

# Selection of autofocus algorithms for printed circuit board automated optical inspection system

**Rizki Putra Prastio, Rodik Wahyu Indrawan**

Robotics and Artificial Intelligence Engineering, Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga, Surabaya, Indonesia

## Article Info

### Article history:

Received Feb 8, 2023

Revised Mar 3, 2023

Accepted Mar 12, 2023

### Keywords:

Autofocus

Automated optical inspection

Printed circuit board

Selection

Sharpness value

## ABSTRACT

P This paper presents an examination study of 11 autofocus algorithms for printed circuit board (PCB) automated optical inspection (AOI). A selection of an optimal algorithm for that application based on some criteria was carried out. Unlike microscopy, PCB optical inspection does not require very high magnification. The object in this work was also different from that of microscopy and thus influenced the image features. We analyzed 47 PCB images, size of 640×480, sequentially captured every 1 mm in the z-direction. This work utilized USB digital microscope, and the magnification was set at ten times. Each algorithm calculated the sharpness values of the image sequences, and the plot of the sharpness profile was created. Moreover, the research also carried out experiments in several strategies, including image resizing and applying the non-local means (NLM) denoising filter to assess the algorithm performance in different situations. The algorithms were examined and ranked based on five criteria, i.e., computation time, full width at half maximum (FWHM), accuracy, number of half maxima, and range. The experimentation results showed that the Brenner gradient worked best for analyzing images both in their original dimension or resized images.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Rizki Putra Prastio

Robotics and Artificial Intelligence Engineering, Faculty of Advanced Technology and Multidiscipline

Universitas Airlangga

Surabaya, Indonesia

Email: r.p.prastio@ftmm.unair.ac.id

## 1. INTRODUCTION

The artificial vision system in this modern era is widely utilized in numerous fields, including surveillance, robot assembly, vehicle guidance, traffic monitoring, biometric measurement, medical application, automated inspection system, and many other applications [1]. The imaging device must be in the place of focus to obtain clear images of the object being imaged. For automated applications, e.g., printed circuit board printed circuit board (PCB) automated optical inspection, the system is required to be self-adjusted in capturing sharp images to accelerate the whole process. Hence, the computer needs a reliable autofocus algorithm to quantify the level of image sharpness. Some reports have demonstrated the implementation of many image quality metrics for actual applications [2]–[9]. As technologies keep evolving, numerous discussions and new methods have been proposed related to this topic.

The study of autofocus algorithms for microscopy applications is abundant. Considering that the typical magnification of microscopy is greater than equal to 100, the issue arises when the algorithm is going to be applied for another application, such as PCB automated optical inspection (AOI) with lower magnification. Since the object and area under observation are much larger than the microscope sample, the optical system requirement does not necessarily have to be equal. In fact, it is indispensable for a microscope

to have high magnification to obtain great structural detail of the microscopic object. The lens is very close to the object under observation. Furthermore, high magnification, leading to a high numerical aperture, limits the axial movement of the optical system [10]. A tiny axial movement of the lens would influence much of the sharpness. So, the lens must be adjusted in a very short distance. On the contrary, PCB optical inspection needs lower magnification and has a wider depth of field. Thus, the axial travel range of the imaging device is larger compared to the microscope lens system. Certainly, these situations influence the condition of the image, and some algorithms might result in poor performance. To achieve the best image clarity at a single position, the focus evaluation function must show a monotonic behavior, i.e., continuously increasing or decreasing. Moreover, it must show a unimodality profile (single peak).

Some extensive studies measuring and ranking the performances of iterative autofocus algorithms for microscopy have been presented in [11]–[13]. Sun *et al.* [11], [12] comprehensively studied various algorithms that were employed for measuring image sharpness from three types of microscopes, i.e., bright field microscope, phase contrast microscope, and differential interference contrast microscope. Although the best overall performance in this work was normalized variance, Sun *et al.* [11], [12] also found that the autofocus algorithm performed differently for different types of microscope images. It suggested that image characteristics formed by each microscope type influenced the performance of the algorithm. The other comparative assessments in different places and times were also carried out. The investigators used different types of microscopes to observe *Mycobacterium tuberculosis* [14], [15]. Osibote *et al.* [14] employed a bright field microscope, whereas Mateos-Perez *et al.* [15] utilized a fluorescence microscope. In addition, the work in [15] examined different algorithms with some filtering strategies. As a result, there is an agreement between the two research that autocorrelation exhibits excellent performance. Mateos-Perez *et al.* [15] also suggests that a method namely midfrequency discrete cosine transform (MDCT), Lee *et al.* [16], is a very good focus measure for microscope imaging systems previously mentioned.

In the search for robust sharpness image metrics, the combination and optimization of existing methods are also explored. Jia *et al.* [17] employed two stages of image processing. The initial process is Gaussian filtering followed by Laplacian edge detection. Then, the image sharpness is quantified by computing the variance of the resultant image from the previous routine. Following that, the mountain climbing search algorithm dictates the step size of the focus knob motor to reach the peak of the sharpness curve. DiMeo *et al.* [18], seeking the best focus position was estimated through a mathematical model: The Gaussian standard deviation curve. Since the focus metric is modeled, it takes only 3 to 4 images as a basis for approximating the motor step size, through a regression, toward the peak point. Therefore, this method is not computationally expensive.

A further technique in computing image clarity is examining the frequency spectrum by means of the Fourier transform. A smooth transition of pixel intensities denotes a poor high-frequency spectrum [19]. Consequently, the photos do not appear to be visually sharp. Image quality measure is determined by calculating the number of pixel values larger than the threshold derived from the frequency spectrum. On the other hand, an alternative approach was introduced by [20]. Rather than calculating the sharpness value, the algorithm represents the image quality by evaluating the degree of blurriness in horizontal and vertical directions. This method implies that sharper images give lower blur values. Following the modern trend, there is also the latest approach emerging. The researchers try to mimic the human ability to determine the focus image using the learning method established in [21]–[25]. This study aims to provide a performance comparison of 11 methods and select the optimal algorithm that researchers or engineers can use for PCB optical inspection applications. We collected 47 PCB images using a USB microscope with a resolution of 640×480 and 10-times magnification. We also investigated the influence of image resizing and the application of non-local means (NLM) denoising [26]–[29] to the overall performance.

## 2. METHOD

### 2.1. Image preparation

The images were collected using a USB digital microscope that was set to 10 times magnification. The imaging system (the microscope) was controlled to travel along *z* axis every 1 mm to obtain sequential images with varied quality. In total, there were 47 image stacks with different sharpness processed with a computer 2.42 GHz CPU and 8 GB of RAM. This paper used the second version of the dataset, which is publicly available in [30].

Some samples of PCB images being analyzed in this work are depicted in Figure 1. Figure 1(a) shows the dataset's finest image when the microscope is in its focus position. Figure 1(b) presents the blurriest image when the microscope is out of focus. As can be seen, the image quality might not be satisfying due to the resolution and illumination. However, this condition might give researchers or engineers insight into selecting the most robust method. The main library for image processing in this work was OpenCV. The dataset was subjected to some treatments, that is, with and without a denoising filter. In this case, we employed NLM denoising provided by OpenCV with filter strength 10, template window size 7, and search window size 21.

The filter parameters used in this research were the values that the inventor applied in their research [26]. NLM denoising was selected as a noise suppressor due to its ability to maintain the high-frequency signals observed in the edge area. Hence, this denoising method removes noise without introducing a blurring effect in the output image to preserve the original sharpness for the next procedure. Furthermore, to see the effect of image degradation, we also performed the analysis with two different image sizes, 100% (original dimension) and 50% of image size. It should be noted that the aspect ratio of the resized image remains constant, and the resultant image is interpolated according to the pixel area relation.

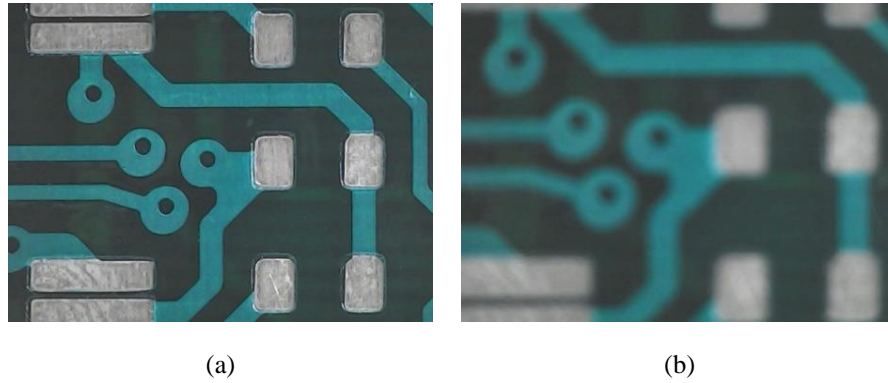


Figure 1. The PCB images used in this work. By manual visual inspection, (a) was the clearest image (at the focus position) and (b) the blurriest image where the lens was very close to the object

## 2.2. Focus measurement algorithm

### 2.2.1. Absolute gradient

As presented in (1) [11], [12], the algorithm sums the absolute difference of a pixel with one neighboring pixel in the  $x$  direction. The other references might apply a threshold so that only the difference values larger than a threshold are counted. However, no threshold was applied in this paper.

$$F_{AbsGrad} = \sum_{Height} \sum_{Width} |i(x+1, y) - i(x, y)| \quad (1)$$

### 2.2.2. Autocorrelation

Autocorrelation describes the degree of dispersion or clusterization from an image. This method is classified as a statistic-based algorithm and expressed as presented (2) [11], [12], [14], [15], where  $i(x, y)$  is the pixel value in a grayscale image. A sharp image is considered clustered, and its autocorrelation value will differ from that of a blurry image.

$$F_{AutoCorr} = \sum_{Height} \sum_{Width} i(x, y) \cdot i(x+1, y) - \sum_{Height} \sum_{Width} i(x, y) \cdot i(x+2, y) \quad (2)$$

### 2.2.3. Blur measurement

Blur measurement (BM) does not calculate the sharpness index. Instead, it measures the amount of blur in the image ranging from 0 to 1. As the name suggests, this metric presents image quality values contrary to most of the algorithms. The blur value gets lower as the image sharpness improves. The method consists of several routines that are explicitly described in [20].

### 2.2.4. Brenner gradient

The brenner gradient algorithm takes a square of the difference between a pixel with its neighbor 2 pixels away. The shown in (3) [11], [12], [14] takes the difference of pixel values in  $x$  direction. It is also possible to take the difference vertically.

$$F_{Brenner} = \sum_{Height} \sum_{Width} (i(x+2, y) - i(x, y))^2 \quad (3)$$

### 2.2.5. Entropy

An image can be thought of as an information set. The amount of information contained in images can be expressed in a value called entropy, as in (4) [11], [12].  $H$  and  $W$  are height and width, respectively:

$$F_{Entropy} = - \sum_{Intensities} \frac{h(i)}{H \cdot W} \log_2 \left( \frac{h(i)}{H \cdot W} \right) \quad (4)$$

where,  $h(i)$  is the histogram of pixel intensity of  $i$ .

### 2.2.6. Energy of Laplacian

The energy of Laplacian is defined as (5) [11], [12], [14], [15]. The convolution of  $i(x, y)$  with  $L$  results in a second derivative of an image that is correlated with the object's edge:

$$F_{EnergyLaplace} = \sum_{Height} \sum_{Width} (i(x, y) * L)^2 \quad (5)$$

with,

$$L = \begin{bmatrix} -1 & 4 & -1 \\ -4 & 20 & -4 \\ -1 & 4 & -1 \end{bmatrix}$$

### 2.2.7. Midfrequency discrete fourier transform

Similar to 2.2.6., that convolves an image with a kernel, MDCT uses a  $4 \times 4$  kernel instead of  $3 \times 3$ . The kernel of MDCT is derived from discrete fourier transform (DCT), as elucidated in [16]. The focus measure is defined as follows:

$$F_{MDCT} = \sum_{Height} \sum_{Width} (i(x, y) * MDCT)^2 \quad (6)$$

with,

$$MDCT = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

### 2.2.8. Squared gradient

Similar to 2.2.1., this algorithm sums the squared difference between neighboring pixels in the  $x$  direction with a distance of 1 [11], [12]. The power of two in (7) results in a steeper gradient than in (1). Consequently, this algorithm provides a bigger difference in sharpness value between images.

$$F_{SqrGrad} = \sum_{Height} \sum_{Width} (i(x+1, y) - i(x, y))^2 \quad (7)$$

### 2.2.9. Standard deviation-based correlation

This algorithm behaves similarly to the (2) expressed in 2.2.2., it takes into account the image dimension and the average of the pixel intensities as stated in (8) [11], [12]. Instead of taking the correlation with two pixels away, this algorithm calculates the correlation between adjacent pixels to its mean value ( $\mu$ ).

$$F_{StdDevCorr} = \sum_{Height} \sum_{Width} i(x, y) \cdot i(x+1, y) - H \cdot W \cdot \mu^2 \quad (8)$$

### 2.2.10. Tenenbaum gradient

Tenenbaum gradient, also known as tenengrad, employs the sobel image detector kernel. It detects the edge of objects within the image both horizontally ( $G_x$ ) and vertically ( $G_y$ ) [11], [12], [14], [15]:

$$F_{Tenengrad} = \sum_{Height} \sum_{Width} (i(x, y) * G_x)^2 + (i(x, y) * G_y)^2 \quad (9)$$

the squared of the convolution is then summed to yield a sharpness value.

### 2.2.11. Variance

As the image gets blurry, the pixel value does not differ drastically from each other. The variance of image [11], [12], [15] describes the spread of pixel values in a dataset measured from the mean ( $\mu$ ). Sharp images theoretically have pixel values that vary greatly.

$$F_{variance} = \frac{1}{H \cdot W} \sum_{Height} \sum_{Width} (i(x, y) - \mu)^2 \quad (10)$$

### 2.3. Selection parameter

In determining the suitability of autofocus algorithms for PCB optical inspection systems, defining some parameters for constructing the overall score is crucial. Since the individual algorithm gives different output values, the focus curve is presented in normalized form. The scoring parameters used in this paper are as follows [11], [12], [14]. The entire method is ranked according to each parameter, and the sum of the rank from each parameter defines the overall performance. Therefore, the optimal algorithm has the lowest score:

- Computation time: it is measured based on the time needed for an algorithm to calculate the sharpness value of a single image.
- Full width at half maximum (FWHM): the distance between two positions where the sharpness index is half the maximum. A smaller value indicates a narrower curve and is preferred.
- Accuracy is defined as the distance between the true focus point determined by the user and the focus point determined according to the image quality computed by the algorithm. Thus, the smaller, the better. The ideal value is 0.
- The number of false maxima: a monotonic focus curve ideally shows a single peak located at the focus point. As seen from the peak point, the plot should show a constantly descending or ascending trend (depending on the method). However, some algorithms might not display this behavior. A good measure provides no false maxima.
- Range: related to parameter number 4, the plot reaches a local minimal/maximum before turning back. The range is measured between two local minimal/maximal. A large range is preferred.

## 3. RESULTS AND DISCUSSION

### 3.1. Original image

In this stage, the sharpness values of the captured images were calculated without applying a filter or image resizing. Thus, the only processing carried out was converting the color images to grayscale mode. The plot of normalized focus curves (original image dimension) calculated by each algorithm is presented in Figure 2. The position of 0 mm indicates the focus point, where the sharpness reaches the peak. One can see that several algorithms did not present a satisfying performance compared to others. They were BM, entropy, energy of Laplacian (Laplace), and standard deviation-based correlation (StdDevCorr). Consequently, these poor-performing algorithms were excluded from the analysis. On the other hand, squared gradient, Brenner gradient, autocorrelation, and Tenengrad were the best candidates, as they showed a narrow Gaussian-shaped profile.

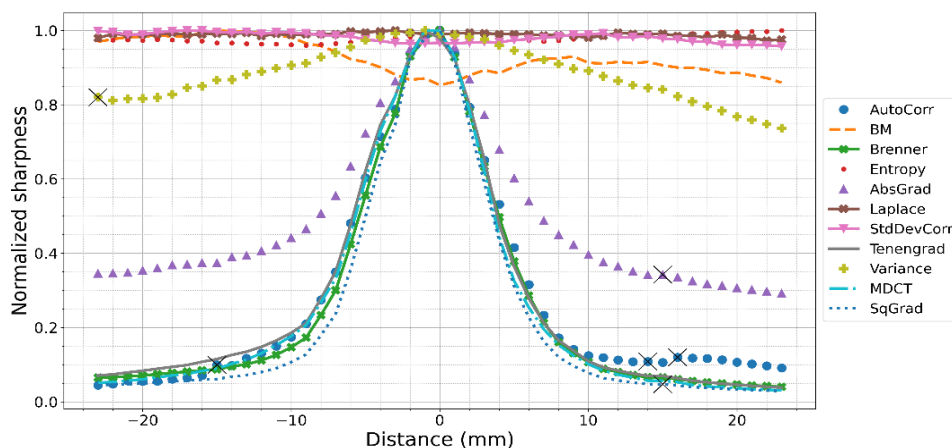


Figure 2. The plot of the sharpness curve from 47 sequential images. No image resizing was performed

Carefully examining the data, we discovered that several false maxima, marked with a black cross, were observed in the plot of autocorrelation (AutoCorr), absolute gradient (AbsGrad), variance, and squared gradient (SqGrad). False maxima are a local peak point where the sharpness value on the right and the left side of this point are lower. For instance, the local maxima in the AbsGrad in Figure 2 appeared at the position of 15 mm. It implies that the sharpness index at that location was higher than the adjacent points. Suppose the search algorithm is not intelligent enough to find the global maxima. The computer will erroneously determine

this position as a focal point in that case. This behavior was also clearly seen in [11], [12]. As we traced from the peak toward the edge of the curve, three local maxima points were seen in the autocorrelation plot. On the left side, upward deflection occurred at -15 mm, while the two other local maxima were visible at 14 mm and 16 mm on the right sight. This could lead to an erroneous determination of the focus position if the optical system travels from the edge of the curve.

A different trend was seen in the remaining methods. A single false maximum was observed at -23 mm in the variance curve. Whereas absolute gradient (AbsGrad) and squared gradient (SqGrad) showed a false maximum on another side, at -15 mm. Brenner, Tenengrad, and MDCT gradient were found as the three algorithms that exhibit no false maxima in this research. In terms of accuracy, the evaluation revealed that the peak of Tenengrad, variance and MDCT were deviating 1 mm off from the true focus position. Table 1 summarizes the performance of the selected algorithms. The value inside parentheses is the rank of the corresponding method according to the parameter being evaluated.

Table 1. The comparison of 7 selected autofocus algorithms based on the performance from each parameter. The dataset dimension was not resized

Parameter	AutoCorr	Brenner	AbsGrad	Tenengrad	Variance	MDCT	SqGrad
Computation time (ms)	10.6 (7)	5.72 (1)	8.04 (4)	8.48 (5)	5.9 (2)	10.4 (6)	7.6 (3)
FWHM (mm)	10 (2)	9 (1)	15 (3)	10 (2)	45 (5)	9 (1)	9 (1)
Accuracy	0 (1)	0 (1)	0 (1)	1 (2)	1 (2)	1 (2)	0 (1)
Number of false maxima	3 (3)	0 (1)	1 (2)	0 (1)	1 (2)	0 (1)	1 (2)
Range (mm)	29 (5)	46 (1)	39 (3)	46 (1)	45 (2)	46 (1)	37 (4)
Total score	18	5	13	11	13	11	11
Global rank	4	1	3	2	3	2	2

In this application, the Brenner gradient undoubtedly provides excellent results among other algorithms since it outperforms the other methods in every aspect. This result agrees with the study carried out by Osibote *et al.* [14] that the Brenner gradient was in first place in terms of computation time. Although the Brenner gradient was not in the first rank for the overall performance in similar research conducted by [8], [11], [12], [14], our study showed otherwise in this application. Similar to Tenengrad and MDCT, the Brenner gradient sees no false maxima along the focus profile, reflecting a monotonic behavior. This profile would prevent the system from mistakenly detecting the focus peak. In the second place, Tenengrad and squared gradient (SqGrad) did not experience a significant disparity in terms of computation time and FWHM. MDCT also offers similar characteristics; however, it performed slowly. One can pre-process the image before sharpness analysis to improve the accuracy of Tenengrad or to increase the range of the squared gradient (SqGrad) algorithm.

### 3.2. Effect of image processing

#### 3.2.1. Under dimension change

Table 1 shows that all the algorithms are considered not computationally expensive since the maximum computation time per image is around 10 ms. To see more possibilities in improving computation time and, at the same time, investigating the effect of the information changes, an image rescaling was conducted. The images were resized to 50% of the original dimension, as depicted in Figure 3. The image was resized using the interpolation method. Figure 3(a) is the image in Figure 1(a) that was reduced to 50%. At the same time, Figure 3(b) is Figure 3(a) that has been stretched to its original size. It is clearly seen that Figure 3(b) is not as sharp as the original image shown in Figure 1(a). The sharpness profile is presented in Figure 4, and the performance ranking is given in Table 2. A visual comparison between Figure 2 and Figure 4 shows a convincing indication that image resizing influences image characteristics. As the algorithms rely heavily on the pixel values, alteration in the image affects the sharpness. Therefore, it is reasonable if the sharpness profile in Figure 4 exhibits a more irregular fashion than the original images. One can see that more false maxima points were observed. Some of them emerged very close to the focus point.

A quantified measure of the effects of image resizing is described in Table 2. Inasmuch as an image size reduction was performed, most algorithms showed improvement in computation time. Conversely, the entire FWHM experienced degradation due to the broadening curve. Under this scenario, the Brenner algorithm still offered the best overall performance. In the aspect of accuracy, Tenengrad and variance performed differently. While Tenengrad precisely detected the peak focus point (improved), the variance algorithm found the peak of sharpness at the position of -3 mm, further away from the true focus position. Another finding in this experiment is the emergence of false maxima in the Brenner gradient and absolute gradient. At the same time, an additional false maxima point is seen in the variance plot at -1 mm. The degradation of the range also occurred in 6 algorithms except for Tenengrad. Regardless of the deterioration of the results, the performance

of three algorithms, i.e., Brenner, Tenengrad, and squared gradient, was still consistent under image dimension variation. Even though SqGrad was at the same rank as MDCT, SqGrad was favored as it was more accurate under image resizing.

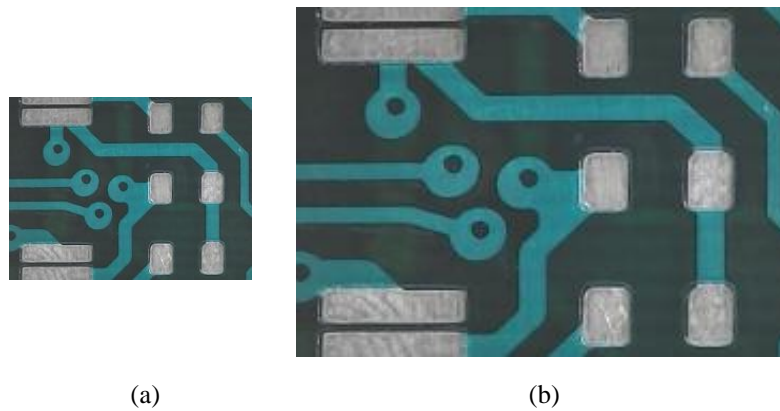


Figure 3. The visual appearance of the resized image; (a) one of the image samples was resized to 50% and (b) the figure in (a) was stretched to the original dimension

Table 2. The algorithm performance for images that were resized by 50%

Parameter	AutoCorr	Brenner	AbsGrad	Tenengrad	Variance	MDCT	SqGrad
Computation time (ms)	6.06 (4)	3.95 (1)	8.13 (7)	7.28 (5)	4.7 (2)	5.45 (3)	7.6 (6)
FWHM (mm)	19 (5)	13 (2)	23 (6)	15 (4)	46 (7)	14 (3)	10 (1)
Accuracy	1 (2)	0 (1)	0 (1)	0 (1)	3 (3)	1 (2)	0 (1)
Number of false maxima	2 (3)	1 (2)	1 (2)	0 (1)	3 (4)	1 (2)	1 (2)
Range (mm)	16 (4)	24 (3)	38 (2)	46 (1)	23 (2)	24 (3)	24 (3)
Total score	18	9	18	12	16	13	13
Global rank	5	1	5	2	4	3	3

Image dimension in this work was not done by image sub-sampling, as demonstrated by Sun *et al.* [12]. Instead, the image dimension was reduced by an interpolation method that modified the pixel values in the resultant image. Smaller dimension images allow rapid computation. However, the result showed that the information reduction led to deterioration in most parameters. The image dimension of 640×480, or equal to 0.3 MP, is reasonably small, and the computer can do rapid computation. Since the image resolution is already low, image resizing is not recommended to avoid the decline in performance.

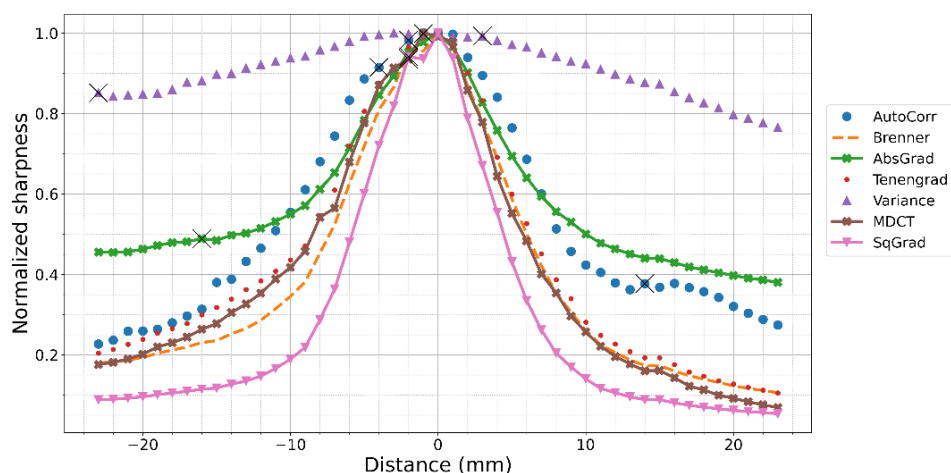


Figure 4. The sharpness curve from images that were resized by 50%



### 3.2.2. Image denoising

A denoising filter, namely NLM filter, was applied to examine its significance on the overall performance. According to the filter parameters described in the method section, it does affect the algorithm's performance. The shortcoming of filter application is increasing the computation time in all algorithms. The performance summary in Table 3 reveals that the processing time per image experienced a dramatic surge, more than 20 times compared to unfiltered images. Many researchers have recognized this time-consuming denoising algorithm, and some alternative methods to accelerate NML filtering have been proposed [31]–[33]. NLM filtering removes the noises differently than a conventional filter. A conventional filter computes the meaning of neighboring pixels (local pixels) being denoised. In contrast, an NLM filter searches for a group of pixels in the entire image region, which the mean value very close to a target pixel. NLM filtering is reported as an excellent technique to preserve image detail. In addition, it does not introduce much blur to the result. The conventional average filter size of  $7 \times 7$  has an algorithm complexity of  $7 \times 7 \times H \times W$ , where  $H$  and  $W$  are height and width, respectively. On the other hand, using the parameter values described in the method section, the complexity of NLM is  $7 \times 7 \times 21 \times 21 \times H \times W$ . Therefore, the upsurge in computation time is reasonable. When one is going to use the NLM denoising method for real-time application, it is recommended to implement parallel processing to speed up the computation [29].

Table 3. The algorithm performance for filtered images, original dimension

Parameter	AutoCorr	Brenner	AbsGrad	Tenengrad	Variance	MDCT	SqGrad
Computation time (ms)	692.9 (7)	434.3 (5)	213.4 (2)	507.8 (6)	255.8 (4)	238.6 (3)	212.9 (1)
FWHM (mm)	10 (4)	9 (3)	13 (5)	10 (4)	46 (6)	7 (1)	8 (2)
Accuracy	0 (1)	0 (1)	0 (1)	1 (2)	1 (2)	0 (1)	0 (1)
Number of false maxima	3 (3)	0 (1)	1 (2)	0 (1)	1 (2)	0 (1)	0 (1)
Range (mm)	35 (3)	46 (1)	37 (4)	46 (1)	45 (2)	46 (1)	46 (1)
Total score	18	11	14	14	16	7	6
Global rank	6	3	4	4	5	2	1

Comparing the performance summary between Table 1 dan Table 3, NLM filtering improved the accuracy of MDCT and eliminated the false maxima of the squared gradient (SqGrad) algorithm. In this situation, Brenner gradient's rank dropped due to its execution time, while the remaining parameter was unchanged. Furthermore, the FWHM of MDCT experienced an improvement since it was seen narrower from 9 mm to 7 mm, as depicted in Figure 5. Nevertheless, there was no noticeable outcome of filtering for the remaining algorithms. The application of NLM denoising in resized images was conducted as well, but the results did not show a satisfying improvement. From Table 3, one can see that NLM denoising strategy had changed the rank. The top three algorithms were SqGrad, MDCT, and Brenner. Due to its time complexity and low impact on performance, the application of NLM denoising is unsuitable for real-time PCB automated optical inspection.

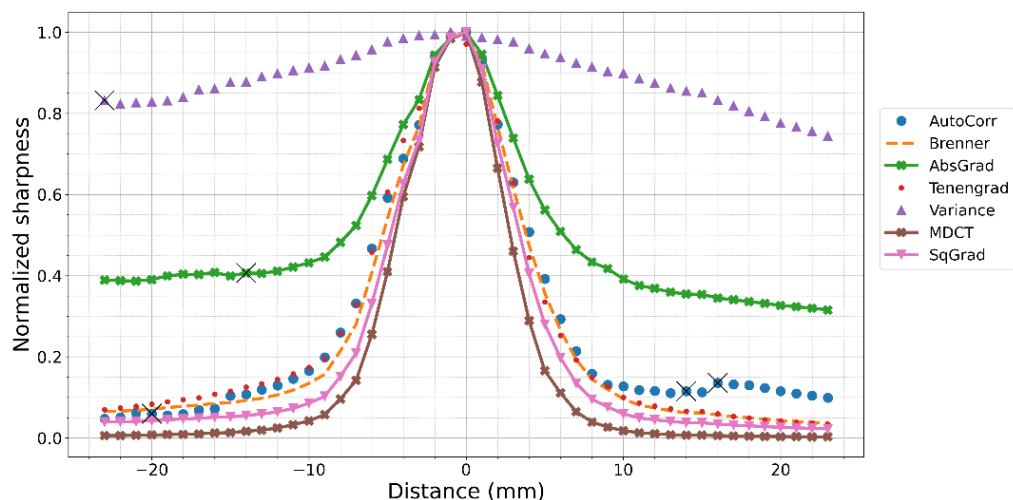


Figure 5. The sharpness curve from filtered images with no image resizing



#### 4. CONCLUSION

This paper presents the study of 11 autofocus algorithms. There were four algorithms, i.e., BM, the energy of Laplacian, entropy, and StdDevCorr, were excluded from analysis due to unsatisfying performance. For the case of unfiltered images, the Brenner gradient was found in the first rank, both in the condition of un-resized image (original dimension) and resized image. Furthermore, the Brenner gradient method offered the fastest computation time under the situation. One can use a squared gradient or Tenengrad with improvements in the second option. In the case of the denoising filter application, the squared gradient was the best. However, that rank was obtained under the use of NLM denoising, which caused a dramatic increase in computation time. Therefore, it is not advisable for real-time usage.

#### ACKNOWLEDGEMENTS

The authors would like to thank Universitas Airlangga for funding the research under grant number 279/UN3.1.17/PT/2021.




#### REFERENCES

- [1] E. R. Davies, "Vision, the challenge," *Computer and machine vision: theory, algorithms, practicalities*. USA: Academic Press, p. 10, 2012.
- [2] J. Zhang and T. Zhang, "Focusing algorithm of automatic control microscope based on digital image processing," *Journal of Sensors*, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/5643054.
- [3] S. Yazdanfar, K. B. Kenny, K. Tasimi, A. D. Corwin, E. L. Dixon, and R. J. Filkins, "Simple and robust image-based autofocus for digital microscopy," *Optics Express*, vol. 16, no. 12, p. 8670, Jun. 2008, doi: 10.1364/oe.16.008670.
- [4] C. Yang, M. Chen, F. Zhou, W. Li, and Z. Peng, "Accurate and rapid auto-focus methods based on image quality assessment for telescope observation," *Applied Sciences (Switzerland)*, vol. 10, no. 2, p. 658, Jan. 2020, doi: 10.3390/app10020658.
- [5] C. T. Liu, Z. X. He, Y. Zhan, and H. C. Li, "Searching algorithm of theodolite auto-focusing based on compound focal judgment," *Eurasip Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 110, Dec. 2014, doi: 10.1186/1687-1499-2014-110.
- [6] S. Liu, M. Liu, and Z. Yang, "An image auto-focusing algorithm for industrial image measurement," *Eurasip Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 70, Dec. 2016, doi: 10.1186/s13634-016-0368-5.
- [7] X. Zhou, X. Wen, Y. Ji, Y. Li, S. Liu, and Z. Liu, "Image reconstruction using autofocus in single-lens system," *Applied Sciences (Switzerland)*, vol. 12, no. 3, p. 1378, Jan. 2022, doi: 10.3390/app12031378.
- [8] M. B. Sanz, F. M. Sánchez, and S. Borromeo, "An algorithm selection methodology for automated focusing in optical microscopy," *Microscopy Research and Technique*, vol. 85, no. 5, pp. 1742–1756, May 2022, doi: 10.1002/jemt.24035.
- [9] T. Werner and J. Carrasco, "Validating autofocus algorithms with automated tests," *Robotics*, vol. 7, no. 3, p. 33, Jun. 2018, doi: 10.3390/robotics7030033.
- [10] F. Shen, L. Hodgson, and K. Hahn, "Digital autofocus methods for automated microscopy," *Methods in enzymology*, vol. 141, pp. 620–632, 2006, doi: 10.1016/S0076-6879(06)14032-X.
- [11] Y. Sun, S. Duthaler, and B. J. Nelson, "Autofocusing in computer microscopy: Selecting the optimal focus algorithm," *Microscopy Research and Technique*, vol. 65, no. 3, pp. 139–149, Oct. 2004, doi: 10.1002/jemt.20118.
- [12] Y. Sun, S. Duthaler, and B. J. Nelson, "Autofocusing algorithm selection in computer microscopy," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2005*, pp. 70–76, doi: 10.1109/IROS.2005.1545017.
- [13] S. Koho, E. Fazeli, J. E. Eriksson, and P. E. Hänninen, "Image quality ranking method for microscopy," *Scientific Reports*, vol. 6, no. 1, p. 28962, Jul. 2016, doi: 10.1038/srep28962.
- [14] O. A. Osibote, R. Dendere, S. Krishnan, and T. S. Douglas, "Automated focusing in bright-field microscopy for tuberculosis detection," *Journal of Microscopy*, vol. 240, no. 2, pp. 155–163, Nov. 2010, doi: 10.1111/j.1365-2818.2010.03389.x.
- [15] J. M. Mateos-Pérez *et al.*, "Comparative evaluation of autofocus algorithms for a real-time system for automatic detection of Mycobacterium tuberculosis," *Cytometry Part A*, vol. 81 A, no. 3, pp. 213–221, Mar. 2012, doi: 10.1002/cyto.a.22020.
- [16] S. Y. Lee, Y. Kumar, J. M. Cho, S. W. Lee, and S. W. Kim, "Enhanced autofocus algorithm using robust focus measure and fuzzy reasoning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1237–1246, Sep. 2008, doi: 10.1109/TCSVT.2008.924105.
- [17] D. Jia, C. Zhang, N. Wu, J. Zhou, and Z. Guo, "Autofocus algorithm using optimized Laplace evaluation function and enhanced mountain climbing search algorithm," *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 10299–10311, Mar. 2022, doi: 10.1007/s11042-022-12191-w.
- [18] P. DiMeo, L. Sun, and X. Du, "Fast and accurate autofocus control using gaussian standard deviation and gradient-based binning," *Optics Express*, vol. 29, no. 13, p. 19862, Jun. 2021, doi: 10.1364/oe.425118.
- [19] K. De and V. Masilamani, "Image sharpness measure for blurred images in frequency domain," *Procedia Engineering*, vol. 64, pp. 149–158, 2013, doi: 10.1016/j.proeng.2013.09.086.
- [20] R. M. Bahy, "Autofocus microscope system based on blur measurement approach," *Journal of Physics: Conference Series*, vol. 1721, no. 1, p. 012058, Jan. 2021, doi: 10.1088/1742-6596/1721/1/012058.
- [21] C. Herrmann *et al.*, "Learning to autofocus," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2020, pp. 2227–2236, doi: 10.1109/CVPR42600.2020.00230.
- [22] Y. Liang, M. Yan, Z. Tang, Z. He, and J. Liu, "Learning to autofocus based on Gradient boosting machine for optical microscopy," *Optik*, vol. 198, p. 163002, Dec. 2019, doi: 10.1016/j.ijleo.2019.163002.
- [23] H. Mir, P. Xu, R. Chen, and P. van Beek, "An autofocus heuristic for digital cameras based on supervised machine learning," *Journal of Heuristics*, vol. 21, no. 5, pp. 599–616, Oct. 2015, doi: 10.1007/s10732-015-9291-4.
- [24] Y. Xiang, Z. He, Q. Liu, J. Chen, and Y. Liang, "Autofocus of whole slide imaging based on convolution and recurrent neural networks," *Ultramicroscopy*, vol. 220, p. 113146, Jan. 2021, doi: 10.1016/j.ultramic.2020.113146.




- [25] C. Feichtenhofer, H. Fassold, and P. Schallauer, "A perceptual image sharpness metric based on local edge gradient analysis," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 379–382, Apr. 2013, doi: 10.1109/LSP.2013.2248711.
- [26] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, Sep. 2011, doi: 10.5201/ipol.2011.bcm\_nlm.
- [27] X. Wang, H. Wang, J. Yang, and Y. Zhang, "A new method for nonlocal means image denoising using multiple images," *PLoS ONE*, vol. 11, no. 7, p. e0158664, Jul. 2016, doi: 10.1371/journal.pone.0158664.
- [28] S. H. Kang and J. Y. Kim, "Application of fast non-local means algorithm for noise reduction using separable color channels in light microscopy images," *International Journal of Environmental Research and Public Health*, vol. 18, no. 6, pp. 1–12, 2021, doi: 10.3390/ijerph18062903.
- [29] F. Huang *et al.*, "A parallel nonlocal means algorithm for remote sensing image denoising on an intel xeon phi platform," *IEEE Access*, vol. 5, pp. 8559–8567, 2017, doi: 10.1109/ACCESS.2017.2696362.
- [30] *PCB optical inspection autofocus*, Mendeley Data, V2, 2023, doi: 10.17632/sz3774jbbp.2.
- [31] V. Karnati, M. Uliyar, and S. Dey, "Fast non-local algorithm for image denoising," in *Proceedings - International Conference on Image Processing, ICIP*, Nov. 2009, pp. 3873–3876, doi: 10.1109/ICIP.2009.5414044.
- [32] J. Lv and X. Luo, "Image denoising via fast and fuzzy non-local means algorithm," *Journal of Information Processing Systems*, vol. 15, no. 5, pp. 1108–1118, 2019, doi: 10.3745/JIPS.02.0122.
- [33] A. Dauwe, B. Goossens, H. Q. Luong, and W. Philips, "A fast non-local image denoising algorithm," in *Proceedings SPIE 6812, Image Processing: Algorithms and Systems VI*, Feb. 2008, vol. 6812, no. 0, pp. 1–8, doi: 10.1117/12.765505.

## BIOGRAPHIES OF AUTHORS



**Rizki Putra Prastio**    born in Pasuruan, Indonesia on 27<sup>th</sup> June 1991, is a lecturer in Robotics and Artificial Intelligence Engineering at Universitas Airlangga, Surabaya, Indonesia. He earned a bachelor's degree in physics from Universitas Airlangga in 2013 and a Master of Electrical Engineering from Institut Teknologi Bandung in 2016. His research interests include biomedical imaging, machine learning, image processing, signal processing, instrumentation, and computer vision. He can be contacted at email: r.p.prastio@ftmm.unair.ac.id.



**Rodik Wahyu Indrawan**    is a lecturer in Robotics and Artificial Intelligence Engineering at the Faculty of Advanced Technology and Multidiscipline Universitas Airlangga, Surabaya, Indonesia. He holds a bachelor of applied science from Politeknik Elektronika Negeri Surabaya and a master of applied science from the same institution. Besides teaching at a university, he also manages several projects from various institutions or industries. Some of his projects include an automated guided vehicle, robotics trainer, and IoT-based scales. His research interests are robotics, mechatronics, and control systems. He can be contacted at email: rodik.w.i@ftmm.unair.ac.id.