel RM and VW-Parallel RM. Incidentally, the algorithm framework of these methods is the same, used initially in AHO. Therefore, as another method with a

different framework, we show the result of the *variational depth from focus reconstruction* (Moeller et al. 2015) (hereafter referred to as VDFF). VDFF uses ML in the focus measure step and total variation minimization as the aggregation step, and produces a realistic depth map. Furthermore, the authors' implementation are published (adrelino: variational-depth-from-focus 2020), so we can easily use VDFF as a suitable reference method.

In our experiment, we use two types of images: artificial images and actual images acquired with a digital microscope. The images are 8bit = $[0, 255]$ images but calculated with floating point precision. Depth maps are scaled within the range $[0, 255]$, the minimum and the maximum depths obtained are mapped to 0 and 255, respectively. Artificial images have a ground truth depth map (GT), so the estimated depth values are scaled by using the scale of GT.

4.1 Memory usage

Theoretically, in the normal RM, the required data amounts to store $\{\Phi, \alpha, \Psi\}$ in (6) are respectively $\{NRZ, NR, NZ\}$, for a total of $N(RZ + R + Z)$, where N is the number of pixels in the whole image. In the proposed Parallel RM, Φ requires temporary storage of only Z (N is not necessary), Ψ and two kinds of summed-area tables require NZ , for a total of $Z + 3NZ$. When $N = 2$ MB, $R = 10$, $Z = 20$, and 4 byte data is used, the memory usage becomes roughly 1.8 GB in RM, but only 0.6 GB in Parallel RM. Therefore, only 1/3 of the memory used in RM is required in Parallel RM.

Practically, Fig. 3 shows the memory usage measured through the Windows OS task manager. The horizontal and vertical axes respectively denote the image size and the memory usage. Regarding four image sizes, the transitions of the memory usage of RM, Parallel RM with 1 thread and 4 threads are shown with the solid lines, whereas the dashed lines indicate the theoretical usage. The measured values (solid lines) almost follow the theoretical values (dashed lines). The drop with respect to RM for the 3MB image size is caused by memory swap. The memory usage of Parallel RM can be reduced to 1/3 of that of RM, regardless of the number of threads.

4.2 Processing time for parallel calculation

We implemented parallel calculation using a CPU and GPU, and measured the processing time for all the processes. The PC used was a Dell Area-51 with an Intel Core i9-7920X CPU, 64GB of DDR4 2666MHz memory and an NVIDIA GeForce 1080 GPU. As for the compiler and parallelizer, we used Microsoft Visual Studio 2017 with MFC for the CPU, and NVIDIA CUDA 9.2 for the GPU.

Figure 4 shows the results. First, one can see the processing times increase linearly as the image size increases. Basically, the Parallel RM is slightly faster than the normal RM, and becomes faster as the number of threads increases, but nonlinearly. The processing time for 4 threads is 1/2 of that for the normal RM. The GPU implementation is much faster, and the processing time is only 1/7 of that for the normal RM.

4.3 Numerical and qualitative evaluation using artificial images

In this study, the two image sets shown in Fig. 5 were used.

Fig. 3 Comparison of memory usage

Fig. 4 Comparison of processing times on a multithread CPU and a GPU

Fig. 5 Artificial image sets used

Table 1 Comparison of RMSE values with the GT depth maps and estimated maps, using Fig. 5

Method		AHO	RM	Parallel RM	VW-parallel RM	VDFF
Depth 1	Center	10.2	8.8	8.3	8.7	4.5
	Whole	11.2	9.3	8.9	9.4	1.9
Depth 1	Center	51.9	16.8	12.5	8.9	2.4
	Whole	64.2	16.2	16.1	9.6	18.3
Depth 2	Center	15.5	20.4	20.4	17.3	24.3
	Whole	13.7	12.5	12.3	11.6	24.4
Depth 3	Whole	66.6	4.0	2.9	4.0	10.5

Set 1: (a) is a texture image of size 700×700 generated by pasting two different metal surface images together. (b) and (c) are artificially generated depth maps of a “slope” and a “staircase”. We generated 11 images having different focuses by blurring each pixel of image (a) based on the prepared depth values, and used Gaussian functions as the point spread function to blur the images.

Set 1 (noisy): Additionally, to check the noise tolerance, we generated 11 noisy images by adding additive white Gaussian noise (with standard deviation: 2.55) to the slope images (b).

Set 2: An image set published by Pertuz (2016), which contains 30 images of size 256×256 with different focuses. (d) is the all-in-focus image generated from all the images.

In this experiment, we perform the evaluation without using aggregation and interpolation of depth values.

4.3.1 Numerical evaluation

Table 1 shows the root mean square error (RMSE) between each ground truth depth map D^* in Fig. 5b, c, e and the depth map D estimated by AHO, RM, Parallel RM, VW-Parallel RM, and VDFF, where RMSE is calculated by

$$\text{RMSE}(D; D^*) := \sqrt{\frac{1}{\sum_{x,y} 1} \sum_{x,y} (D(x, y) - D^*(x, y))^2}. \quad (23)$$

A lower RMSE value indicates better estimation accuracy. In addition, the same table shows the RMSE calculated from the central 100×100 pixels including edges with large luminance variation. From the results, RM, Parallel RM, and VW-Parallel RM give better results than AHO. A comparison between RM and the proposed Parallel RM shows almost the same result.

In comparison with VDFF, VDFF has higher noise robustness, and produces very smooth images. However, the shapes in some regions are not correctly obtained depending on the input image (Fig. 6c-* in the qualitative evaluation). On the other hand, VW-Parallel RM shows better results.

4.3.2 Qualitative evaluation

Figure 6 shows a part of the depth maps obtained from each method. From the results, RM and Parallel RM have similar performance, whereas VW-Parallel RM slightly improves the depth around the edges in the central part. This shows the effectiveness of the proposed approximation (16). We show a more detailed verification of the approximation in Sect. 4.5.

Fig. 6 Qualitative evaluation using artificial images

Fig. 7 Qualitative evaluation using noisy artificial images

4.3.3 Qualitative evaluation for noisy image set

Figure 7 shows resulting depth maps obtained from the noisy images in Set 1 (noisy).

For the comparison of the original modified Laplacian (ML) in (19) and our modification in (20), we show the result of ML in (c). One can see that our modification (d) gives better results than (c), and is highly robust to noise.

Compared with RM (b), Parallel RM (d) gives almost similar results, whereas VW-Parallel RM (e) shows significantly improved noise tolerance. This can also be confirmed from the results in Table 1 because the search for window radius is not aborted at a pixel with noise, and the RM value at the maximum radius is used, so the influence of noise was reduced.

Figure 8 shows the focus measure curves at a dark pixel and a pixel near the edge in the center of the Set 1 (noisy) image in Fig. 5. In addition to the focus measures based on RM, we show the result of a variance-based focus measure method (Śliwiński et al. 2013) for comparison. The vertical axis represents the focus measure. Each curve's vertical scale is normalized so that the maximum value becomes 1. The horizontal axis represents the image index, which is also related to depth. The dashed line near the center represents the depth's

Fig. 8 Comparison of focus measure curves of Set1 image in Fig. 5

ground truth (GT). The ideal curve has a unimodal shape with the maximum value near the GT depth. In the noise-free images (*-1), most methods give unimodal curves with the peak position near GT. In the noisy images (*-2) and (*-3), the unimodal shapes become unclear, and some local maxima appear. The local maxima are often had higher values than

Fig. 9 Qualitative evaluation using actual images. From top to bottom, **a**— an electronic substrate, **b**— a molded item consisting of metal and resin, **c**— marguerite flowers, and **d**— a wing of a winged ant

the true maximum near GT, leading to the wrong estimation of the depth value. On the other hand, VW-Parallel RM curves tend to retain their unimodal shape, and VW-Parallel RM is robust to Gaussian and Poisson noise. The significant improvement in VW-Parallel RM comes from that the process window size is decided so that the curve has a clear unimodal peak, aforementioned in Sect. 3.3.

4.4 Qualitative evaluation using actual images

We show the results using actual images taken with a digital microscope (MXK-Scope by Kikuchi Optics Co., Ltd.) in Fig. 9. The image sizes are 800×600 for images (a-*) and 852×640 for (b-*), (c-*), and (d-*). Additionally, (e-*) is an image of size 1920×1080 used in VDDF, and the image set is available from the web (Haarbach 2017).

The performance of Parallel RM seems to be almost the same as RM, but comparing (a-3) with (a-4), Parallel RM is slightly degraded on the bottom side of the image, which has few textures. However, in VW-Parallel RM (a-5), this part became smoother than RM, i.e., image

Fig. 10 Images used for the validation of the approximation

boundaries were excluded. A larger window radius was applied to the texture-less regions, so the effectiveness of VW-Parallel RM can be seen.

In comparison with VDFF for reference, VDFF gives good results with a smooth appearance even with texture-less regions. However, VDFF gives bad estimation results in (c-6) and (e-6). In (c-6), the complex shapes are over smoothed, and the valleys cannot be observed. In (c-6), the depth of the white teddy bear is indistinguishable from that of the white background wall. Additionally, the width of the plant's stem becomes wider than it appears in (e-1). On the other hand, although the results of VW-Parallel RM are noisy compared to those of VDFF, VW-Parallel RM gives the shape almost accurately even with texture-less regions.

4.5 Validation of the approximation

We validate the appropriateness of the approximation in Sects. 3.1 and 3.2 by using actual images. As an example, we show the results of images in Fig. 10 taken with different focus settings. Incidentally, the in-focus regions of the images are: for $z = 3$, the bottom dark region, and for $z = 7$, the nondark region except for the central depression. The image for $z = 5$ does not have an in-focus region. At the same pixel coordinates (x, y) , we compare the focus measure value $\Phi_z^r(x, y)$ of RM (4) with that of Parallel RM (16), and show the relationship in Figs. 11, 12 and 13. The horizontal and vertical axes respectively denote the values of RM and Parallel RM, and the details are as follows.

First, we compared the values with respect to z with a fixed window radius equal to $r = 5$ and a pixel interval equal to $d = 3$, and show the result in Fig. 11. One can see the shape of each distribution has a clear principle axis (red line obtained by principal component analysis), and there is a linear correlation between the values. Although the range and the gradient of the distribution are different from each other, it only affects the amplitude of the focus measure curve, and does not affect the peak position.

Next, we compared the values with respect to r with $z = 7$ and $d = 3$. Figure 12 shows the results, and linear correlation can be seen especially with values of $r \leq 5$.

Finally, we compared the values with respect to d with $z = 7$ and $r = 5$. Figure 13 shows the results, and linear correlation also can be seen especially with values of $d \leq 5$.

From these results, it was shown that the approximation becomes a good alternative calculation when using small values of d , and $d = 3$ will contribute to give almost the same result as RM.

Fig. 11 Correlation between the RM values of RM and Parallel w.r.t. z for fixed values of r and d

Fig. 12 Correlation between the RM values of RM and Parallel w.r.t. r for fixed values of z and d

Fig. 13 Correlation between the RM values of RM and Parallel w.r.t. d for fixed values of r and z

5 Conclusions

The conventional RM method has two problems: large memory usage and long processing time. To address these problems, we proposed a Parallel RM method that can calculate and integrate the focus measure value at each pixel. An approximation to the averaging process can reduce the number of processes from two to one. Verification showed that the performance of Parallel RM is sufficiently comparable to RM.

This pixel-wise process significantly reduced the processing time and also memory usage by a third.

We also proposed a VW-parallel RM that changes the processing window size on a pixel-wise basis. The method was able to remarkably improve the restoration performance of

noisy images, and also improve the restoration performance in texture-less regions and the restoration accuracy of complicated shapes.

Funding This work was supported by JSPS (Japan Society for the Promotion of Science) KAKENHI, Japan Grant Number JP18K11351.

Declarations

Conflict interests The authors declare that they have no competing interests.

Authors' contribution YM developed the main framework of the proposed method. KS was corresponding author and equal contributor, and discussed about the method. YI has implemented parallel processing program on GPU. KT is a professor in the laboratory where the first author was previously a student and discussed about the method.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: Derivation of the RM value

The derivation of the calculation of the RM value is originally described in our previous method (Matsubara et al. 2017), but written in a different language, so we briefly describe the derivation in this section.

When assuming that in-focus regions have clear textures and unique luminance variations, it is difficult to interpolate the variation from the neighboring luminance values, so the difference between the center pixel value and the mean value of neighbors will be large. Therefore, we consider the probability of estimating a pixel value from the mean value, and use the logarithmic gradient to define the difficulty of estimation as the degree of focusing. We also assume that the luminance variation, especially in dark regions, follows a Poisson distribution (Blanter and Büttiker 2000), and consider the aforementioned also follows a Poisson distribution.

Let J be an observed image and $M = K * I$, ($M > 0$) be an image in which the mean value of neighbors is calculated at each pixel, where K is a box filter, I is a noise-less ideal image, and $*$ denotes filtering. The probability that a luminance value is estimated from the mean value is given by

$$\text{Poisson}(J|M) := \prod_x \frac{M(x)^{J(x)}}{J(x)!} \exp(-M(x)), \quad (24)$$

where x denotes the pixel index. Hereafter, (x) is omitted for simplicity. Then we compute the gradient w.r.t. I after calculating $-\log$. First, the calculation of $-\log$ is given by

$$-\log \text{Poisson}(J|M) = \sum_x \{M - J \circ \log M + \log J!\}, \quad (25)$$

where $\circ$ denotes element-wise multiplication. Next, we substitute $K * I$ for M , and calculate the differential w.r.t. I :

$$\begin{aligned} \frac{\partial}{\partial I} \sum_x \{M - J \circ \log M + \log J!\} &= K \otimes 1 - K \otimes \frac{J}{K * I} \\ &= K \otimes \left(1 - \frac{J}{K * I}\right) = K * \left(1 - \frac{J}{K * I}\right), \end{aligned} \quad (26)$$

where $\otimes$ denotes convolution, which is the same as correlation $*$ when using the symmetric box filter kernel to K . In this gradient, we substitute $I = J$ and replace $(\cdot)$ by absolute $|\cdot|$, and finally obtain the calculation of the RM value:

$$\begin{cases} K * \left|1 - \frac{J}{K * J}\right| &= \frac{1}{N} \sum_{x' \in \mathcal{N}(x)} \text{RM}(x'), \\ \text{RM}(x) &= \left|1 - \frac{J(x)}{J(x)}\right|, \\ \overline{J}(x) &= \frac{1}{N} \sum_{x' \in \mathcal{N}(x)} J(x'). \end{cases} \quad (27)$$

References

- adrelino: Variational-depth-from-focus. (2020). <https://github.com/adrelino/variational-depth-from-focus>.
- Ahmad, M. B., & Choi, T.-S. (2007). Application of three dimensional shape from image focus in LCD/TFT displays manufacturing. *IEEE Transactions on Consumer Electronics*, 53, 1–4.
- An, Y., Kang, G., Kim, I.-J., Chung, H.-S., & Park, J. (2008). Shape from focus through Laplacian using 3D window. In *Proceedings of international conference on future generation communication network*, Vol. 2, pp. 46–50.
- Blanter, Y. M., & Büttiker, M. (2000). Shot noise in mesoscopic conductors. *Elsevier Physics Reports*, 336(1–2), 1–166.
- Frommer, Y., Ben-Ari, R., & Kiryati, N. (2015). Shape from focus with adaptive focus measure and high order derivatives. In *Proceedings of the British machine vision conference (BMVC)*, pp. 134–113412.
- Haarbach, A. (2017). Variational-depth-from-focus: devCam sequences. <https://doi.org/10.5281/zenodo.438196>.
- Kautsky, J., Flusser, J., Zitová, B., & Šimberová, S. (2002). A new wavelet-based measure of image focus. *Pattern Recognition Letters*, 23(14), 1785–1794.
- Krotkov, E. (1988). Focusing. *International Journal of Computer Vision*, 1(3), 223–237.
- Lewis, J. P. (1995). Fast template matching. In *Proceedings of VIS interface*, pp. 120–123.
- Malik, A. S., & Choi, T.-S. (2007). Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery. *Pattern Recognition*, 40(1), 154–170.
- Matsubara, Y., Shirai, K., & Tanaka, K. (2017). Depth from focus with adaptive focus measure using gray level variance based on Poisson distribution. *The Institute of Image Electronics Engineers of Japan (IEEEJ)*, 46(2), 273–282 ((in Japanese)).
- Moeller, M., Benning, M., Schonlieb, C., & Cremers, D. (2015). Variational depth from focus reconstruction. *IEEE Transactions on Image Processing (TIP)*, 24(12), 5369–280935378.
- Nayar, S. K., & Nakagawa, Y. (1990). Shape from focus: An effective approach for rough surfaces. In *Proceedings of IEEE international conference on robotics automation (ICRA)*, pp. 218–225.
- Pertuz, S. (2016). Shape from focus. <http://www.mathworks.com/matlabcentral/fileexchange/55103-shape-from-focus>.
- Pertuz, S., Puig, D., & Garcia, M. A. (2013). Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5), 1415–1432.
- Śliwiński, P., Berezowski, K., & Patronik, P. (2013). Efficiency analysis of the autofocus algorithm based on orthogonal transforms. *Journal of Computer and Communications (JCC)*, 1(6), 41–45.
- Subbarao, M., Choi, T.-S., & Nikzad, A. (1993). Focusing techniques. *Optical Engineering*, 32(11), 2824–2836.
- Thelen, A., Frey, S., Hirsch, S., & Hering, P. (2009). Improvements in shape-from-focus for holographic reconstructions with regard to focus operators, neighborhood-size, and height value interpolation. *IEEE Transactions on Image Processing (TIP)*, 18(1), 151–157.
- Xie, H., Rong, W., & Sun, L. (2006). Wavelet-based focus measure and 3-D surface reconstruction method for microscopy images. In *The Institute of image information and television engineers (ITE)*, Technical report, pp. 229–234.

Yang, G., & Nelson, B. J. (2003). Wavelet-based autofocus and unsupervised segmentation of microscopic images. In *Proceedings of IEEE/RSJ international conference on intelligent robots and systems*, Vol. 3, pp. 2143–2148.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yoichi Matsubara received the B.S. degree in solid-state physics from Tohoku University in 1991 and the Ph.D. degree in engineering from Shinshu University in 2019. From 1991 to 2019, he was a system engineer in private companies. Since 2019, he is a Professor at Nagano Prefecture Nanshin Institute of Technology. His research interests include optical inspection system, image processing and computer vision.

Keiichihiro Shirai received the B.E., M.E., and D.Eng. degrees from Keio University, Yokohama, Japan, in 2001, 2003, and 2006, respectively. He has been with Shinshu University, Japan, as Associate Professor. His current research interests include signal processing, image processing, and computer vision.

Yuya Ito received the M.E. degree in computer science from Shinshu University in 2020. His research interests are in the field of image restoration, high-speed computing and parallel computing.

Kiyoshi Tanaka received his B.S. and M.S. degrees in Electrical Engineering and Operations Research from the National Defense Academy in 1984 and 1989, respectively. In 1995, he joined the Department of Electrical and Electronic Engineering, Faculty of Engineering, Shinshu University, currently he is a full professor in the academic assembly (Institute of Engineering) of Shinshu University. He is the Vice-President of Shinshu University as well as the director of the Center for Global Education and Collaboration (GEC) of Shinshu University. His research interests include image and video processing, 3D point cloud processing, information hiding, human visual perception, evolutionary computation, multiobjective optimization, smart grids, and their applications.